Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота бакалавра

на тему: «Розробка веб-додатку на основі фреймворку Django для керування та організації списку завдань»

Виконав: студент групи __ІПЗ-19-1____
Спеціальність 121 "Інженерія програмного забезпечення»

Кривуля В.С.
(прізвище та ініціали)
Керівник к.т.н., доцент Чупілко Т.А.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент Дніпропетровський державний університет внутрішніх справ
(місце роботи)
доцент кафедри інформаційних технологій
(посада)
к. т. н., доцент Насонова С.С
(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро - 2023

АНОТАЦІЯ

Кривуля В.С. Розробка веб-додатку на основі фреймворку Django для керування та організації списку завдань.

Пояснювальна записка: 100 с., 20 рисунків, 8 таблиць, 18 джерел.

Об'єкт розробки: веб-додаток для керування та організації списку завдань.

Предмет розробки : веб – орієнтовані програмні засоби для проектування та розробки веб-додатку для керування та організації списку завдань.

Мета проекту: розробка веб-додатку для автоматизації процесів керування та організації списку завдань.

У першому розділі наведено теоретичні відомості про використані інструменти для вирішення задачі, проведено аналіз та порівняльну характеристика технологій для розробки проектованого ПЗ, зроблено вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу.

У другому розділі проведено аналіз предметної області та специфікації вимог, надана коротка характеристика об’єкту автоматизації та предметної галузі, проведено огляд аналіз існуючих аналогів, наведено специфікацію вимог до системи.

У третьому розділі проведено практичну реалізацію проектованої системи, надано інформаційне, технічне та системне забезпечення розробки.

Актуальність розробки веб-сервісу визначається великим попитом на подібні системи, оскільки автоматизація робить адміністрування завданнями більш гнучкими, зрозумілими, відкритими й стандартизованими.

Список ключових слів: АВТОМАТИЗАЦІЯ, ВЕБ-ДОДАТОК, БАЗА ДАНИХ, PYTHON, HTML, CSS, SQLITE, DJANGO.

ABSTRACT

Krivulya V.S. Development of a web application based on the Django framework to manage and organize a task list.

Explanatory note: 100 pages, 20 figures, 8 tables, 18 sources.

Object of development: a web application for managing and organizing a list of tasks.

Development subject: web-based software tools for designing and developing a web application for managing and organizing a to-do list.

The goal of the diploma project: development of a web application for automating the processes of managing and organizing a list of tasks.

The first section provides theoretical information about the tools used to solve the problem, an analysis and comparative characteristics of technologies for the development of the designed software are carried out, a selection of tools for solving the task is made based on the analysis.

In the second section, an analysis of the subject area and specification of requirements was carried out, a brief description of the automation object and the subject area was provided, an overview and analysis of existing analogues was carried out, and a specification of requirements for the system was given.

In the third section, practical implementation of the designed system was carried out, informational, technical and system development support was provided. The relevance of web service development is determined by the great demand for such systems, as automation makes the administration of tasks more flexible, understandable, open and standardized.

Keywords list: AUTOMATION, WEB APPLICATION, DATABASE, PYTHON, HTML, CSS, SQLITE, DJANGO.

ЗМІСТ

ВСТУП

Автоматизація будь-якого виду надання послуг – завдання на сьогоднішній день відоме та поширене. У сучасному світі не одна галузь не може обійтися без інтеграції в свою роботу інформаційної системи в якомусь її представленні, оскільки це істотно підвищує як зручність і якість взаємодії всередині неї, так і значно підвищує охоплення клієнтської аудиторії та можливості взаємодії з нею.

Актуальність роботи, що полягає у розробці веб-сервісу, визначається великим попитом на подібні системи, оскільки автоматизація робить адміністрування завданнями більш гнучкими, зрозумілими, відкритими й стандартизованими.

Можна говорити про велику кількість успішно впроваджених проектів у різних галузях, у різних за масштабом процесах, з застосуванням різних засобів впровадження процесів автоматизації. Коли ми говоримо про розробку веб-додатку на основі фреймворку Django ми маємо на увазі розробку інтернет-сервісу, який надасть можливість користувачу автоматизовано керувати своїми процесами. Фактично поняття веб-додатку стало невід'ємною частиною галузі адміністрування часом. Результати аналізу існуючих даних свідчать, що нині більшість як великих мережевих компаній, або невеликих локальних тією чи іншою мірою переглянули свій підхід до структури та процесів керування процесами завдань у бік автоматизації.

Метою кваліфікаційної роботи є удосконалення процесу розробки веб-додатку для керування та організації списку завдань шляхом розробки та впровадження Web-сервісу.

Для досягнення мети в роботі поставлено й вирішено такі теоретичні й практичні завдання:

- Провести аналіз предметної області для вибору найкращих інструментів для вирішення поставленого завдання
- Провести аналіз необхідних матеріалів для роботи з мовами програмування
- Провести аналіз необхідних матеріалів для роботи з базами даних
- Обрати архітектури веб – додатку
- Провести аналіз предметної області
- Створення інтерфейсу користувача веб-додатку
- Створення бекенд частини програми для взаємодії з базою даних
- Тестування системи
- Аналіз результатів роботи

Об'єктом дослідження є веб-додаток для керування та організації списку завдань.

Предметом дослідження є веб – орієнтовані програмні засоби для проектування та розробки веб-додатку для керування та організації списку завдань.

Методи дослідження. У процесі роботи застосовувались загальнонаукові та спеціальні методи, які дозволили вивчити предмет та об'єкт дослідження, дослідити напрями та шляхи оптимізації процесу проектування та розробки веб-додатку для керування та організації списку завдань.

Практичне значення одержаних результатів. Практичне значення веб-сервісу визначається великим попитом на подібні системи, оскільки автоматизація робить адміністрування завданнями більш гнучкими, зрозумілими, відкритими й стандартизованими, а також надає значні переваги: мобільність, безконтактність, ціна, масштабованість, оперативність швидкість, ефективність і т.ін.

Структура роботи: вступ, 3 розділи, висновки, список використаних джерел, додаток А, додаток В.

РОЗДІЛ 1.
ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ВИКОРИСТАНІ ІНСТРУМЕНТИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ

1.1 Аналіз існуючих варіантів вирішення поставленого завдання

Коли починається етап розробки веб-додатку для керування та організації списку завдань перше з чим розробник обов'язково зіштовхнеться це HTML (мова гіпертекстової розмітки) і CSS (каскадні таблиці стилів), але це всього лише інструменти для розмітки сторінки, для побудови чогось більшого розробнику обов'язково доведеться зробити вибір мови програмування, що найбільше підходить для вирішення поставленого завдання.

HTML та CSS є мовами розмітки, які використовуються для створення електронних документів (або сторінок) та для розробки сайтів відповідно. Тим часом, мови веб-програмування складніші і можуть бути поділені на дві категорії: внутрішні та зовнішні мови веб-розробки.

Зазвичай веб-розробка включає кодування на стороні сервера (внутрішня частина), кодування на стороні клієнта (зовнішня частина) та технології баз даних [1].

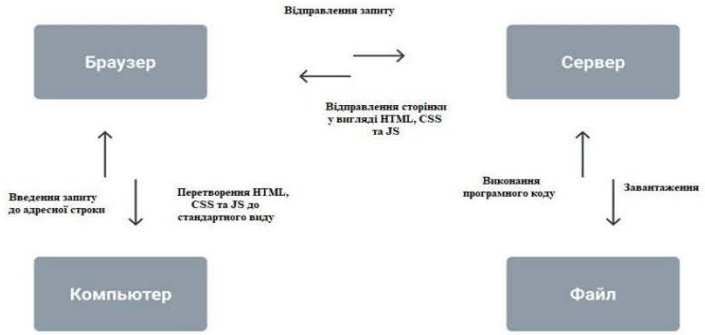


Рисунок 1.1 - Схема роботи типового веб-додатку

Як приклад візьмемо LinkedIn - найбільшу у світі професійну мережу. Щоб дозволити кандидатам з різних регіонів спілкуватися з іншими, шукати роботу або навіть набувати нових навичок, LinkedIn був запрограмований трьома різними мовами веб-програмування: JavaScript, Java та Scala, а потім використав Voldemort як розподілену базу даних для зберігання величезної кількості профілів.

Відповідно, мови для веб-розробки можна визначити як "складні логічні інструкції та процеси", які допомагають створювати сайти, що відповідають певним вимогам.

Отже, розглянемо найпопулярніші мови веб-розробки та визначимо, яка з них найкраще підійде для вирішення нашого завдання [2].

JavaScript. Щоразу, коли мова заходить про веб-розробку, швидше за все, у 9 із 10 разів використовується JavaScript. Згідно з щорічними звітами різних популярних платформ, таких як Stack Overflow та Octoverse, JavaScript є однією з найбільш кращих та провідних мов програмування у світі технологій.

Одна з основних причин цього полягає в тому, що конкретна мова може використовуватися як для веб - розробки переднього плану, так і для веб-розробки внутрішнього інтерфейсу.

Дивлячись на деякі попередні тенденції та статистику, можна сказати, що популярність Node.js якимось чином збільшила використання JavaScript як внутрішньої мови для веб-розробки. Тим часом, мова надає вам чудові функції для серверної розробки, таких як полегшена мова сценаріїв, динамічна типізація, інтерпретація, підтримка об'єктно-орієнтованого програмування, перевірка на стороні клієнта, величезна підтримка спільноти та багато іншого.

Фреймворки JavaScript для серверної веб-розробки: Next.js, Express, MeteorJS і т.д.

Популярні веб-сайти, що використовують JavaScript: Facebook, Google, eBay і т.д [3].

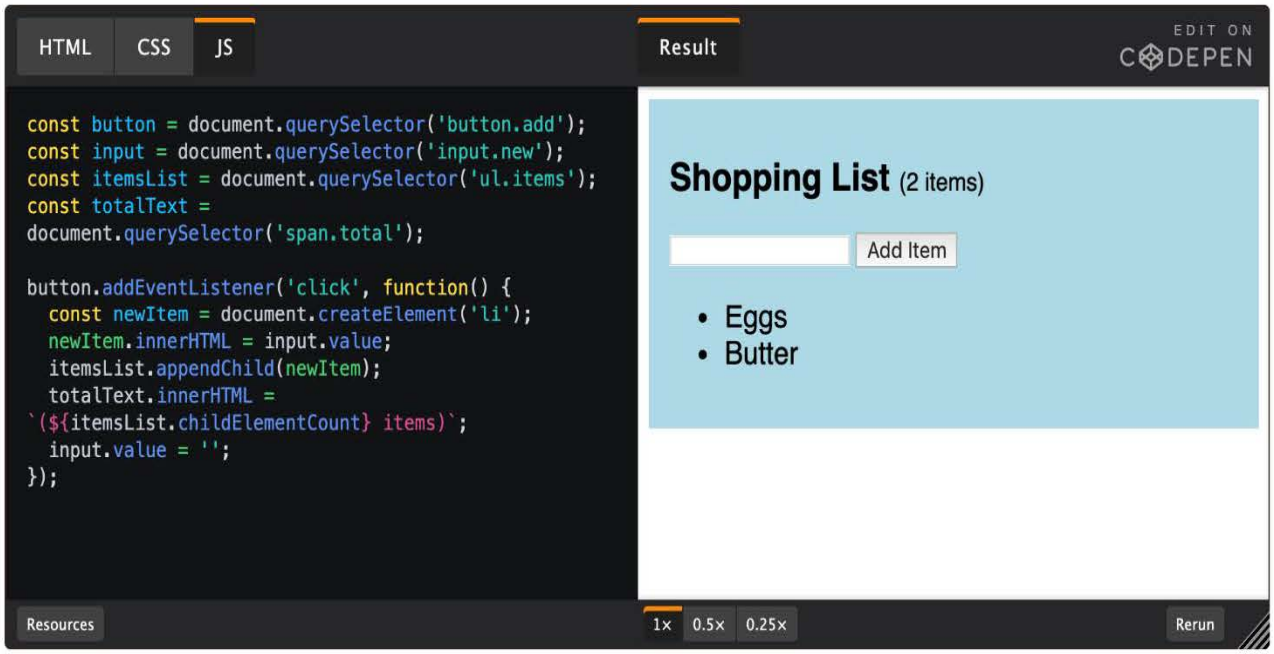


Рисунок 1.2 - Приклад рорробки на JavaScript

Python. Хоча Python досить відомий серед людей своєю сумісністю з передовими технологіями, такими як машинне навчання, IoT, наука про дані тощо, ця збагачувальна мова програмування широко використовується і дуже підходить для серверної веб - розробки.. Навіть один із провідних IT-гігантів сучасності Google значною мірою покладається на Python, і це одна з трьох основних мов, які використовуються Google (дві інші — Java та C++). Однією з основних переваг використання Python для веб-розробки є величезний набір стандартних бібліотек, які роблять роботу розробників порівняно простішою та ефективнішою. Більш проста інтеграція з іншими мовами, підтримка програмування з графічним інтерфейсом, переносимість та багато інших факторів роблять його більш переважною мовою серед веб-розробників.

Фреймворки Python для серверної веб-розробки: Django, Flask, Pyramid тощо.

Популярні веб-сайти, що використовують Python: Spotify, Pinterest, Instacart тощо [4].

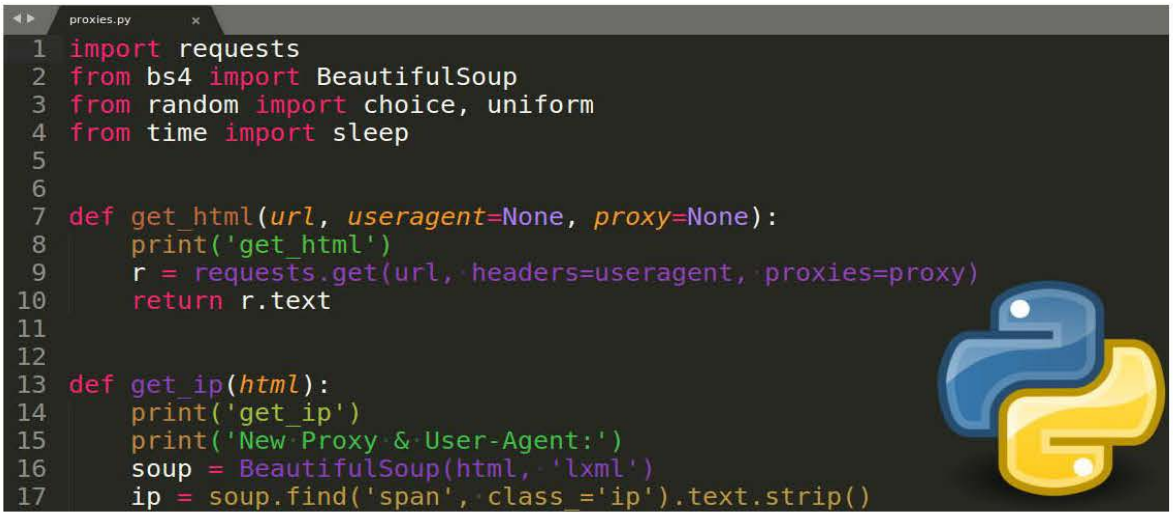


Рисунок 1.3 - Приклад розробки на Python

PHP. PHP є дуже популярною мовою у світі веб-розробки. Ця серверна мова сценаріїв з відкритим вихідним кодом створена в 1994 році і спеціально використовується для веб-розробки. Оскільки це мова, що інтерпретується, вона також не потребує компілятора, а також може працювати практично у всіх основних операційних системах, таких як Windows, Linux, macOS, Unix і т. д., а також має багато переваг, таких як : простоту в освоєнні, крос-платформну сумісність, функції ООП, підтримка різних стандартних баз даних, таких як MySQL, SQLite і т. д., величезна підтримка спільноти та багато іншого. Крім цього, PHP дуже безпечний як мова сценаріїв на стороні сервера, оскільки в PHP доступно безліч хеш-функцій для шифрування даних користувача. Зокрема, для починающих розробників буде доречно обрати PHP для серверної веб-розробки.

PHP-фреймворки для серверної веб-розробки: Laravel, CodeIgniter, Symfony і т.д.

Популярні веб-сайти, що використовують PHP: WordPress, MailChimp, Flickr і т.д [5].

```php
<?php

namespace Email;

class Client
{
    public function send(string $emailAddress, Message $message): bool
    {
        if (!$this->validateParameters($emailAddress, $message)) {
            return false;
        }

        return $this->sendToMailServer([
            'to' => $emailAddress,
            'text' => $message->text(),
        ]);
    }
```

Рисунок 1.4 - Приклад розробки на PHP

Ruby. Ruby - ще одна мова програмування, яка відмінно підходить для веб-розробки. Подібно до PHP і Python, Ruby також простий в освоєнні і підходить для початківців.

Що робить Ruby особливим для веб-розробки, так це середовище Ruby on Rails, на якому працюють такі сайти, як Github, Shopify, Airbnb, Groupon, GoodReads та Kickstarter.

Rails – це середовище модель-представлення-контролер (MVC), що надає стандартні структури для бази даних, веб-служби та веб-сторінок. Він заохочує та полегшує використання веб-стандартів, таких як JSON або XML, для передачі даних та HTML, CSS та JavaScript для взаємодії з користувачем.

Сама мова є високорівневою та чисто об'єктно-орієнтованою, що означає, що «кожне значення є об'єктом», і в Ruby немає примітивних типів даних. Він також має строгу динамічну типізацію та автоматичне складання сміття – форму для управління пам'яттю.

Синтаксис Ruby порівняно схожий на синтаксис Python та Perl і досить гнучкий. Хоча це робить Ruby легким для читання програмістами, він може легко створювати непередбачувані помилки під час виконання, які важко налагоджувати [6].

```ruby
require "time"

class InvoiceItem
attr_reader :id, :item_id, :invoice_id, :created_at

attr_accessor :quantity, :unit_price, :updated_at

def initialize(params)
@id = params[:id].to_i
@item_id = params[:item_id].to_i
@invoice_id = params[:invoice_id].to_i
@quantity = params[:quantity].to_i
@unit_price = BigDecimal(params[:unit_price])
@created_at = Time.parse(params[:created_at].to_s)
@updated_at = Time.parse(params[:updated_at].to_s)
end

def unit_price_to_dollars
@unit_price.to_f
end
end

```

Рисунок 1.5 - Приклад розробки на Ruby

Java. Java – ще одна зразкова мова програмування для серверної веб-розробки. Об'єктно-орієнтована мова програмування широко використовується для розробки веб-додатків масштабу підприємства, а також для розробки додатків для Android, настільних додатків, наукових додатків тощо. Основною перевагою використання Java є те, що він працює за принципом «написати один раз і запустити будь-де», тобто. скомпільований код Java може виконуватись на будь-якій платформі, що підтримує Java, без необхідності повторної компіляції.

Java Frameworks для серверної веб-розробки: Spring, Struts, Grails.

Популярні веб-сайти, що використовують Java: LinkedIn, IRCTC, Yahoo і т.д [7].

```python
def get_symbols(file_name):
    with open(file_name, "r") as in_file:
        records = []
        count = 0
        symbol_set = ""
        for line in in_file:
            symbol_set = symbol_set + line[:-1] + ','
            count = count + 1
            if count % 50 == 0:
                records.append(symbol_set)
                symbol_set = ""

        symbols.append(symbol_set)
        return records
```

Рисунок 1.6 - Приклад розробки на Java

1.2 Аналіз існуючих СУБД для вирішення поставленої задачі

Незалежно від того, яка інформаційна система та з використанням яких мов програмування буде проектуватися, для повного її функціювання будуть використані бази даних.

Наведемо приклад бази даних: онлайн телефонний довідник використовує базу даних для зберігання даних про людей, телефонні номери та інші контактні дані. Постачальник електроенергії використовує базу даних для управління виставленням рахунків, проблемами, пов'язаними з клієнтами, обробкою даних про несправності тощо.

Розглянемо також Facebook. Він повинен зберігати, обробляти та подавати дані, пов'язані з учасниками, їх друзями, діями учасників, повідомленнями, рекламою та багатьом іншим. Можна надати безліч прикладів використання баз даних [10].

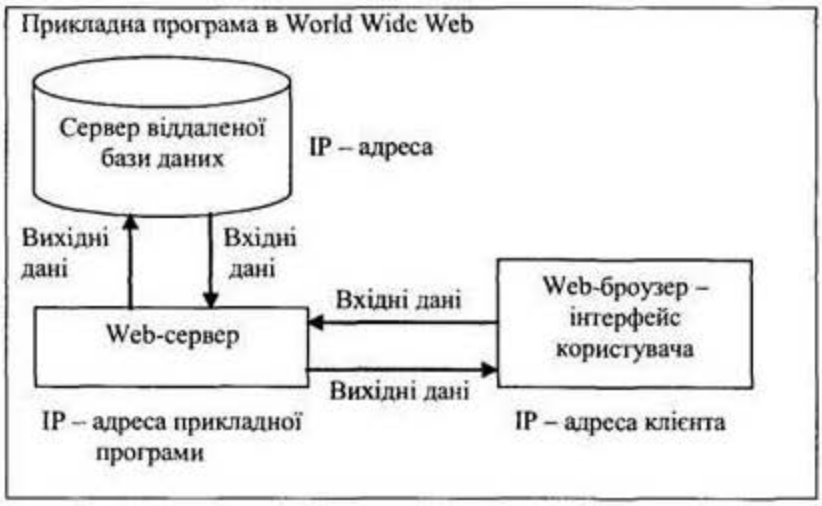


Рисунок 1.7 - Схема роботи бази даних

Отже, розглянемо найпопулярніші системи управління базами даних та визначимо, яка з них найкраще підійде для вирішення нашого завдання.

SQLite. Це одна з найпопулярніших систем реляційних баз даних. Спочатку рішення з відкритим вихідним кодом, SQLite тепер належить

корпорації Oracle. Сьогодні SQLite є основою прикладного програмного забезпечення LAMP. Це означає, що він є частиною стека Linux, Apache, SQLite та Perl/PHP/Python. Маючи під капотом C і C++, SQLite добре працює з системними платформами, як Windows, Linux, MacOS, IRIX та іншими.

