ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп’ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

**КУРСОВА РОБОТА**

з дисципліни

«Web-дизайн та web-програмування»

на тему:

Розробка лендінгової сторінки використовуючи технології React

Студента ІІІ курсу групи ІПЗ-*31*

спеціальності

121 – Інженерія програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Анатолій ЛУЦЬКО

(підпис)

Керівник: асистент Анастасія ВОЗНЮК

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Наталія ЛІЩИНА

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Анастасія ВОЗНЮК

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Андрій ЯЩУК

(підпис)

Луцьк 2024

**Луцький національний технічний університет**

***Факультет комп’ютерних та інформаційних технологій***

***Кафедра інженерії програмного забезпечення***

Ступінь вищої освіти: *бакалавр*

Спеціальність: *121 «Інженерія програмного забезпечення»*

Дисципліна: *«Web-дизайн та web-програмування»*

Курс *ІІІ* Група *ІПЗ-31* Семестр *V*

***ЗАВДАННЯ***

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

|  |
| --- |
| *Луцько Анатолій Іванович* |

(прізвище, ім’я, по батькові)

1. Тема роботи\_Розробка лендінгової сторінки використовуючи технології React

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник роботи: *асистент Вознюк Анастасія Вадимівна*

2. Строк подання студентом роботи *27 листопада 2024 р.*

3. Вихідні дані до роботи *завдання на курсову роботу*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)\_аналіз вимог до лендінгової сторінки та її функціоналу; вибір та впровадження технологій для створення лендінгової сторінки; розробка та оптимізація інтерфейсу користувача;інтеграція з зовнішніми сервісами та бібліотеками; тестування та деплоймент лендінгової сторінки.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перелік графічного матеріалу *кількість рисунків*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6. Дата видачі завдання *24 вересня 2024 р.*

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ з/п** | **Назва етапів курсової роботи** | **Строк**  **виконання**  **етапів роботи** | **Примітка** |
| 1 | Отримання теми курсової роботи | 24.09.2024 |  |
| 2 | Узгодження постановки задачі з керівником | 01.10.2024 |  |
| 3 | Вибір макета сайту та узгодження функціоналу сайту з керівником | 08.10.2024 |  |
| 4 | Огляд та аналіз літератури з питань курсової роботи | 15.10.2024 |  |
| 5 | Вибір засобів розробки | 22.10.2024 |  |
| 6 | Розробка сайту | 19.11.2024 |  |
| 7 | Тестування сайту та розміщення на хостингу | 26.11.2024 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 8 | Підготовка пояснювальної записки | 26.11.2024 |  |
| 9 | Здача курсової роботи на перевірку | 27.11.2024 |  |
| 10 | Захист |  |  |

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | \_\_\_\_Анатолій ЛУЦЬКО\_\_\_  (власне ім’я ПРІЗВИЩЕ) |
| Керівник роботи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | Анастасія ВОЗНЮК |

**АНОТАЦІЯ**

Луцько А.І. Розробка лендінгової сторінки використовуючи технології React. Курсова робота з дисципліни «Web-дизайн та web-програмування» студента спеціальності 121 «Інженерія програмного забезпечення». – Луцький національний технічний університет. – Луцьк, 2024. – 30 с.

Пояснювальна записка до курсової роботи містить: 30 стор., 12 рис., 10 джерел.

Мета курсової роботи — створення інтерактивної, адаптивної та естетично привабливої лендінгової сторінки з використанням сучасних технологій веб-розробки для просування продуктів чи послуг.

Лендінг створено із застосуванням бібліотеки React, стильове оформлення реалізовано через Tailwind CSS із додатковим використанням React-icons для іконографіки. Готовий проєкт розгорнуто на платформі Vercel, що забезпечує стабільний хостинг та швидке завантаження сторінок.

Пояснювальна записка до курсової роботи складається з двох розділів: аналіз предметної області та практична розробка лендінгової сторінки.

У розділі «Аналіз предметної області» розглянуто сучасні тенденції створення лендінгових сторінок, їх роль у маркетингу та бізнесі, а також переваги використання React і Tailwind CSS для досягнення високої адаптивності та інтерактивності.

У розділі «Практична розробка лендінгової сторінки» детально описано процес створення веб-продукту, зокрема організацію компонентів, інтеграцію стилів, застосування React-icons, а також розгортання на хостингу Verce.

Ключові слова: React, Tailwind CSS, React-icons, Vercel, інтерактивний дизайн, веб-розробка, маркетинговий інструмент.

