

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 «Створення Web-застосунку інструментами React у середовищі
розробки Visual Studio Code»

Виконав: студент групи _____ ПЗ20-1 _____

Спеціальність 121 «Інженерія програмного забезпечення» _____

_____ Шевченко Ілля Петрович _____

(прізвище та ініціали)

Керівник _____ проф. Корнєєв М. В. _____

(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент _____ Університет митної справи та фінансів _____

(місце роботи)

в.о. завідувача кафедри Кібербезпеки та інформаційних
технологій _____

(посада)

к.т.н., доцент кафедри кібербезпеки та інформаційних
технологій Прокопович-Ткаченко Д. І. _____

(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2024

АНОТАЦІЯ

Шевченко І. П. Створення Web-застосунку інструментами React у середовищі розробки Visual Studio Code.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 121 «Комп'ютерні науки» – Університет митної справи та фінансів, Дніпро, 2023.

Дана кваліфікаційна робота присвячена розробці адаптивного web-застосунку для навчального закладу, що можна буде використати для створення web-сайтів різних закладів освіти. Розглянуто загальні поняття щодо web-застосунків, їх актуальність і розвиток. Досліджено сайти-авторитети, які дали основний план створення універсального web-застосунку. Реалізовано проєкт інструментами бібліотеки React у середовищі розробки Visual Studio Code, що забезпечило високу ефективність і гнучкість у процесі розробки. Використання React дозволило створити динамічний та інтерактивний web-застосунок із можливістю швидкої адаптації під специфічні потреби кожного навчального закладу.

Результатом створення та розробки є універсальний web-застосунок, який можна адаптувати під різні навчальні заклади. Було створено оптимізовані алгоритми для швидкої роботи web-сайту. Даний web-застосунок має сучасний та інтуїтивно зрозумілий дизайн, що спрощує пошук інформації користувачам.

Ключові слова: web-застосунок, Visual Studio Code, VS Code, React, React.js., web-дизайн, web-сайт.

ABSTRACT

I.P. Shevchenko. Creating a Web application with React tools in the Visual Studio Code development environment.

Qualification work for obtaining a bachelor's degree in the specialty 121 «Computer Science» – University of Customs and Finance, Dnipro, 2023.

This qualification work is devoted to the development of an adaptive web application for an educational institution, which can be used to create websites of various educational institutions. General concepts of web applications, their relevance and development are considered. Authoritative sites that provided the basic plan for creating a universal web application were studied. The project was implemented using React library tools in the Visual Studio Code development environment, which ensured high efficiency and flexibility in the development process. The use of React made it possible to create a dynamic and interactive web application with the ability to quickly adapt to the specific needs of each educational institution.

The result of creation and development is a universal web application that can be adapted to different educational institutions. Optimized algorithms were created for fast website operation. This web application has a modern and intuitive design, which simplifies the search for information for users.

Keywords: web application, Visual Studio Code, VS Code, React, React.js., web design, website.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.2 Актуальність задачі реалізації адаптивного web-застосунку.....	11
1.2.1 Типи web-сайтів	14
1.2.2 Класифікація web-сайтів.....	16
1.2.3 Підхід для створення адаптивного для різних навчальних закладів web-сайту.....	17
1.3 Аналіз подібних Web-застосунків.....	18
1.3.1 Аналіз web-сайту Університету Митної Справи та Фінансів	19
1.3.2 Аналіз web-сайту Дніпровського Технічно-Економічного Фахового Коледжу.....	22
1.4 Використання web-застосунків	24
1.4.1 Використання web-сайту Київського Політехнічного Інституту	25
РОЗДІЛ 2. РОЗРОБКА КОДУ ПО МАКЕТУ З FIGMA ІНСТРУМЕНТАМИ REACT У СЕРЕДОВИЩІ VISUAL STUDIO CODE	29
2.1 Особливості роботи з бібліотекою React	29
2.2 Аналіз особливостей використання Framework React.js.....	31
2.3 Середовище розробки Visual Studio Code	36
2.4 Особливості проектування програмного продукту у середовищі Visual Studio Code	38
РОЗДІЛ 3. РЕАЛІЗАЦІЯ КОДУ ПО МАКЕТУ З FIGMA ІНСТРУМЕНТАМИ REACT.....	41
3.1 Створення React-компонентів та сторінок.....	41
3.2 Налаштування маршрутизації React-Router.....	45
3.3 Підключення сторонніх бібліотек для інтерактивної взаємодії користувача.....	47
3.4 Результат роботи web-застосунку.....	50

	5
ВИСНОВОК.....	57
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	61

ВСТУП

Актуальність розробки. Web-застосунки навчальних закладів є невід'ємною частиною навчального процесу, дозволяючи будь-яким закладам освіти ефективно взаємодіяти зі студентами, абітурієнтами, школярами, викладачами та іншими співробітниками. Вони надають зручний доступ до різноманітних освітніх ресурсів, сприяючи підвищенню якості навчання та зручності управління освітнім процесом.

Однією з ключових переваг web-застосунків для навчальних закладів є їхній потенціал підвищити залученість школярів або студентів і покращити їхній досвід навчання. За допомогою таких застосунків учні можуть легко отримувати доступ до навчальних матеріалів, виконувати завдання та отримувати оцінки в онлайн-форматі, що сприяє активній участі та інтерактивному навчанню.

Крім того, web-застосунки дозволяють покращити комунікацію між учасниками освітнього процесу. Вони забезпечують засоби для швидкого обміну інформацією між студентами та викладачами, організацію онлайн-конференцій, вебінарів та інших подій, що сприяють зміцненню спільноти навчального закладу.

Ще однією важливою складовою web-застосунків для навчальних закладів є їхня здатність забезпечувати індивідуалізований підхід до навчання. Завдяки таким застосункам студенти можуть мати доступ до персоналізованих навчальних матеріалів, дистанційних курсів та інших ресурсів, що відповідають їхнім потребам та інтересам.

Розвиток і використання web-застосунків у навчальних закладах сприяє підвищенню ефективності освітнього процесу, забезпечує широкі можливості для залучення студентів та сприяє розвитку сучасного освітнього середовища.

Відповідно до матеріалу з наукової публікації (статті) Пономаренко І.В «Веб-сайт як ефективний інструмент інтернет-маркетингу для досягнення стратегічних цілей компанії» [1], можна виділити багато важливої інформації щодо ринку web-застосунків у сьогоденні.

Загалом, сучасне функціонування web-сайтів нерозривно пов'язане з активною комунікацією з цільовою аудиторією через Інтернет. Особливо це стає актуальним у контексті швидкого розвитку цифрового маркетингу, коли навчальні заклади змушені впроваджувати інноваційні технології для створення власних унікальних дизайнів і web-сторінок [2].

Для досягнення успіху у цьому напрямку, важливо проводити дослідження різних джерел і використовувати аналітичні дані, щоб максимально ефективно використовувати web-сайт як потужний інструмент для покращення комунікації та подачі інформації.

Зараз спостерігається активний розвиток передачі інформації в Інтернеті. Впровадження інноваційних технологій призводить до трансформації web-сайтів та форумів, які стають важливими інструментами для різних навчальних закладів.

Основними каналами комунікації підприємств зі своєю аудиторією в Інтернеті є web-сайти та сторінки у соціальних мережах. Кожен з цих каналів має свої переваги та недоліки, тому під час розробки web-сайту важливо чітко встановити доцільність їх застосування і відповідно обрати цілі та функції для даного web-застосунку [3].

Інструментом для розробки є бібліотека React.

Одним із найкращих варіантів для створення web-застосунку є використання бібліотеки React. Вона є потужним інструментом для розробки інтерактивних і динамічних web-інтерфейсів. Дана бібліотека дозволяє створювати зручні та ефективні інтерфейси користувача, що забезпечують приємний досвід безпосередньо користувача під час використання web-застосунку [4].

Крім того, React базується на ідеї компонентної архітектури, що спрощує розробку, тестування і підтримку web-застосунків. Можна створювати невеликі і повторно використововані компоненти, що сприяє швидкості і зручності розробки [5].

React дозволяє створювати динамічні інтерфейси з асинхронним оновленням сторінок, що особливо корисно для навчальних застосунків, які потребують живої взаємодії з користувачем, наприклад, відправлення запитів, отримання даних і оновлення вмісту без перезавантаження сторінки.

Швидкість React дозволяє створювати швидкі застосунки. Це є важливою характеристикою для створення web-сайту навчального закладу, що має обробляти великий обсяг даних та інтерактивних операцій.

Також React забезпечує можливості інтеграції з іншими сервісами та API, що розширює функціональність web-застосунку та полегшує його використання. При розробці власного web-застосунку на основі React для навчального закладу важливо не лише спланувати архітектуру і визначити функціональні вимоги, але й створити привабливий інтерфейс, відповідний потребам користувачів, забезпечити тестування і підтримку для забезпечення стабільної роботи застосунку[6].

Середовищем розробки є Visual Studio Code.

Вибір Visual Studio Code для розробки web-застосунку на базі React є досить популярним і обґрунтованим вибором. VS Code – це потужний, безкоштовний редактор коду, який здобув широку популярність серед програмістів та розробників завдяки своїм функціональностям та можливостям [7].

Однією з переваг VS Code є його легкість використання і налаштування. Він постачається зі значною кількістю корисних розширень (extensions) та інструментів, які спрощують розробку застосунків. Наприклад, розширення для React та JSX надають підтримку автодоповнення, перевірку синтаксису, аналіз коду та інші корисні функції, які допомагають зробити розробку ефективнішою та швидшою.

Крім того, VS Code підтримує інтеграцію з різними інструментами розробки, такими як Git для контролю версій коду, розширення для роботи з Docker контейнерами, можливість налаштування зовнішніх інструментів для відлагодження (debugging) та автоматичного тестування коду [8].

Ще однією суттєвою перевагою VS Code є його активна спільнота користувачів та розробників, яка постійно створює нові розширення та плагіни, що розширюють функціональність редактора. Це дозволяє легко знаходити рішення для будь-яких задач розробки, включаючи роботу з React та створення web-застосунків.

Отже, використання бібліотеки React є одним із найкращих варіантів, оскільки вона дозволяє створювати зручні та ефективні інтерфейси користувача з використанням компонентної архітектури, динамічних оновлень сторінок і швидкості реагування. Для успішної реалізації такого проєкту, Visual Studio Code є оптимальним інструментом, оскільки він поєднує у собі потужність, зручність і можливості розширення, що необхідні для створення високоякісного web-застосунку.

Метою роботи є реалізація універсального web-застосунку за допомогою інструментів бібліотеки React у середовищі розробки Visual Studio Code.

У відповідності до поставленої мети в даній роботі ставилися та вирішувалися наступні завдання дослідження:

1. Розглянути класифікацію та типи web-сайтів.
2. Проаналізувати сайти-авторитети, що будуть прикладами для створення універсального web-застосунку.
3. Створення власного сценарію для реалізації універсального web-сайту для навчальних закладів.
4. Безпосередньо реалізація проєкту з використанням інструментів бібліотеки React у середовищі розробки Visual Studio Code.
5. Аналіз виконаної роботи.

Об'єктами дослідження є web-застосунки сайтів-авторитетів.

Предметом реалізації є інструменти бібліотеки React у середовищі розробки Visual Studio Code.

Практичне значення отриманих результатів – створення універсального web-застосунку, який можна адаптувати під інший навчальний заклад.

Структура роботи:

Розділ 1 Аналіз предметної області.

Розділ 2 Розробка коду по макету з figma інструментами react у середовищі visual studio code.

Розділ 3 Реалізація коду по макету з figma інструментами react.

Робота складається зі вступу, 3-х розділів, висновків, списку використаних джерел із 45 найменувань. Загальний обсяг роботи 65 сторінок, 33 рисунків.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність задачі реалізації адаптивного web-застосунку

Сучасне функціонування web-сайтів, загалом, передбачає активну комунікацію з цільовою аудиторією, переважно через Інтернет. Відбуваючись в умовах активної діджиталізації маркетингу, навчальні заклади мають впроваджувати інноваційні технології для розроблення власних дизайнів і сторінок. Для досягнення цих цілей важливо приділяти увагу дослідженню різних джерел за допомогою дослідження web-сайту як ефективного інструменту для покращення подачі інформації [9].

У сучасних умовах активізується розвиток передачі інформації в Інтернеті. Інтенсивне впровадження інноваційних технологій призводить до трансформації web-сайтів і форумів, основні інструменти яких широко застосовуються різними навчальними закладами [10].

Основними каналами комунікації підприємств з цільовою аудиторією в Інтернеті є web-сайти та головних сторінок і сторінок у соціальних мережах. Кожен з цих каналів має свої переваги та недоліки, тому під час розроблення web-сайту важливо чітко встановити доцільність застосування і відповідним чином визначити цілі та функції даного web-застосунку.

Створення web-застосунку для навчального закладу різного рівня освіти в Україні є доволі актуальною задачею в сучасному контексті з багатьох причин. Це відповідає наростаючим потребам і вимогам сьогоденного суспільства, яке все більше використовує цифрові технології у всіх сферах життя. Використання web-застосунків спрощує доступ до інформації та сприяє ефективній комунікації між різними учасниками освітнього процесу [11].