Плюси SQLite:

Безкоштовне встановлення. Версія SQLite для спільноти доступна для безкоштовного завантаження. Версія SQLite Community Edition з базовим набором інструментів для індивідуального використання є гарним варіантом для початку.

Простий синтаксис та невелика складність. Структура та стиль SQLite дуже прості. Розробники навіть вважають SQLite базою даних із людською мовою. SQLite часто використовується у тандемі з мовою програмування PHP.

Хмарна сумісність. Орієнтована на бізнес за своєю природою та спочатку розроблена для Інтернету, SQLite підтримується найпопулярнішими хмарними провайдерами. Він доступний на таких провідних платформах, як Amazon, Microsoft та інші [11].

PostgreSQL. Ця система управління базами даних поділяє свою популярність із MySQL. Це об'єктно-реляційна СУБД, в якій об'єкти користувача та табличні підходи об'єднуються для створення більш складних структур даних. Крім того, PostgreSQL має багато спільного з MySQL. Він спрямований на зміцнення стандартів відповідності та розширюваності. Отже, він може обробляти будь-яке робоче навантаження як для одномашинних продуктів, так і для складних додатків.

Плюси PostgreSQL:

Відмінна масштабованість. Вертикальна масштабованість є відмінністю PostgreSQL, на відміну від СУБД MySQL. Враховуючи, що майже будь-яке спеціалізоване програмне рішення має тенденцію до

зростання, що призводить до розширення бази даних, цей конкретний варіант, безумовно, сприяє зростанню та розвитку бізнесу.

Підтримка типів даних користувача. PostgreSQL спочатку підтримує велику кількість типів даних за промовчанням, таких як JSON, XML, H-Store та інші. PostgreSQL використовує цю перевагу, будучи однією з небагатьох реляційних баз даних із сильною підтримкою функцій NoSQL.

Підтримка з відкритим вихідним кодом та спільнотою. Postgres має повністю відкритий вихідний код та підтримується спільнотою, що зміцнює його як повноцінну екосистему [14].

1.3 Вибір інструментів для вирішення поставленого завдання

Для написання візуальної частини інтерфейсу веб-додатку для керування та організації списку завдань буде використано HTML як основну розмітку для сторінок, які відображатимуться перед користувачем та CSS для стилізації цих сторінок та надання їм необхідного нам вигляду. В якості мови для виконання бекенд частини буде використаний Python, проведений аналіз показав, що фреймфорк Django для веб-розробки на його основі найбільш підходить для виконання поставленого завдання, так як має досить зрозумілу структуру, є абсолютно вільно використовуваним, має хороші показники в області продуктивності і відмінно підходить для написання веб - додатків, що недвозначно підтверджується частотою його застосування. В якості СУБД була обрана SQLite, так як вона достатньо проста у використанні, має великий функціонал, безпечна, має хорошу швидкість роботи, а також за замовченням інтегрована у Django. Як IDE для розробки було обрано Pycharm- інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains.

1.4 Висновок до розділу

У даному розділі наведено теоретичні відомості про використані інструменти для вирішення задачі. Перераховані різні мови програмування та їх ключові моменти. Проведено аналіз існуючих варіантів вирішення поставленого завдання, наведено порівняльну характеристику технологій для розробки проектованого веб-додатку, проведено аналіз існуючих СУБД для вирішення поставленої задачі, наведено порівняльну характеристику можливих субд для використання в розробці проектованого веб-додатку. В результаті проведено вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу. Вибір Django надає вбудований адміністративний інтерфейс, який дозволяє легко створювати, змінювати і видаляти дані в базі даних. Це значно спрощує розробку адміністративних панелей для веб-додатків. Висока безпека Django завдяки вбудованим заходам безпеки, такі як захист від вразливостей введення даних і захист від міжсайтового скриптування (XSS). Він також забезпечує механізми автентифікації і авторизації, які допомагають зберегти безпеку вашого додатка. Django надає ORM, яке дозволяє розробникам працювати з базою даних за допомогою об'єктно-орієнтованого підходу. Це спрощує взаємодію з базою даних і дозволяє легко змінювати тип бази даних без необхідності змінювати код додатка. Тим часом, SQLite може бути вбудованим безпосередньо в додатки, що означає, що немає необхідності в окремому сервері або процесі, що запускається незалежно. Він надає простий інтерфейс для управління базами даних без необхідності встановлення окремого серверу. Це робить його ідеальним варіантом для додатків, які вимагають локального збереження даних.Загалом, SQLite є простим у використанні та переносним реляційним базовим системою даних, який підходить для невеликих проектів або додатків з низьким навантаженням.

РОЗДІЛ 2.
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СПЕЦИФІКАЦІЇ ВИМОГ ВЕБ-ДОДАТКУ ДЛЯ КЕРУВАННЯ ТА ОРГАНІЗАЦІЇ СПИСКУ ЗАВДАНЬ

2.1 Обгрунтування теми кваліф роботи

Актуальність обраної теми кваліфікаційної роботи полягає в тому, що з поширенням дистанційної роботи відповідальність за контроль робочих завдань переноситься на самого робітника. У таких умовах робітник має бути відповідальним, організованим та дисциплінованим, щоб забезпечити ефективне виконання робочих завдань та досягнення поставлених цілей. Програмне забезпечення для керування списком завдань може стати ефективним інструментом для дистанційних робітників, щоб організувати свій робочий процес та забезпечити виконання завдань у встановлені терміни.

Крім того, з пандемією COVID-19 багато компаній переходять на дистанційну роботу для збереження здоров'я своїх працівників та підтримки бізнес-процесів. Програмне забезпечення для керування списком завдань може стати ефективним рішенням для керування та контролю робочих завдань, щоб забезпечити безперебійну роботу на дистанції.

Робота програміста вдома може бути важливою складовою для досягнення успіху в IT-індустрії. Дистанційна робота дозволяє програмістам працювати з будь-якого місця, що дає змогу забезпечити гнучкість та зручність у роботі. Програмне забезпечення для керування списком завдань для робітників на дистанційній праці може забезпечити більш ефективний спосіб організації та контролю над завданнями, що потрібні для виконання роботи.

Розробка програмного забезпечення для керування списком завдань може бути корисною не лише для програмістів, але й для менеджерів

проектів, які працюють з командами на дистанційній основі. Ця програма може допомогти керівникам проектів відстежувати прогрес роботи, планувати завдання та взаємодіяти з командою, навіть якщо вони не знаходяться в одному офісі.

Крім того, розробка програмного забезпечення для керування списком завдань може допомогти забезпечити безпеку даних. За допомогою цієї програми можна зберігати дані про завдання на хмарному сервері, що забезпечує їх доступність з будь-якого місця та забезпечує їх збереження в безпечному місці.

Отже, розробка програмного забезпечення для керування списком завдань для робітників на дистанційній праці є важливою темою, оскільки вона може допомогти у забезпеченні ефективної організації робочого процесу та контролю за виконанням завдань на дистанції.

2.2 Коротка характеристика об'єкту веб-додаток для керування та організації списку завдань

Проектований об’єкт є веб – додатком для автоматизації процесів керування та організації списку завдань.

Веб-додаток дає можливість умовним користувачам зареєструватися у системі, авторизуватися у системі якщо користувача вже зареєстровано, створити новий проект, обрати які користувачі будуть виконувати завдання у поточному проекті, адмініструвати процес виконання завдань, редагувати профіль, отримувати статистику щодо поточних виконаних завдань тощо.

На сьогоднішній день існує великий попит на системи автоматизації процесів керування завданнями, тому потрібно мати власну стратегію, щоб забезпечити тривале та ефективне існування свого бізнесу на ринку послуг та зручність процесів керування. Для того, щоб користувачеві виконувати

адміністрування завдань між умовною командою співробітників, йому необхідно безпосередньо контактувати з кожним з них з кожного питання та та використовувати безліч паперових документів, що значно сповільнює взаємодію всередині команди. Для вирішення цієї проблеми найбільш ефективним способом буде створення веб-додатку з всіма функціями, описаними вище.

Дана система дозволить користувачеві отримати всю необхідну інформацію про наявні проекти та завдання, що актуальні до виконання, інформацію щодо них, а також створювати та адмініструвати свої власні проекти та завдання.

2.3 Опис предметної області

Предметною областю даної дипломної є створення веб – додатку для автоматизації процесів керування та організації списку завдань.

Для того, щоб повністю представити функціонал додатку, виділено такі основні процеси:

- Реєстрація нових користувачів;
- Авторизація для вже зареєстрованих користувачів;
- Перегляд списку проектів та завдань, які актуальні для даного користувача
- Можливість адмініструвати активні власні завдання
- Можливість керувати процесом виконання активних завдань
- Можливість календарного відображення дедлайнів завдань;
- Можливість відображення статистики щодо виконаних завдань
- Можливість автоматичного формування та завантаження звіту щодо виконаних завдань

На рисунку 2.1 наведена діаграма процесів програмного продукту, що розробляється

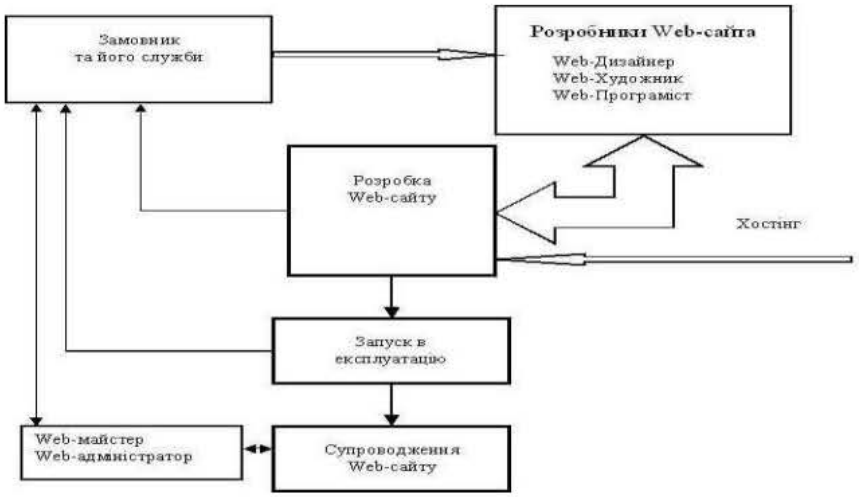


Рис 2.1 - Діаграма процесів розроблюваного програмного продукту

Детальніше розглянемо основний з представлених вище процесів.

Можливість керування завданнями надає користувачеві можливість авторизуватися у додатку, створити новий проект, згідно якого будуть надаватися завдання, або обрати існуючий, створити нове завдання, обрати виконавців завдання, або залишити виконавцем себе, ввести інформацію щодо завдання, дедлайн виконання, отримати сформоване завдання на інтерфейс кожного виконавця у вигляді динамічних блоків статусу виконання та календарного плану виконання.

Основні етапи процесу:

- Авторизація у додатку
- Створення проекту, або вибір існуючого
- Створення завдання
- Вибір виконавців завдання
- Введення інформації щодо завдання
- Отримання інформації щодо статусу виконання
- Отримання календарного плану виконання

Характеристика процесу керування завданнями представлена у таблиці 2.1.

Таблиця 2.1

Характеристика процесу керування завданнями

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Керування завданнями
Основні учасники	Користувач, База даних
Вхідна подія	Запит на керування завданнями
Вихідна подія	Відображення сформованого завдання
Вихідні документи	Календарний план, статус виконання
Клієнт бізнес процесу	Користувач

2.4 Огляд та аналіз існуючих аналогів

Перш ніж почати розробляти певну систему, потрібно виконати пошук її аналогів, визначити їх переваги та недоліки, щоб не допустити ті ж помилки та використовувати переваги проаналізованих систем. Предметною галуззю проектованої системи є керування завданнями у команді.

Проаналізовано три веб-ресурси, що реалізують функції предметної галузі:

1) Веб-додаток «Мегаплан».

2) Веб-додаток «GanttPro»

3) Веб-додаток «Ganter»

Розглянемо докладніше подані вище аналоги. Першим розглянемо додаток Мегаплан:

На рисунку 2.2 зображено вигляд головної сторінки веб-ресурсу (див. Додаток В, ст 100. Рисунок 2.2 - Вигляд головної сторінки ресурсу).

Цей додаток має зручний інтерфейс, за допомогою якого користувач може легко зорієнтуватися у подальших діях.

На цьому ресурсі відображається стрічка актуальних завдань обраного проекту, невелика навігація по додатку представлена у вигляді вертикального меню. При натисканні вкладки відкривається докладна інформація, прикріплена до них. Є можливість створення нових завдань, вибір виконавців, контроль виконання існуючих завдань.

Мінусами є ціна використання та занадто застарілий функціонал та інтерфейс.

Наступним розглянемо веб – додаток GanttPro.

На рисунку 2.3 зображено вид головної сторінки веб-ресурсу:

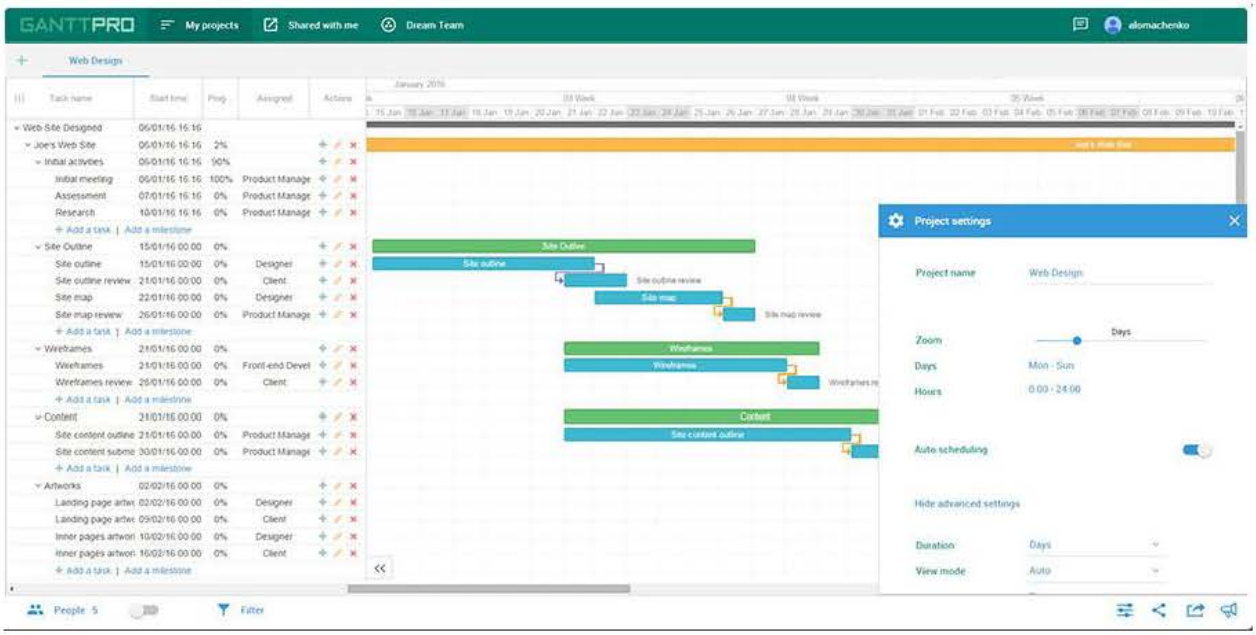


Рисунок 2.3 - Вигляд головної сторінки веб-ресурсу

На даному ресурсі відображається список активних завдань, навігація по додатку представлена у вигляді горизонтального меню, є інформація про поточні завдання, можливість створення підзавдань всередині основних завдань. При натисканні вкладки відкривається докладна інформація, прикріплена до них. Додаток має оновлений інтуїтивний інтерфейс та зручну систему відображення прогресу виконання завдань. Також є можливість завантажити звіт щодо процесу виконання поточних завдань.

Останнім розглянемо додаток Ganter:

На рисунку 2.4 зображено головну сторінку веб-ресурсу

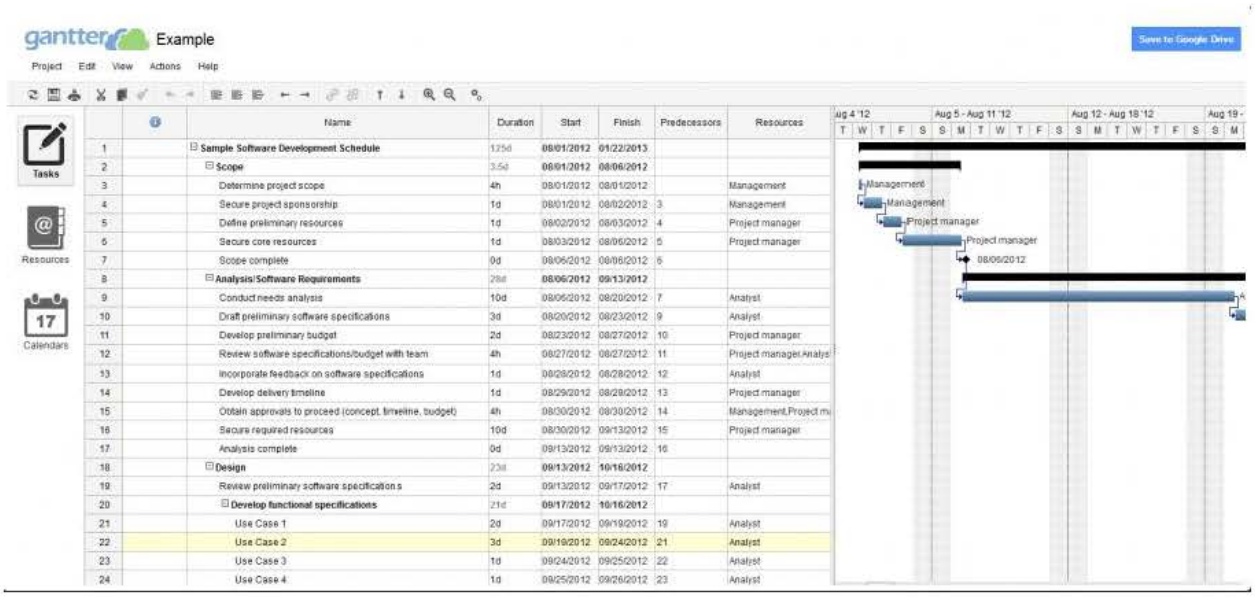


Рисунок 2.4 -Вигляд головної сторінки веб-ресурсу

На даному ресурсі відображається кнопка реєстрації та авторизації користувачів, навігація по додатку представлена у вигляді горизонтального меню. Додаток надає можливість адміністрування поточних завдань між виконавцями, контроль процесу виконання, відображення календарного плану виконання та має можливість імпорту списку завдань. Перевагою

даного додатку є безкоштовність використання. Основними недоліками є застарілий інтерфейс та невеликий функціонал.

Порівняльна характеристика зазначених вище програмних продуктів представлена у таблиці 2.2

Таблиця 2.2

Порівняльна характеристика програмних продуктів

Фірма розробник	Aiwe.pl	Reklama.lviv	Webera
Назва програмного продукту	Мегаплан	GanttPro	Ganter
Функціональність	На цьому ресурсі відображається стрічка актуальних завдань обраного проекту, невелика навігація по додатку представлена у вигляді вертикального меню. При натисканні вкладки відкривається докладна інформація, прикріплена до них. Є можливість створення нових завдань, вибір виконавців, контроль виконання	На даному ресурсі відображається список активних завдань, навігація по додатку представлена у вигляді горизонтального меню, є інформація про поточні завдання, можливість створення підзавдань всередині основних завдань. При натисканні вкладки відкривається докладна інформація, прикріплена до них. Додаток має оновлений інтуїтивний інтерфейс та зручну систему	На даному ресурсі відображається кнопка реєстрації та авторизації користувачів , навігація по додатку представлена у вигляді горизонтального меню. Додаток надає можливість адміністрування поточних завдань між виконавцями, контроль процесу виконання, відображення календарного плану виконання та має

	існуючих завдань.	відображення прогресу виконання завдань.	можливість імпорту списку завдань.
Інтерфейс користувача	Є інтуїтивно зрозумілим, колірна гама підібрана невдало.	Не є інтуїтивно зрозумілим, колірна гама підібрана невдало.	Не є інтуїтивно зрозумілим, колірна гама підібрана невдало.
Переваги	Мінімалістичний інтерфейс	Великий набір функціоналу, зручність	Безкоштовність
Недоліки	Ціна використання , застарілий функціонал та інтерфейс	Ціна використання	Застарілий інтерфейс, невеликий функціонал.

Після огляду систем-аналогів було проаналізовано всі переваги та недоліки цих ресурсів. До переваг відносяться такі функції, як можливість створення нових завдань та розподіл їх на підзавдання, керування виконавцями завдань, керування строками виконання, авторизація у додатку, реєстрація у додатку, можливість отримання звіту щодо процесів виконання, оптимальне відображення процесів виконання та календарного плану на інтерфейсі користувача.

Всі вище наведені переваги та недоліки були враховані під час проектування та розробки веб-додатку для керування та організації списку завдань з передбачуваною назвою «Diploma Tasks».

2.5 Специфікація вимог до системи

Специфікація вимог для програмної системи – це повний опис поведінки системи, що розробляється. Вона включає безліч прецедентів, що описують всі взаємодії, які користувачі мають з програмним забезпеченням.

Прецеденти також відомі як функціональні вимоги. Крім прецедентів специфікація вимог також містить нефункціональні вимоги. Специфікації вимог до розроблюваної системи наведено далі.

У таблиці 2.3 представлений глосарій основних термінів, що використовуються при розробці веб-додатку для керування та організації списку завдань.