[ВСТУП](#_wtwoszrp3bqx) 6

[РОЗДІЛ 1  
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ](#_1fob9te) 8

[1.1 Огляд та аналіз сучасних Front End технологій. Вибір засобів розробки лендінгової сторінки](#_b95kednnhrce) 8

[1.2 Постановка завдання на курсову роботу](#_ptv0w7smmizb) 13

[РОЗДІЛ 2  
РОЗРОБКА ЛЕНДІНГОВОЇ СТОРІНКИ](#_tyjcwt) 14

[2.1 Обґрунтування вибору технологій і засобів вирішення поставленого завдання](#_fujwgu4zsh35) 14

[2.2 Етапи розробки лендінгової сторінки](#_4d34og8) 16

[2.3 Практична реалізація лендінгової сторінки](#_2s8eyo1) 18

[2.4 Тестування лендінгової сторінки](#_p2thko459656) 25

[ВИСНОВКИ](#_iui8jjhs1cpu) 30

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ](#_sonmrs5wewfq) 32

**ВСТУП**

Сучасний розвиток інформаційних технологій та інтернету сприяє зростанню потреби у створенні ефективних веб-рішень, які можуть забезпечити якісний користувацький досвід, адаптивність і високу продуктивність. Одним із таких рішень є лендінгові сторінки, що виступають інструментом маркетингових кампаній, спрямованих на залучення уваги аудиторії до конкретного продукту чи послуги. Лендінгові сторінки характеризуються лаконічним дизайном, чітко структурованою інформацією та інтерактивними елементами, які стимулюють відвідувачів до здійснення певної дії, наприклад, заповнення форми чи купівлі товару.

Метою цієї курсової роботи є розробка лендінгової сторінки з використанням сучасного фреймворку React, що забезпечує високий рівень динамічності, зручності підтримки та розширення. Реалізація проекту передбачає вивчення можливостей React, оптимальне застосування його бібліотек і інструментів, а також впровадження найкращих практик у розробці інтерфейсу користувача.

Розробка лендінгової сторінки на базі React передбачає використання низки сучасних технологій і інструментів. React, як один із найбільш популярних JavaScript-фреймворків, відзначається простотою у використанні та широкими можливостями для створення компонентного підходу. Завдяки цьому фреймворку можна реалізувати інтерактивні елементи, які не потребують постійного перезавантаження сторінки, що значно покращує продуктивність і забезпечує плавний користувацький досвід.

У процесі розробки було використано бібліотеку framer-motion для управління анімаціями та створення плавних переходів між сторінками, а також Styled Components для стилізації компонентів, що дозволяє зберігати стиль і структуру коду. Окрім того, для забезпечення адаптивності та інтеграції іконок у дизайн інтерфейсу була використана бібліотека react-icons, що значно полегшує роботу з мультимедійними елементами.

З метою розробки і тестування проекту застосовано інструменти Visual Studio Code, одна з найкращих IDE для роботи з JavaScript, а також Git для контролю версій і зберігання коду в репозиторії.

Крім технічних аспектів, особлива увага приділяється питанням оптимізації лендінгової сторінки для пошукових систем (SEO) та забезпечення її кросбраузерної сумісності. Це дозволяє гарантувати, що сторінка буде доступною і зручною для широкої аудиторії, незалежно від пристрою чи браузера.

Таким чином, ця курсова робота охоплює повний цикл створення лендінгової сторінки: від визначення цілей та планування до реалізації з використанням React і пов'язаних інструментів. Результатом дослідження стане готовий продукт, який відповідає сучасним вимогам до веб-дизайну та розробки, а також слугує прикладом ефективного використання технологій фронтенд-розробки.

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд та аналіз сучасних Front End технологій. Вибір засобів розробки лендінгової сторінки

Сучасна розробка фронтенду є невід'ємною частиною веб-розробки, яка поєднує в собі естетику дизайну, високу продуктивність та інтерактивність. За останнє десятиліття технології, які використовуються для створення веб-інтерфейсів, зазнали значних змін, що дозволило розробникам створювати складні, динамічні та ефективні веб-додатки. Основою успішного проєкту стає вибір інструментів та технологій, які забезпечують швидкість розробки, масштабованість та легкість у підтримці. Цей розділ детально розглядає основні інструменти, фреймворки, бібліотеки та середовища розробки, які відіграють ключову роль у створенні сучасних фронтенд-додатків.

Історія фронтенд-розробки бере свій початок з початку 1990-х років, коли перші веб-сторінки створювалися за допомогою HTML. Ці сторінки були статичними і не мали інтерактивних елементів. У середині 1990-х років з'явився CSS, що дозволило відокремити візуальне оформлення від структури документа. У 1995 році мова JavaScript вперше побачила світ, надаючи можливість додавати до сторінок інтерактивність.