Крім того, web-застосунки дозволяють оптимізувати та автоматизувати значну частину освітнянських процесів в навчальному закладі, що полегшує роботу адміністраторів, викладачів та студентів або учнів. Вони

забезпечують зручний і доступний інтерфейс для взаємодії з різними сервісами та ресурсами, такими як розклад занять, матеріали для навчання, електронні бібліотеки тощо.

Web-сайт дозволяє покращити внутрішні комунікації в середовищі навчального закладу. Він створює можливості для взаємодії між студентами, викладачами і адміністрацією через онлайн-форуми, чати та електронні форми зворотнього зв'язку.

Також, web-застосунок для навчального закладу виступає як потужний інструмент для маркетингу та набору нових учнів. Він може вмістити інформацію про освітні програми, напрями класів, шкільні проєкти, будь-які інші можливості та різні привабливі аспекти навчального закладу.

Загалом, створення web-застосунку для навчального закладу можна віднести до стратегічного кроку для покращення якості освіти, підвищення ефективності управління та підвищення привабливості навчального закладу для студентів та інших учасників освітнього процесу [12].

Створення універсального web-сайту для навчального закладу, який відповідатиме потребам різних типів освітніх установ, є складним завданням, що вимагає глибокого аналізу та планування. Основний підхід полягає в тому, щоб створити дизайн та функціональність, які будуть зрозумілі й корисні для різних категорій користувачів.

Важливо зазначити, що такий web-застосунок буде корисним саме для тих закладів освіти, у яких немає фінансування чи можливості створити власний web-сайт для навчального закладу. Таким чином, використавши універсальний варіант, у них з'явиться можливість легко адаптувати під свій навчальний заклад і додати туди всю необхідну інформацію.

Аналіз потреб користувачів є ключовим етапом. Варто визначити, які групи користувачів будуть використовувати сайт (учні, учителя, батьки, адміністрація) і які їхні очікування від web-сайту. Це визначить структуру та зміст сайту.

Сегментація контенту грає важливу роль у web-сайті для навчального закладу. Web-сайт повинен включати розділи, які відповідають різним потребам користувачів: навчальні програми, розклад занять, матеріали для самостійної, лабораторної чи контрольної роботи, новини, контактна інформація та інше.

Однією з важливих характеристик є адаптивний дизайн. Web-сайт повинен оптимально відображатися на різних пристроях (комп'ютерах, планшетах, смартфонах), щоб забезпечити комфортний перегляд користувачами.

Простота та доступність є ключовими аспектами успішного web-сайту для навчального закладу. Інтерфейс повинен бути інтуїтивно зрозумілим для всіх категорій користувачів [13].

Інтерактивність та залучення користувачів також мають велике значення. Включення елементів інтерактивності, таких як форуми, чати, можливість відправки запитів або звернень, збільшує відчуття взаємодії та зацікавленості користувачів.

Безпека та конфіденційність даних – це один із важливих аспектів. Важливо забезпечити захист персональної інформації користувачів, особливо у випадках, коли це стосується неповнолітніх [14].

Інтеграція з іншими системами, такими як системи управління навчальним процесом, електронні бібліотеки чи системи дистанційного навчання, може зробити web-сайт більш функціональним і зручним для користувачів [15].

Загальний підхід до створення універсального web-сайту для навчального закладу полягає в поєднанні зручного дизайну, доступності для різних користувачів та врахуванні специфіки освітнього процесу для досягнення успішного впровадження і використання сайту.

1.1.1 Типи web-сайтів

Web-сайти для навчальних закладів є, загалом, важливою складовою освітнього середовища, надаючи широкий спектр інтернет-ресурсів для підтримки навчання, спілкування та спільноти учнів, викладачів та батьків. Ці ресурси розроблені з метою полегшення доступу до освітніх матеріалів та інформації, сприяння ефективній комунікації всередині навчальної спільноти та забезпечення високоякісного навчального досвіду. Загальна схема видів web-сайтів поділяється на гілки: за технологією, за власником, за наповненням, за призначенням (див. рис. 1) [16].

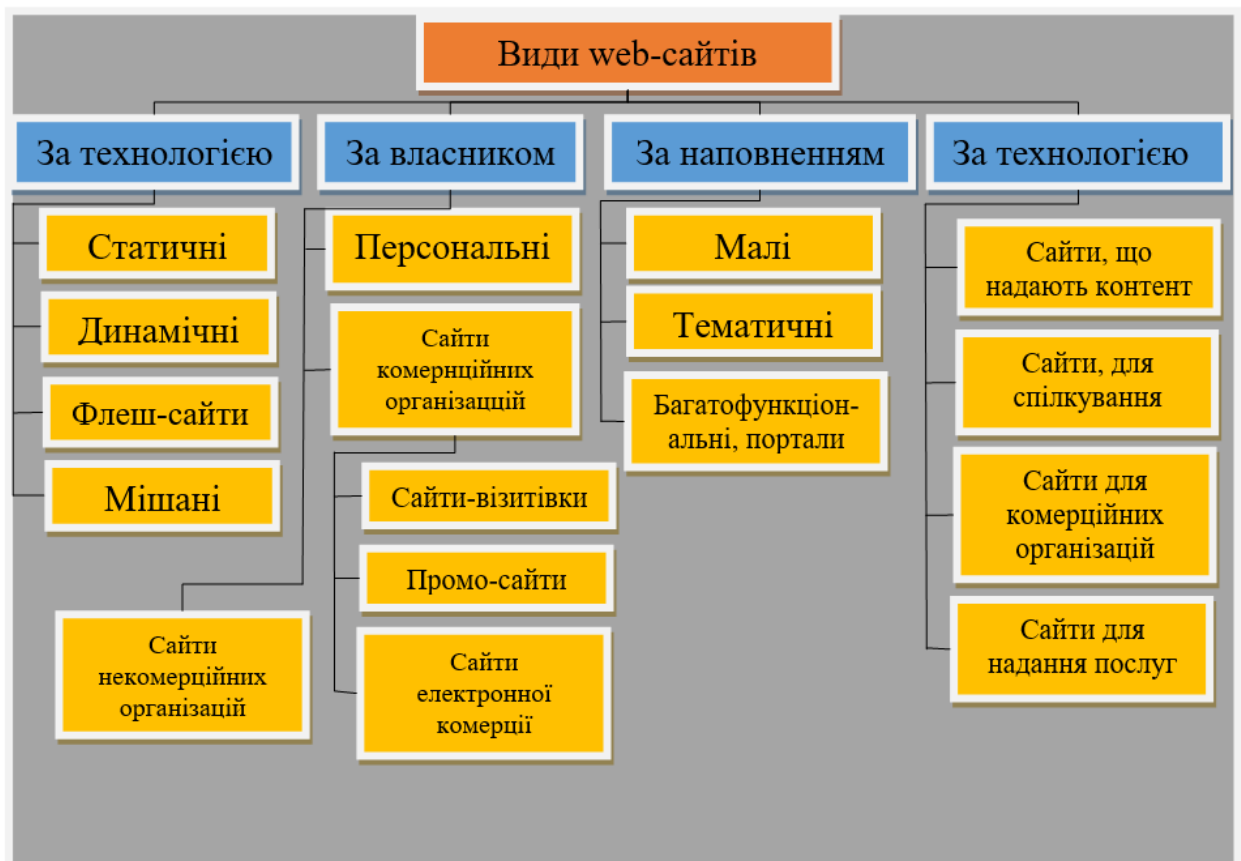


Рисунок 1 – Схема «Види web-сайтів»

Основні типи web-сайтів для навчальних закладів включають офіційні web-ресурси установ, які містять загальну інформацію про навчальний заклад, його програми, правила та контактну інформацію. Ці сайти є

візитною карткою навчального закладу і дозволяють потенційним студентам, батькам та іншим зацікавленим особам отримати повну інформацію про нього [17].

Платформи електронного навчання є ще одним типом web-сайтів для навчальних закладів. Вони призначені для організації дистанційного навчання, надання доступу до навчальних матеріалів, завдань та тестів, а також для спілкування з учителями та іншими учасниками навчального процесу [18].

Деякі web-сайти для навчальних закладів спеціалізуються на забезпеченні зв'язку з громадськістю та батьками. Вони надають інформацію про події, розклади зустрічей, поради щодо освітнього процесу та інше, щоб батьки були інформовані про життя навчального закладу [19].

Також існують web-сайти для демонстрації можливостей навчального закладу, які містять інформацію про програми, напрями класів, навчальні програми та відгуки випускників. Ці ресурси спрямовані на привертання нових учнів та популяризацію навчального закладу серед потенційних абітурієнтів [20].

Загальна спрощена схема типів сайтів виділяє 5 типів (див. рис. 2).

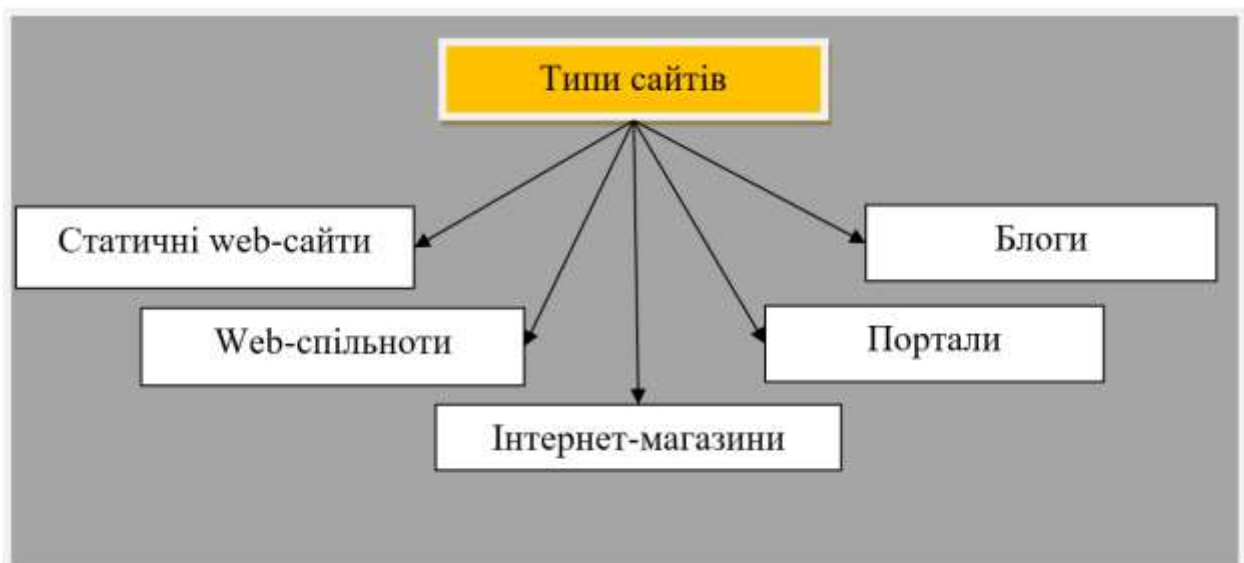


Рисунок 2 – Загальна схема типів сайтів

Загалом, web-сайти для навчальних закладів мають різноманітні функції та завдання, спрямовані на підтримку освітнього процесу та забезпечення зручного інтерфейсу для користувачів. Вони відіграють важливу роль у полегшенні доступу до інформації, підтримці спільноти навчального закладу та покращенні якості навчання і виховання учнів [21].

1.1.2 Класифікація web-сайтів

Web-застосунки можна класифікувати за різними критеріями, такими як їх призначення, функціональність, характеристики та тип власності. Класифікація web-сайтів для навчальних закладів спеціалізується на особливостях, які є специфічними для освітньої сфери [22].

Основні типи класифікації web-сайтів для навчальних закладів включають: за типом освітнього закладу, за призначенням, за функціональністю, за технічними характеристиками, за спеціалізацією.

Web-сайти можуть бути призначені, загалом, для університетів, коледжів, шкіл, дошкільних навчальних закладів або інших освітніх установ залежно від рівня та типу освіти, яку вони надають.

Web-сайти можуть служити для різних цілей, таких як інформаційні ресурси про навчальний заклад, платформи електронного навчання (LMS), web-портали для учнів, вчителів та батьків, а також web-сайти для набору нових учнів.

Вони можуть мати різноманітні функції, включаючи доступ до навчальних матеріалів, курсів, розкладів занять, організацію спілкування та співпраці між учасниками навчального процесу .

Web-сайти можуть бути класифіковані за їх технічними характеристиками, такі як типи платформ, використовувані технології, адаптивність для різних пристроїв. Приклади технологій для розробки web-застосунків: HTML/CSS, JavaScript, React, PHP тощо [23].

Web-сайти можуть спеціалізуватися на конкретних освітніх програмах, додаткових послугах, спортивних чи культурних заходах в навчальному закладі.

Класифікація web-сайтів для навчальних закладів допомагає визначати їхню специфіку і відповідність потребам користувачів, забезпечуючи зручний доступ до інформації та покращення комунікаційних процесів всередині навчальної спільноти. Кожен тип web-сайту має свої особливості та вимоги до дизайну, функціональності та інтерфейсу, спрямовані на досягнення різних, або конкретних цілей відповідно створеного освітнього закладу.