Таблиця 2.3

Глосарій основних використовуваних термінів

Веб-сторінка	Інформаційний ресурс, доступний в Інтернеті, який можна переглянути за допомогою веб-браузера
Ключові слова	Слова, що мають суттєве смислове навантаження, вони є ключем при пошуку інформації в інтернеті або на сторінці сайту.
Веб-додатку для керування та організації списку завдань	Веб – сервіс , основним призначенням якого є автоматизація процесів керування та організації завдань у команді.
Браузер	Програмне забезпечення, яке дозволяє користувачеві переглядати інформацію в Інтернеті.
СУБД	Програмне забезпечення, що надає користувачеві можливість створити, редагувати, зберегти та видалити інформацію з баз даних.
Web-server	Апаратне забезпечення, що надає користувачеві свої ресурси та доступ до сервісів.
Адміністратор	Користувач системи , який підтримує роботи системи.
Клієнт	Особа, яка використовує систему для отримання доступу до певних послуг
HTML 5	Мова розмітки

Python	Мова серверної частини
Django	Фреймворк для серверної розробки

Проектований веб-додаток має такі функції:

- Реєстрація нових користувачів;
- Авторизація для вже зареєстрованих користувачів;
- Перегляд списку проектів та завдань, які актуальні для даного користувача
- Можливість адмініструвати активні власні завдання
- Можливість керувати процесом виконання активних завдань
- Можливість календарного відображення дедлайнів завдань;
- Можливість відображення статистики щодо виконаних завдань
- Можливість автоматичного формування та завантаження звіту щодо виконаних завдань
-

Таблиця 2.4

Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги	Приорітет	Складність	Контакт
1	Перегляд списку актуальних проектів та завдань	Обов'язковий	Висока	Користувач, База даних
2	Можливість адміністрування активних завдань	Обов'язковий	Висока	Користувач, База даних
3	Можливість керувати процесом виконання активних завдань	Обов'язковий	Висока	Користувач, База даних

4	Можливість календарного відображення дедлайнів завдань	Обов’язковий	Висока	Користувач, База даних
5	Можливість відображення статистики щодо виконаних завдань	Рекомендовано	Низька	Користувач, База даних
6	Можливість автоматичного формування та завантаження звіту щодо виконаних завдань	Рекомендовано	Середня	Користувач, База даних
7	Авторизація	Обов’язковий	Висока	Користувач
8	Реєстрація	Обов’язковий	Висока	Користувач

Таблиця 2.5

Специфікація нефункціональних вимог :

Ідентифікатор вимоги	Назва вимоги	Приорітет	Складність	Контакт
1	Основні вимоги до застосування нової системи щодо інших систем	Опціональний	Низька	Користувач Адміністратор

2	Вимоги до відповідності стандартам графічного інтерфейсу користувача	Рекомендовано	Низька	Користувач Адміністратор
3	Доступність	Обов’язковий	Середня	Адміністратор
4	Середній час безвідмовної роботи	Обов’язковий	Середня	Адміністратор
5	Точність	Обов’язковий	Середня	Адміністратор
6	Використання ресурсів	Рекомендовано	Середняя	Адміністратор
7	Вимоги до технології програмування	Рекомендовано	Середня	Адміністратор

Значення нефункціональних вимог:

- Даний проект реалізований мовою програмування Python за допомогою веб – фреймворку Django. В якості СУБД використано Sqlite, а також мови розмітки HTML та CSS.
- Середній час безвідмовної роботи становить 3 місяці;
- Оперативна пам'ять 512Mb;
- процесор із частотою не менше 2.50GHz;
- дискове місце на стороні сервера 5-10Gb.

2.6 Висновок до розділу

У даному розділі було проведено аналіз предметної області та специфікації вимог веб-додатку для керування та організації списку завдань тому, що визначення предметної області має важливе значення, оскільки воно визначає функціональність, потреби користувачів, типи даних, що обробляються, та особливості взаємодії з іншими системами або сервісами. Коректне розуміння предметної області дозволяє створити веб-додаток, який найкращим чином задовольнятиме потреби користувачів та пропонуватиме необхідні функції та можливості. Проведено розгляд теми кваліфікаційної роботи, наведено коротку характеристику об'єкту проектованого веб-додатку. Проведено опис предметної області та огляд і аналіз існуючих аналогів, розглянуто їх переваги та недоліки. Проведено вивчення веб-сервісів аналогів для розвитку власного веб-сервісу. Після огляду систем-аналогів було проаналізовано всі переваги та недоліки цих ресурсів. Це допоможе знайти нішу або розробити унікальні функції, які відрізнятимуть сервіс. Аналізуючи веб-сервіси, ми виявили їхні сильні та слабкі сторони. Це дасть ідеї щодо покращення свого власного веб-сервісу. Також можна навчитися від їхніх успіхів і помилок, вдосконалити дизайн, функціонал або впровадити нові ідеї. Також виявлено вимоги до системи серверу які мають велике значення для ефективної роботи веб-сервісу. Врахування цих вимог допомагає забезпечити оптимальну продуктивність, безпеку та доступність для користувачів. Достатня кількість оперативної пам'яті дозволяє серверу ефективно обробляти багато одночасних запитів та зберігати кешовані дані для покращення продуктивності. Вимоги до сховища даних можуть варіюватися залежно від типу та обсягу даних, які зберігаються в веб-сервісі. Необхідно вибрати підходячу базу даних або інше сховище, яке відповідає потребам щодо швидкості, масштабованості та безпеки даних.

РОЗДІЛ 3.
ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТОВАНОГО ПРОГРАМНОГО ПРОДУКТУ

3.1 Інформаційне забезпечення системи

Система, що розробляється, повинна бути інтегрована з веб-орієнтованим продуктом, а це означає, що потрібно розробити рекомендаційну структуру бази даних, в якій будуть міститися дані, необхідні для функціювання веб-додатку. Щоб реалізувати весь функціонал, а саме: збереження галереї зображень, збереження відгуків, збереження інформації і всі відомості про них, а також заявки, що надходять на сайт. Інтеграція бази даних потрібна для оптимізації роботи веб-додатку з критично важливими даними, що зберігаються під час використання сайту.

Для цього проекту було прийнято рішення, в якості СУБД використовувати MySQL.

Система є веб-сервісом , тому дані будуть знаходитись у базі даних. Вони підтягуватимуться для подальшого відображення користувачеві, тому необхідно мати постійний зв'язок сервера з базою даних та клієнта із сервером.

Для цього рішення було обрано локальне розміщення бази даних разом із веб-ресурсом. Першим кроком є створення таблиці ідентифікаторів, яка наведена у таблиці 3.1.

Таблиця 3.1

Таблиця ідентифікаторів

Об'єкт	Власність	Тип	Розмірність	Ідентифікатор
Таблиця	Айді	varchar	100	ID

користувачів	Пошта	Varchar	100	Mail
	Логін	Varchar	100	Login
	Пароль	Varchar	100	Password
Таблиця проектів	Айді	varchar	100	ID
	Логін адміна	varchar	100	Login
	Виконавці	varchar	100	Performers
	Назва	varchar	100	Name
	Опис	Varchar	500	Descr
	Дедлайн	Varchar	500	Deadline
Таблиця завдань у проектах	Айді	Varchar	100	ID
	Проект	Varchar	100	Project
	Назва завдання	Varchar	100	Name
	Опис завдання	Varchar	100	Descr
	Дедлайн завдання	Varchar	100	Deadline
	Виконавець завдання	Varchar	100	Performer
	Статус завдання	Varchar	100	Status

3.2 Обґрунтування вибору технічних засобів

Одним з найважливіших факторів, який необхідно врахувати під час розробки програми, є відповідність потреб у ресурсах наявному технічному забезпеченні. Програма має коректно працювати на сумісному з ним обладнанні.

Реалізація проекту проводиться в системі розробки сценаріїв, включає інтерпретатор мови, набір функцій для доступу до баз даних та різних служб Інтернет. Використання цього продукту пред'являє такі вимоги до обладнання та програмного забезпечення [17]:

— процесор Intel Pentium III 866 МГц та вище (рекомендується Intel Pentium IV / Celeron 1800 МГц);

— оперативна пам'ять 1024 Мбайт та вище;

— дисковий простір щонайменше 100 Мб;

— відеокарта 256Мб;

— мінімальна роздільна здатність екрана 1024x768;
— 32-розрядна операційна система Windows (NT/2000/XP/Vista/Seven/10);
— підтримка технології Flash;
— Інтернет-браузер: Opera, Mozilla Firefox, Internet Explorer,Google Chrome;
— маніпулятор "миша";
— клавіатура.

Для коректного запуску програмного продукту на сервері, визначено наступні вимоги до серверного обладнання та програмного забезпечення:

— Операційна система – Linux (CentOs, Debian тощо) .
— Python, Django
— Оперативна пам'ять від 2 Гб.
— Наявність SSD
— Процесор щонайменше 2-х ядер.
— Vesta, ISPmanager

3.2.1 Обґрунтування вибору ОС та протоколу обміну даними

Для сервера вибрано операційну систему Windows Server Standard Edition, вона розроблена спеціально для невеликих відділів компаній та забезпечує ефективне створення загального доступу до файлів та принтерів, безпечне підключення до Інтернету, централізоване розгортання настільних додатків та веб-рішення для організації взаємодії співробітників, партнерів, клієнтів.

Сервер Windows Server Standard Edition забезпечує високий рівень надійності, масштабованості та безпеки.

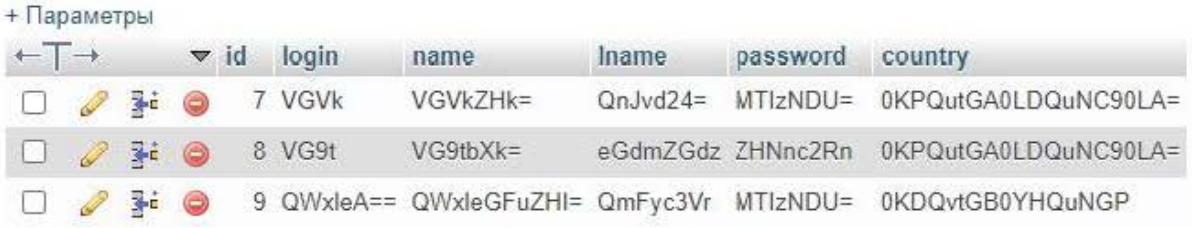
Для робочих станцій обрано операційну систему Microsoft Windows 10, так як вона має: високий рівень безпеки, включаючи можливість шифрування файлів та папок із метою захисту корпоративної інформації; підтримку мобільних пристроїв для забезпечення можливості працювати автономно або підключатися до комп'ютера у віддаленому режимі; вбудована підтримка високопродуктивних багатопроцесорних систем; можливість роботи з серверами Microsoft Windows Server та системами управління підприємствами; ефективна взаємодія з іншими користувачами у всьому світі завдяки можливостям багатомовної підтримки. Програма має український інтерфейс, що зручно всім користувачів підприємства.

3.2.2 Заходи захисту від несанкціонованого доступу до системи

Для забезпечення інформаційної безпеки особистих даних користувачів, ці дані буде зашифровано за допомогою алгоритмів симетричного шифрування. Симетричне шифрування передбачає використання одного й того самого ключа і для шифрування, і для розшифрування. До симетричних алгоритмів застосовуються дві основні вимоги: повна втрата всіх статистичних закономірностей в об'єкті шифрування та відсутність лінійності. Прийнято розділяти симетричні системи на блокові та потокові. У блокових системах відбувається розбиття вихідних даних на блоки з подальшим перетворенням ключем. У потокових системах виробляється певна послідовність (вихідна гамма), яка у подальшому накладається саме повідомлення, і шифрування даних відбувається потоком у міру генерування гами. Схема зв'язку з використанням симетричної криптосистеми представлена малюнку. Де M - відкритий текст, K - секретний ключ, що передається по закритому каналу, En(M) - операція шифрування, а Dk(M) - операція розшифрування.

Операція підстановки виконує першу вимогу до симетричного шифру, позбавляючись будь-яких статистичних даних шляхом перемішування бітів повідомлення за певним заданим законом. Перестановка необхідна до виконання другої вимоги – надання алгоритму нелінійності. Досягається за рахунок заміни певної частини повідомлення заданого обсягу на стандартне значення шляхом звернення до вихідного масиву.

Симетричні системи мають свої переваги, так і недоліки перед асиметричними. До переваг симетричних шифрів відносять високу швидкість шифрування, меншу необхідну довжину ключа за аналогічної стійкості, велику вивченість і простоту реалізації. Недоліками симетричних алгоритмів вважають насамперед складність обміну ключами через велику ймовірність порушення секретності ключа при обміні, який необхідний, і складність управління ключами у великій мережі.



	id	login	name	lname	password	country
	7	VGVk	VGVkZHk=	QnJvd24=	MTIzNDU=	0KPQutGA0LDQuNC90LA=
	8	VG9t	VG9tbXk=	eGdmZGdz	ZHNnc2Rn	0KPQutGA0LDQuNC90LA=
	9	QWxleA==	QWxleGFuZHI=	QmFyc3Vr	MTIzNDU=	0KDQvtGB0YHQuNGP

Рис. 3.1 - Зберігання зашифрованих даних у базі даних

3.3 Архітектура проектованого програмного продукту

Під час проведення аналітичної частини роботи було вирішено використовувати веб – фреймфорк Django для реалізації проектованого програмного продукту, тому розглянемо архітектуру майбутнього Django проекту.

Архітектура Django схожа на «Модель-Уявлення-Контролер» (MVC). Контролер класичної моделі MVC приблизно відповідає рівню, який у Django називається уявлення (англ. View), а презентаційна логіка реалізується в

Django рівнем Шаблонів (англ. Template). Через це рівневу архітектуру Django можна представити як "Модель-Шаблон-Представлення" (MTV). Архітектура проектованого продуту Django представлена малюнку 3.1.

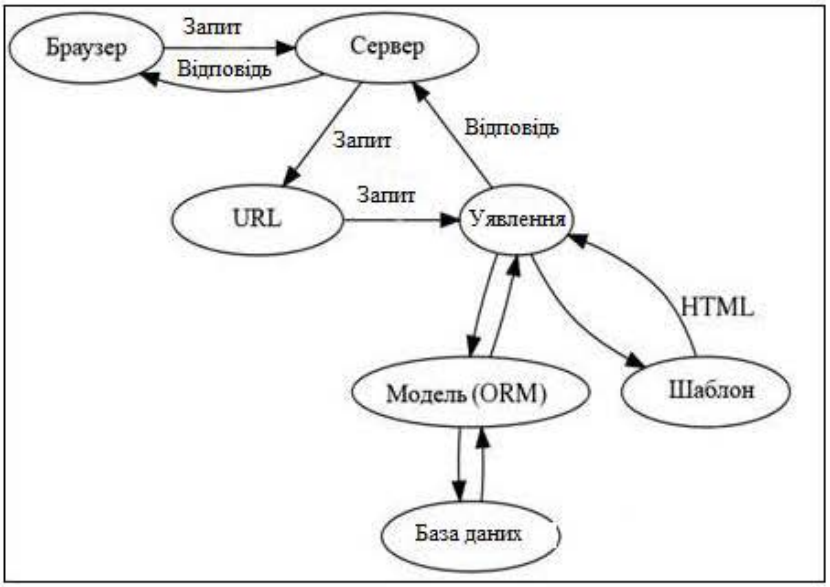


Рисунок 3.2 - Архітектура проектованого продуту Django

Архітектура Django складається з чотирьох основних компонентів:

1) Модель даних є серцевиною будь-якого сучасного веб-додатку. Модель - це найважливіша частина програми, яка постійно звертається до даних за будь-якого запиту з будь-якої сесії. Будь-яка модель є стандартним Python класом. Об'єктно-орієнтований mapper (ORM) забезпечує таким класам доступ безпосередньо до баз даних. Якби не було ORM, довелося писати запити безпосередньо на SQL. Модель забезпечує полегшений механізм доступу до шару даних, інкапсулює бізнес-логіку. Модель не залежить від конкретної програми. Даними можна маніпулювати навіть із командного рядка, не використовуючи при цьому веб-сервер;

2) Уявлення виконують різноманітні функції, зокрема контролюють запити користувача, видають контекст залежно з його ролі. View - це

звичайна функція, яка викликається у відповідь на запит якоїсь адреси (URL) та повертає контекст;

3) Шаблони є формою представлення даних. Шаблони мають свою власну просту метамову і є одним з основних засобів виведення на екран;

4) URL механізм зовнішнього доступу до уявлень (view). Вбудовані в URL регулярні вирази роблять механізм досить гнучким. При цьому одне подання може бути налаштоване на кілька URL, надаючи доступ до різних програм. Тут підтримується філософія закладок: URL стають самодостатніми та починають жити незалежно від уявлення.

3.4 Програмна реалізація моделей завдань

Відповідно до нашого завдання, першими реалізуємо моделі (класи) проектів та завдань та детально опишемо їх. Реалізуємо перший клас Project для створення проекту завдань користувачем (див. Додаток А: Models.py ст. 60).

Реалізований клас надає можливість створення нового проекту з полями імені, опису, деталей, власника, виконавців та фото проекту. Клас реалізовано на основі класу models, який використовується у Django для взаємодії з базою даних. Інформацію щодо створеного проекту буде занесено до бази даних для подальшої взаємодії.

Наступним кроком буде реалізовано клас для створення відповідно самих завдань у проекті для подальшого керування ними (див. Додаток А: Models.py ст. 60).

Реалізований клас надає можливість створення завдання з полями назви, опису, власника, часу старту виконання, часу дедлайну, статусу виконання та віповідного проекту. Клас так само реалізовано на основі класу models, який використовується у Django для взаємодії з базою даних. Інформацію щодо створеного завдання буде занесено до бази даних для подальшої взаємодії.

Наступним реалізуємо клас для отримання інформації щодо статусу виконання завдань відповідно поточному користувачу (див. Додаток А: Models.py ст. 58).

Реалізований клас надає можливість отримати інформацію щодо кожного актуального для поточного користувача проекту та завдань у ньому для формування звіту. Логіка класу реалізує отримання інформації щодо назви актуальних проектів завдань для поточного користувача, виконавців, статусу виконання для формування звіту щодо поточного стану виконання завдань.

Наступним реалізуємо клас для отримання інформації щодо поточного користувача (див. Додаток А: Models.py ст. 58-59).

Реалізований клас надає можливість отримати інформацію щодо поточно авторизованого користувача та всіх проектів завдань для нього для формування звіту. Логіка класу реалізує отримання інформації щодо статусу поточних проектів авторизованого користувача для формування звіту щодо поточного стану виконання завдань.

3.5 Програмна реалізація логіки проекту

Моделі реалізовано, тому наступним кроком необхідно провести реалізацію безпосередньо логіки взаємодії користувача з сервером. У views.py містяться всі можливі керуючи функції. Вони контролюють який шаблон слід використовувати та які дані та методи моделі слід використовувати. Вся логіка додатка та всі можливі підрахунки зосереджені у цих функціях. View, або уявлення, це те місце, де буде реалізовано «логіку» роботи проектованого додатка. Вони взаємодіють з інформацією з моделей, які було створено раніше, і передають її в шаблони.

Насамперед реалізуємо логіку першої взаємодії користувача з додатком:

```
def index(request):
    if request.user.is_authenticated:
        return redirect('boards')
    else:
        return redirect('signIn')
```

Реалізована функція виконує логіку перевірки користувача на авторизацію. Якщо користувача авторизовано, його буде перенаправлено на сторінку boards, тобто головну сторінку додатку. В зворотньому випадку – користувача буде перенесено на сторінку проходження процесу авторизації signIN, що є необхідною умовою використання додатку.

Якщо користувач перший раз використовує додаток, йому потрібно пройти процес реєстрації. Реалізуємо логіку реєстрації у наступному класі: Views.py (див. Додаток А, ст. 95).

Даний клас реалізує логіку реєстрації користувача. Після перевірки користувача і підтвердження, що його не авторизовано – методом post передаються параметри, які користувач вказав при реєстрації, а саме логін, пароль та email, користувачу надається стандартне фото для профілю. Далі наведена функція виконує збереження наведених особистих даних користувача у базу даних. Якщо процес пройдено успішно – користувача буде додано до бази даних та перенаправлено до головної сторінки додатку. Якщо такого користувача вже зареєстровано – функція поверне помилку з відповідним текстом повідомлення, та не авторизує користувача.

Якщо користувача вже зареєстровано у додатку, йому потрібно пройти процес авторизації. Реалізуємо логіку авторизації у наступному класі: Views.py (див. Додаток А, ст. 95-96).

Даний клас реалізує логіку авторизації користувача. Після перевірки користувача і підтвердження, що його не авторизовано – методом post передаються параметри, які користувач вказав при авторизації, а саме логін та пароль. Далі наведена функція виконує перевірку на наявність такого

користувача у базі даних. Якщо процес пройдено успішно – користувача буде авторизовано та перенаправлено до головної сторінки додатку. Якщо наведеної пари логіну та паролю не знайдено – функція поверне помилку з відповідним текстом повідомлення, та не авторизує користувача.

Якщо користувача авторизовано у додатку, він завжди повинен мати можливість вийти з облікового запису. Реалізуємо логіку де-авторизації у наступному класі:

```
class SignOut(View):
    def get(self, request):
        logout(request)
        return redirect('signIn')
```

Даний клас робить запит на видалення сесії користувача та перенаправлення його на сторінку авторизації.

Весь необхідний функціонал для виконання процесів авторизації користувача реалізовано, наступним кроком реалізуємо безпосередньо логіку взаємодії з проектами та завданнями.

Першим реалізуємо клас для взаємодії з моделлю створення проекту та отримання інформації щодо поточних проектів (див. Додаток А: Views.py ст.59).

Даний клас реалізує логіку відображення інформації щодо актуальних проектів та створення нових проектів поточним користувачем. Функція get після перевірки користувача і підтвердження, що його авторизовано робить запит щодо інформації про поточного користувача та списку активних проектів у базі даних додатку. Далі за допомогою циклу проекти ітеруються та якщо поточний користувач є власником проекту або його виконавцем – додаються до списку активних проектів.