Згодом, із появою браузерів, які підтримують DOM-маніпуляції, почали з'являтися перші бібліотеки, як-от jQuery, що значно полегшили роботу з JavaScript. Проте стрімкий розвиток веб-технологій припав на останнє десятиліття, коли з'явилися сучасні фреймворки та бібліотеки, як-от React, Angular і Vue.js. Вони дозволили розробникам створювати масштабовані, продуктивні та багатофункціональні додатки. Топ найкращих фреймворів можна побачити на рисунку 1.1.

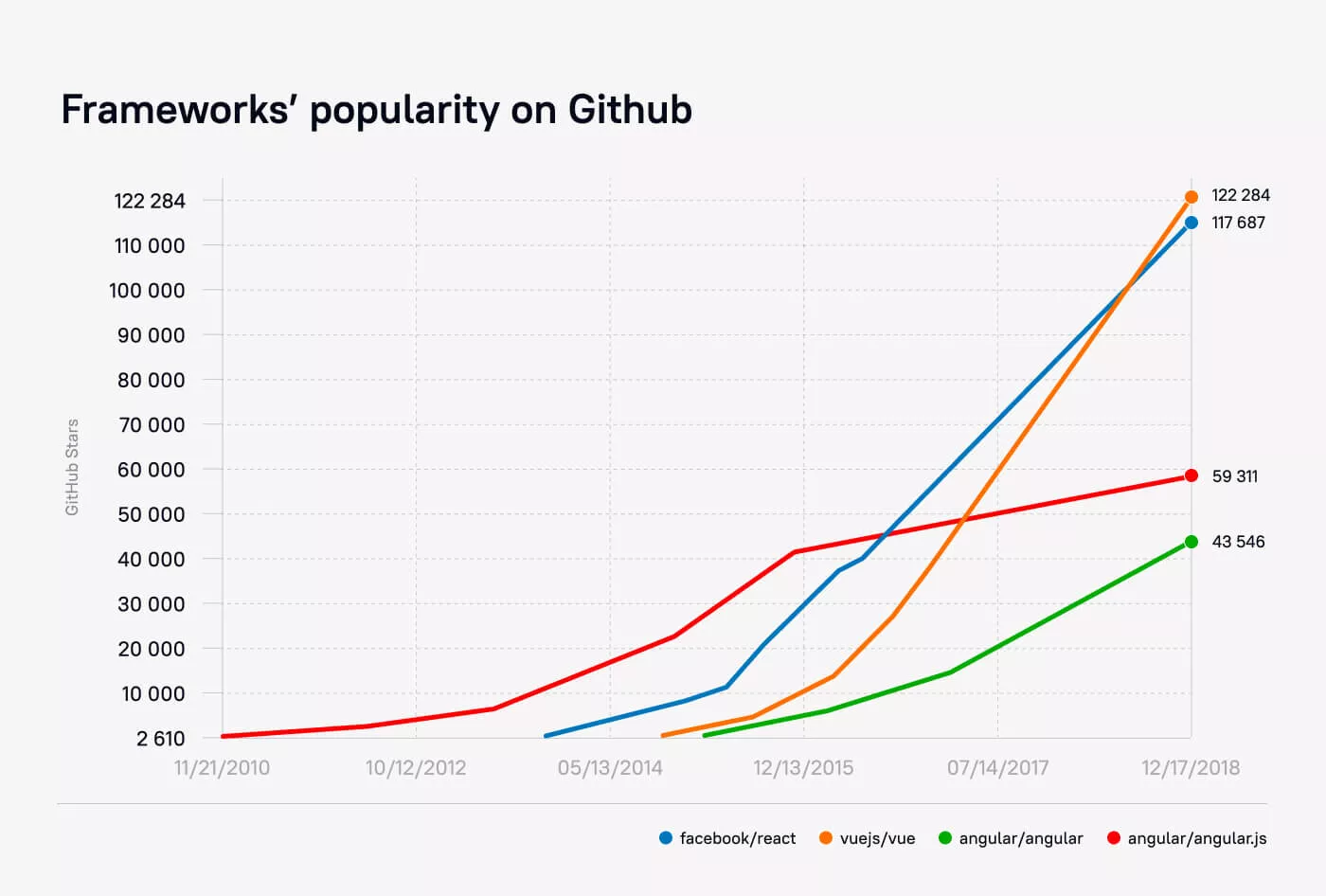


Рисунок 1.1 – Графік популярності фреймворків на [Github](https://en.wikipedia.org/wiki/GitHub) [1]

Однією з найпопулярніших технологій для розробки інтерфейсів є бібліотека React, розроблена Facebook. React відзначається компонентним підходом, який дозволяє розробникам створювати повторно використовувані елементи, зменшуючи обсяг коду та підвищуючи його читабельність. Основною перевагою React є його віртуальний DOM, який оптимізує оновлення інтерфейсу та забезпечує високу продуктивність навіть для складних додатків. Додатково React активно підтримується спільнотою, що забезпечує швидкий доступ до багатьох плагінів, бібліотек та документації.