1.1.3 Підхід для створення адаптивного для різних навчальних закладів web-сайту

Створення універсального web-сайту для навчальних закладів відкриває можливості адаптації під різні типи освітніх установ, зокрема школи, які не мають можливостей для самостійної розробки та підтримки web-сайтів [24].

Перш за все, універсальний підхід дозволяє широко використовувати загальні модулі та функціонал для різних шкіл. Наприклад, система управління розкладом занять чи віртуальні класні кімнати можуть бути стандартизовані та легко адаптовані до конкретних потреб будь-якої школи.

Друге, уніфікований web-сайт дозволяє школам заощадити ресурси, необхідні для розробки та підтримки власного web-простору. Вони можуть скористатися готовими рішеннями, що вже мають усі необхідні функції, і зосередитися на інших аспектах освітнього процесу.

Крім того, універсальний web-сайт може бути простим у використанні і зміні, навіть для тих закладів, які не мають технічних спеціалістів у своєму складі. Школи можуть легко модифікувати зміст і структуру сайту, використовуючи інтуїтивний інтерфейс адміністрування.

Уніфікований підхід дозволяє створювати спільні простори для обміну досвідом та ресурсами між різними школами. Це сприяє підвищенню якості освіти і створює сприятливе середовище для співпраці та інновацій у навчальній сфері.

1.2 Аналіз подібних Web-застосунків

Розробка web-сайту для навчального закладу є складним процесом, який передбачає ретельне пропрацювання та аналіз різних варіантів дизайну та функціоналу. При розробці такого проєкту зазвичай розглядаються різні ідеї та концепції, а потім обирається оптимальний варіант, що відповідає потребам спільноти даних навчальних закладів і цілям навчального закладу в цілому.

Головною задачею при створенні універсального адаптивного web-застосунку є розробка відповідної структури, що дозволить інтуїтивно користуватися web-застосунком і дозволить легко підлаштовувати його під будь-який інший web-сайт.

У процесі вивчення та аналізу різних варіантів сторінок навчальних закладів враховуються такі фактори, як дизайн і користувацький досвід. Важливо розглядати різні стилі дизайну та інтерфейси, які забезпечать зручність користувачів при взаємодії з сайтом. Це може бути класичний, консервативний дизайн або сучасний, стильний інтерфейс з використанням останніх трендів у дизайні web-сайтів.

Функціональність також є важливим аспектом. Варіанти сторінок повинні враховувати потреби різних категорій користувачів, таких як студенти, викладачі, абітурієнти, батьки та інші. Наприклад, на сайті можуть бути розділи з розкладами занять, інформацією про факультети та програми, новинами та подіями університету, онлайн-курсами, бібліотечними ресурсами тощо.

Також важливо враховувати адаптивність. Оскільки багато користувачів переглядають сайти на мобільних пристроях, web-сайт повинен мати адаптивний дизайн, що забезпечить коректне відображення сторінок на різних пристроях і розмірах екранів.

1.2.1 Аналіз web-сайту Університету Митної Справи та Фінансів

Розглядаючи багато різних варіантів, найголовнішим авторитетом для наслідування був Університет Митної Справи та Фінансів. У нього є багато важливих тем для дослідження.

Дизайн web-сайту УМСФ грає важливу роль у візуальному сприйнятті користувачів. Важливо оцінити його чистоту, зручність навігації, кольорову палітру, використання шрифтів та загальний стиль. Дизайн є збалансованим, професійним і привабливим для аудиторії. Також, він відповідає корпоративному образу університету.

Технічна частина дозволяє всім користувачам переходити за будь-якими посиланнями без жодних затримок. Це пояснюється ефективним написанням коду даного web-застосунку та сучасного обладнання для підтримки роботи серверу.

У web-сайті Університету Митної Справи та Фінансів присутній ряд функціональних складових, які сприяють зручності користувачів і забезпечують доступ до різноманітних освітніх ресурсів. Для створення досить універсального web-сайту для інших різних навчальних закладів, необхідно виділяти основні елементи, що будуть фігурувати також у різних інших web-застосунках. Такі структури будуть досить інтуїтивно зрозумілі користувачам і будуть спрощувати використання web-сайту для навчального закладу.

Основні елементи web-сайту включають початкову сторінку, меню опцій, розділ новин та статистичні дані. Кожна з цих складових доповнює функціонал усього web-сайту та гармонійно поєднує це все привабливим

для будь-яких користувачів дизайном, що дає змогу інтуїтивно знаходити необхідну інформацію.

Початкова сторінка УМСФ (див. рис. 3) є важливим елементом веб-сайту, який першим зустрічає користувача. Вона, як у всіх інших випадках, повинна бути привабливою та інформативною, надаючи швидкий доступ до основних розділів та ресурсів.

На цій сторінці можуть бути розміщені актуальні новини, оголошення або події університету, що дозволяє користувачам швидко ознайомитися з важливою інформацією.



Рисунок 3 – Головна сторінка веб-сайту УМСФ

Меню опцій веб-сайту УМСФ (див. рис. 4) розширює можливості користувачів для навігації та пошуку потрібної інформації.



Рисунок 4 – Меню опцій веб-сайту

Через це меню користувачі можуть отримати доступ до різних розділів, таких як «Про нас», «Структура», «Вступ. освіта», «Наука», «Студентське життя», «International activities».

Меню забезпечує легкий доступ до різних ресурсів та інформації, що підвищує зручність взаємодії з web-сайтом.

Розділ новин на web-сайті (див. рис. 5) УМСФ містить актуальні події, оголошення та інші новини з життя університету. Цей розділ є важливим для сповіщення користувачів про останні події та розвиток університетського життя. Користувачі можуть швидко отримувати новини через цей розділ.

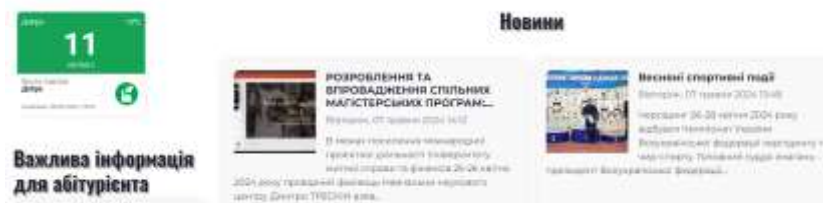


Рисунок 5 – Новини сайту УМСФ

Статистичні дані (див. рис. 6), які представлені на web-сайті УМСФ, можуть включати інформацію про кількість студентів, викладачів, напрями навчання, результати досліджень та інші важливі показники. Ця інформація дозволяє користувачам отримати уявлення та діяльність університету, підвищуючи прозорість і доступність інформації про навчальний заклад.

Набір на місце державного замовлення ЗВО м. Дніпра
за даними vstup.edu.gov.ua
Бакалавр

Вид навч. освіти	2019		2020		2021		2022		2023	
	місц.	рейтинг	місц.	рейтинг	місц.	рейтинг	місц.	рейтинг	місц.	рейтинг
Національний технічний університет "Дніпровська політехніка"	818	1	891	1	842	1	898	1	798	1
Дніпровський національний університет імені Олеся Гончара	665	2	665	2	761	2	768	2	470	2
Гірничий державний університет ім.і. Гоголя	0		0		0		571	3	513	3
Криворізький національний університет	406	4	387	3	285	5	491	4	378	4
Дніпровський державний аграрно-економічний університет	323	6	274	6	383	6	382	6	333	5
Криворізький державний педагогічний університет	473	3	379	4	557	3	589	5	312	7
ДНУЗ "Придніпровський державний академічний університет та професійний коледж"	271	7	370	7	223	10	294	7	308	6
Дніпровський державний технічний університет	184	11	188	11	183	11	204	10	226	9
Дніпровський державний педагогічний університет	202	10	216	10	224	9	251	9	212	8
ДНУЗ "Хмельницький державний економіко-технологічний університет"	233	9	220	9	298	8	310	8	182	10
Державний університет економіки і технологій	0		128	12	123	12	136	11	149	11
Дніпровський державний академічний університет культури і спорту	48	12	84	13	80	13	80	12	101	12
Національна академія керівних кадрів України	328	5	380	5	332	4				
Дніпровський національний університет економіки і технологій ім. академіка В. Лазаренка	368	8	291	8	279	7				

Рисунок 6 – Статистичні дані УМСФ

Ці функціональні компоненти web-сайту УМСФ сприяють забезпеченню ефективної комунікації зі студентами, викладачами та іншими зацікавленими особами. Вони створюють зручний інструмент для навігації та отримання необхідної інформації про університет.

1.2.2 Аналіз web-сайту Дніпровського Технічно-Економічного Фахового Коледжу

Основний варіант повинен відповідати найкращим практикам у web-розробці, відображати індивідуальність університету та забезпечувати ефективну реалізацію всіх потрібних функцій і можливостей.

Такий підхід до розробки web-сайту дозволяє створити продукт, який відповідає всім потребам університетської спільноти, підвищує рівень комунікації і забезпечує зручний доступ до інформації і ресурсів. Вибір додаткового основного варіанту сторінки – це крок у напрямку сучасного та інноваційного підходу до представлення університету в онлайн-середовищі.

Додатковим основним прикладом для створення web-застосунку інструментами React у середовищі розробки Visual Studio Code було обрано web-сайт Дніпровського Технічно-Економічного Фахового Коледжу (<https://dtek.dp.ua/>) (див. рис. 7).



Рисунок 7 – Головна сторінка ДТЕК

Даний web-застосунок є важливим інформаційним ресурсом, що містить розклад занять, актуальні новини, інформацію про вступ і події, що

спрямовані на підвищення ефективності комунікації в університетському середовищі.

Основна актуальність створення подібного web-сайту полягає в підвищенні взаємодії між учасниками освітнього процесу. Він створює платформу для спілкування, обміну інформацією та розвитку академічного середовища (див. рис. 8).

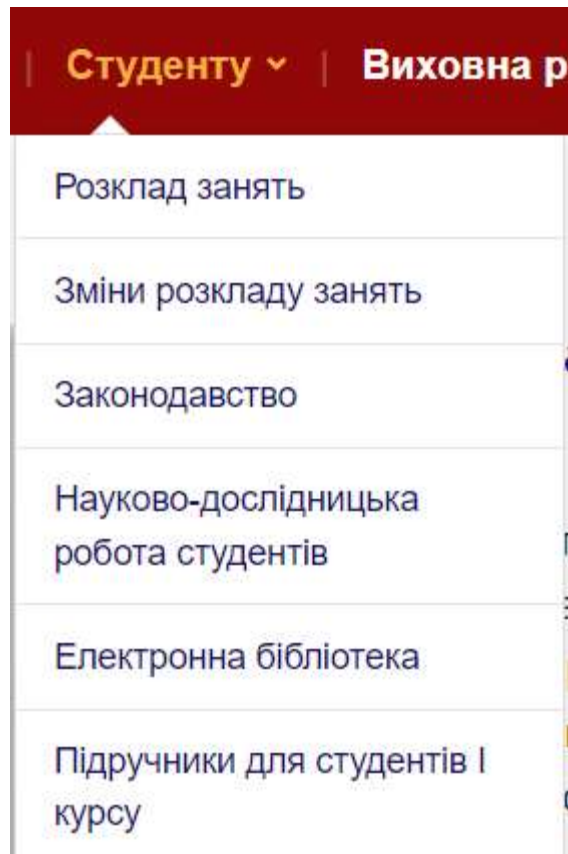


Рисунок 8 – Вкладка «Студенту» з різними опціями

Крім того, web-сайт є ефективним інструментом маркетингу та рекрутингу нових студентів. Він презентує інформацію про освітні програми, факультети, можливості стипендій та допомагає в адаптації до університетського життя.

Створення сучасного web-сайту для університету, як у прикладі Дніпровського Технічно-Економічного Фахового Коледжу, є важливим кроком для підвищення якості освіти, покращення управління та залучення

нових студентів. Цей інноваційний підхід демонструє прогресивний розвиток університету та його готовність до викликів сучасного освітнього середовища.

1.3 Використання web-застосунків

Web-застосунки стали невід'ємною складовою сучасної освіти, привносячи значні переваги для університетських спільнот. Однією з ключових переваг є зручний доступ до різноманітної інформації для студентів, викладачів, абітурієнтів і адміністраторів. Web-застосунки дозволяють легко отримувати розклад занять, результати, матеріали для навчання, оголошення та інші важливі дані, що сприяє підвищенню продуктивності і організованості університетського життя.

Крім того, web-застосунки стимулюють покращення комунікації всередині університетського середовища. Вони забезпечують зручні інструменти для обміну повідомленнями, спілкування в групах, проведення обговорень та співпраці над проектами. Це сприяє покращенню взаєморозуміння, співпраці та колективної роботи всіх учасників освітнього процесу.

Оптимізація управління є ще однією вагомою перевагою web-застосунків університетів. Вони автоматизують адміністративні процеси, такі як реєстрація студентів, контроль навчального процесу, ведення обліку успішності та інші аспекти управління. Це дозволяє ефективно використовувати ресурси та підвищує ефективність адміністрування університету.