Функція post відповідає за безпосередньо логіку взаємодії з моделлю створення нових проектів. Після виклику та перевірки користувача і підтвердження, що його авторизовано – функція отримує дані щодо нового

проекту завдань методом POST, а саме назву проекту, опис, деталі, власника, список id виконавців та фото проекту. Наступним кроком функція робить запит до моделі для безпосередньо збереження нового проекта у базі даних та у випадку успіху перенаправляє користувача на головну сторінку додатку.

Коли логіку створення проекту завдань реалізовано, необхідно надати користувачеві можливість видалити вже не активний проект. Реалізуємо для цього наступну логіку :

```
class MangeProject(View):
    def post(self, request, id):
        Project.objects.filter(id=id).delete()

        response = JsonResponse({"message": "OK"})
        response.status_code = 200
        return response
```

Даний клас реалізує логіку видалення проекту завдань за його id. Після вибору необхідного проекту та натискання кнопки «Видалити» користувачу буде виведено вікно в підтвердженням дії, після чого проект буде видалено з бази даних.

Логіку взаємодії користувача з проектами завдань реалізовано, наразі перейти до реалізації логіки взаємодії безпосередньо з завданнями.

Першим реалізуємо клас для взаємодії з моделлю створення завдань та отримання інформації щодо поточних завдань (див. Додаток А: Views.py ст.59-60).

Даний клас реалізує логіку відображення інформації щодо актуальних завдань та створення нових завдань у проектах, які було створено поточним користувачем. Функція get після перевірки користувача і підтвердження, що його авторизовано робить запит щодо інформації про поточного користувача та інформації щодо активних завдань у базі даних додатку. Далі ці дані

надсилаються на сторінку запиту tasks.html, де будуть використані для формування інформації щодо поточних завдань.

Функція post відповідає за безпосередньо логіку взаємодії з моделлю створення нових завдань. Після виклику та перевірки користувача і підтвердження, що його авторизовано – функція отримує дані щодо нового завдання методом POST, а саме назву проекту, опис, виконавця, статус та час дедлайну. Наступним кроком функція робить запит до моделі для безпосередньо збереження нового завдання у базі даних та у випадку успіху перенаправляє користувача на сторінку взаємодії з поточними завданнями.

Коли логіку створення завдань реалізовано, необхідно надати користувачеві можливість адмініструвати поточні завдання, а саме змінювати статус, час дедлайну тощо. Реалізуємо для цього наступну логіку (див. Додаток А: Views.py ст.59-61).

Даний клас реалізує логіку адміністрування поточного завдання користувачем, що створив його.

Після вибору необхідного проекту та натискання кнопки «редагувати» , буде викликано функцію post, яка виконає перевірку користувача і підтвердження, що його авторизовано. У разі успішного підтвердження за запитом буде отримано інформацію щодо поточного користувача та типу редагування методом POST. Далі функція розпізнає яку саме дію хоче виконати користувач та в залежності від цього отримає параметри зміни, а саме id завдання, новий статус, новий час дедлайну тощо. Наступним кроком функція виконає перевірку, чи є даний користувач власником даного завдання, та у випадку підтвердження виконає взаємодію з відповідною моделлю. У зворотньому випадку на інтерфейс користувача буде виведено повідомлення з помилкою.

Логіку взаємодії з завданнями реалізовано,останнім кроком реалізуємо логіку взаємодії зі створенням звітності щодо виконання завдань поточного користувача (див. Додаток А: Views.py ст.57-59).

Даний клас реалізує логіку відображення загальної інформації щодо поточного користувача для використання у шаблоні звіту. Функція get після перевірки користувача і підтвердження, що його авторизовано робить запит щодо інформації про поточного користувача та інформації про нього з бази даних додатку. Функція робить запит на отримання даних щодо усіх проектів користувача, завдань що були створені їм, статусу їх виконання, статусу виконання завдань безпосередньо самим користувачем. Далі усі отримані дані перенаправляються до сторінки report.html для формування звіту.

3.6 Програмна реалізація маршрутів

URLs (Uniform Resource Locators) у Django - це механізм маршрутизації запитів веб-додатку, який визначає, який код буде виконано для кожної конкретної URL-адреси.

Кожен URL Django має свій унікальний ідентифікатор, званий URL-шаблоном. URL-шаблон - це рядок, який визначає формат URL-адреси, а також пов'язаний з ним обробник (view) та його параметри.

Унікальність URL-адрес Django забезпечується шляхом використання так званих "namespaces" і "app_name" при визначенні URL-шаблонів. Це дозволяє уникнути конфліктів між URL-адресами, визначеними в різних програмах, і гарантує, що кожна URL-адреса має унікальний ідентифікатор.

Реалізуємо маршрутизацію для системи автентифікації користувача:

```
urlpatterns = [
    path('', index, name='index'),
    path('signIn', SignIn.as_view(), name="signIn"),
    path('signUp', SignUp.as_view(), name='signUp'),
```

Також реалізуємо маршрутизацію для безпосередньо системи взаємодії з завданнями:

```
urlpatterns = [
    path('', Projects.as_view(), name='boards'),
    path('<id>', Tasks.as_view(), name='tasks'),
    path('<id>/delete', MangeProject.as_view()),
    path('<id>/task', ManegeTasks.as_view())
]
```

3.7 Програмна реалізація сторінок додатку

Templates (шаблони) Django - це файли, які визначають структуру і зовнішній вигляд веб-сторінок, які будуть відображатись на сторінці користувача. Шаблони використовуються разом з уявленнями (views), щоб створювати динамічні веб-сторінки, які можуть бути змінені залежно від даних, отриманих з бази даних чи інших джерел.

Унікальність шаблонів Django забезпечується шляхом використання ієрархії успадкування. Шаблони можуть бути успадковані від інших шаблонів, що дозволяє керувати загальними елементами веб-сторінок, такими як шапка та підвал, з одного місця. Це спрощує підтримку та зміну зовнішнього вигляду сайту.

Крім того, Django також надає можливість визначення унікальних шаблонів, використовуючи тег {% extends %}, який вказує, який шаблон є основним (base template), та тег {% block %}, який дозволяє визначити блоки, які можна замінювати у наслідуваних шаблонах. Django шаблони підтримують спадкування, що дозволяє створювати базові шаблони зі спільними елементами та успадковувати їх у більш конкретних шаблонах. Це спрощує управління та модифікацію шаблонів та сприяє повторному використанню коду.

Таким чином, використання наслідування та блоків у шаблонах дозволяє створювати унікальні веб-сторінки, які можуть бути багаторазово використані в різних частинах сайту, що підвищує ефективність розробки та підтримки веб-додатків.

3.7.1 Програмна реалізація сторінки авторизації

Першою реалізуємо сторінку авторизації користувача. На цій сторінці була розміщена форма для авторизації у додатку. Введені у форму дані будуть перевірені на співпадання з базою даних користувачів , та якщо вони співпадають , користувача буде переміщено до головної сторінки додатку.

На сторінці реалізується підключення до бази даних, створення форми з полями імені та паролю та перевірку їх на наявність у базі даних сайту. Код сторінки наведено далі:

```
<div class="d-flex flex-column-fluid flex-column flex-center">
    <!--begin::Signin-->
    <div class="login-form login-signin py-11">
        <!--begin::Form-->
        <form class="form" novalidate="novalidate"
id="kt_login_signin_form"
              method="post">{% csrf_token %}
```

У даній частині коду реалізовано ініціювання форми авторизації та отримання csrf токену. CSRF (Cross-Site Request Forgery) токен Django - це механізм захисту від атаки типу CSRF, яка може виникнути, коли зловмисник відправляє підроблений запит від імені авторизованого користувача на сайт, де цей користувач зареєстрований.

Django CSRF-токен є унікальним рядком, який генерується щоразу, коли користувач відкриває сторінку з формою. Токен потім додається в приховане поле форми і cookie-файл користувача. Коли форма відправляється на сервер, Django порівнює значення CSRF-токена в файлі cookie користувача і в прихованому полі форми. Якщо значення не збігаються, запит відхиляється (див. Додаток А: Auth.html ст.64).

У даній частині коду реалізовано форму авторизації користувача у додатку. Загальний вигляд сторінки авторизації наведено на рисунку 3.2.

Рисунок 3.3 – Загальний вигляд сторінки авторизації

3.7.2 Програмна реалізація сторінки реєстрації

Далі реалізуємо сторінку реєстрації користувача. На цій сторінці була розміщена форма для реєстрації у додатку. Реєстрація обов'язкова для подальшого використання сервісу.

Логіка сторінки відповідає за перевірку даних з реєстраційної форми на валідність та якщо вони валідні – відправлення даних до бази даних користувачів. Код сторінки наведено в Додатку А: Auth.html ст.64.

У даній частині коду реалізовано форму реєстрації користувача у додатку. Загальний вигляд сторінки авторизації наведено на рисунку 3.3.

Рисунок 3.4 – Загальний вигляд сторінки реєстрації

3.7.3 Програмна реалізація сторінки проектів

На сторінці проектів було розміщено блоки з інформацією щодо активних проектів завдань поточного користувача. Інформація щодо проектів знаходиться у базі даних та витягується на сторінку за допомогою моделі, що було реалізовано раніше. Код сторінки наведено далі:

```
<div class="topbar-item">
                        <div class="btn btn-icon btn-hover-
transparent-white d-flex align-items-center btn-lg px-md-2 w-md-
auto"
                             id="kt_quick_user_toggle">
                            <span class="text-white opacity-70
font-weight-bold font-size-base d-none d-md-inline mr-1">Ласкаво
просимо,</span>
                            <span class="text-white opacity-90
font-weight-bolder font-size-base d-none d-md-inline mr-
4">{{ user.username }}</span>
                            <span class="symbol symbol-35">
    <span class="symbol-label text-white font-size-h5 font-
weight-bold bg-white-o-30">{{ first }}</span>
```

У даній частині коду реалізовано навігаційне меню з можливістю швидкого доступу до профілю користувача. Меню відображає логін користувача та при натисканні виводить на інтерфейс меню швидкого доступу (див. Додаток А: Tasks.html ст.64-67).

У даній частині коду реалізовано блоки з інформацією щодо поточних проектів. У кожному блоці відображено назву проекту, його постер, опис, деталі, виконавців та кількість поточних завдань у проекті (див. Додаток А: Projects.html ст.66-79).

У даній частині коду реалізовано меню швидкого доступу до сторінок та невелика інформація щодо користувача. Загальний вигляд сторінки проектів завдань наведено на рисунку 3.4 (див. Додаток В ст. 96: Рисунок 3.5 –Загальний вигляд сторінки проектів завдань)

3.7.4 Програмна реалізація сторінки завдань

На сторінці завдань було розміщено блоки з інформацією щодо активних завдань у обраному проекті. Інформація щодо завдань знаходиться у базі даних та витягується на сторінку за допомогою моделі, що було реалізовано раніше.Користувач може адмініструвати статус завдань, виконавців та строки виконання за допомогою спеціальної панелі. Код сторінки наведено в Додатку А: Tasks.html ст. 85.

Дана частина коду відповідає за відображення блоків зі статусами виконання завдань. У блоках статусів відображено поточні завдання та їх виконавці. Власник проекту може адмініструвати завдання всередині блоків, змінюючи їх статус (див. Додаток А: Tasks.html ст. 85-86.)

Дана частина коду відповідає за відображення форми для додавання нового завдання до поточного проекту. Форма містить поля назви, опису, дедлайну та вибору виконавців зі списку зареєстрованих у додатку користувачів. Інформація щодо завдань знаходиться у базі даних та завантажується за допомогою моделі, реалізованої раніше. (див. Додаток А: Tasks.html ст. 86-87.)

Дана частина коду реалізує календар, на якому зображено графік дедлайнів виконання активних завдань. Інформація щодо дедлайнів знаходиться у базі даних та завантажується за допомогою моделі, реалізованої раніше. Загальний вигляд сторінки адміністрування завдань наведено на рисунку 3.5 (див Додаток В ст. 98: Рисунок 3.6 – Загальний вигляд сторінки адміністрування завдань).

3.7.5 Програмна реалізація сторінки формування звіту

На сторінці формування звіту було розміщено інформацію щодо поточного користувача. Інформація знаходиться у базі даних та витягується

на сторінку за допомогою моделі, що було реалізовано раніше. На сторінці у вигляді звіту буде відображено усі проекти та завдання, в роботі над якими є поточний користувач, статус їх виконання та загальна інформація щодо них. Код сторінки наведено в Додатку А: Report.html ст.80-84.

Дана частина коду відповідає за формування звіту, який можна завантажити для розпечатування, натиснувши на відповідну кнопку «Завантажити звіт» знизу сторінки. Загальний вигляд сторінки формування звіту щодо поточних завдань наведено на рисунку 3.7 (див. Додаток В ст. 100: Загальний вигляд сторінки формування звіту щодо поточних завдань).

3.8 Програмна реалізація обробки помилок

Обробка помилок у Django - це процес обробки та відображення помилок, які можуть виникнути під час роботи веб-програми. Django має механізми для обробки та відображення різних видів помилок, таких як помилки HTTP, помилки бази даних, помилки шаблонів та інші.

Унікальність обробки помилок у Django полягає у використанні спеціальних класів та функцій для кожного типу помилок, що дозволяє обробляти помилки більш гнучко та точно.

Для реалізованого веб – додатку буде доречно реалізувати обробку помилок 404 та 500.

Помилка 404, також відома як "сторінка не знайдена", є помилкою HTTP, яка виникає, коли веб-сервер не може знайти запитуваний ресурс. Ця помилка виникає, коли користувач запитує сторінку або файл, який не існує на сервері або з якоїсь причини недоступний.

Програмна реалізація обробки помилки 404 наведена далі:

```
<div class="d-flex flex-column flex-root">
    <div class="error error-3 d-flex flex-row-fluid bgi-size-cover bgi-
position-center"
         style="background-image: url({% static '/media/error/bg3.jpg' %});">
```

```
        <div class="px-10 px-md-30 py-10 py-md-0 d-flex flex-column justify-
content-md-center">
            <h1 class="error-title text-stroke text-transparent">404</h1>
            <p class="display-4 font-weight-boldest text-white mb-12">Чому ви
тут ?</p>
            <p class="font-size-h1 font-weight-boldest text-dark-75">Вибачте,
ми не можемо знайти сторінку, яку ви шукаєте.</p>
            <p class="font-size-h4 line-height-md">У введеній URL-адресі або
на сторінці, на якій ви перебуваєте, може бути орфографічна помилка, або
сторінка може більше не існувати.</p>
```

Помилка 500 також відома як "внутрішня помилка сервера", є помилкою HTTP, яка виникає, коли на сервері відбувається внутрішня помилка, яка не дозволяє серверу обробити запит.

Код 500 - це стандартний код відповіді HTTP, який повертається сервером у разі виникнення цієї помилки. Веб-сервер передає цей код браузеру клієнта, який потім відображає повідомлення про помилку "500 Внутрішня помилка сервера" на екрані користувача.

Помилки 500 можуть виникати з різних причин, наприклад, через помилки в програмному коді серверної частини програми, неправильні установки сервера або неполадки з базою даних. Ця помилка може бути також спричинена перевантаженням сервера або відсутністю ресурсів на сервері.

Для вирішення проблеми з помилкою 500 необхідно проводити ретельний аналіз журналів сервера та переглянути код програми, щоб визначити причину помилки. У деяких випадках причина помилки може бути пов'язана з недоступністю бази даних або неправильною конфігурацією сервера. Програмна реалізація обробки помилки 500 наведена далі:

```
<div class="d-flex flex-column flex-root">
    <div class="error error-6 d-flex flex-row-fluid bgi-size-cover bgi-
position-center"
         style="background-image: url({% static '/media/error/bg6.jpg' %});">
        <div class="d-flex flex-column flex-row-fluid text-center">
            <h1 class="error-title font-weight-boldest text-white mb-12"
style="margin-top: 12rem;">Oops...</h1>
            <p class="display-4 font-weight-bold text-white">Схоже, щось
пішло не так. Ми працюємо над цим :(</p>
```

3.9 Висновок до розділу

У даному розділі було проведено практичну реалізацію проектованого програмного продукту.

Вибрали ОС Windows Server Standard Edition для серверу. Вона має деякі характеристики та функції, які роблять її відповідною для потреб бізнесу та підприємств. Windows Server Standard Edition підтримує різні робочі навантаження, включаючи файлові сервери, друковані сервери, веб-сервери, бази даних та багато іншого. Вона надає інструменти та функціональність для ефективного розгортання та управління різними типами серверів.

Реалізація маршрутів в Django дозволяє керувати тим, як URL-адреси відображаються та яким переглядам передаються запити користувачів. Це дозволяє організувати структуру веб-додатку та забезпечити правильне маршрутизацію запитів.

Обробка помилок дозволяє забезпечити більш зрозумілу та дружню інтеракцію з користувачем. Замість отримання загальних повідомлень про помилку, ви можете надати користувачам конкретні та зрозумілі повідомлення про проблему, що сталася, а також пропонувати варіанти виправлення. Також обробка помилок допомагає зберігати репутацію вашого веб-додатку. Швидка та ефективна реакція на помилки, виправлення їх та надання користувачам належних пояснень сприяє позитивному враженню та задоволеності від використання сервісу.

Проведено обґрунтування вибору технічних засобів та обґрунтування вибору ОС та протоколу обміну даними. Проведено заходи захисту від несанкціонованого доступу до системи. Обрано та описано архітектуру проектованого програмного продукту. Для реалізації функціоналу додатку було проведено програмну реалізацію моделей завдань та логіки проекту.

ВИСНОВКИ

Під час виконання дипломної роботи була проведена робота з вивчення технологій розробки веб - додатків з використанням технологій Python, Django,SQLite, HTML, CSS. Було проведено дослідження аналогічних веб-сервісів для розвитку власного веб-сервісу. Після огляду цих систем були аналізовані переваги та недоліки, що надають можливість знайти нішу або розробити унікальні функції, які відрізнятимуть сервіс. В процесі аналізу виявили сильні та слабкі сторони інших веб-сервісів, що надало нам ідеї для покращення нашого власного веб-сервісу. Ми також можемо навчитися від їхніх успіхів і помилок, вдосконалюючи дизайн, функціонал або впроваджуючи нові ідеї. Крім того, ми визначили вимоги до системи серверу, які є важливими для ефективної роботи нашого веб-сервісу. Врахування цих вимог допомагає забезпечити оптимальну продуктивність, безпеку та доступність для користувачів. Наявність достатньої кількості оперативної пам'яті дозволяє серверу ефективно обробляти багато одночасних запитів та зберігати кешовані дані для поліпшення продуктивності. Вимоги до сховища даних залежать від типу та обсягу даних, що зберігаються в нашому веб-сервісі. Нам потрібно обрати підходящу базу даних або інше сховище, яке відповідає нашим потребам щодо швидкості, масштабованості та безпеки даних. За підсумками виконання роботи було розроблено веб-додаток для керування та організації списку завдань. Основна мета такого додатку - надати можливість користувачам зберігати, організовувати та керувати своїми завданнями зручним способом.

Основні функціональні можливості, які можна використовувати в такому веб-додатку:

1. Реєстрація та аутентифікація користувачів: Додаток може дозволити користувачам створювати облікові записи, входити до системи та мати персоналізований доступ до своїх списків завдань.

2. Створення завдань: Користувачі можуть створювати нові завдання, вказувати їх заголовки, описи, терміни виконання, пріоритети та інші атрибути.

3. Організація списку завдань: Додаток може надати функції для організації завдань у вигляді списків, міток або категорій. Користувачі можуть групувати та сортувати свої завдання для зручності.

4. Оновлення та видалення завдань: Користувачі повинні мати можливість оновлювати та видаляти свої завдання, встановлювати нові терміни виконання або змінювати інші атрибути.

Створена система дозволяє користувачу використовувати функціонал додатку за допомогою інтерфейсу користувача, керувати вмістом додатку та бази даних, змінювати, видаляти та додавати інформацію щодо завдань, які актуальні для поточного користувача. При створенні системи були враховані всі плюси та мінуси систем-аналогів, вибраних для порівняння.

Було проведено дослідження доступних варіантів для вирішення поставленої задачі, зроблено порівняльний аналіз технологій розробки та СУБД, що можуть бути використані для проектованого веб-додатку. На основі цього аналізу було зроблено вибір необхідних інструментів для реалізації завдання. Оскільки порушень не виявлено, всі функції відповідають вимогам, тому веб-ресурс готовий до повної функціональної роботі, тобто система готова для впровадження, та може бути використана для автоматизації процесів керування та організації списку завдань. Цей веб-додаток може бути корисним інструментом для індивідуального використання або для колективної співпраці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Python та SQLite Web Development (4th Edition)", Luke Welling, Laura Thomson 848 стор., 3 іл.; ISBN 978-5-8459-1574-0, 978-0-672-32916-6.

2. Веб Database Application », 2nd Edition By David Lane, Hugh E. Вільямс. © O'Reilly, May 2004. ISBN: 0-596-00543-1.

3. CMS List. Огляд cms. Сайт про системи керування сайтом. [Електронний ресурс] Режим доступу: http://www.cmslist.ua

4. SQL 4 – Строкові функції [Електронний ресурс] Режим доступу: http://www.codenet.ua/db/mysql/mystring4

5. Web-розробка [Електронний ресурс] Режим доступу: http://fcit.tneu.org/web-rozrobka/

6. Бази даних: розробка та управління: Книга / Хансем Г., Хансем Дж. - М: Біном, 2010. - 704 с.