Крім підвищення ефективності, web-застосунки сприяють інноваціям у навчальному процесі. Вони дозволяють впроваджувати нові технології, створювати інтерактивні навчальні матеріали, підтримувати сучасний підхід до навчання і забезпечувати доступ до онлайн-ресурсів. Це сприяє розвитку креативності та підвищенню якості освіти.

Не останню роль грає глобальний доступ і гнучкість web-застосунків. Вони забезпечують можливість навчання і співпраці студентів та викладачів з будь-якої точки світу, надаючи гнучкість у часі та місці занять. Це розширює можливості університету залучати студентів з різних країн і стимулює міжнародний обмін досвідом і знаннями.

1.3.1 Використання web-сайту Київського Політехнічного Інституту

Яскравим прикладом використання web-сайтів для університету є сайт одного з найвідоміших університетів України – Київського Політехнічного Інституту імені Ігоря Сікорського (<https://www.kpi.ua/>) (див. рис. 9).

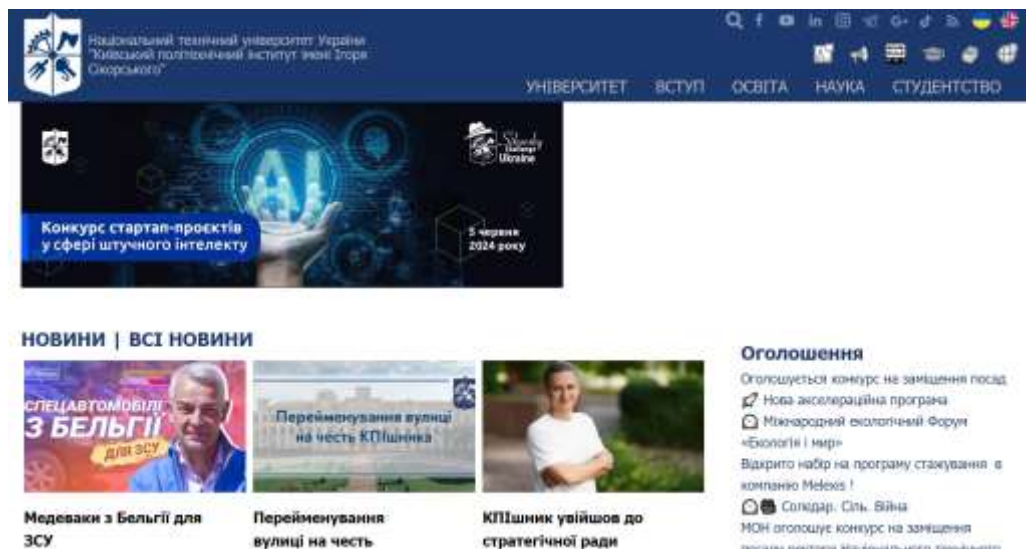


Рисунок 9 – Головна сторінка КПІ

Він користується великою популярністю серед користувачів з різних груп і має декілька переваг, зокрема щодо його інтерфейсу.

Інтерфейс сайту КПІ є досить інтуїтивно зрозумілим та зручним для використання. Головна сторінка містить чітку навігаційну структуру з легким доступом до основних розділів, таких як інформація про університет, факультети та кафедри, новини та події, а також важливі розділи для студентів, абітурієнтів та викладачів.

Дизайн сайту КПІ відповідає сучасним стандартам і виглядає привабливо. Використання приємних кольорів, чітких шрифтів і зображень допомагає створити зручне і привабливе візуальне сприйняття, що робить користування сайтом приємним і ефективним.

Даний web-сайт є універсальним джерелом інформації для всього університету. Кожен бажаючий абітурієнт може знайти інформацію щодо вступу (див. рис. 10).



Рисунок 10 – Вкладка вступу до КПІ

Інші корисні функції включають розклад занять, актуальні новини (див. рис. 11), результати сесії, розклад вступних випробувань та інше.

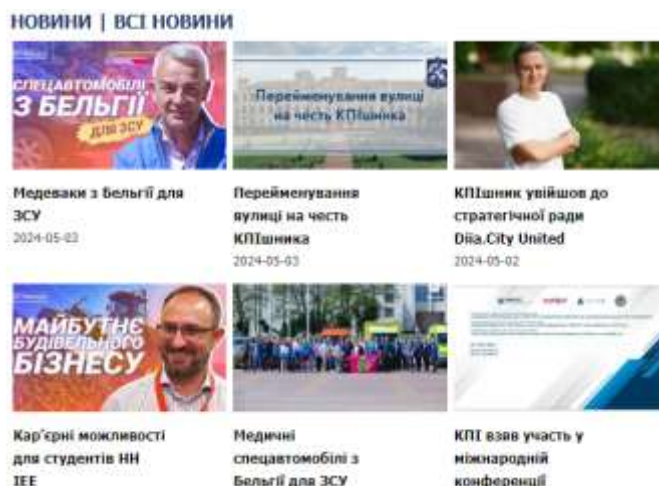


Рисунок 11 – Вкладка новин КПІ

Web-застосунок також має приклад привабливих зображень у розділі корисних посилань (див. рис. 12), що показують безпосередньо однойменні

опції. Такий спосіб створення web-сайтів дозволяє користувачам знаходити необхідну інформацію максимально легко.



Рисунок 12 – Корисні посилання

Одним із корисних підходів до створення привабливого дизайну та щоб заохотити користувача приєднатися до спільноти навчального закладу є привернення уваги цікавими науковими проєктами та ініціативами (див. рис. 13).



Рисунок 13 – Наукові проєкти та ініціативи КНУ

Узагальнюючи, web-застосунки для університетів не лише спрощують життя учасників освітнього процесу, а й відкривають широкі можливості для

покращення навчання, управління та розвитку освітньої сфери. Їх впровадження є ключовим етапом в модернізації середовища навчального закладу та підвищенні його конкурентоспроможності в сучасному світі.

Також важливо помітити, що більшість web-застосунків навчальних закладів, мають відносно схожий функціонал. Такі вкладки, як: вступ, новини, розклад занять, інформація про навчання, навчальні дисципліни, корисні посилання тощо, є майже в усіх сайтах. Можна зробити висновок, що для створення універсальної моделі web-застосунку навчального закладу, можна притримуватися вже готових варіантів сайтів, як прикладів для створення власного.

РОЗДІЛ 2. РОЗРОБКА КОДУ ПО МАКЕТУ З FIGMA ІНСТРУМЕНТАМИ REACT У СЕРЕДОВИЩІ VISUAL STUDIO CODE

2.1 Особливості роботи з бібліотекою React

Бібліотека React є однією з найпопулярніших технологій JavaScript, використовуваною для побудови користувацьких інтерфейсів. Її основна перевага полягає у компонентному підході до розробки, що дозволяє розбити складні інтерфейси на маленькі, повторно використовувані компоненти [25].

Однією з ключових переваг роботи з React є його компонентна архітектура. Цей підхід дозволяє розбити інтерфейс на невеликі, незалежні компоненти, які легко управляти окремо [26]. Такий підхід спрощує процес розробки, тестування і підтримки web-застосунків [27].

Другою важливою особливістю React є використання віртуального DOM (Document Object Model), що забезпечує ефективне оновлення інтерфейсу. Зміни спочатку відбуваються в віртуальному DOM, і тільки потім React маніпулює реальним DOM, що дозволяє досягти високої швидкодії та ефективності [28].

Третьою важливою особливістю є використання JSX (JavaScript XML) синтаксису, який дозволяє писати код, схожий на HTML, для опису структури компонентів. Це полегшує створення і розуміння інтерфейсів [29].

Крім того, React дозволяє створювати динамічні інтерфейси, які реагують на зміни даних без перезавантаження сторінки [30]. Це забезпечує більш плавну та інтерактивну взаємодію з користувачем.

У процесі роботи з бібліотекою React розробники зазвичай стикаються з рядом завдань, таких як створення компонентів, управління станом, рендерінг компонентів, обробка подій, та розуміння методів життєвого циклу компонентів для управління їх станом та відображенням [31].

Усі ці особливості та переваги роботи з бібліотекою React дозволяють розробникам побудувати ефективні, швидкі та масштабовані web-застосунки, які відповідають сучасним вимогам і забезпечують зручне взаємодію з користувачами [32]. Володіння навичками роботи з React є важливою складовою для фронтенд-розробників у сучасному IT-середовищі загалом.

Щоб розпочати роботу з бібліотекою React, перший крок – це встановлення необхідних інструментів і середовища розробки. Основними компонентами, які потрібно встановити для початку роботи з React, є Node.js і npm (Node Package Manager), які дозволяють керувати залежностями проєкту та встановлювати необхідні пакети [33].

Node.js – це середовище виконання JavaScript, яке дозволяє виконувати JavaScript поза браузером. Можна завантажити та встановити Node.js з офіційного web-сайту (nodejs.org). Після встановлення з'являється доступ до npm для керування пакетами [34].

Після встановлення Node.js та npm можна створити новий проєкт React. Для цього використовується команда `npm create-react-app`, яка створює нову структуру проєкту React з необхідними налаштуваннями та залежностями.

Після створення проєкту React можна перейти до його каталогу та встановити додаткові залежності. Для розробки React зазвичай використовуються деякі додаткові пакети, такі як `react-router-dom` для маршрутизації, `axios` для взаємодії з API, `redux` для керування станом тощо [35]. Встановлення цих пакетів здійснюється за допомогою команди `npm install package-name`.

Після встановлення залежностей розробник може запустити React проєкт. Зазвичай для цього використовують команду `npm start`, яка запускає розробний сервер і відкриває даний проєкт у браузері. Розробний сервер React автоматично відстежує зміни у вихідних файлах і перезавантажує сторінку при необхідності.

Після успішного запуску користувач готовий приступити до розробки React застосунку. Для роботи потрібно відкрити зручний редактор для

розробки, наприклад, Visual Studio Code, та можна починати редагувати компоненти, створювати нові файли та додавати функціональність до застосунку [36].

Встановлення бібліотеки React та налаштування середовища розробки, загалом, є першими кроками у створенні сучасних web-застосунків на основі React. Цей процес дозволяє швидко розпочати роботу з React і використовувати всі його переваги для створення потужних інтерфейсів користувача.

2.2 Аналіз особливостей використання Framework React.js

React.js є одним з найпопулярніших JavaScript фреймворків для створення інтерактивних інтерфейсів користувача. Він відомий своєю ефективністю, гнучкістю й широким спектром функціональних можливостей, які забезпечують зручну розробку сучасних web-застосунків [37].

Компонентна архітектура є однією з основних особливостей React.js. Фреймворк працює на основі розбиття інтерфейсу на невеликі незалежні компоненти, що дозволяє підтримувати структуру програми чистою, організованою і повторно використовуваною [38].

Важливою характеристикою React є використання віртуального DOM (Document Object Model), що дозволяє оптимізувати оновлення інтерфейсу. React зберігає віртуальну копію DOM у пам'яті і автоматично визначає оптимальний спосіб оновлення реального DOM на основі змін стану компонентів.

JSX синтаксис є ще однією важливою особливістю React.js. JSX дозволяє вбудовувати HTML-подібний код безпосередньо в JavaScript, що робить код React більш читабельним і зрозумілим, а також сприяє швидкішому розвитку інтерфейсів.

Управління станом і властивостями є ключовою концепцією в React. Компоненти можуть мати власний стан (state) і приймати властивості (props)

від батьківських компонентів, що дозволяє організувати інтерактивність та обмін даними між компонентами.

Односторонній потік даних є ще однією важливою концепцією React, де дані передаються зверху вниз (з батьківського до дочірнього компонента). Цей підхід спрощує відлагодження програми і дозволяє створювати чітко визначені інтерфейси.

React також має різні методи життєвого циклу компонентів, такі як `componentDidMount`, `componentDidUpdate`, `componentWillUnmount` тощо. Ці методи дозволяють виконувати певні дії в певні моменти життєвого циклу компонентів, наприклад, ініціалізувати дані після монтажу компоненту або видаляти підписи на події перед розмонтажем.

Широке співтовариство розробників і активна екосистема різноманітних бібліотек і інструментів сприяють швидкому розвитку застосунків на React. Це забезпечує доступ до багатьох корисних рішень, які допомагають вирішувати різні завдання під час розробки web-застосунків на цьому фреймворку.

React також відомий своєю активною підтримкою і постійними оновленнями. Розробники регулярно вдосконалюють фреймворк, додаючи нові функції, оптимізуючи продуктивність і покращуючи безпеку. Це робить React одним із переважних виборів для створення сучасних web-застосунків з багатофункціональним інтерфейсом.

Перед початком розробки web-застосунків важливо ретельно вивчити підходи до використання фреймворків і визначити особливості JavaScript, які стануть передумовами для успішної реалізації проєкту.

Існують два основних шляхи використання фреймворків. Перший – це `Multi Page Application (MPA)`, де кожна URL повертає окрему HTML-сторінку, згенеровану сервером. Додавання такого фреймворку, як `React.js`, може підсилити окремі сторінки, вимагаючи додаткового імпорту на кожній з них. Загалом, використання багатосторінкового підходу з додаванням

фреймворку передбачає створення інстанцій з кодом для кожної сторінки, де він застосовується.

Проте не кожен фреймворк підтримує такий підхід. Альтернативою є Single Page Application (SPA), коли сервер повертає лише одну сторінку для всіх URL, які можуть бути до нього надіслані. Ця сторінка містить JavaScript-застосунок, набагато більший за обсягом, ніж той, який був би доданий до окремих сторінок.

Для користувача це означає, що він отримує різні сторінки, проте зміни відбуваються за рахунок коду, а сторінка не перезавантажується. Усі дії відбуваються швидше, оскільки не потрібно чекати на завантаження нової сторінки для кожного шляху. Навіть у випадках, коли потрібно отримати або записати дані у базу, JavaScript виконує це «за кулісами», не змушуючи користувача залишати односторінковий web-застосунок.

Прикладами МРА можуть бути сайти типу Facebook, де кожна сторінка завантажується знову, в той час як прикладом SPA є сервіс Gmail, де сторінка завантажується лише один раз, а зміни відбуваються без перезавантаження.

Перевагами використання багатосторінкових застосунків є їх оптимізація під пошукові системи (SEO). Наприклад, Facebook вибрав такий підхід через необхідність індексації кожної сторінки пошуковими системами. Пошукові роботи Google зустрічають виклики при скануванні сторінок з великою кількістю JavaScript-коду, оскільки вони не можуть передбачити, яким чином змінюватиметься вміст односторінкового застосунку. Тому багатосторінкові застосунки, як МРА, стають кращим вибором, особливо коли потрібно забезпечити доступність не лише головної сторінки.

До переваг багатосторінкових застосунків також варто віднести їх віковість, що призводить до наявності багатьох шаблонів, посібників, готових рішень та розробок у сфері безпеки web-ресурсів. Крім того, існує велика ймовірність, що навіть при вимкненому JavaScript користувач отримає значну частину контенту сторінки, що не може бути сказано про застосунок, який повністю залежить від JavaScript.

Однак у багатосторінкових застосунках є свої недоліки. Наприклад, затримка у відгуку, оскільки кожне перезавантаження сторінки при кожному кліку може ускладнювати користування сайтом, особливо на мобільних пристроях.

Односторінкові застосунки мають ряд переваг, серед яких висока швидкість реагування на дії користувача. Оскільки не потрібно очікувати завантаження нових сторінок після кожного кліку, взаємодія відбувається миттєво, що робить користування приємним для клієнтів. Також слід зазначити, що в односторінкових застосунках краще відокремлюється бізнес-логіка від логіки представлення, що змушує розробників чітко розмежовувати фронтенд та бекенд.

Ще однією перевагою є можливість підтримки застосунку при тимчасовому відсутності підключення до Інтернету. Всі необхідні ресурси завантажуються одразу, тому будуть недоступні лише ті частини, які потребують зв'язку з сервером. Проте найбільшим недоліком є втрата переваги з оптимізації пошукових систем. Якщо це критично для застосунку, використання односторінкового підходу може стати складнішим, але не неможливим. Нові технології, які використовують попередньо зрендерені частини застосунку на сервері для оптимізації пошукових систем, з'являються щодня, але їхня реалізація вимагає додаткових зусиль.

Крім того, слід врахувати можливі проблеми з безпекою, оскільки інформації про цей патерн менше, і насторожує те, що застосунок не буде працювати без увімкненого JavaScript, що може вплинути на певний відсоток користувачів.

JavaScript є найбільш популярною мовою сценаріїв для створення скриптів у web-браузерах, які вбудовуються у web-сторінки. JavaScript, що є зареєстрованою торговою маркою Sun Microsystems, Inc., був створений для «оживлення» web-сторінок. Програми на JavaScript називаються скриптами. Вони можуть бути написані прямо в HTML-сторінки і автоматично запускатися при завантаженні сторінки. Скрипти постачаються і

зберігаються як простий текст і не вимагають спеціальної підготовки чи компіляції для запуску. У цьому відношенні JavaScript значно відрізняється від Java.

Початково JavaScript був названий «Живим скриптом», але через популярність Java на той час вирішили, що така назва допоможе зберегти нову мову як «молодшого брата» Java. Однак з часом JavaScript став повністю самостійною мовою зі своїм власним стандартом ECMAScript і тепер не має нічого спільного з Java.

JavaScript зазвичай використовується як мова сценаріїв клієнта, тобто код JavaScript вбудований безпосередньо у HTML-сторінки. Коли користувач запитує HTML-сторінку з JavaScript, сценарій надсилається браузеру, який повинен виконати відповідні дії. Оскільки сценарій розміщується у HTML, скрипти можуть бути доступні для перегляду і копіювання будь-яким користувачем, хто відвідує сторінку. Однак ця прозорість є перевагою, оскільки можете вивчати та використовувати JavaScript, знайдений на web-сторінках.

JavaScript можна використовувати не лише у web-браузерах, а й у різних контекстах поза ними. Наприклад, JavaScript на стороні сервера Netscape, спочатку створений як мова CGI, може виконувати ті ж завдання, що й Perl або ASP. Немає жодних причин, чому JavaScript не можна використовувати для написання складних програм. Однак на цьому web-сайті ми розглядаємо виключно використання JavaScript у web-браузерах.

В даний час JavaScript працює не лише у браузерах, а також на сервері та на різних пристроях за допомогою спеціальних програм, таких як JavaScript Mechanical. У браузері вбудовано двигун JavaScript, який іноді називають віртуальною машиною JavaScript. Різні браузери використовують різні движки зі своїми «кодовими назвами», наприклад: V8 – у Chrome і Opera; SpiderMonkey – у Firefox.

Сучасний JavaScript є «безпечною» мовою програмування. Він не надає низькорівневого доступу до пам'яті або процесора, оскільки був розроблений для використання у браузерях, де це не є необхідним.

Можливості JavaScript значно залежать від середовища, в якому він використовується. Наприклад, у середовищі Node.js JavaScript підтримує функції, які дозволяють читати або записувати файли, взаємодіяти з мережею та інше.

У браузері JavaScript може здійснювати різні дії, пов'язані з взаємодією на web-сайті, з користувачем та сервером, такі як: додавання нового HTML на сторінку, зміна вмісту, стилів; реагування на дії користувача (клік мишею, переміщення вказівника, натискання клавіш); відправлення мережових запитів на віддалені сервери, завантаження файлів (технології AJAX та COMET); робота з файлами cookie, виведення питань користувачеві, показ повідомлень; збереження даних на стороні клієнта (локальне зберігання).

Загалом, вибір React.js є чудовим для створення web-застосунків через його здатність ефективно керувати станом застосунку, пропонуючи компонентний підхід до розробки, що полегшує підтримку та розширення коду. React також забезпечує швидке оновлення інтерфейсу за рахунок віртуального DOM, що робить його ідеальним вибором для реалізації інтерактивних web-застосунків з великою кількістю динамічного контенту.

2.3 Середовище розробки Visual Studio Code

Visual Studio Code (VS Code) є одним з найпопулярніших та потужних середовищ розробки (IDE) для програмістів усіх рівнів. Його популярність пояснюється рядом переваг, що роблять його ідеальним вибором для роботи над різноманітними проєктами [39].

Почнемо з основних функцій даного середовища. VS Code відомий своєю легкістю встановлення та налаштування. Користувач може завантажити його швидко з web-сайту розробника, і встановлення зазвичай

займає хвилини. Після запуску можна побачити інтуїтивний інтерфейс з інструментами редагування коду, інтегрованим терміналом, та потужною панеллю розширень.

Однією з ключових переваг VS Code є його розширюваність. Можна додати різні розширення для покращення функціональності редактора, включаючи підтримку мов програмування, систем контролю версій, засоби відладки, та багато іншого. Широкий вибір розширень дозволяє налаштувати VS Code під потреби користувача.

Другою важливою особливістю є підтримка відладки прямо в середовищі. VS Code має вбудовані засоби відладки для різних мов, які дозволяють встановлювати точки зупинки, аналізувати змінні, та виконувати код по кроцях, що значно полегшує виявлення та виправлення помилок.

Крім того, VS Code надає розширену підтримку системи контролю версій, що дозволяє працювати з репозиторіями Git безпосередньо з редактора. Можна переглядати відгалуження, робити коміти, витягувати та злити зміни, що спрощує спільну роботу над проектами з командою.

Окрім цього, VS Code доступний для операційних систем Windows, macOS та Linux, і має ідентичний інтерфейс у всіх них, що полегшує адаптацію користувачів при зміні платформи. Програма має інтегрований термінал, який дозволяє взаємодіяти з командним рядком (CMD, PowerShell, Linux Bash) безпосередньо в середовищі розробки.

VS Code пропонує режими збереження файлів за замовчуванням (явне або автоматичне), а також великий вибір налаштувань зовнішнього вигляду, що дозволяє змінювати теми оформлення, розмір вікон та шрифтів.

Це потужне середовище підтримує редагування файлів і папок у складі проєктів, автоматично визначаючи тип проєкту за наявністю в папці певних файлів (таких як `package.json`, `tsconfig.json`), що дозволяє активувати багато корисних функцій, таких як IntelliSense та навігація по коду.

Усі ці функції та можливості роблять Visual Studio Code ідеальним інструментом для розробки web-застосунків, дозволяючи працювати продуктивно та зручно над проєктами будь-якої складності.

Не слід також забувати про широкий спектр інструментів підтримки для розробки web-застосунків. VS Code має підтримку HTML, CSS, JavaScript, а також популярних фреймворків та бібліотек, таких як React, Angular, Vue.js, що робить його ідеальним вибором для фронтенд-розробки.

Загалом, VS Code є потужним, легким у використанні та розширюваним середовищем розробки з багатьма корисними функціями, які спрощують та покращують роботу програмістів. Це відмінний інструмент для будь-якого розробника, хто працює з програмним кодом незалежно від їх досвіду та області розробки.

2.4 Особливості проєктування програмного продукту у середовищі Visual Studio Code

Особливості проєктування програмного продукту у даному середовищі охоплюють широкий спектр можливостей і інструментів, що дозволяють розробникам створювати, тестувати і оптимізувати програмні рішення з високою ефективністю і зручністю [40].

Visual Studio Code є повноцінним інтегрованим середовищем розробки, яке об'єднує в собі редактор коду, підтримку мов програмування, інструменти для роботи з версіями коду, засоби відлагодження та багато іншого. Його інтерфейс приємний для користування і підтримує велику кількість налаштувань (див. рис. 14). Це дозволяє новим користувачам, або тим, які давно не використовували дане середовище, швидко зорієнтуватися в усіх налаштуваннях то створювати власний продукт. Також, однією з основних переваг VS Code є його розширюваність. За допомогою розширень (extensions) розробники можуть розширити функціональність редактора,

додавши підтримку для різних мов програмування, фреймворків, інструментів розробки, аналізаторів коду та інших корисних інструментів.

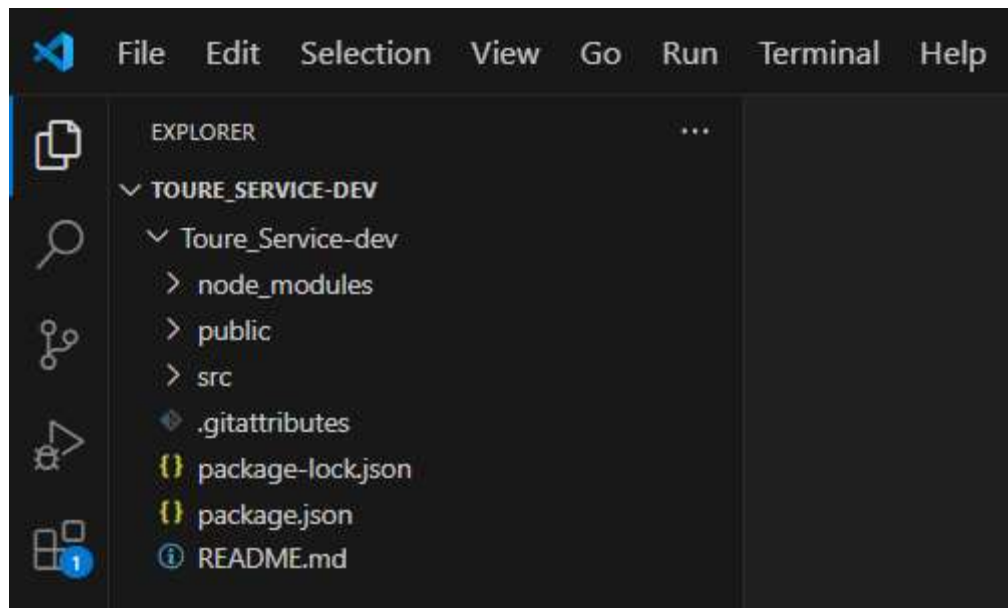


Рисунок 14 – Інтерфейс Visual Studio Code

Visual Studio Code підтримує спільну роботу над проєктами через розширення, які забезпечують інтеграцію з сервісами спільної роботи, такими як GitHub, GitLab, Bitbucket тощо (див. рис. 15).