7. Технології розробки та тестування програм [Електронний ресурс]. Режим доступу: http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php

8. Види тестування ПЗ [Електронний ресурс]. Режим доступу: http://qalearning.com.ua/theory/lectures/material/testing-types-functional/

9. Перенесення файлів із localhost на сервер [Електронний ресурс]. Режим доступу: http://joomlaportal.ua/faq/installation-and-update/51-perenos-sajta-slocalhost-na-server

10. Архітектура клієнт-сервер [Електронний ресурс] Режим доступу: http://inter.ptngu.com/

11. hrliga.com [Електронний ресурс]. – Режим доступу: https://hrliga.com/index.php?module=profession&op=view&id=1676

12. lifewire.com [Електронний ресурс]. – Режим доступу: https:// ww.lifewire.com/what-is-a-web-application-3486637

13. en.yeeply.com [Електронний ресурс]. – Режим доступу: https://en.yeeply.com/blog/advantages-and-disadvantages-of-webappdevelopment/

14. svitla.com [Електронний ресурс]. – Режим доступу: https://svitla.com/blog/web-application-architecture

15. slideshare.net [Електронний ресурс]. – Режим доступу: https://www.slideshare.net/ssusere5f319/web-66422826

16. wikipedia.org [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Доменна_система_імен

ДОДАТОК А

Лістинг програми:

Models.py

```
from task_manager.models import Project, Task, User


class ProjectInfo:
    def __init__(self, project):
        self.project = project
        self.name = project.name
        all_tasks = project.task_set.all()
        self.users = []

        self.tasks = len(all_tasks)
        for id in project.get_members():
            user = User.objects.filter(id=id).first()
            self.users.append(user)

        self.t = 0
        self.d = 0
        self.i = 0
        self.o = 0

        for task in all_tasks:
            if task.status == 'T':
                self.t = self.t + 1

            elif task.status == 'D':
                self.d = self.d + 1

            elif task.status == 'I':
                self.i = self.i + 1

            elif task.status == 'O':
                self.o = self.o + 1

        all_tasks = self.t + self.d + self.i + self.o

        if all_tasks != 0:
            self.progress = int((self.o * 100) / all_tasks)
        else:
            self.progress = 0


class UserInfo:
    def __init__(self, user):
        self.user = user
        self.todo = 0
        self.doing = 0
        self.done = 0
        self.progress = 0

    def analyze_project(self, project):
        all_tasks = project.task_set.all()
        for task in all_tasks:
            if task.assigned_to == self.user:
```

```
                if task.status == 'T':
                    self.todo = self.todo + 1

                elif task.status == 'D':
                    self.doing = self.doing + 1

                elif task.status == 'I':
                    self.doing = self.doing + 1

                elif task.status == 'O':
                    self.done = self.done + 1

        all_tasks = self.todo + self.doing + self.done

        if all_tasks != 0:
            self.progress = int((self.done * 100) / all_tasks)
        else:
            self.progress = 0


class UserInProject:
    def __init__(self, user, project):
        self.u_info = UserInfo(user)
        self.name = project.name
        self.u_info.analyze_project(project)
```

Views.py

```
from django.shortcuts import render, redirect
from django.views import View
from datetime import datetime
from task_manager.models import Project
from .models import ProjectInfo, UserInfo, UserInProject


class Report(View):
    def get(self, request):
        if not request.user.is_authenticated:
            return redirect('signIn')

        user = request.user
        projects = Project.objects.all()
        p_info_list = []
        u_info = UserInfo(user)
        user_in_projects = []

        for p in projects:
            if p.owner == user or user.id in p.get_members():
                p_info = ProjectInfo(p)
                u_info.analyze_project(p)
                p_info_list.append(p_info)
                user_in_projects.append(UserInProject(user, p))

        data = {"user": user,
                "first": user.username[0],
                "p_info": p_info_list,
                "u_info": u_info,
                "u_in_p": user_in_projects,
                'time': datetime.today()
```

```
                }
        return render(request, 'report.html', data)
```

Models.py

```
from django.db import models
from django.contrib.auth.models import User
import json


class Project(models.Model):
    name = models.CharField(max_length=50)
    description = models.CharField(max_length=200)
    details = models.TextField()
    owner = models.ForeignKey(User, on_delete=models.CASCADE)
    members = models.CharField(max_length=500)
    profile_photo = models.CharField(max_length=200,
default='/static/media/project-logos/1.png')

    def get_members(self):
        return json.loads(self.members)


class Task(models.Model):
    name = models.CharField(max_length=50)
    description = models.CharField(max_length=200)
    assigned_to = models.ForeignKey(User, on_delete=models.CASCADE)
    status_choices = (
        ('T', 'To Do'),
        ('D', 'Doing'),
        ('I', 'In Test'),
        ('O', 'Done'),
        ('B', 'Blocked'),
        ('L', 'Deleted')
    )
    status = models.CharField(max_length=1, choices=status_choices)
    start_time = models.DateField(null=True)
    end_time = models.DateField()
    project = models.ForeignKey(Project, on_delete=models.CASCADE)
```

Views.py

```
import datetime
import json
import random

from django.contrib.auth.models import User
from django.db.models import Q
from django.http import JsonResponse
from django.shortcuts import render, redirect
from django.views import View

from reports.models import ProjectInfo
from .models import Task, Project


class Projects(View):
    def get(self, request):
        if not request.user.is_authenticated:
```

```
            return redirect('signIn')

        user = request.user
        projects = Project.objects.all()
        list = []

        for p in projects:
            if p.owner == user or user.id in p.get_members():
                list.append(ProjectInfo(p))

        data = {"user": user,
                "first": user.username[0],
                "other users": User.objects.filter(~Q(id=user.id)).all(),
                "projects": list,
                }
        return render(request, 'projects.html', data)

    def post(self, request):
        if not request.user.is_authenticated:
            return redirect('signIn')

        name = request.POST['name']
        description = request.POST['desc']
        details = request.POST['details']
        owner = request.user
        user_ids = request.POST.getlist('users', [])

        ids = []
        for id in user_ids:
            ids.append(int(id))

        n = random.randint(1, 7)
        pf_url = f'/media/project-logos/{n}.png'

        proj = Project.objects.create(name=name, description=description,
details=details, owner=owner,
                                      members=json.dumps(ids),
profile_photo=pf_url)
        proj.save()

        return redirect('boards')


class MangeProject(View):
    def post(self, request, id):
        Project.objects.filter(id=id).delete()

        response = JsonResponse({"message": "OK"})
        response.status_code = 200
        return response


class Tasks(View):
    def get(self, request, id):
        if not request.user.is_authenticated:
            return redirect("signIn")

        proj = Project.objects.filter(id=id).first()
        user = request.user
        users = User.objects.filter(Q(id__in=proj.get_members()) |
Q(id=proj.owner.id))
```

```
        data = {"user": user,
                "first": user.username[0],
                "other_users": users,
                "tasks": proj.task_set.all(),
                'proj': proj,
                "can_add": user == proj.owner
                }
        return render(request, 'tasks.html', data)

    def post(self, request, id):
        if not request.user.is authenticated:
            return redirect('signIn')

        name = request.POST['name']
        description = request.POST['desc']
        assigned_to = request.POST['users']
        status = 'T'
        end_time = request.POST['date']

        task = Task(name=name, description=description,
assigned_to_id=assigned_to, status=status,
                    end_time=end_time, project_id=id)
        task.save()

        return redirect('tasks', id=id)


class ManegeTasks(View):
    def post(self, request, id):
        if not request.user.is authenticated:
            response = JsonResponse({"error": "Недійсний користувач"})
            response.status_code = 403
            return response

        user = request.user

        type = request.POST['type']
        if type == 'edit_status':
            task id = request.POST['task id']
            status = request.POST['board_id']

            task = Task.objects.filter(id=task_id).first()

            if status in ['O', 'B', 'L'] or task.status in ['O', 'B', 'L']:
                if user == task.project.owner:
                    task.status = status
                    task.save()

                else:
                    response = JsonResponse({"error": "Ви на маєте доступу на
цю дію"})
                    response.status_code = 403
                    return response
            else:
                if user == task.assigned_to or user == task.project.owner:
                    task.status = status
                    if status == 'D':
                        task.start_time = datetime.datetime.today().date()
                    task.save()
                else:
                    response = JsonResponse({"error": "Ви на маєте доступу на
```

```
цю дію"})
                    response.status_code = 403
                    return response

            response = JsonResponse({"message": "OK"})
            response.status_code = 200
            return response

        if type == 'edit end time':

            task_id = request.POST['task_id']
            end_time = request.POST['new_end_time']

            task = Task.objects.filter(id=task_id).first()

            if user == task.project.owner:
                task.end_time = end_time
                task.save()

                response = JsonResponse({"message": "OK"})
                response.status_code = 200
                return response

            else:
                response = JsonResponse({"error": "Ви на маєте доступу на цю
дію"})
                response.status_code = 403
                return response
```

Auth.html

```
{% extends 'base.html' %}
{% load static %}

{% block css %}
    <link href="{% static '/css/pages/login/login-2.css' %}" rel="stylesheet"
type="text/css"/>
{% endblock %}

{% block title %}Task Manager Login{% endblock %}


{% block content %}

    <!--begin::Main-->
    <div class="d-flex flex-column flex-root">
        <!--begin::Login-->
        <div class="login login-2 login-signin-on d-flex flex-column flex-lg-
row flex-column-fluid bg-white"
             id="kt_login">
            <!--begin::Aside-->
            <div class="login-aside order-2 order-lg-1 d-flex flex-row-auto
position-relative overflow-hidden">
                <!--begin: Aside Container-->
                <div class="d-flex flex-column-fluid flex-column justify-
content-between py-9 px-7 py-lg-13 px-lg-35">
                    <!--begin::Aside body-->
                    <div class="d-flex flex-column-fluid flex-column flex-
center">
                        <!--begin::Signin-->
```

```
                        <div class="login-form login-signin py-11">
                            <!--begin::Form-->
                            <form class="form" novalidate="novalidate"
id="kt_login_signin_form"
                                  method="post">{% csrf token %}
                                <!--begin::Title-->
                                <div class="text-center pb-8">
                                    <h2 class="font-weight-bolder text-dark
font-size-h2 font-size-h1-lg">Вхід</h2>
                                    <span class="text-muted font-weight-bold
font-size-h4">чи
                                <a href="" class="text-primary font-weight-
bolder"

id="kt_login_signup">Зареєструватись</a></span>
                                </div>
                                <!--end::Title-->
                                <!--begin::Form group-->
                                <div class="form-group">
                                    <label class="font-size-h6 font-weight-
bolder text-dark">Логін</label>
                                    <input id="in-email"
                                           class="form-control form-control-
solid h-auto py-7 px-6 rounded-lg"
                                           type="text"
                                           name="username"
autocomplete="off"/>
                                </div>
                                <!--end::Form group-->
                                <!--begin::Form group-->
                                <div class="form-group">
                                    <div class="d-flex justify-content-
between mt-n5">
                                        <label class="font-size-h6 font-
weight-bolder text-dark pt-5">Пароль</label>
                                        <a href="javascript:;"
                                           class="text-primary font-size-h6
font-weight-bolder text-hover-primary pt-5"
                                           id="kt login forgot">Забули
пароль ?</a>
                                    </div>
                                    <input id="in-password"
                                           class="form-control form-control-
solid h-auto py-7 px-6 rounded-lg"
                                           type="password" name="password"
autocomplete="off"/>
                                </div>
                                <!--end::Form group-->
                                <!--begin::Action-->
                                <div class="text-center pt-2">
                                    <button id="kt_login_signin_submit"
                                            class="btn btn-dark font-weight-
bolder font-size-h6 px-8 py-4 my-3">Увійти
                                    </button>
                                </div>
                                <!--end::Action-->
                            </form>
                            <!--end::Form-->
                        </div>
                        <!--end::Signin-->
                        <!--begin::Signup-->
```

```
                        <div class="login-form login-signup pt-11">
                            <!--begin::Form-->
                            <form class="form" novalidate="novalidate"
id="kt_login_signup_form"
                                  method="post">{% csrf token %}
                                <!--begin::Title-->
                                <div class="text-center pb-8">
                                    <h2 class="font-weight-bolder text-dark
font-size-h2 font-size-h1-lg">Зареєструватись</h2>
                                    <p class="text-muted font-weight-bold
font-size-h4">Введіть свої дані для створення
                                        вашого
                                        акаунту</p>
                                </div>
                                <!--end::Title-->
                                <!--begin::Form group-->
                                <div class="form-group">
                                    <input class="form-control form-control-
solid h-auto py-7 px-6 rounded-lg font-size-h6"
                                           type="text" placeholder="Логін"
name="username" autocomplete="off"/>
                                </div>
                                <!--end::Form group-->
                                <!--begin::Form group-->
                                <div class="form-group">
                                    <input class="form-control form-control-
solid h-auto py-7 px-6 rounded-lg font-size-h6"
                                           type="email" placeholder="Email"
name="email" autocomplete="off"/>
                                </div>
                                <!--end::Form group-->
                                <!--begin::Form group-->
                                <div class="form-group">
                                    <input class="form-control form-control-
solid h-auto py-7 px-6 rounded-lg font-size-h6"
                                           type="password"
placeholder="Пароль" name="password" autocomplete="off"/>
                                </div>
                                <!--end::Form group-->
                                <!--begin::Form group-->
                                <div class="form-group">
                                    <input class="form-control form-control-
solid h-auto py-7 px-6 rounded-lg font-size-h6"
                                           type="password"
placeholder="Підтвердіть пароль" name="cpassword"
                                           autocomplete="off"/>
                                </div>
                                <!--end::Form group-->
                                <!--begin::Form group-->
                                <div class="form-group d-flex flex-wrap flex-
center pb-lg-0 pb-3">
                                    <button type="button"
id="kt_login_signup_submit"
                                            class="btn btn-primary font-
weight-bolder font-size-h6 px-8 py-4 my-3 mx-4">
                                        Зареєструватись
                                    </button>
                                    <button type="button"
id="kt_login_signup_cancel"
                                            class="btn btn-light-primary
font-weight-bolder font-size-h6 px-8 py-4 my-3 mx-4">
```

```
                                        Скасувати
                                    </button>
                                </div>
                                <!--end::Form group-->
                            </form>
                            <!--end::Form-->
                        </div>
                        <!--end::Signup-->
                        <!--begin::Forgot-->
                        <div class="login-form login-forgot pt-11">
                            <!--begin::Form-->
                            <form class="form" novalidate="novalidate"
id="kt login forgot form">
                                <!--begin::Title-->
                                <div class="text-center pb-8">
                                    <h2 class="font-weight-bolder text-dark
font-size-h2 font-size-h1-lg">Забули
                                        пароль
                                    </h2>
                                    <p class="text-muted font-weight-bold
font-size-h4">Введіть свою електронну адресу, щоб відновити свій
                                        пароль</p>
                                </div>
                                <!--end::Title-->
                                <!--begin::Form group-->
                                <div class="form-group">
                                    <input class="form-control form-control-
solid h-auto py-7 px-6 rounded-lg font-size-h6"
                                           type="email" placeholder="Email"
name="email" autocomplete="off"/>
                                </div>
                                <!--end::Form group-->
                                <!--begin::Form group-->
                                <div class="form-group d-flex flex-wrap flex-
center pb-lg-0 pb-3">
                                    <button type="button"
id="kt_login_forgot_submit"
                                            class="btn btn-primary font-
weight-bolder font-size-h6 px-8 py-4 my-3 mx-4">
                                        Готово
                                    </button>
                                    <button type="button"
id="kt login forgot cancel"
                                            class="btn btn-light-primary
font-weight-bolder font-size-h6 px-8 py-4 my-3 mx-4">
                                        Скасувати
                                    </button>
                                </div>
                                <!--end::Form group-->
                            </form>
                            <!--end::Form-->
                        </div>
                        <!--end::Forgot-->
                    </div>
                    <!--end::Aside body-->
                    <!--end: Aside footer for desktop-->
                </div>
                <!--end: Aside Container-->
            </div>
            <!--begin::Aside-->
            <!--begin::Content-->
```

```
            <div class="content order-1 order-lg-2 d-flex flex-column w-100
pb-0" style="background-color: #B1DCED;">
                <!--begin::Title-->
                <div class="d-flex flex-column justify-content-center text-
center pt-md-5 pt-sm-5 px-lg-0 pt-5 px-7">
                    <h3 class="display4 font-weight-bolder my-7 text-dark"
style="color: #986923;">Diploma Task Manager</h3>
                    <p class="font-weight-bolder font-size-h2-md font-size-lg
text-dark opacity-70">
                        For Diploma <Work></Work>
                    </p>
                </div>
                <!--end::Title-->
                <!--begin::Image-->
                <div class="content-img d-flex flex-row-fluid bgi-no-repeat
bgi-position-y-bottom bgi-position-x-center"
                     style="background-image: url({% static
'/media/svg/illustrations/login-visual-2.svg' %});"></div>
                <!--end::Image-->
            </div>
            <!--end::Content-->
        </div>
        <!--end::Login-->
    </div>
    <!--end::Main-->

{% endblock %}


{% block js %}
    <script src="{% static '/js/pages/custom/login/login-
general.js' %}"></script>
{% endblock %}
```