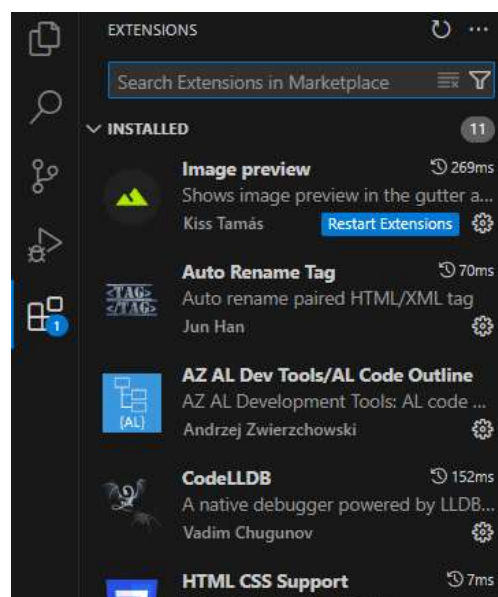


Рисунок 15 – Приклад розширень для VS Code

Visual Studio Code має підтримку популярних систем керування версіями, таких як Git. Це дозволяє розробникам працювати з репозиторіями, виконувати коміти, переглядати відмінності між версіями файлів та керувати гілками безпосередньо з інтерфейсу VS Code.

Відлагодження коду також є важливою частиною проєктування програмного продукту, і VS Code надає зручні інструменти для цього (див. рис. 16). За допомогою вбудованих засобів або розширень можна ставити точки зупинки, відстежувати значення змінних, виконувати крок за кроком виконання коду та вирішувати проблеми в програмі.

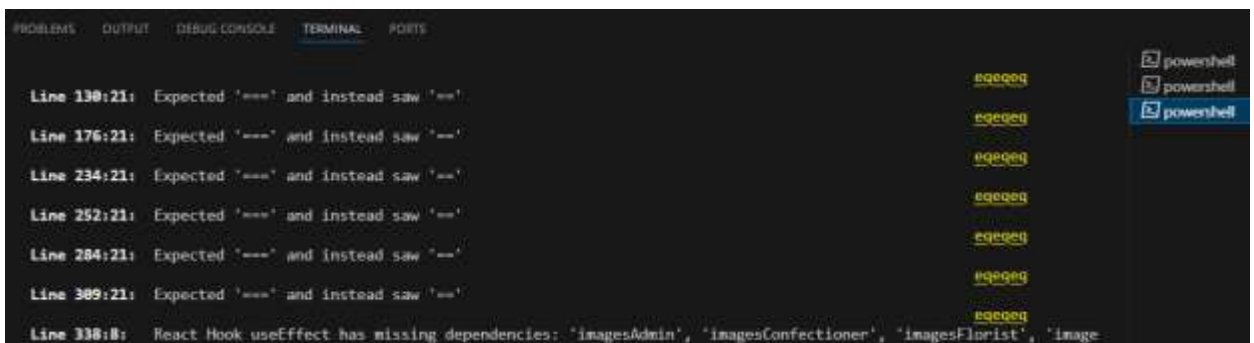


Рисунок 16 – Приклад роботи терміналу VS Code

Також важливою особливістю є можливість налаштування та запуску різних автоматизованих задач безпосередньо з редактора VS Code. Наприклад, скрипти збірки проєкту, тестування чи розгортання програмного забезпечення.

VS Code підтримує багато мов програмування, включаючи JavaScript, TypeScript, Python, Java, C#, C++, PHP, Ruby, і багато інших. Кожна мова має свої розширення, які полегшують роботу з нею, забезпечуючи підсвічування синтаксису, автодоповнення коду, перевірку помилок тощо

Загалом, ці особливості дозволяють створювати програмні продукти в середовищі Visual Studio Code ефективно та з комфортом, забезпечуючи високу продуктивність та якість розробки.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ КОДУ ПО МАКЕТУ З FIGMA ІНСТРУМЕНТАМИ REACT

3.1 Створення React-компонентів та сторінок

Знання структури React-компонентів є ключовим для успішного розроблення web-інтерфейсів за допомогою бібліотеки React [41]. У React існують два основних типи компонентів: функціональні та класові.

Функціональні компоненти – це прості функції JavaScript, які приймають props (властивості) як аргументи і повертають JSX (розширений синтаксис JavaScript для створення React елементів). Вони є простими у використанні та здатні швидко оновлювати інтерфейс (див. рис. 17).

```
▼ const MyFunctionalComponent = (props) => {  
  ▼ return (  
    ▼   
      <div>  
        <h1>{props.title}</h1>  
        <p>{props.description}</p>  
      </div>  
    );  
  };  
};
```

Рисунок 17 – Приклад props

Класові компоненти, натомість, є класами JavaScript, що розширюють базовий клас `React.Component`. Вони мають власний стан (`state`) та методи життєвого циклу React, які дозволяють керувати станом компонентів та взаємодіяти з іншими частинами застосунку (див. рис. 18).

Компоненти можуть приймати дані через props (властивості), які передаються в компонент вигляді атрибутів. Це дозволяє компонентам бути параметризованими та більш загальними у використанні.

Життєвий цикл класових компонентів включає методи, які викликаються на різних етапах життєвого циклу компонентів, такі як `componentDidMount()`, `componentDidUpdate()`, `componentWillUnmount()`. Це дозволяє виконувати певні дії відповідно до змін стану компонентів [42].

```

class MyComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  render() {
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Increment
        </button>
      </div>
    );
  }
}

```

Рисунок 18 – Приклад класових компонентів

Узагальнюючи, функціональні компоненти є простими та ефективними для простих відображень, тоді як класові компоненти надають більше можливостей для управління станом та життєвим циклом компонентів. Обидва типи компонентів використовуються в React в залежності від потреб застосунку.

У React хуки (hooks) – це спеціальні функції, які дозволяють використовувати стан (state) та інші можливості React у функціональних компонентах. Вони були введені з метою полегшити роботу зі станом і життєвим циклом компонентів у функціональних компонентах.

`useState()` – цей хук дозволяє додавати локальний стан (state) до функціональних компонентів. Він приймає початкове значення стану і

повертає поточне значення стану та функцію для його оновлення (див. рис. 19).

```
import React, { useState } from 'react';

const Counter = () => {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
};
```

Рисунок 19 – Приклад useState()

useEffect() – цей хук дозволяє виконувати певні дії (такі як взаємодія з API або підписка на події) у функціональних компонентах в реакції на зміни стану або після рендерингу компоненту (див. рис. 20).

```
import React, { useState, useEffect } from 'react';

const DataFetchingComponent = () => {
  const [data, setData] = useState(null);

  useEffect(() => {
    fetch('https://api.example.com/data')
      .then(response => response.json())
      .then(data => setData(data));
  }, []);

  return (
    <div>
      {data ? <p>Data: {data}</p> : <p>Loading...</p>}
    </div>
  );
};
```

Рисунок 20 – Приклад useEffect()

Методи життєвого циклу (lifecycle methods) у класових компонентах React - це спеціальні методи, які автоматично викликаються на різних етапах життєвого циклу компонента. Вони надають можливість виконувати певні дії в певний момент часу, наприклад, після створення компонента, після оновлення стану або перед видаленням компонента з DOM [43].

`constructor()` – це метод, який викликається при створенні компонента. Використовується для ініціалізації стану та прив'язки методів (див. рис. 21).

```
import React, { Component } from 'react';

class MyClassComponent extends Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  render() {
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Increment
        </button>
      </div>
    );
  }
}
```

Рисунок 21 – Приклад методу життєвого циклу `constructor()`

`componentDidMount()` – цей метод викликається після того, як компонент був вперше відображений на екрані. Використовується для виконання дій після завантаження компонента.

`componentDidUpdate()` – цей метод викликається після оновлення компонента (після того, як було змінено стан або властивості).

`componentWillUnmount()` - цей метод викликається перед тим, як компонент буде видалений з DOM. Використовується для очищення ресурсів, таких як підписки на події або таймери (див. рис. 22).

```

    componentDidMount() {
      // Виконати дії після завантаження компонента
    }

    componentDidUpdate(prevProps, prevState) {
      // Виконати дії після оновлення компонента
    }

    componentWillUnmount() {
      // Виконати дії перед видаленням компонента
    }

```

Рисунок 22 – Приклади методів `componentDidMount()`,

Узагальнюючи, хуки в функціональних компонентах та методи життєвого циклу в класових компонентах дозволяють розробникам керувати станом та поведінкою компонентів у React, забезпечуючи більшу гнучкість і продуктивність при розробці web-інтерфейсів [44].

3.2 Налаштування маршрутизації React-Router

React Router – це бібліотека, яка дозволяє користувачам вбудовувати маршрутизацію в React застосунки. Вона надає потужні інструменти для визначення та керування шляхами (routes) у застосунку, щоб забезпечити навігацію між різними сторінками або компонентами. Розглянемо детальніше налаштування маршрутизації з використанням React Router. Для встановлення необхідно використати команду `npm install react-router-dom` у терміналі.

Основний компонент `BrowserRouter` використовує HTML5 History API для маршрутизації на стороні клієнта з використанням `pushState` і

replaceState. Він зазвичай обгортає зпмтосунок, щоб відслідковувати зміни URL. Компонент Route визначає маршрут, який співпадає з URL і відображає відповідний компонент. Switch вибирає перший дочірній <Route> або <Redirect>, що співпадає з поточним URL. Компоненти Link та NavLink використовуються для навігації між сторінками.

Налаштування маршрутів в React Router відбувається за допомогою компонента <Route> і <Switch> (див. рис. 23). Користувач визначає маршрути, де кожен маршрут має свій шлях (path), який співпадає з URL, і компонент, який відображається при співпадінні маршруту.

```

import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
import HomePage from './HomePage';
import AboutPage from './AboutPage';

const App = () => {
  return (
    <Router>
      <Switch>
        <Route path="/" exact component={HomePage} />
        <Route path="/about" component={AboutPage} />
      </Switch>
    </Router>
  );
};

export default App;

```

Рисунок 23 – Приклад налаштування маршрутів

Параметри маршруту дозволяють розробникам передавати різні динамічні дані у компоненти створюваного застосунку. Прикладом до цього може бути використання розробником :userId для передачі різних параметрів, які потім можна отримати у компоненті за допомогою props.match.params.userId.

Для навігації між сторінками використовуються компоненти <Link> або <NavLink>, де можна вказати URL, на який потрібно перейти (див. рис.

24). Ці компоненти автоматично змінюють URL при натисканні і дозволяють легко переходити між різними сторінками застосунку.

```
import { Link } from 'react-router-dom';

const Navigation = () => {
  return (
    <nav>
      <ul>
        <li>
          <Link to="/">Home</Link>
        </li>
        <li>
          <Link to="/about">About</Link>
        </li>
      </ul>
    </nav>
  );
};
```

Рисунок 24 – Приклад навігації між сторінками

React Router дозволяє створювати потужні маршрутизаційні системи в React застосунку, що дозволяє легко керувати навігацією між різними частинами інтерфейсу користувача. Використовуючи його компоненти і API, можна створити динамічні та інтерактивні застосунки зі складною навігацією.

3.3 Підключення сторонніх бібліотек для інтерактивної взаємодії користувача

Підключення сторонніх бібліотек для інтерактивної взаємодії користувача є важливою складовою розробки сучасних web-застосунків, оскільки ці бібліотеки дозволяють розширювати функціональність та спрощувати роботу з різноманітними взаємодіями користувачів. Вони надають готові рішення для реалізації складних ефектів, анімацій, інтерактивних елементів та інших компонентів [45].

Однією з найпопулярніших бібліотек для інтерактивної взаємодії користувача є React-Bootstrap. Ця бібліотека базується на популярному фреймворку Bootstrap і надає готові компоненти, які можна використовувати для швидкого розгортання елементів інтерфейсу. Вона забезпечує адаптивність, доступність та стилізацію компонентів, що значно полегшує роботу з дизайном web-застосунків.

Основні переваги React-Bootstrap включають в себе гнучкість та легкість використання готових компонентів Bootstrap у ваших React-проєктах. Розробник може легко інтегрувати такі елементи, як кнопки, форми, картки, навігаційні панелі та багато інших безпосередньо у різні React-компоненти.

React-Bootstrap надає також можливість налаштовувати компоненти за допомогою властивостей (props) React, що дозволяє змінювати їх зовнішній вигляд та поведінку в залежності від потреб застосунку. Крім того, вона підтримує темізацію і має широкий набір параметрів для керування стилями та поведінкою компонентів.

React-Bootstrap є популярним вибором для розробників React через свою простоту та гнучкість. Вона дозволяє використовувати потужні можливості Bootstrap у ваших React-проєктах без необхідності додаткового коду і використовує сучасні розробки web-застосунків на базі React.

Бібліотека React-Bootstrap також має детальну документацію та велику спільноту розробників, що сприяє швидкому вивченню її можливостей та розвитку навичок роботи з нею. Вона підтримується активною командою розробників і регулярно оновлюється з урахуванням нових вимог і стандартів розробки.

Ще однією популярною бібліотекою є Material-UI для React. Вона пропонує готові компоненти, які дотримуються дизайну Material Design від Google. Material-UI дозволяє легко створювати модерні інтерфейси, забезпечуючи різноманітні елементи, такі як кнопки, картки, меню та інші, з великою кількістю налаштувань.

Вона дозволяє швидко створювати сучасні та естетично привабливі інтерфейси з мінімальними зусиллями. Основними особливостями Material-UI є широкий набір готових компонентів, які відповідають стандартам Material Design. Ці компоненти включають кнопки, текстові поля, картки, меню, діалогові вікна, таблиці, тулбари та інші, і вони мають сучасний вигляд, що сприяє однорідності дизайну застосунку.

Material-UI також вражає гнучкістю налаштувань. Кожен компонент має широкий набір параметрів, які дозволяють змінювати їх зовнішній вигляд та поведінку. Користувач може легко налаштовувати кольори, розміри, типи кнопок, стилі тексту та інші аспекти. Крім того, Material-UI підтримує темізацію, що дозволяє швидко змінювати вигляд застосунку шляхом налаштування теми, включаючи кольори, шрифти та пропорції.

Важливою перевагою Material-UI є його відповідність стандартам Material Design, розробленим Google. Це забезпечує однорідність і сучасний вигляд інтерфейсу. Бібліотека добре підтримує адаптивний дизайн, що дозволяє створювати інтерфейси, які працюють на різних пристроях і розмірах екрану.

Material-UI широко використовується в спільноті розробників React через свою простоту та продуктивність. Вона має велику кількість документації та прикладів, що допомагає розробникам вивчити її можливості та швидко і ефективно використовувати для створення привабливих і функціональних web-застосунків.

Також популярною бібліотекою є Ant Design, яка також надає готові компоненти для React. Ant Design має великий вибір стилізованих компонентів, що дозволяє швидко розгортати складні інтерфейси. Вона підтримує стандартизовані дизайн-патерни та дозволяє створювати стильні і сучасні web-застосунки.

Ant Design – це інтерфейсна бібліотека для React, розроблена компанією Ant Design Team. Вона надає набір готових компонентів і стилів для швидкої побудови сучасних web-інтерфейсів у проєктах на React.

Основні переваги Ant Design полягають у його широкому виборі готових компонентів, які охоплюють різноманітні аспекти web-розробки, включаючи кнопки, форми, таблиці, каруселі, меню, модальні вікна, іконки та багато іншого. Ці компоненти мають сучасний дизайн і можуть легко інтегруватися у ваші проєкти.

Ant Design також пропонує розширені можливості конфігурації компонентів через використання властивостей (props), що дозволяє змінювати їх зовнішній вигляд та поведінку. Крім того, бібліотека має підтримку темізації, що дозволяє змінювати стиль компонентів за допомогою набору кольорів та параметрів.

Один із головних принципів Ant Design – це простота використання. Вона надає зручний API для роботи з компонентами, що дозволяє ефективно будувати складні інтерфейси з мінімальними зусиллями. Бібліотека також підтримує розробку на основі компонентного підходу, що сприяє повторному використанню коду і підтримці стандартів розробки web-застосунків.

Ant Design має детальну документацію з прикладами використання кожного компонента і надає широкий спектр можливостей для створення привабливих та функціональних інтерфейсів. Вона підтримується активною спільнотою розробників і має велику кількість сторонніх розширень та плагінів для розширення її можливостей.

Отже, дані бібліотеки дозволяють розробникам швидко впроваджувати інтерактивні та привабливі елементи web-застосунків без необхідності писати весь код з нуля. Вони спрощують розробку, дозволяють ефективно використовувати ресурси та полегшують підтримку і розвиток проєктів.

3.4 Результат роботи web-застосунку

У результаті розробки макету Figma та подальшій реалізації інструментами бібліотеки React у Visual Studio Code, було створено web-застосунок, що має декілька основних структур.

Ураховуючи всі розглянуті типи web-застосунків у Розділі 1, було притримано всіх стилей, які можна буде адаптувати під web-сайти різних навчальних закладів. Такі функції, як: меню вибору опцій, головне зображення на початку з назвою навчального закладу, головна інформація про навчальний заклад, зображення даного навчального закладу та посилання на різну інформацію про них, нагороди даного навчального закладу з посиланнями, новини даного закладу освіти разом із зображеннями, описом і посиланнями, а також інтерактивний перегляд вчителів даного закладу.

Із самого верху присутнє меню вибору опцій (див. рис. 25).



Рисунок 25 – Меню вибору опцій

У самому центрі присутня назва web-застосунку з фоновим рисунком (див. рис. 26).



Рисунок 26 – Початкове зображення web-застосунку

У web-сайті присутня вкладинка «Головне», у якій є багато головної інформації про навчальний заклад (див. рис. 27). Туди можна додавати різну

корисну інформацію для абітурієнтів, про професії даного навчального закладу та профорієнтацію.

ГОЛОВНЕ



Професії

Вибери професію майбутнього разом із Дніпровським центром професійно-технічної освіти туристичного сервісу, де ти розкриєш свій потенціал та зможеш здійснити мрію про успішну кар'єру в обраній галузі. Отримай професійні навички, необхідні для досягнення успіху у своїй професійній діяльності та стань експертом у своєму ділі разом із нашим коледжем.

[ДЕТАЛЬНІШЕ →](#)



Абітурієнту

Поглиби свої знання у світі туризму з Дніпровським центром професійно-технічної освіти туристичного сервісу, де ти відкриєш для себе нові горизонти та зможеш ступити до мрії про кар'єру в галузі туристичного сервісу. Обери професійну освіту, яка допоможе тобі досліджувати та збагачувати світ туризму разом з нами!

[ДЕТАЛЬНІШЕ →](#)



Профорієнтація

Отримай відповіді на всі свої питання щодо майбутньої кар'єри разом із Дніпровським центром професійно-технічної освіти туристичного сервісу, де ми допоможемо тобі зрозуміти та обрати найбільш відповідну професію для твоєї успішної майбутньої кар'єри. Довіряйся нашому досвіду та підтримці в пошуках своєї шляхи в професійній сфері та долучися до спільноти успішних випускників!

[ДЕТАЛЬНІШЕ →](#)

Рисунок 27 – Зовнішній вигляд вкладки «Головне»

На web-сайті створено вкладку про навчальні майстерні та кабінети з зображеннями та анімацією при наведенні курсору миші (див. рис. 28).

Навчальні майстерні та кабінети Центру та гуртожиток

[ДЕТАЛЬНІШЕ →](#)

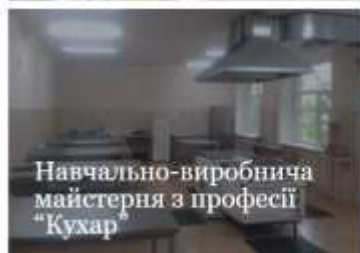


Рисунок 28 – Зовнішній вигляд вкладки майстерень і кабінетів

Було додано вкладинку нагород даного навчального закладу (див. рис. 29). Додавання вкладинки «Нагороди» може сприяти підвищенню престижу навчального закладу, показуючи досягнення учнів, викладачів або співробітників. Це може стати додатковим мотиватором для студентів прагнути досягати високих результатів і брати участь у заходах та проєктах. Крім того, наявність вкладинки з нагородами може позитивно вплинути на імідж закладу серед студентів, батьків та громадськості.



НАГОРОДИ

Нагороди – це визнання досягнень та заслуг. Вони мотивують людей до нових висот, а також слугують символом пошани та вдячності. Існують різні типи нагород: державні, міжнародні, галузеві, громадські. Їх вручають за видатні успіхи в науці, мистецтві, спорті, бізнесі, волонтерській діяльності та інших сферах.

[ДЕТАЛЬНІШЕ →](#)

Рисунок 29 – Вкладинка «Нагороди»

У web-застосунці присутня вкладинка навчально-практичних центрів для даного навчального закладу (див. рис. 30). Ця вкладка дозволить навчальному закладу зібрати інформацію про партнерські угоди та програми навчання, що дозволить студентам знайти інформацію про можливості стажування, вакансії, доступні через партнерство з іншими організаціями.

НАВЧАЛЬНО-ПРАКТИЧНІ ЦЕНТРИ

Під девізом «Якісно! Професійно! Гостинно!» працюють навчально-практичні центри, для яких підготували конкурентоспроможних кадрів стала закономірним наслідком вкладання сил, натхнення і душі педагогічного колективу Дніпровського центру професійно-технічної освіти туристичного сервісу.

[ДЕТАЛЬНІШЕ →](#)



Рисунок 30 – Зовнішній вигляд вкладинки «Навчально-практичні центри»

Також, було створено вкладинку новин навчального закладу (див. рис. 31). Створення вкладки «Новини навчального закладу» важливо для забезпечення студентів та співробітників актуальною інформацією про події, оголошення та новини, пов'язані з навчальним закладом.

Новини навчального закладу



Разом з учнями та персоналом Дніпровського коледжу професійної освіти туристичного сервісу ми відзначили День вишиванки з особливим захватом та патріотизмом. Усі разом ми вдягнули національні вишиванки, щоб символізувати нашу національну єдність та гордість за нашу спадщину. Цей день став для нас не лише святом культури, але й дотриманням національних традицій, що об'єднує нас як спільноту.

[ДЕТАЛЬНІШЕ](#)



У Дніпровському коледжі професійної освіти туристичного сервісу ми щиро відзначили День волонтера з великою вдячністю та повагою до всіх, хто дарує свій час та зусилля для допомоги іншим. Цей день став важливим для нас, оскільки ми визнаємо та цінимо внесок волонтерів у розвиток нашого суспільства та підтримку тих, хто потребує допомоги. Відзначаючи цей день, ми висловлюємо свою глибоку вдячність та підтримку всім волонтерам, які роблять світ кращим для нас усіх.

[ДЕТАЛЬНІШЕ](#)



У Дніпровському коледжі професійної освіти туристичного сервісу ми відзначили День козацтва з великою пишністю та відчуттям патріотизму. Цей день став для нас символом відданості національним традиціям та героїчним подвигам наших предків. Ми шануємо спадщину козацтва та віддаємо належне шанування тим, хто відстоював свободу та незалежність нашої держави. Відзначаючи цей день, ми відчуваємо гордість бути частиною великої історії та спадкоємцями величного козацького духу.

[ДЕТАЛЬНІШЕ](#)

Рисунок 31 – Зовнішній вигляд вкладки «Новини навчального закладу»

Ця вкладка дозволить навчальному закладу регулярно оновлювати студентів про останні події, важливі анонси лекцій, семінарів, культурних заходів та спортивних змагань. Вона також може бути платформою для публікації важливих повідомлень, пов'язаних з академічними аспектами, а також оголошень про можливість стажування та робочі місця.

Отже, вкладка «Новини» стане важливим елементом web-застосунку, що забезпечить доступ до свіжої та цікавої інформації для всіх користувачів навчального закладу, сприяючи активній комунікації та взаємодії університетської спільноти.

У web-застосунці було додано інтерактивний перегляд вчителів даного навчального закладу (див. рис. 32).



Рисунок 32 – Зовнішній вигляд вкладки «Колектив навчального закладу»

При гортанні зображень відбувається анімаційний перехід до наступних у вигляді «кімнати музею» (див. рис. 33).



Рисунок 33 – Зовнішній вигляд анімованих переходів при гортанні сторінок

У результаті розробки макету у Figma та реалізації за допомогою React у Visual Studio Code було створено web-застосунок з основними структурами.

Він містить меню вибору опцій, головне зображення з назвою навчального закладу, інформацію про заклад, новини, вкладки про майстерні та нагороди, що додають престижу. Даний web-застосунок також забезпечує інтерактивний перегляд вчителів та важливі оголошення для студентів і співробітників.

Таким чином, цей web-застосунок є цінним інструментом для спілкування і взаємодії університетської спільноти, забезпечуючи доступ до різноманітної та цікавої інформації для всіх користувачів навчального закладу.

ВИСНОВОК

Дана кваліфікаційна робота передбачала детальне дослідження web-застосунків, включаючи їх різноманітні типи та функціонал. Під час виконання дослідження було оглянуто різні класифікації web-застосунків залежно від їхніх цілей, призначення та характеристик.

Дослідження включало аналіз різних типів web-застосунків, таких як інформаційні портали, електронні платформи для освіти, інтерактивні навчальні середовища, системи управління навчанням та багато інших. Кожен тип має свої особливості та функціонал, спрямований на забезпечення певних освітніх або інформаційних потреб користувачів.

Крім того, в рамках даної роботи було проведено детальне дослідження авторитетних web-сайтів у сфері освіти. Авторитетні сайти включають університетські ресурси, інформацію про освітні процеси, наукові досягнення та інші важливі аспекти. Також було досліджено основні функції, які важливо додати до універсального web-застосунку, щоб можна було його зручно адаптувати до різних інших навчальних закладів.

Дані дослідження дозволили отримати глибокі знання про сучасні тенденції web-розробки для освітніх цілей та встановити зв'язок між типами web-застосунків та їх функціональними характеристиками.

Для реалізації даного проєкту було обрано використання бібліотеки React. Дана потужна бібліотека для розробки інтерфейсів, забезпечує швидку та ефективну розробку динамічних інтерфейсів, які відповідають потребам сучасних користувачів.