Projects.html

```
{% extends 'base.html' %}
{% load static %}
{% block title %}Boards{% endblock %}
{% block content %}

    <!--begin::Header Mobile-->
    <div id="kt_header_mobile" class="header-mobile">
        <!--begin::Toolbar-->
        <div class="d-flex align-items-center">
            <button class="btn p-0 burger-icon burger-icon-left ml-4"
id="kt_header_mobile_toggle">
                <span></span>
            </button>
            <button class="btn btn-icon btn-hover-transparent-white p-0 ml-3"
id="kt_header_mobile_topbar_toggle">
                <span class="svg-icon svg-icon-xl">
                    <!--begin::Svg Icon |
path:/media/svg/icons/General/User.svg-->
                    <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" width="24px"
                         height="24px" viewBox="0 0 24 24" version="1.1">
                        <g stroke="none" stroke-width="1" fill="none" fill-
rule="evenodd">
                            <polygon points="0 0 24 0 24 24 0 24"/>
```

```
                        <path d="M12,11 C9.790861,11 8,9.209139 8,7
C8,4.790861 9.790861,3 12,3 C14.209139,3 16,4.790861 16,7 C16,9.209139
14.209139,11 12,11 Z"
                                    fill="#000000" fill-rule="nonzero"
opacity="0.3"/>
                        <path d="M3.00065168,20.1992055
C3.38825852,15.4265159 7.26191235,13 11.9833413,13 C16.7712164,13
20.7048837,15.2931929 20.9979143,20.2 C21.0095879,20.3954741 20.9979143,21
20.2466999,21 C16.541124,21 11.0347247,21 3.72750223,21 C3.47671215,21
2.97953825,20.45918 3.00065168,20.1992055 Z"
                                    fill="#000000" fill-rule="nonzero"/>
                    </g>
                </svg>
                    <!--end::Svg Icon-->
            </span>
        </button>
    </div>
    <!--end::Toolbar-->
</div>
<!--end::Header Mobile-->

<div class="d-flex flex-column flex-root">
    <!--begin::Page-->
    <div class="d-flex flex-row flex-column-fluid page">
        <!--begin::Wrapper-->
        <div class="d-flex flex-column flex-row-fluid wrapper"
id="kt_wrapper">
            <!--begin::Header-->
            <div id="kt_header" class="header header-fixed">
                <!--begin::Container-->
                <div class="container d-flex align-items-stretch justify-
content-between">
                    <!--begin::Topbar-->
                    <div class="topbar">
                        <!--begin::User-->
                        <div class="dropdown">
                            <!--begin::Toggle-->
                            <div class="topbar-item">
                                <div class="btn btn-icon btn-hover-
transparent-white d-flex align-items-center btn-lg px-md-2 w-md-auto"
                                    id="kt_quick_user_toggle">
                                    <span class="text-white opacity-70
font-weight-bold font-size-base d-none d-md-inline mr-1">Ласкаво
просимо,</span>
                                    <span class="text-white opacity-90
font-weight-bolder font-size-base d-none d-md-inline mr-
4">{{ user.username }}</span>
                                    <span class="symbol symbol-35">
                                        <span class="symbol-label text-white
font-size-h5 font-weight-bold bg-white-o-30">{{ first }}</span>
                                    </span>
                                </div>
                            </div>
                            <!--end::Toggle-->
                        </div>
                        <!--end::User-->
                    </div>
                    <!--end::Topbar-->
                </div>
                <!--end::Container-->
            </div>
```

```
                <!--end::Header-->

                {% block body %}

                    <!--begin::Content-->
                    <div class="content d-flex flex-column flex-column-fluid"
id="kt content">
                        <!--begin::Subheader-->
                        <div class="subheader py-2 py-lg-12 subheader-
transparent" id="kt_subheader">
                            <div class="container d-flex align-items-center
justify-content-between flex-wrap flex-sm-nowrap">
                                <!--begin::Info-->
                                <div class="d-flex align-items-center flex-
wrap mr-1">
                                    <!--begin::Heading-->
                                    <div class="d-flex flex-column">
                                        <!--begin::Title-->
                                        <h2 class="text-white font-weight-
bold my-2 mr-5">Ваші завдання</h2>
                                        <h5 class="text-success font-weight-
bold">Створюйте та керуйте вашими завданнями тут !</h5>
                                        <!--end::Title-->
                                    </div>
                                    <!--end::Heading-->
                                </div>
                                <!--end::Info-->
                                <!--begin::Toolbar-->
                                <div class="d-flex align-items-center">
                                    <!--begin::Button-->
                                    <a href="/report"
                                       class="btn btn-transparent-white font-
weight-bold py-3 px-6 mr-2">Звіт</a>
                                    <!--end::Button-->
                                    <!--begin::Dropdown-->
                                    <div class="dropdown dropdown-inline ml-
2" data-toggle="tooltip"
                                         title="Add New Project or Team"
                                         data-placement="top">

                                        <!-- Button  modal-->
                                        <button type="button" class="btn btn-
white font-weight-bold py-3 px-6"
                                                data-toggle="modal"
                                                data-target="#exampleModal">
                                            Новий проект
                                        </button>

                                        <!-- Modal-->
                                        <div class="modal fade"
id="exampleModal" tabindex="-1" role="dialog"
                                             aria-
labelledby="exampleModalLabel" aria-hidden="true">

                                            <div class="modal-dialog"
role="document">
                                                <div class="modal-content">
                                                    <div class="modal-
header">
                                                        <h5 class="modal-
title" id="exampleModalLabel">Нове завдання
```

```
                                                                    </h5>
                                                                    <button type="button"
class="close" data-dismiss="modal"
                                                                            aria-
label="Close">
                                                                        <i aria-
hidden="true" class="ki ki-close"></i>
                                                                    </button>
                                                                </div>
                                                                <div class="modal-body">
                                                                    <form
id="add_board_from" method="post">{% csrf_token %}
                                                                        <div class="form-
group">
                                                                            <label>Назва
                                                                                <span
class="text-danger">*</span></label>
                                                                            <input
id="iname" name="name" type="text"
class="form-control form-control-lg"
placeholder="Назва завдання">
                                                                        </div>
                                                                        <div class="form-
group">
                                                                            <label>Опис
                                                                                <span
class="text-danger">*</span></label>
                                                                            <input
id="idesc" name="desc" type="text"
class="form-control form-control-lg"
placeholder="Опис завдання">
                                                                        </div>
                                                                        <div class="form-
group">
                                                                            <label
for="Textarea">Деталі
                                                                                <span
class="text-danger">*</span></label>
                                                                            <textarea
name="details" class="form-control"
id="idetails"
rows="3"
placeholder="Деталі завдання"></textarea>
                                                                        </div>
                                                                        <div class="form-
group">
<label>Виконавці
                                                                                <span
class="text-danger">*</span></label><br>
                                                                            <select
name="users" style="width: 100%"
class="form-control form-control-lg select2"
```

```
id="kt_select2_3"

name="param" multiple="multiple">
                                                                    {% for ou
in other_users %}

<option value="{{ ou.id }}">{{ ou.username }}</option>
                                                                    {%
endfor %}
                                                                </select>
                                                            </div>
                                                            <div class="form-
group">
                                                                <button
type="button"

class="btn btn-light-primary font-weight-bold"
                                                                        data-
dismiss="modal">Скасувати
                                                                </button>
                                                                <button
id="psubmit" type="submit"

class="btn btn-primary font-weight-bold">Зберегти
                                                                    зміни
                                                                </button>
                                                            </div>
                                                        </form>
                                                    </div>

                                                </div>
                                            </div>
                                        </div>

                                    </div>
                                    <!--end::Dropdown-->
                                </div>
                                <!--end::Toolbar-->
                            </div>
                        </div>
                        <!--end::Subheader-->
                        <!--begin::Entry-->
                        <div class="d-flex flex-column-fluid">
                            <!--begin::Container-->
                            <div class="container">
                                {% for proj in projects %}
                                    {% if forloop.first %}
                                        <div class="row">{% endif %}

                                    <div class="col-xl-4">
                                        <!--begin::Card-->
                                        <div class="card card-custom gutter-b
card-stretch">

                                            <!--begin::Body-->
                                            <div class="card-body">
                                                <!--begin::Info-->
                                                <div class="d-flex align-items-
center">

                                                    <!--begin::Pic-->
                                                    <div class="flex-shrink-0 mr-
```

```
4 symbol symbol-60 symbol-circle">
                                                    <img src="{% static
proj.project.profile_photo %}"
                                                         alt="image"/>
                                                </div>
                                                <!--end::Pic-->
                                                <!--begin::Info-->
                                                <div class="d-flex flex-
column mr-auto">
                                                    <!--begin: Title-->
                                                    <div class="d-flex flex-
column mr-auto">
                                                        <a
href="/boards/{{ proj.project.id }}"
                                                           class="text-dark
text-hover-primary font-size-h4 font-weight-bolder mb-
1">{{ proj.project.name }}</a>
                                                        <span class="text-
muted font-weight-bold">{{ proj.project.description }}</span>
                                                    </div>
                                                    <!--end::Title-->
                                                </div>
                                                <!--end::Info-->
                                                <!--begin::Toolbar-->
                                                <div class="card-toolbar mb-
7">
                                                    <div class="dropdown
dropdown-inline" data-toggle="tooltip"
                                                         title="Quick
actions" data-placement="left">
                                                        <a href="#"
                                                           class="btn btn-
clean btn-hover-light-primary btn-sm btn-icon"
                                                           data-
toggle="dropdown" aria-haspopup="true"
                                                           aria-
expanded="false">
                                                            <i class="ki ki-
bold-more-hor"></i>
                                                        </a>
                                                        <div class="dropdown-
menu dropdown-menu-sm dropdown-menu-right">
                                                            <!--
begin::Navigation-->
                                                            <ul class="navi
navi-hover">
                                                                <li
class="navi-item">
                                                                    <a
href="javascript:void(0)"
id="{{ proj.project.id }}/delete"
class="navi-link aff">
                                                                <span class="navi-
icon">
                                                                    <i
class="flaticon2-delete"></i>
                                                                </span>
                                                                                <span
class="navi-text">Видалити</span>
```

```
                                                                    </a>
                                                                </li>
                                                            </ul>
                                                            <!--
end::Navigation-->
                                                        </div>
                                                    </div>
                                                </div>
                                                <!--end::Toolbar-->
                                            </div>
                                            <!--end::Info-->
                                            <!--begin::Description-->
                                            <div class="mb-10 mt-5 font-
weight-bold">{{ proj.project.details }}</div>
                                            <!--end::Description-->
                                            <!--begin::Progress-->
                                            <div class="d-flex mb-5 align-
items-cente">
                                                <span class="d-block font-
weight-bold mr-5">Прогрес</span>
                                                <div class="d-flex flex-row-
fluid align-items-center">
                                                    <div class="progress
progress-xs mt-2 mb-2 w-100">
                                                        <div class="progress-
bar bg-success" role="progressbar"
                                                             style="width:
{{ proj.progress }}%;" aria-valuenow="50"
                                                             aria-
valuemin="0"
                                                             aria-
valuemax="100"></div>
                                                    </div>
                                                    <span class="ml-3 font-
weight-bolder">{{ proj.progress }}%</span>
                                                </div>
                                            </div>
                                            <!--ebd::Progress-->
                                            <div class="d-flex flex-column
flex-lg-fill float-left mb-7">
                                                <span class="font-weight-
bolder mb-4">Учасники</span>
                                                <div class="symbol-group
symbol-hover">
                                                    {% for user in
proj.users %}
                                                        <div class="symbol
symbol-30 symbol-circle"
                                                             data-
toggle="tooltip"
                                                             title="" data-
original-title="{{ user.username }}">
                                                            <img alt="Pic"
                                                                 src="{%
static user.profile.profile_photo %}">
                                                        </div>
                                                    {% endfor %}
                                                </div>
                                            </div>

                                        </div>
```

```
                                        <!--end::Body-->
                                        <!--begin::Footer-->
                                        <div class="card-footer d-flex align-
items-center">
                                            <div class="d-flex">
                                                <div class="d-flex align-
items-center mr-7">
                                                <span class="svg-icon svg-icon-
gray-500">
                                                    <!--begin::Svg Icon |
path:/media/svg/icons/Text/Bullet-list.svg-->
                                                    <svg
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" width="24px"
height="24px" viewBox="0 0 24 24" version="1.1">
                                                        <g stroke="none" stroke-
width="1" fill="none"
                                                                                fill-
rule="evenodd">
                                                            <rect x="0" y="0"
width="24" height="24"/>
                                                            <path d="M10.5,5 L19.5,5
C20.3284271,5 21,5.67157288 21,6.5 C21,7.32842712 20.3284271,8 19.5,8 L10.5,8
C9.67157288,8 9,7.32842712 9,6.5 C9,5.67157288 9.67157288,5 10.5,5 Z M10.5,10
L19.5,10 C20.3284271,10 21,10.6715729 21,11.5 C21,12.3284271 20.3284271,13
19.5,13 L10.5,13 C9.67157288,13 9,12.3284271 9,11.5 C9,10.6715729
9.67157288,10 10.5,10 Z M10.5,15 L19.5,15 C20.3284271,15 21,15.6715729
21,16.5 C21,17.3284271 20.3284271,18 19.5,18 L10.5,18 C9.67157288,18
9,17.3284271 9,16.5 C9,15.6715729 9.67157288,15 10.5,15 Z"
fill="#000000"/>
                                                            <path d="M5.5,8
C4.67157288,8 4,7.32842712 4,6.5 C4,5.67157288 4.67157288,5 5.5,5
C6.32842712,5 7,5.67157288 7,6.5 C7,7.32842712 6.32842712,8 5.5,8 Z M5.5,13
C4.67157288,13 4,12.3284271 4,11.5 C4,10.6715729 4.67157288,10 5.5,10
C6.32842712,10 7,10.6715729 7,11.5 C7,12.3284271 6.32842712,13 5.5,13 Z
M5.5,18 C4.67157288,18 4,17.3284271 4,16.5 C4,15.6715729 4.67157288,15 5.5,15
C6.32842712,15 7,15.6715729 7,16.5 C7,17.3284271 6.32842712,18 5.5,18 Z"
fill="#000000" opacity="0.3"/>
                                                        </g>
                                                    </svg>
                                                                        <!--end::Svg
Icon-->
                                                </span>
                                                            <a
href="/boards/{{ proj.project.id }}"
                                                                class="font-weight-
bolder text-primary ml-2">
                                                                 {{ proj.tasks }}
Завдань активно</a>
                                            </div>
                                        </div>
                                    </div>
                                    <!--end::Footer-->
                                </div>
                                <!--end:: Card-->
                            </div>
```

```
                                {% if forloop.counter|divisibleby:3 %}</div>
                                    <div class="row">{% endif %}
                                {% if forloop.last %}</div>{% endif %}
                                {% endfor %}

                                <!--begin::Pagination-->
                                <div class="d-flex justify-content-between
align-items-center flex-wrap">
                                    {% for proj in projects %}
                                        {% if forloop.first %}
                                            <div class="d-flex flex-wrap mr-
3">
                                                <a href="#" class="btn btn-
icon btn-sm btn-light-primary mr-2 my-1">
                                                    <i class="ki ki-bold-
double-arrow-back icon-xs"></i>
                                                </a>
                                                <a href="#" class="btn btn-
icon btn-sm btn-light-primary mr-2 my-1">
                                                    <i class="ki ki-bold-
arrow-back icon-xs"></i>
                                                </a>
                                                <a href="#"
                                                   class="btn btn-icon btn-sm
border-0 btn-hover-primary mr-2 my-1">1</a>
                                                <a href="#" class="btn btn-
icon btn-sm btn-light-primary mr-2 my-1">
                                                    <i class="ki ki-bold-
arrow-next icon-xs"></i>
                                                </a>
                                                <a href="#" class="btn btn-
icon btn-sm btn-light-primary mr-2 my-1">
                                                    <i class="ki ki-bold-
double-arrow-next icon-xs"></i>
                                                </a>
                                            </div>
                                            <div class="d-flex align-items-
center">
                                                <select class="form-control
form-control-sm text-primary font-weight-bold mr-4 border-0 bg-light-primary"
                                                        style="width: 75px;">
                                                    <option
value="10">10</option>
                                                    <option
value="20">20</option>
                                                    <option
value="30">30</option>
                                                    <option
value="50">50</option>
                                                    <option
value="100">100</option>
                                                </select>
                                                <span class="text-muted">На
сторінці</span>
                                            </div>
                                        {% endif %}
                                    {% endfor %}
                                </div>
                                <!--end::Pagination-->
                            </div>
                            <!--end::Container-->
```

```
                        </div>
                        <!--end::Entry-->
                    </div>
                    <!--end::Content-->

                {% endblock %}
                <!--begin::Footer-->
                <div class="footer bg-white py-4 d-flex flex-lg-column"
id="kt footer">
                    <!--begin::Container-->
                    <div class="container d-flex flex-column flex-md-row
align-items-center justify-content-between">
                        <!--begin::Copyright-->
                        <div class="text-dark order-2 order-md-1">
                            <span class="text-muted font-weight-bold mr-
2">2023©</span>
                            <span class="text-dark-75 text-hover-
primary">Diploma Tasks</span>
                        </div>
                        <!--end::Copyright-->
                        <!--begin::Nav-->
                        <div class="nav nav-dark order-1 order-md-2">
                            <span class="nav-link pr-3 pl-0">About</span>
                            <span class="nav-link px-3">Team</span>
                            <span class="nav-link pl-3 pr-0">Contact</span>
                        </div>
                        <!--end::Nav-->
                    </div>
                    <!--end::Container-->
                </div>
                <!--end::Footer-->
            </div>
            <!--end::Wrapper-->
        </div>
        <!--end::Page-->
    </div>

    <!-- begin::User Panel-->
    <div id="kt quick user" class="offcanvas offcanvas-right p-10">
        <!--begin::Header-->
        <div class="offcanvas-header d-flex align-items-center justify-
content-between pb-5">
            <h3 class="font-weight-bold m-0">User Profile
                <small class="text-muted font-size-sm ml-2"></small></h3>
            <a href="#" class="btn btn-xs btn-icon btn-light btn-hover-
primary" id="kt_quick_user_close">
                <i class="ki ki-close icon-xs text-muted"></i>
            </a>
        </div>
        <!--end::Header-->
        <!--begin::Content-->
        <div class="offcanvas-content pr-5 mr-n5">
            <!--begin::Header-->
            <div class="d-flex align-items-center mt-5">
                <div class="symbol symbol-100 mr-5">
                    <div class="symbol-label"
                        style="background-image:url({% static
user.profile.profile_photo %})"></div>
                    <i class="symbol-badge bg-success"></i>
                </div>
                <div class="d-flex flex-column">
```

```
                    <a href="#"
                       class="font-weight-bold font-size-h5 text-dark-75
text-hover-primary">{{ user.username }}</a>
                    <div class="navi mt-2">
                        <a href="#" class="navi-item">
                         <span class="navi-link p-0 pb-2">
                            <span class="navi-icon mr-1">
                               <span class="svg-icon svg-icon-lg svg-icon-
primary">
                                    <!--begin::Svg Icon |
path:/media/svg/icons/Communication/Mail-notification.svg-->
                                    <svg xmlns="http://www.w3.org/2000/svg"

xmlns:xlink="http://www.w3.org/1999/xlink" width="24px" height="24px"
                                                        viewBox="0 0 24 24"
version="1.1">
                                        <g stroke="none" stroke-width="1"
fill="none" fill-rule="evenodd">
                                            <rect x="0" y="0" width="24"
height="24"/>
                                            <path d="M21,12.0829584
C20.6747915,12.0283988 20.3407122,12 20,12 C16.6862915,12 14,14.6862915 14,18
C14,18.3407122 14.0283988,18.6747915 14.0829584,19 L5,19 C3.8954305,19
3,18.1045695 3,17 L3,8 C3,6.8954305 3.8954305,6 5,6 L19,6 C20.1045695,6
21,6.8954305 21,8 L21,12.0829584 Z M18.1444251,7.83964668 L12,11.1481833
L5.85557487,7.83964668 C5.4908718,7.6432681 5.03602525,7.77972206
4.83964668,8.14442513 C4.6432681,8.5091282 4.77972206,8.96397475
5.14442513,9.16035332 L11.6444251,12.6603533 C11.8664074,12.7798822
12.1335926,12.7798822 12.3555749,12.6603533 L18.8555749,9.16035332
C19.2202779,8.96397475 19.3567319,8.5091282 19.1603533,8.14442513
C18.9639747,7.77972206 18.5091282,7.6432681 18.1444251,7.83964668 Z"
                                                                fill="#000000"/>
                                            <circle fill="#000000" opacity="0.3"
cx="19.5" cy="17.5" r="2.5"/>
                                        </g>
                                    </svg>
                                        <!--end::Svg Icon-->
                               </span>
                            </span>
                            <span class="navi-text text-muted text-hover-
primary">{{ user.email }}</span>
                         </span>
                        </a>
                        <a href="/signOut" class="btn btn-sm btn-light-
primary font-weight-bolder py-2 px-5">Вийти</a>
                    </div>
                </div>
            </div>
            <!--end::Header-->
            <!--begin::Separator-->
            <div class="separator separator-dashed mt-8 mb-5"></div>
            <!--end::Separator-->
            <!--begin::Nav-->
            <div class="navi navi-spacer-x-0 p-0">
                <!--begin::Item-->
                <a href="/report" class="navi-item">
                    <div class="navi-link">
                        <div class="symbol symbol-40 bg-light mr-3">
                            <div class="symbol-label">
                            <span class="svg-icon svg-icon-md svg-icon-
success">
```

```
                                <!--begin::Svg Icon |
path:/media/svg/icons/General/Notification2.svg-->
                                <svg xmlns="http://www.w3.org/2000/svg"

xmlns:xlink="http://www.w3.org/1999/xlink" width="24px" height="24px"
                                                    viewBox="0 0 24 24"
version="1.1">
                                    <g stroke="none" stroke-width="1"
fill="none" fill-rule="evenodd">
                                        <rect x="0" y="0" width="24"
height="24"/>
                                        <path d="M13.2070325,4
C13.0721672,4.47683179 13,4.97998812 13,5.5 C13,8.53756612 15.4624339,11
18.5,11 C19.0200119,11 19.5231682,10.9278328 20,10.7929675 L20,17
C20,18.6568542 18.6568542,20 17,20 L7,20 C5.34314575,20 4,18.6568542 4,17
L4,7 C4,5.34314575 5.34314575,4 7,4 L13.2070325,4 Z"
                                                                fill="#000000"/>
                                        <circle fill="#000000" opacity="0.3"
cx="18.5" cy="5.5" r="2.5"/>
                                    </g>
                                </svg>
                                            <!--end::Svg Icon-->
                            </span>
                            </div>
                        </div>
                        <div class="navi-text">
                            <div class="font-weight-bold">Профіль</div>
                            <div class="text-muted">Account settings and
more</div>
                        </div>
                    </div>
                </a>
                <!--end:Item-->
                <!--begin::Item-->
                <a href="/boards" class="navi-item">
                    <div class="navi-link">
                        <div class="symbol symbol-40 bg-light mr-3">
                            <div class="symbol-label">
                            <span class="svg-icon svg-icon-md svg-icon-
primary">
                                <!--begin::Svg Icon |
path:/media/svg/icons/Communication/Mail-opened.svg-->
                                <svg xmlns="http://www.w3.org/2000/svg"

xmlns:xlink="http://www.w3.org/1999/xlink" width="24px" height="24px"
                                                    viewBox="0 0 24 24"
version="1.1">
                                    <g stroke="none" stroke-width="1"
fill="none" fill-rule="evenodd">
                                        <rect x="0" y="0" width="24"
height="24"/>
                                        <path d="M6,2 L18,2 C18.5522847,2
19,2.44771525 19,3 L19,12 C19,12.5522847 18.5522847,13 18,13 L6,13
C5.44771525,13 5,12.5522847 5,12 L5,3 C5,2.44771525 5.44771525,2 6,2 Z M7.5,5
C7.22385763,5 7,5.22385763 7,5.5 C7,5.77614237 7.22385763,6 7.5,6 L13.5,6
C13.7761424,6 14,5.77614237 14,5.5 C14,5.22385763 13.7761424,5 13.5,5 L7.5,5
Z M7.5,7 C7.22385763,7 7,7.22385763 7,7.5 C7,7.77614237 7.22385763,8 7.5,8
L10.5,8 C10.7761424,8 11,7.77614237 11,7.5 C11,7.22385763 10.7761424,7 10.5,7
L7.5,7 Z"
                                                                fill="#000000"
opacity="0.3"/>
```

```
                                        <path d="M3.79274528,6.57253826 L12,12.5
L20.2072547,6.57253826 C20.4311176,6.4108595 20.7436609,6.46126971
20.9053396,6.68513259 C20.9668779,6.77033951 21,6.87277228 21,6.97787787
L21,17 C21,18.1045695 20.1045695,19 19,19 L5,19 C3.8954305,19 3,18.1045695
3,17 L3,6.97787787 C3,6.70173549 3.22385763,6.47787787 3.5,6.47787787
C3.60510559,6.47787787 3.70753836,6.51099993 3.79274528,6.57253826 Z"
                                                    fill="#000000"/>
                                    </g>
                                </svg>
                                            <!--end::Svg Icon-->
                            </span>
                            </div>
                        </div>
                        <div class="navi-text">
                            <div class="font-weight-bold">Завдання</div>
                            <div class="text-muted">останні завдання</div>
                        </div>
                    </div>
                </a>
                <!--end:Item-->
            </div>
            <!--end::Nav-->
            <!--begin::Separator-->
            <div class="separator separator-dashed my-7"></div>
            <!--end::Separator-->
            <!--begin::Notifications-->
            <div>
                <!--begin:Heading-->
                <h5 class="mb-5">Recent Notifications</h5>
                <!--end:Heading-->
                <!--begin::Item-->
                <div class="d-flex align-items-center bg-light-warning
rounded p-5 gutter-b">
                    <span class="svg-icon svg-icon-warning mr-5">
                       <span class="svg-icon svg-icon-lg">
                          <!--begin::Svg Icon |
path:/media/svg/icons/Home/Library.svg-->
                          <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
                                        width="24px" height="24px" viewBox="0 0
24 24" version="1.1">
                             <g stroke="none" stroke-width="1" fill="none"
fill-rule="evenodd">
                                <rect x="0" y="0" width="24" height="24"/>
                                <path d="M5,3 L6,3 C6.55228475,3 7,3.44771525
7,4 L7,20 C7,20.5522847 6.55228475,21 6,21 L5,21 C4.44771525,21 4,20.5522847
4,20 L4,4 C4,3.44771525 4.44771525,3 5,3 Z M10,3 L11,3 C11.5522847,3
12,3.44771525 12,4 L12,20 C12,20.5522847 11.5522847,21 11,21 L10,21
C9.44771525,21 9,20.5522847 9,20 L9,4 C9,3.44771525 9.44771525,3 10,3 Z"
                                                fill="#000000"/>
                                <rect fill="#000000" opacity="0.3"
                                                transform="translate(17.825568,
11.945519) rotate(-19.000000) translate(-17.825568, -11.945519)"
                                                x="16.3255682" y="2.94551858"
width="3" height="18" rx="1"/>
                             </g>
                          </svg>
                                <!--end::Svg Icon-->
                       </span>
                    </span>
                     <div class="d-flex flex-column flex-grow-1 mr-2">
```

```
                        <a href="#" class="font-weight-normal text-dark-75
text-hover-primary font-size-lg mb-1">Another
                            purpose persuade</a>
                        <span class="text-muted font-size-sm">Due in 2
Days</span>
                    </div>
                    <span class="font-weight-bolder text-warning py-1 font-
size-lg">+28%</span>
                </div>
                <!--end::Item-->
            </div>
            <!--end::Notifications-->
        </div>
        <!--end::Content-->
    </div>
    <!-- end::User Panel-->

{% endblock %}

{% block js %}
    <script src="{% static '/js/pages/my-script.js' %}"></script>
    <script>
        $('.aff').on("click", function () {
            var $div = $(this);
            swal.fire({
                text: "Are You Sure ?",
                icon: "error",
                buttonsStyling: false,
                confirmButtonText: "Yes !",
                customClass: {
                    confirmButton: "btn font-weight-bold btn-light-primary"
                }
            }).then(function (result) {
                if (result.isConfirmed) {

                    let url = $div.attr('id');

                    $.ajax(`/boards/${url}`, {
                        headers: {"X-CSRFToken": $(`{% csrf token %}`).val()},
                        type: "POST",
                        success: (data) => {
                            window.location.replace('/')
                        },
                        error: function (data) {
                            $('[data-switch=true]').bootstrapSwitch();
                            let content = {};
                            content.message = data.responseJSON.error;
                            $.notify(content, {
                                type: 'danger',
                                placement: {
                                    from: 'top',
                                    align: 'left'
                                },
                                offset: {
                                    x: 30,
                                    y: 30
                                },
                            });
                        }
                    });
                }
```

```
            });
        });
    </script>
{% endblock %}
```