Однією з ключових переваг React є його компонентна архітектура, що дозволяє розбити складні інтерфейси на маленькі, повторно використовувані компоненти. Це спрощує процес розробки, тестування і підтримки застосунків, забезпечуючи гнучкість та швидкість в розробці. Крім того,

React забезпечує можливість асинхронного оновлення сторінок, що особливо важливо для web-застосунків, що вимагають живої взаємодії з користувачем.

Вибір Visual Studio Code як середовища розробки також є вдалим рішенням. Цей редактор коду володіє потужними функціональними можливостями, які допомагають розробникам створювати, відлагоджувати і тестувати web-застосунки з високою продуктивністю. VS Code підтримує широкий спектр розширень для розробки на React, забезпечуючи автодоповнення коду, підтримку синтаксису JSX, аналіз коду та інші корисні інструменти.

Важливо відзначити, що обидва інструменти мають активні спільноти користувачів і розробників, що постійно вносять покращення та створюють нові розширення. Це сприяло швидкому вирішенню проблем та розвитку високоякісних web-застосунків. Крім того, використання React разом із Visual Studio Code забезпечило універсальний та потужний набір інструментів для розробки web-застосунку.

Таким чином, обрання React і Visual Studio Code є стратегічним кроком для успішної реалізації проєкту web-застосунку для навчальних закладів. Ці технології дозволили створити ефективний, гнучкий і сучасний web-сайт, який відповідає потребам користувачів і сприяє покращенню якості освіти.

Результатом створення та розробки універсального web-застосунку для навчальних закладів є інноваційна платформа, яка може бути легко адаптована під різні потреби освітніх установ. В ході розробки були вдосконалені алгоритми, що забезпечують оптимізацію та швидку роботу web-сайту навчального закладу. Оптимізація включала в себе впровадження передових методів кешування, оптимізацію запитів до бази даних та покращення швидкості завантаження сторінок.

Основна мета при розробці цього web-застосунку полягала у створенні сучасного та інтуїтивно зрозумілого інтерфейсу. Дизайн був створений з урахуванням потреб користувачів, забезпечуючи легкий доступ до необхідної інформації та інструментів.

Завдяки цьому підходу користувачі мають зручну навігацію по веб-сайту, швидко знаходять потрібні ресурси та отримують необхідну інформацію без зайвих зусиль.

Додатково, веб-застосунок був розроблений з урахуванням сучасних тенденцій у дизайні та веб-розробці. Використання React дозволило створити динамічний та масштабований інтерфейс, який підтримує різноманітні функціональні можливості, такі як інтерактивність, анімація та швидка відповідь на дії користувача. Компонентний підхід дозволив розбити складність веб-застосунку на окремі частини, що спрощує розробку та підтримку коду.

Таким чином у результаті виконання кваліфікаційної роботи була досягнута її мета – створення адаптивного веб-застосунку для навчального закладу. У роботі було виконано наступне:

- розглянуто загальні поняття щодо реалізації адаптивних веб-застосунків, їх актуальність, популярність і розвиток
- проаналізовано сайти-авторитети, їх дизайни та функціонал,
- виявлено та досліджено основні функції та об'єкти адаптивного дизайну,
- реалізовано адаптивний веб-застосунок для навчального закладу інструментами бібліотеки React у Visual Studio Code.

Таким чином, створення універсального веб-застосунку для навчальних закладів з використанням React і Visual Studio Code дало змогу реалізувати потужний, швидкий та безпечний інструмент, який може задовольнити потреби навіть найвимогливіших користувачів. Такий підхід дозволив створювати сучасне та ефективне рішення для освітніх закладів, що сприяє підвищенню якості освіти та забезпечує зручну взаємодію зі студентами, викладачами та іншими учасниками освітнього процесу.

Отже, у результаті даної кваліфікаційної роботи було проведено докладне дослідження веб-застосунків, їх типів та функціоналу, а також класифікацій, що базуються на їхніх характеристиках та призначенні. Для

реалізації проєкту були обрані технології React і Visual Studio Code. В рамках проєкту було розроблено універсальний web-застосунок для навчальних закладів, який забезпечує швидку роботу та оптимізацію web-сайту. Основний акцент був зроблений на створенні сучасного та інтуїтивно зрозумілого дизайну, що спрощує взаємодію користувачів з інформацією.

Використання React і Visual Studio Code дало змогу створити потужний, швидкий та безпечний інструмент для освітніх закладів, який забезпечує зручну взаємодію зі студентами та іншими учасниками освітнього процесу, сприяючи підвищенню якості освіти.

Кваліфікаційна робота виконана у відповідності до стандарту спеціальності 121 «Інженерія програмного забезпечення» і демонструє володіння такими компетентностями, як:

- Здатність оцінювати актуальність задачі та розуміння важливості її виконання.
- Здатність досліджувати сайти-авторитети та виділяти ключові аспекти для реалізації адаптивного web-застосунку.
- Здатність реалізовувати створення web-застосунків.

Серед результатів навчання, визначених стандартом, кваліфікаційна робота реалізовує наступні:

- Використання бібліотеки React.js.
- Робота в середовищі розробки Visual Studio Code.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Пономаренко І. В. Веб-сайт як ефективний інструмент інтернет-маркетингу для досягнення стратегічних цілей компанії [Електронний ресурс] // І. В. Пономаренко, А. С. Сапян // Інфраструктура ринку. – 2019 – Вип. 31. – С. 354-361.
2. Бурило Ю. П. Веб-сайт як інформаційний ресурс та об'єкт права інтелектуальної власності [Електронний ресурс] // Науковий вісник Ужгородського національного університету – 2015. С. 67-70.
3. Northell A Complete Web Development Guide [Навчальний посібник] // 2021 – С. 50.
4. Robin Wieruch. The Road to React: Your journey to master plain yet pragmatic React.js [Навчальний посібник] // Independently published – 2018 – С. 286.
5. Mikhail Sakhniuk. React and React Native - Fifth Edition: Build cross-platform JavaScript and TypeScript apps for the web, desktop, and mobile [Навчальний посібник] // Packt Publishing – 2024 – С. 508.
6. Дмитрієв В. С. Розробка сайту з продажу нерухомості засобами React та Node.js: кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» [Електронний ресурс] // Наук. керівник О. М. Мильцев. Запоріжжя, ЗНУ – 2022 – С. 65.
7. Шуст С.І. Випускна робота на тему: Створення сайту для загальноосвітньої школи №1 м. Середини-Буди [Електронний ресурс] // Завідувач випускаючої кафедри Довбиш А.С. Суми – 2020 – С. 15-17.
8. Wikipedia. Visual Studio Code [Електронний ресурс] // 2024
9. Урбанович М.С. Розробка web-сайта загальноосвітньої школи за допомогою системи управління змістом cms joomla [Електронний ресурс] // І.В. Рассоха, к.ф.-м. н., доцент. Полтавський національний технічний університет імені Юрія Кондратюка – 2019 – С. 138.

10. Фірсова О. А. Web-аналітика як інструмент digital-marketing : робота на здобуття кваліфікаційного рівня бакалавра [Електронний ресурс]// Наук. кер. О. С. Квілінський. Суми: Сумський державний університет – 2021 – С. 30.
11. Шаргін Я. В. Веб-сайт автошколи Сумської обласної організації Всеукраїнської спілки [Електронний ресурс] // Наук. кер. О. В. Шутилева. Суми :Сумський державний університет – 2022 – С. 77.
12. Федотова Н. А. Веб-сайт дитячої музичної школи ім. П.І. Чайковського м. Тростянець [Електронний ресурс] // Наук. кер. С. М. Ващенко. Суми : СумДУ – 2021 – С. 104.
13. Халлиєв М.Г. Інформаційний веб-портал «Центру первинної медико-санітарної допомоги м. Суми» [Електронний ресурс] // М.Г. Халлиєв наук. керівник І.В. Шелехов. Суми: СумДУ – 2020 – С. 54.
14. Морозов М.Ю. Інформаційна система обліку пацієнтів ГепатаЦентру медичного інституту СумДУ [Електронний ресурс] // М.Ю. Морозов; наук. керівник О.Б. Берест. Суми: СумДУ – 2020 – С. 59.
15. Волошин В.В. Веб-додаток організації продажів на основі технології React JS [Електронний ресурс // В.В Волошин; наук. керівник О.Б. Проценко Суми: СумДУ, 2020. – С. 57.
16. Geometric. Типи сайтів: які бувають види сайтів та їх різниця, класифікація [Електронний ресурс] // ІТ Рейтинг України – 2023.
17. WebTune. Види вебсайтів та їх функціонал [Електронний ресурс] // WebTune – 2023.
18. Webcase. Які типи сайтів існують? [Електронний ресурс] // Webcase – 2022.
19. Apricode. Види сайтів [Електронний ресурс] // Apricode – 2017.
20. Avada Media. Типи Сайтів [Електронний ресурс] // Avada-media – 2022
21. Voll digital agency. Види сайтів [Електронний ресурс] // Voll.com – 2021.

- 22.Cases. Класифікація вебсайтів [Електронний ресурс] // cases.media – 2020.
- 23.Reactjs. Доступність [Електронний ресурс] // uk.legacy.reactjs – 2024.
- 24.Відділ освіти Бородянської селищної ради. Вебсайт закладу освіти – що необхідно знати [Електронний ресурс] // otg.osvitabor.gov.ua – 2024.
- 25.Santana C. R. React 18 Design Patterns and Best Practices - Fourth Edition: Design, build, and deploy production-ready web applications with React by leveraging industry-best practices [Електронний ресурс] // Packt Publishing – 2023 – С. 524.
- 26.Schwarzmueller M. React Key Concepts: Consolidate your knowledge of React's core features [Електронний ресурс] // Packt Publishing – 2022 – С. 590.
- 27.Makarevich N. Advanced React: Deep dives, investigations, performance patterns and techniques [Електронний ресурс] // Independently published – 2023 – С. 319.
- 28.Sakhniuk M. React and React Native - Fifth Edition: Build cross-platform JavaScript and TypeScript apps for the web, desktop, and mobile [Електронний ресурс] // Packt Publishing –2024 –С. 508.
- 29.Rippon C. Learn React with TypeScript: A beginner's guide to reactive web development with React 18 and TypeScript [Електронний ресурс] // Packt Publishing –2023 – С. 474.
- 30.Springer S. React: The Comprehensive Guide [Електронний ресурс] – Rheinwerk Computing – 2023 – С. 676.
- 31.Griffinths D. React Cookbook: Recipes for Mastering the React Framework [Електронний ресурс] // O'Reilly Media – 2021 – С. 510.
- 32.Qiu J. React Anti-Patterns: Build efficient and maintainable React applications with test-driven development and refactoring [Електронний ресурс] // Packt Publishing – 2024 – С. 483.

- 33.Hinkula J. Full Stack Development with Spring Boot 3 and React - Fourth Edition: Build modern web applications using the power of Java, React, and TypeScript [Електронний ресурс] // Packt Publishing – 2023 – С. 454.
- 34.Alickovic A. React Application Architecture for Production: Learn best practices and expert tips to deliver enterprise-ready React web apps [Електронний ресурс] // Packt Publishing – 2023 – С. 230.
- 35.Adedeji O. Full-Stack Flask and React: Learn, code, and deploy powerful web applications with Flask 2 and React 18 [Електронний ресурс] // Packt Publishing – 2023 – С. 408.
36. McAllister J. React JS: A Complete 3-in-1 Guide for All Skill Levels [Електронний ресурс] // Independently published – 2024 – С. 510.
- 37.UAITLAB. Порівняльний аналіз основних фреймворків: Django, Ruby on Rails, Laravel та Express.js [Електронний ресурс] // IT Рейтинг України – 2024.
- 38.Черніцький Е. Вебсайт інтернет-магазину “StyleShop” на платформі React.js з використанням CSS та HTML [Електронний ресурс] // Національний Авіаційний Університет – 2023 – С. 67.
- 39.Visual Studio Code. Using React in Visual Studio Code [Електронний ресурс] // code.visualstudio – 2024.
- 40.Денисюк В. В. Мобільний додаток системи управління дипломними проектами [Електронний ресурс] // Київський Політехнічний Інститут імені Ігоря Сікорського – 2019 – С. 82.
- 41.Stefanov S. React: Up & Running: Building Web Applications [Електронний ресурс] // – O'Reilly Media – 2021 – С. 230.
- 42.Klingman L. React Programming: The Big Nerd Ranch Guide [Електронний ресурс] – Big Nerd Ranch Guides –2023 – С. 1095.
- 43.Benedict A. C. Professional React Native: Expert techniques and solutions for building high-quality, cross-platform, production-ready apps [Електронний ресурс] // Packt Publishing – 2022 – С. 268.

44. Boduch A., Derks. R., Sakhniuk M. React and React Native - Fourth Edition: Build cross-platform JavaScript applications with native power for the web, desktop, and mobile [Электронный ресурс] // Packt Publishing – 2022 – С. 606.
45. Krause M. The Complete Developer: Master the Full Stack with TypeScript, React, Next.js, MongoDB, and Docker [Электронный ресурс] // No Starch Press – 2024 С. 344.