Report.html

```
{% extends 'projects.html' %}
{% load static %}

{% block title %}Reports{% endblock %}

{% block body %}

    <div class="d-flex flex-column-fluid">
        <!--begin::Container-->
        <div class="container">
            <!-- begin::Card-->
            <div class="card card-custom overflow-hidden">
                <div class="card-body p-0">
                    <!-- begin: Invoice-->
                    <!-- begin: Invoice header-->
                    <div class="row justify-content-center bgi-size-cover
bgi-no-repeat py-8 px-8 py-md-27 px-md-0"
                         style="background-image: url({% static
'/media/bg/bg-6.jpg' %});">
                        <div class="col-md-9">
                            <div class="d-flex justify-content-between pb-10
pb-md-20 flex-column flex-md-row">
                                <h1 class="display-4 text-white font-weight-
boldest mb-10">Звіт</h1>
                                <div class="d-flex flex-column align-items-
md-end px-0">
                                    <span class="text-white d-flex flex-
column align-items-md-end opacity-70">
                                        <span>Diploma Task
Manager</span>
                                        <span>Created For
Diploma</span>
                                    </span>
                                </div>
                            </div>
                            <div class="border-bottom w-100 opacity-
20"></div>
                            <div class="d-flex justify-content-between text-
white pt-6">
                                <div class="d-flex flex-column flex-root">
                                    <span class="font-weight-bolde mb-
2r">ДАТА & ЧАС</span>
                                    <span class="opacity-
70">{{ time }}</span>
                                </div>
                                <div class="d-flex flex-column flex-root">
                                    <span class="font-weight-bolder mb-
2">ЗГЕНЕРОВАНО ДЛЯ КОРИСТУВАЧА</span>
                                    <div class="d-flex align-items-center">
                                        <div class="symbol symbol-50 mr-5">
                                            <div class="symbol-label"
                                                 style="background-
image:url({% static user.profile.profile_photo %})"></div>
```

```
                                        </div>
                                        <div class="d-flex flex-column">
                                            <a href="#"
                                               class="font-weight-bold font-
size-h5 text-dark-75 text-hover-primary">{{ user.username }}</a>
                                            <div class="navi mt-2">
                                                <a href="#" class="navi-
item">
                                                    <span class="navi-link p-
0 pb-2">
                                                        <span class="navi-
text text-muted text-hover-primary">{{ user.email }}</span>
                                                    </span>
                                                </a>
                                            </div>
                                        </div>
                                    </div>

                                </div>
                                <div class="d-flex flex-column flex-root">
                                    <span class="font-weight-bolder mb-
2">ВКЛЮЧАЮЧИ</span>
                                    <span class="opacity-70">Інформація щодо
завдань<br>Інформація щодо користувача</span>
                                </div>
                            </div>
                        </div>
                    </div>
                    <!-- end: Invoice header-->
                    <!-- begin: Invoice body-->
                    <div class="row justify-content-center py-8 px-8 py-md-10
px-md-0">
                        <div class="col-md-9">
                            <div class="table-responsive">
                                <span class="font-weight-bolder font-size-h3
mb-6">Статуси усіх проектів :</span>
                                <table class="table">
                                    <thead>
                                    <tr>
                                        <th class="pl-0 font-weight-bold
text-muted text-uppercase">Project</th>
                                        <th class="text-right font-weight-
bold text-muted text-uppercase">TO DO</th>
                                        <th class="text-right font-weight-
bold text-muted text-uppercase">In Progress
                                        </th>
                                        <th class="text-right font-weight-
bold text-muted text-uppercase">In Test
                                        </th>
                                        <th class="text-right font-weight-
bold text-muted text-uppercase">Done
                                        </th>
                                        <th class="text-right pr-0 font-
weight-bold text-muted text-uppercase">PROGRESS
                                        </th>
                                    </tr>
                                    </thead>
                                    <tbody>
                                    {% for p in p_info %}
                                        <tr class="font-weight-boldest
border-bottom-0 font-size-lg">
```

```
                                            <td class="border-top-0 pl-0 py-
4">{{ p.name }}</td>
                                            <td class="border-top-0 text-
right py-4">{{ p.t }}</td>
                                            <td class="border-top-0 text-
right py-4">{{ p.d }}</td>
                                            <td class="border-top-0 text-
right py-4">{{ p.i }}</td>
                                            <td class="border-top-0 text-
right py-4">{{ p.o }}</td>
                                            <td class="text-danger border-
top-0 pr-0 py-4 text-right">{{ p.progress }}
                                                %
                                            </td>
                                        </tr>
                                    {% endfor %}
                                    </tbody>
                                </table>
                            </div>
                        </div>
                    </div>
                    <!-- end: Invoice body-->
                    <!-- begin: Invoice footer-->
                    <div class="row justify-content-center bg-gray-100 py-8
px-8 py-md-10 px-md-0">
                        <div class="col-md-9">
                            <div class="d-flex justify-content-between flex-
column flex-md-row font-size-lg">
                                <div class="d-flex flex-column mb-10 mb-md-
0">
                                    <div class="font-weight-bolder font-size-
lg mb-3">Інформація про користувача :</div>
                                    <div class="d-flex justify-content-
between mb-3">
                                        <span class="mr-15 font-weight-
bold">ВСЬОГО ЗАВДАНЬ ДЛЯ ВИКОНАННЯ :</span>
                                        <span class="text-
right">{{ u_info.todo }}</span>
                                    </div>
                                    <div class="d-flex justify-content-
between mb-3">
                                        <span class="mr-15 font-weight-
bold">ВСЬОГО ВИКОНУЄТЬСЯ ЗАВДАНЬ:</span>
                                        <span class="text-
right">{{ u_info.doing }}</span>
                                    </div>
                                    <div class="d-flex justify-content-
between">
                                        <span class="mr-15 font-weight-
bold">ВСЬОГО ВИКОНАНО ЗАВДАНЬ :</span>
                                        <span class="text-
right">{{ u_info.done }}</span>
                                    </div>
                                </div>
                                <div class="d-flex flex-column text-md-
right">
                                    <span class="font-size-lg font-weight-
bolder mb-1">ПРОГРЕС</span>
                                    <span class="font-size-h2 font-weight-
boldest text-danger mb-1">{{ u_info.progress }} %</span>
                                    <span>Including All Tasks in All
```

```
Projects</span>
                                </div>
                            </div>
                        </div>
                    </div>

                    <div class="row justify-content-center py-8 px-8 py-md-10
px-md-0">
                        <div class="col-md-9">
                            <div class="table-responsive">
                                <span class="font-weight-bolder font-size-h3
mb-2">Статус Користувача у Проектах :</span>
                                <table class="table">
                                    <thead>
                                    <tr>
                                        <th class="pl-0 font-weight-bold
text-muted text-uppercase">Проект</th>
                                        <th class="text-right font-weight-
bold text-muted text-uppercase">ДО ВИКОНАННЯ</th>
                                        <th class="text-right font-weight-
bold text-muted text-uppercase">ВИКОНУЄТЬСЯ</th>
                                        <th class="text-right font-weight-
bold text-muted text-uppercase">ВИКОНАНО
                                        </th>
                                        <th class="text-right pr-0 font-
weight-bold text-muted text-uppercase">ПРОГРЕС
                                        </th>
                                    </tr>
                                    </thead>
                                    <tbody>
                                    {% for u in u_in_p %}
                                        <tr class="font-weight-boldest
border-bottom-0 font-size-lg">
                                            <td class="border-top-0 pl-0 py-
4">{{ u.name }}</td>
                                            <td class="border-top-0 text-
right py-4">{{ u.u_info.todo }}</td>
                                            <td class="border-top-0 text-
right py-4">{{ u.u info.doing }}</td>
                                            <td class="border-top-0 text-
right py-4">{{ u.u_info.done }}</td>
                                            <td class="text-danger border-
top-0 pr-0 py-4 text-right">{{ u.u info.progress }}
                                                %
                                            </td>
                                        </tr>
                                    {% endfor %}
                                    </tbody>
                                </table>
                            </div>
                        </div>
                    </div>
                    <!-- end: Invoice footer-->
                    <!-- begin: Invoice action-->
                    <div class="row justify-content-center py-8 px-8 py-md-10
px-md-0">
                        <div class="col-md-9">
                            <div class="d-flex justify-content-between">
                                <a href="/boards" class="btn btn-light-
primary font-weight-bold">Назад
                                </a>
```

```
                                    <button type="button" class="btn btn-primary
font-weight-bold"
                                            onclick="window.print();">Завантажити
звіт
                                    </button>
                                </div>
                            </div>
                        </div>
                        <!-- end: Invoice action-->
                        <!-- end: Invoice-->
                    </div>
                </div>
                <!-- end::Card-->
            </div>
            <!--end::Container-->
        </div>

{% endblock %}
```

Tasks.html

```
{% extends 'projects.html' %}
{% load static %}

{% block title %}Tasks{% endblock %}

{% block css %}
    <link href="{% static '/plugins/custom/kanban/kanban.bundle.css' %}"
rel="stylesheet" type="text/css"/>
    <link href="{% static
'/plugins/custom/fullcalendar/fullcalendar.bundle.css' %}" rel="stylesheet"
type="text/css">
{% endblock %}

{% block body %}

    <!--begin::Content-->
    <div class="content d-flex flex-column flex-column-fluid"
id="kt_content">
        <!--begin::Subheader-->
        <div class="subheader py-2 py-lg-12 subheader-transparent"
id="kt_subheader">
            <div class="container d-flex align-items-center justify-content-
between flex-wrap flex-sm-nowrap">
                <!--begin::Info-->
                <div class="d-flex align-items-center flex-wrap mr-1">
                    <!--begin::Heading-->
                    <div class="d-flex flex-column">
                        <!--begin::Title-->
                        <h2 class="text-white font-weight-bold my-2 mr-
5">Завдання в "{{ proj.name }}"</h2>
                        <!--end::Title-->
                        <!--begin::Breadcrumb-->
                        <div class="d-flex align-items-center font-weight-
bold my-2">
                            <!--begin::Item-->
                            <a href="/boards" class="opacity-75 hover-
opacity-100">
                                <i class="flaticon2-shelter text-white icon-
1x"></i>
```

```
                            </a>
                            <!--end::Item-->
                        </div>
                        <!--end::Breadcrumb-->
                    </div>
                    <!--end::Heading-->
                </div>
                <!--end::Info-->
            </div>
        </div>
        <!--end::Subheader-->

        <!--begin::Entry-->
        <div class="d-flex flex-column-fluid">
            <!--begin::Container-->
            <div class="container">
                <!--begin::Card-->
                <div class="card card-custom gutter-b">
                    <div class="card-header">
                        <div class="card-title">
                            <h3 class="card-label">Завдання</h3>
                        </div>
                    </div>
                    <div class="card-body">
                        <div id="kt_kanban_4"></div>
                        <div class="mt-4">
                            {% if can_add %}
                                <!-- Button  modal-->
                                <button type="button" class="btn font-weight-
bold btn-light-primary mr-5"
                                        data-toggle="modal"
                                        data-target="#exampleModal">
                                    Додати Завдання
                                </button>
                                <!-- Modal-->
                                <div class="modal fade" id="exampleModal"
tabindex="-1" role="dialog"
                                     aria-labelledby="exampleModalLabel"
aria-hidden="true">

                                    <div class="modal-dialog"
role="document">
                                        <div class="modal-content">
                                            <div class="modal-header">
                                                <h5 class="modal-title"
id="exampleModalLabel">Додати нове завдання</h5>
                                                <button type="button"
class="close" data-dismiss="modal"
                                                        aria-label="Close">
                                                    <i aria-hidden="true"
class="ki ki-close"></i>
                                                </button>
                                            </div>
                                            <div class="modal-body">
                                                <form id="add_board_from"
method="post">{% csrf_token %}
                                                    <div class="form-group">
                                                        <label>Назва
                                                            <span
class="text-danger">*</span></label>
                                                        <input id="iname"
```

```
name="name" type="text"
                                                    class="form-
control form-control-lg"

placeholder="Task">
                                                </div>
                                                <div class="form-group">
                                                    <label>Опис
                                                        <span
class="text-danger">*</span></label>
                                                    <input id="idesc"
name="desc" type="text"
                                                        class="form-
control form-control-lg"

placeholder="Details ...">
                                                </div>
                                                <div class="form-group">
                                                    <label>Оберіть
дедлайн
                                                        <span
class="text-danger">*</span></label>
                                                    <div class="input-
group date">
                                                        <input
name="date" type="text"

class="form-control form-control-lg"

id="kt datepicker 3"/>
                                                        <div
class="input-group-append">
                                                           <span
class="input-group-text">
                                                            <i class="la
la-calendar"></i>
                                                           </span>
                                                        </div>
                                                    </div>
                                                </div>
                                                <div class="form-group">
                                                    <label>Оберіть
виконавців
                                                        <span
class="text-danger">*</span></label><br>
                                                    <select name="users"
style="width: 100%"
                                                            class="form-
control form-control-lg select2"

id="kt_select2_3"
                                                            name="param">
                                                        {% for ou in
other_users %}
                                                            <option
value="{{ ou.id }}">{{ ou.username }}</option>
                                                        {% endfor %}
                                                    </select>
                                                </div>
                                                <div class="form-group">
                                                    <button type="button"
```

```
                                                                    class="btn
btn-light-primary font-weight-bold"
                                                                    data-
dismiss="modal">Скасувати
                                                            </button>
                                                            <button id="psubmit"
type="submit"
                                                                    class="btn
btn-primary font-weight-bold">Зберегти зміни
                                                            </button>
                                                        </div>
                                                    </form>
                                                </div>

                                            </div>
                                        </div>
                                    </div>
                                {% endif %}

                            </div>
                        </div>
                    </div>
                    <div class="card card-custom">
                        <div class="card-header">
                            <div class="card-title">
                                <h3 class="card-label">
                                    Графік виконання
                                </h3>
                            </div>
                        </div>
                        <div class="card-body">
                            <div id="kt_calendar"></div>
                        </div>
                    </div>
                    <!--end::Card-->

                </div>
                <!--end::Container-->
            </div>

            {% for t in tasks %}
                <div class="modal fade hidden" id="task-details-{{ t.id }}"
tabindex="-1" role="dialog"
                     aria-labelledby="exampleModalLabel" aria-hidden="true">

                    <div class="modal-dialog" role="document">
                        <div class="modal-content">
                            <div class="modal-header">
                                <h5 class="modal-title"
id="exampleModalLabel">{{ t.description }}</h5>
                                <button type="button" class="close" data-
dismiss="modal"
                                        aria-label="Close">
                                    <i aria-hidden="true" class="ki ki-
close"></i>
                                </button>
                            </div>
                            <div class="modal-body">
                                <div class="d-flex align-items-center">
                                    <div class="symbol symbol-100 mr-5">
                                        <div class="symbol-label"
```

```
                                        style="background-image:url({%
static t.assigned_to.profile.profile_photo %})"></div>
                                    <i class="symbol-badge bg-success"></i>
                                </div>
                                <div class="d-flex flex-column">
                                    <a href="#"
                                       class="font-weight-bold font-size-h5
text-dark-75 text-hover-primary">{{ t.assigned_to.username }}</a>
                                    <div class="navi mt-2">
                                        <a href="#" class="navi-item">
                            <span class="navi-link p-0 pb-2">
                               <span class="navi-icon mr-1">
                                  <span class="svg-icon svg-icon-lg svg-icon-
primary">
                                     <!--begin::Svg Icon |
path:/media/svg/icons/Communication/Mail-notification.svg-->
                                    <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" width="24px" height="24px"
                                                viewBox="0 0 24 24"
version="1.1">
                                        <g stroke="none" stroke-width="1"
fill="none" fill-rule="evenodd">
                                            <rect x="0" y="0" width="24"
height="24"/>
                                            <path d="M21,12.0829584
C20.6747915,12.0283988 20.3407122,12 20,12 C16.6862915,12 14,14.6862915 14,18
C14,18.3407122 14.0283988,18.6747915 14.0829584,19 L5,19 C3.8954305,19
3,18.1045695 3,17 L3,8 C3,6.8954305 3.8954305,6 5,6 L19,6 C20.1045695,6
21,6.8954305 21,8 L21,12.0829584 Z M18.1444251,7.83964668 L12,11.1481833
L5.85557487,7.83964668 C5.4908718,7.6432681 5.03602525,7.77972206
4.83964668,8.14442513 C4.6432681,8.5091282 4.77972206,8.96397475
5.14442513,9.16035332 L11.6444251,12.6603533 C11.8664074,12.7798822
12.1335926,12.7798822 12.3555749,12.6603533 L18.8555749,9.16035332
C19.2202779,8.96397475 19.3567319,8.5091282 19.1603533,8.14442513
C18.9639747,7.77972206 18.5091282,7.6432681 18.1444251,7.83964668 Z"
                                                fill="#000000"/>
                                            <circle fill="#000000" opacity="0.3"
cx="19.5" cy="17.5" r="2.5"/>
                                        </g>
                                    </svg>
                                            <!--end::Svg Icon-->
                                </span>
                            </span>
                            <span class="navi-text text-muted text-hover-
primary">{{ t.assigned_to.email }}</span>
                        </span>
                                    </a>
                                </div>

                                <span class="label label-inline label-
light-info">Start Time: {{ t.start_time }}</span>
                                <span class="label label-inline label-
light-success">End Time: {{ t.end_time }}</span>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        {% endfor %}
```

```
        <!--end::Entry-->
    </div>
    <!--end::Content-->

{% endblock %}

{% block js %}
    <script src="{% static
'/plugins/custom/kanban/kanban.bundle.js' %}"></script>
    <script src="{% static
'/plugins/custom/fullcalendar/fullcalendar.bundle.js' %}"></script>
    <script src="{% static '/js/pages/my-script.js' %}"></script>

    <script>
        "use strict";

        // Class definition
        var KTKanbanBoardDemo = function () {
            // Private functions
            var _demo4 = function () {
                var kanban = new jKanban({
                    element: '#kt_kanban_4',
                    gutter: '0',
                    click: function (el) {
                        let eid = $(el).data("eid");
                        $(`#task-details-${eid}`).modal('show');
                    },
                    boards: [
                        {
                            'id': 'T',
                            'title': 'To Do',
                            'class': 'light-danger',
                            'item': [
                                {% for t in tasks %}
                                    {% if t.status == 'T' %}
                                        {
                                            'id': {{ t.id }},
                                            'title': `
                                <div class="d-flex align-items-center">
                                    <div class="symbol symbol-success mr-3">
                                        <img alt="Pic" src="{% static
t.assigned_to.profile.profile_photo %}" />
                                    </div>
                                    <div class="d-flex flex-column align-
items-start">
                                        <span class="text-dark-50 font-
weight-bold mb-1">{{t.name}}</span>
                                        <span class="label label-inline
label-light-dark font-weight-bold">{{t.end_time}}</span>
                                    </div>
                                </div>`
                                        },
                                    {% endif %}
                                {% endfor %}
                            ]
                        },
                        {
                            'id': 'D',
                            'title': 'In Progress',
                            'class': 'light-primary',
                            'item': [{% for t in tasks %}
```

```
                                {% if t.status == 'D' %}
                                    {
                                        'id': {{ t.id }},
                                        'title': `
                                <div class="d-flex align-items-center">
                                    <div class="symbol symbol-success mr-3">
                                        <img alt="Pic" src="{% static
t.assigned_to.profile.profile_photo %}" />
                                    </div>
                                    <div class="d-flex flex-column align-
items-start">
                                        <span class="text-dark-50 font-
weight-bold mb-1">{{t.name}}</span>
                                        <span class="label label-inline
label-light-success font-weight-bold">{{t.end_time}}</span>
                                    </div>
                                </div>`
                                    },
                                {% endif %}
                            {% endfor %}
                            ]
                        },
                        {
                            'id': 'I',
                            'title': 'In Test',
                            'class': 'light-primary',
                            'item': [{% for t in tasks %}
                                {% if t.status == 'I' %}
                                    {
                                        'id': {{ t.id }},
                                        'title': `
                                <div class="d-flex align-items-center">
                                    <div class="symbol symbol-success mr-3">
                                        <img alt="Pic" src="{% static
t.assigned_to.profile.profile_photo %}" />
                                    </div>
                                    <div class="d-flex flex-column align-
items-start">
                                        <span class="text-dark-50 font-
weight-bold mb-1">{{t.name}}</span>
                                        <span class="label label-inline
label-light-warning font-weight-bold">{{t.end_time}}</span>
                                    </div>
                                </div>`
                                    },
                                {% endif %}
                            {% endfor %}
                            ]
                        },
                        {
                            'id': 'O',
                            'title': 'Done',
                            'class': 'light-success',
                            'item': [{% for t in tasks %}
                                {% if t.status == 'O' %}
                                    {
                                        'id': {{ t.id }},
                                        'title': `
                                <div class="d-flex align-items-center">
                                    <div class="symbol symbol-success mr-3">
                                        <img alt="Pic" src="{% static
```

```
t.assigned_to.profile.profile_photo %}" />
                                    </div>
                                    <div class="d-flex flex-column align-
items-start">
                                        <span class="text-dark-50 font-
weight-bold mb-1">{{t.name}}</span>
                                        <span class="label label-inline
label-light-primary font-weight-bold">{{t.end_time}}</span>
                                    </div>
                                </div>`
                                },
                            {% endif %}
                        {% endfor %}
                        ]
                    },
                    {
                        'id': 'B',
                        'title': 'Blocked',
                        'class': 'light-danger',
                        'item': [{% for t in tasks %}
                            {% if t.status == 'B' %}
                                {
                                    'id': {{ t.id }},
                                    'title': `
                                <div class="d-flex align-items-center">
                                    <div class="symbol symbol-success mr-3">
                                        <img alt="Pic" src="{% static
t.assigned_to.profile.profile_photo %}" />
                                    </div>
                                    <div class="d-flex flex-column align-
items-start">
                                        <span class="text-dark-50 font-
weight-bold mb-1">{{t.name}}</span>
                                        <span class="label label-inline
label-light-danger font-weight-bold">{{t.end_time}}</span>
                                    </div>
                                </div>`
                                },
                            {% endif %}
                        {% endfor %}
                        ]
                    },
                    {
                        'id': 'L',
                        'title': 'Deleted',
                        'class': 'light-dark',
                        'item': [{% for t in tasks %}
                            {% if t.status == 'L' %}
                                {
                                    'id': {{ t.id }},
                                    'title': `
                                <div class="d-flex align-items-center">
                                    <div class="symbol symbol-success mr-3">
                                        <img alt="Pic" src="{% static
t.assigned_to.profile.profile_photo %}" />
                                    </div>
                                    <div class="d-flex flex-column align-
items-start">
                                        <span class="text-dark-50 font-
weight-bold mb-1">{{t.name}}</span>
                                        <span class="label label-inline
```

```
label-light-danger font-weight-bold">{{t.end_time}}</span>
                                    </div>
                                </div>`
                                    },
                                {% endif %}
                            {% endfor %}
                            ]
                        }
                    ],
                    dragendEl: function (el) {
                        let eid = $(el).data("eid");
                        let bid =
$(el.parentElement.parentElement).data("id");
                        jQuery.ajax(`/boards/${eid}/task`, {
                            headers: {"X-CSRFToken": $(`{%
csrf_token %}`).val()},
                            type: 'POST',
                            data: {
                                "_token": "{{ csrf_token }}",
                                type: "edit_status",
                                task_id: eid,
                                board_id: bid
                            },
                            error: (data) => {
                                $('[data-switch=true]').bootstrapSwitch();
                                let content = {};
                                content.message = data.responseJSON.error;
                                $.notify(content, {
                                    type: 'danger',
                                    placement: {
                                        from: 'bottom',
                                        align: 'left'
                                    },
                                    offset: {
                                        x: 30,
                                        y: 30
                                    },
                                });
                            }
                        });
                    },
                });
            }

            // Public functions
            return {
                init: function () {
                    _demo4();
                }
            };
        }();

        var KTCalendarBasic = function () {

            return {
                //main function to initiate the module
                init: function () {
                    var todayDate = moment().startOf('day');
                    var YM = todayDate.format('YYYY-MM');
                    var YESTERDAY = todayDate.clone().subtract(1,
'day').format('YYYY-MM-DD');
```

```
                    var TODAY = todayDate.format('YYYY-MM-DD');
                    var TOMORROW = todayDate.clone().add(1,
'day').format('YYYY-MM-DD');

                    var calendarEl = document.getElementById('kt_calendar');
                    var calendar = new FullCalendar.Calendar(calendarEl, {
                        plugins: ['bootstrap', 'interaction', 'dayGrid',
'timeGrid', 'list'],
                        themeSystem: 'bootstrap',

                        isRTL: KTUtil.isRTL(),

                        header: {
                            left: 'prev,next today',
                            center: 'title',
                            right: 'dayGridMonth,timeGridWeek,timeGridDay'
                        },

                        height: 800,
                        contentHeight: 780,
                        aspectRatio: 3,  // see:
https://fullcalendar.io/docs/aspectRatio

                        nowIndicator: true,
                        {#now: TODAY + 'T09:25:00', // just for demo#}

                        views: {
                            dayGridMonth: {buttonText: 'month'},
                            timeGridWeek: {buttonText: 'week'},
                            timeGridDay: {buttonText: 'day'}
                        },

                        defaultView: 'dayGridMonth',
                        defaultDate: TODAY,

                        editable: true,
                        eventLimit: true, // allow "more" link when too many
events
                        navLinks: true,
                        events: [
                            {% for t in tasks %}
                                {
                                    id: '{{ t.id }}',
                                    title: '{{t.name}}',
                                    start: new Date('{{ t.end_time }}'),
                                    end: new Date('{{ t.end_time }}'),
                                    description: '{{t.description}}',
                                    {% if t.status == 'T' %}
                                        className: "fc-event-solid-warning
fc-event-light",
                                    {% endif %}
                                    {% if t.status == 'D' %}
                                        className: "fc-event-solid-primary
fc-event-light",
                                    {% endif %}
                                    {% if t.status == 'I' %}
                                        className: "fc-event-solid-info fc-
event-light",
                                    {% endif %}
                                    {% if t.status == 'O' %}
                                        className: "fc-event-solid-success
```

```
fc-event-light",
                                    {% endif %}
                                    {% if t.status == 'B' %}
                                        className: "fc-event-solid-danger fc-
event-light",
                                    {% endif %}
                                    {% if t.status == 'L' %}
                                        className: "fc-event-solid-dark  fc-
event-light",
                                    {% endif %}
                                },
                            {% endfor %}
                        ],

                        eventDrop: function (info) {
                            let id = info.event.id;
                            let year = info.event.start.getFullYear();
                            let month = info.event.start.getMonth() + 1;
                            let day = info.event.start.getDate()
                            console.log(id, `${year}-${month}-${day}`)

                            jQuery.ajax(`/boards/${id}/task`, {
                                headers: {"X-CSRFToken": $(`{%
csrf_token %}`).val()},
                                type: 'POST',
                                data: {
                                    "_token": "{{ csrf_token }}",
                                    type: "edit end time",
                                    task_id: id,
                                    new end time: `${year}-${month}-${day}`
                                },
                                error: (data) => {
                                    $('[data-switch=true]').bootstrapSwitch();
                                    let content = {};
                                    content.message = data.responseJSON.error;
                                    $.notify(content, {
                                        type: 'danger',
                                        placement: {
                                            from: 'bottom',
                                            align: 'left'
                                        },
                                        offset: {
                                            x: 30,
                                            y: 30
                                        },
                                    });
                                }
                            });
                        },

                        eventRender: function (info) {
                            var element = $(info.el);

                            if (info.event.extendedProps &&
info.event.extendedProps.description) {
                                if (element.hasClass('fc-day-grid-event')) {
                                    element.data('content',
info.event.extendedProps.description);
                                    element.data('placement', 'top');
                                    KTApp.initPopover(element);
                                } else if (element.hasClass('fc-time-grid-
```

```
event')) {
                                    element.find('.fc-title').append('<div
class="fc-description">' + info.event.extendedProps.description + '</div>');
                                } else if (element.find('.fc-list-item-
title').lenght !== 0) {
                                    element.find('.fc-list-item-
title').append('<div class="fc-description">' +
info.event.extendedProps.description + '</div>');
                                }
                            }
                        });

                        calendar.render();
                    }
                };
            }();

            jQuery(document).ready(function () {
                KTKanbanBoardDemo.init();
                KTCalendarBasic.init();
            });

        </script>
{% endblock %}
```

Views.py

```
from django.shortcuts import render
from django.views import View
from django.contrib.auth import authenticate, login, logout
from django.shortcuts import redirect
from django.http import JsonResponse
from django.contrib.auth.models import User
from .models import Profile
import random


def index(request):
    if request.user.is_authenticated:
        return redirect('boards')
    else:
        return redirect('signIn')
```

```
class SignIn(View):
    def get(self, request):
        if request.user.is_authenticated:
            return redirect('boards')
        else:
            return render(request, 'auth.html')
```

```
    def post(self, request):
```

```
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('boards')
```

```
        else:
            response = JsonResponse({"error": "Будь ласка, перевірте введені
дані"})
            response.status_code = 403
            return response
```

```
class SignUp(View):
    def get(self, request):
        if request.user.is_authenticated:
            return redirect('boards')
        else:
            return redirect('signIn')
```

```
    def post(self, request):
        try:
            username = request.POST['username']
            email = request.POST['email']
            password = request.POST['password']
```

```
            user = User.objects.create_user(username, email, password)
            user.save()
```

```
            login(request, user)
```

```
            n = random.randint(16, 45)
            pf_url = f'/media/users/{n}.jpg'
            pf = Profile(user=user, profile_photo=pf_url)
            pf.save()
```

```
            return redirect('boards')
```

```
        except:
            response = JsonResponse({"error": "Такого користувача вже
зареєстровано"})
            response.status_code = 403
            return response
class SignOut(View):
    def get(self, request):
        logout(request)
        return redirect('signIn')
```

ДОДАТОК В

Зображення до розділів:

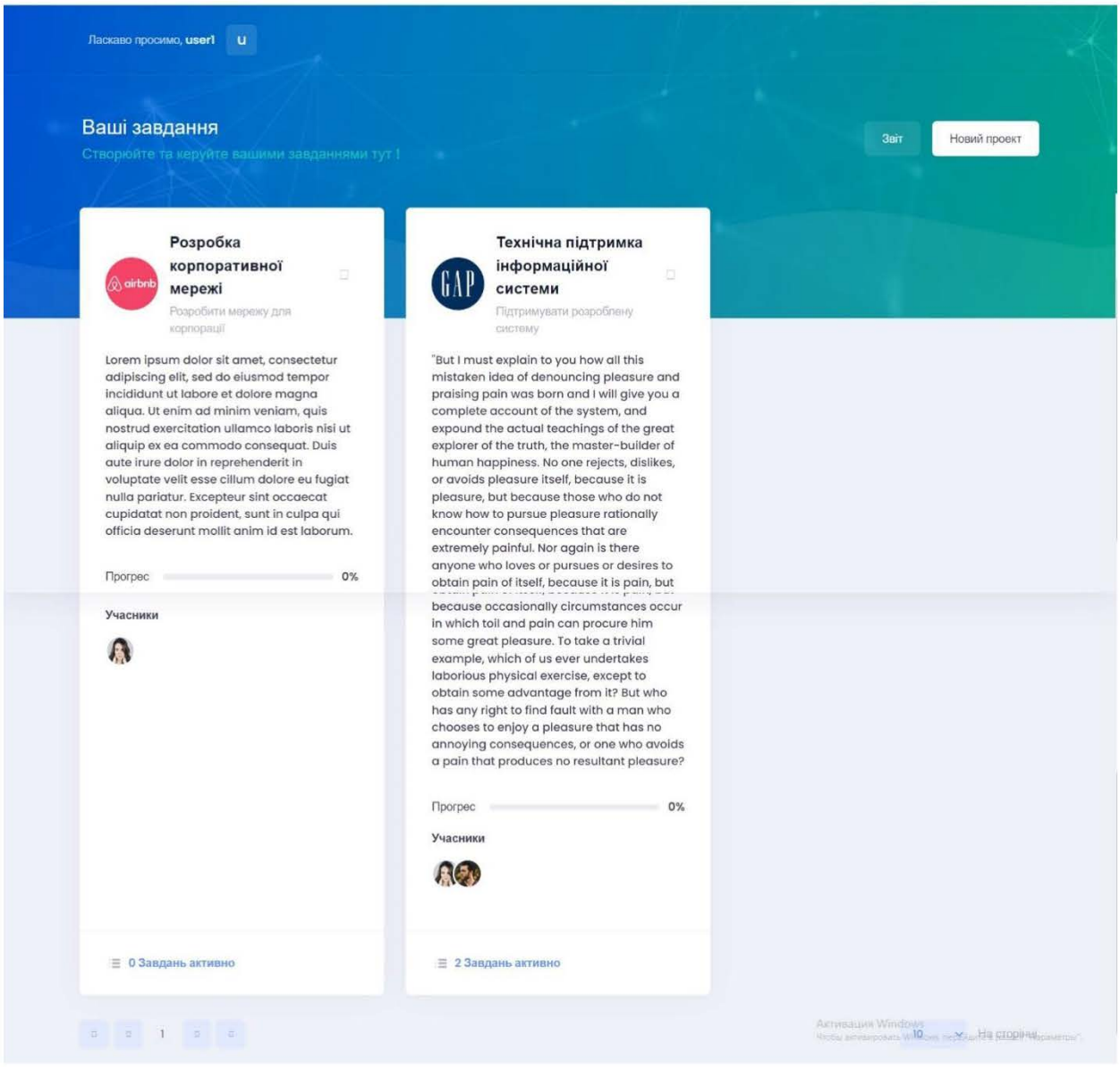


Рисунок 3.5 – Загальний вигляд сторінки проектів завдань

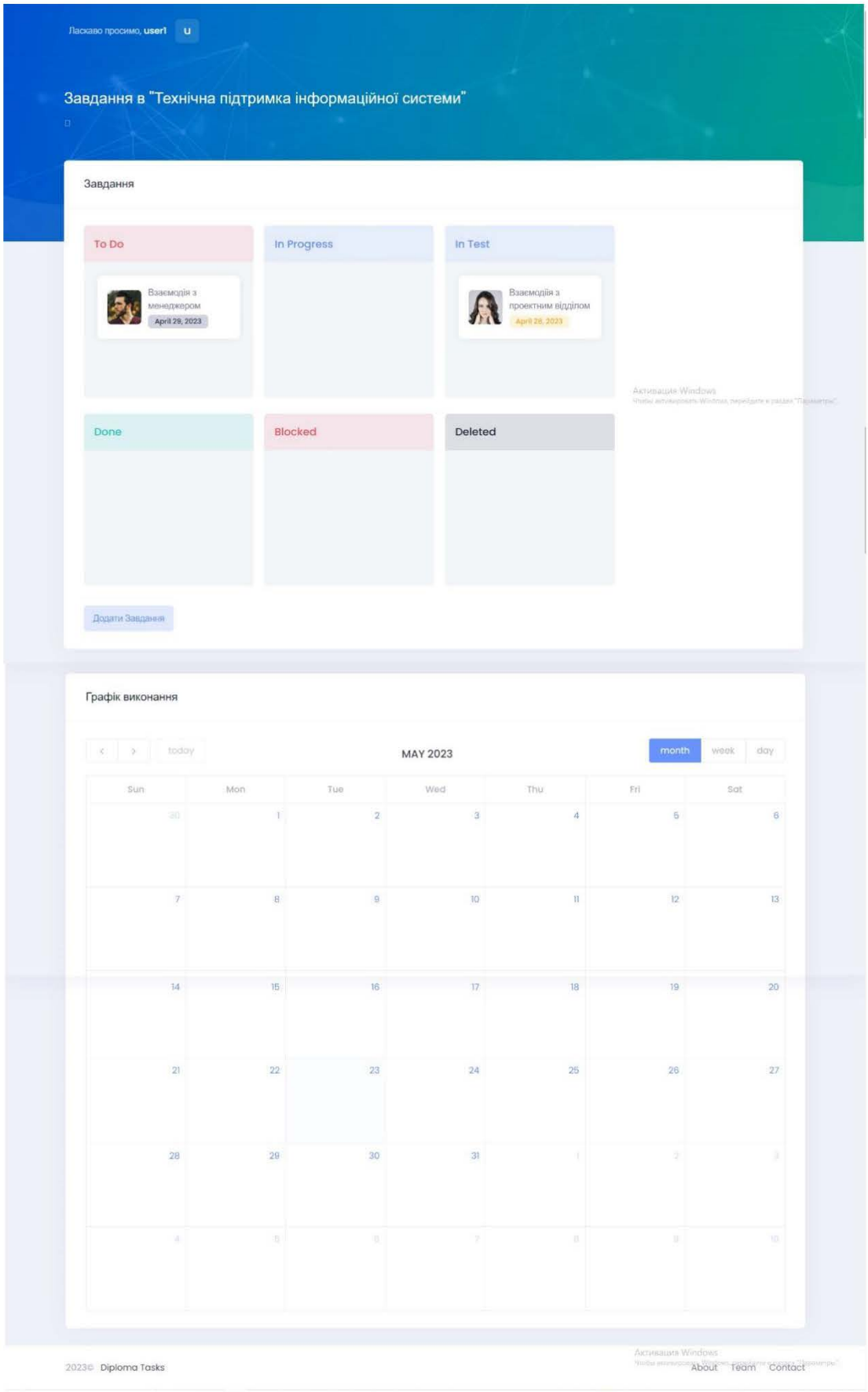


Рисунок 3.6 – Загальний вигляд сторінки адміністрування завдань

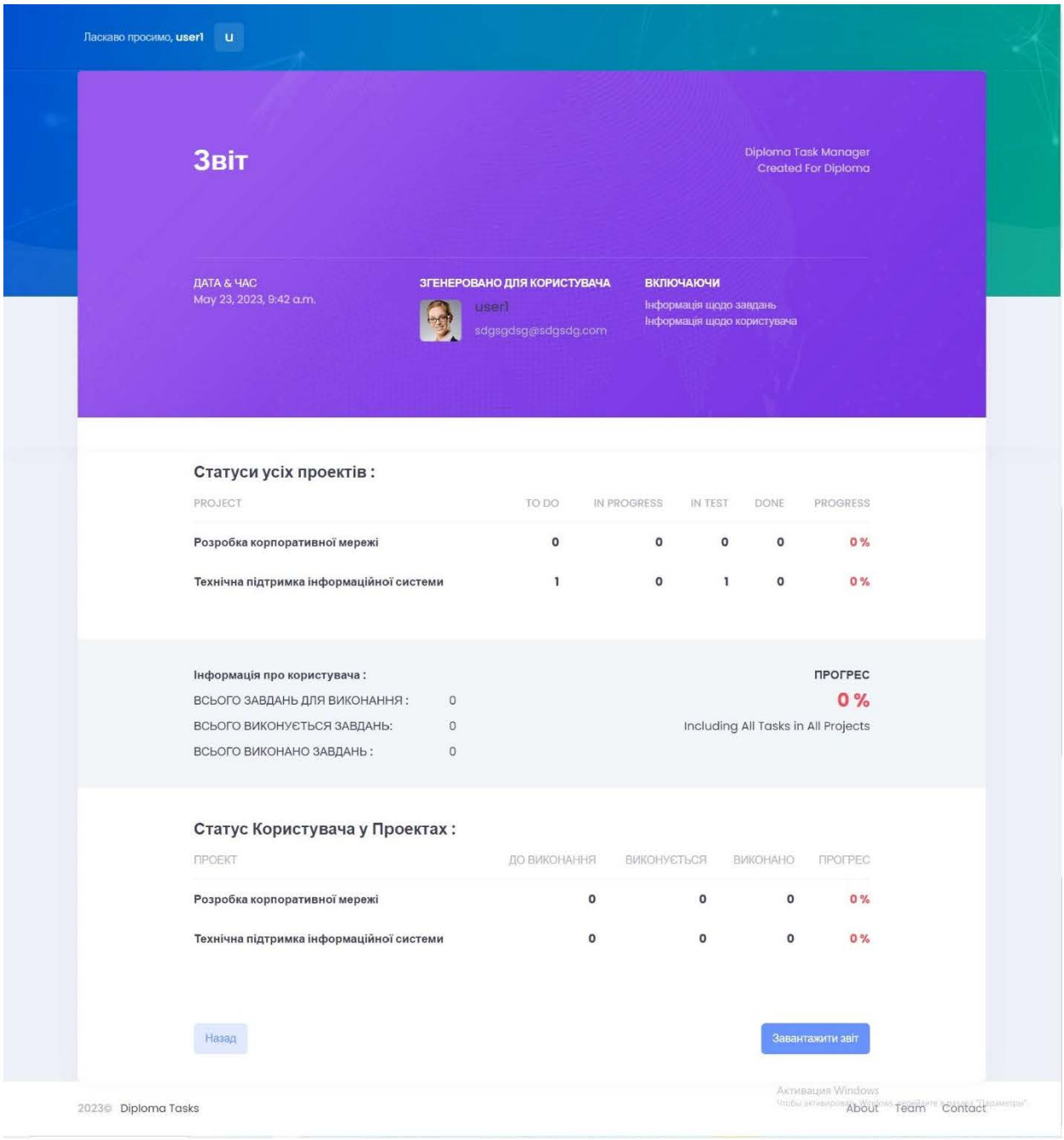


Рисунок 3.7 - Загальний вигляд сторінки формування звіту щодо поточних завдань

Рисунок 2.2 - Вигляд головної сторінки ресурсу