Втім, React має і недоліки. Оскільки це лише бібліотека для створення інтерфейсів, розробники повинні самостійно вибирати інші інструменти для управління станом, маршрутизації та збору проєкту. Цей аспект може ускладнити процес для новачків, які ще не мають досвіду в роботі з екосистемою React. Незважаючи на це, React залишається основним вибором для багатьох компаній завдяки його гнучкості та можливостям масштабування.

Поряд із React, розробники часто використовують Angular та Vue.js. Angular, розроблений Google, є потужним фреймворком, який включає в себе всі необхідні інструменти для створення масштабованих веб-додатків. Angular використовує TypeScript, що додає статичну типізацію і підвищує стабільність коду. Фреймворк також відзначається високою продуктивністю завдяки вдосконаленій системі змінних зон (zones) та оптимізованому механізму зміни стану. Проте складність Angular може відлякати новачків, оскільки він має стрімку криву навчання (рис. 1.2).



Рисунок 1.2 – Відвідуваність Angular-документів [2]

Vue.js, навпаки, є легким і простим у використанні. Цей фреймворк забезпечує інтуїтивно зрозумілий API та хорошу документацію, що робить його ідеальним вибором для невеликих проєктів та лендінгових сторінок. Vue.js підтримує двосторонню прив'язку даних, що дозволяє легко синхронізувати модель і вид, проте він поступається Angular та React у плані екосистеми та підтримки великих проєктів.

Фреймворки мають суттєві переваги порівняно з використанням "ванільного" JavaScript. Зокрема, вони забезпечують більш структурований підхід до розробки, значно зменшуючи час, необхідний на вирішення рутинних завдань. Крім того, сучасні фреймворки мають інструменти для автоматизації, перевірки та оптимізації коду. Проте використання ванільного JavaScript може бути виправданим для невеликих проєктів, де складність технологій є недоречною.

Для ефективної роботи розробника важливе значення має вибір інтегрованого середовища розробки (IDE). Найпопулярнішими інструментами є Visual Studio Code, WebStorm та Sublime Text. Visual Studio Code відзначається своєю легкістю, широкими можливостями розширення та активною спільнотою. Інтеграція з Git, доступ до багатьох розширень і підтримка TypeScript роблять його універсальним рішенням для розробників на React, Angular чи Vue.js.

WebStorm, розроблений компанією JetBrains, пропонує розширені можливості автодоповнення, вбудовану підтримку TypeScript та інтеграцію з популярними фреймворками. WebStorm особливо підходить для великих проєктів, де критично важлива стабільність і точність роботи. Sublime Text, хоч і менш функціональний у порівнянні з VS Code чи WebStorm, залишається популярним серед розробників завдяки швидкодії та простоті використання.

Розвиток фронтенд-технологій продовжує набирати обертів, і очікується, що у найближчі роки основна увага буде приділятися покращенню продуктивності та інтеграції з новими платформами. Однією з перспективних тенденцій є використання WebAssembly, яке дозволяє запускати на веб-сторінках код, написаний на різних мовах програмування, наприклад C++ або Rust. Це відкриває нові можливості для створення високопродуктивних веб-додатків.

Іншим важливим напрямком є розвиток інструментів для автоматизації процесів, таких як турбо-пакети (Turbopack), що дозволяють скоротити час компіляції та оптимізації коду. Також зростає популярність server-side rendering (SSR) і static site generation (SSG), що забезпечують швидке завантаження сторінок та покращують SEO.

Вибір технологій для створення лендінгової сторінки залежить від вимог до проєкту. Якщо потрібна швидка розробка та підтримка інтерактивних елементів, найкращим вибором буде React завдяки його гнучкості та багатій екосистемі. Angular може бути доцільним для проєктів, які потребують складної логіки, тоді як Vue.js стане відмінним варіантом для невеликих і середніх проєктів.

Серед IDE рекомендується Visual Studio Code, оскільки він забезпечує баланс між функціональністю та легкістю у використанні. Для оптимізації процесу розробки слід використовувати інструменти, такі як Prettier для форматування коду, ESLint для його перевірки та Webpack або Vite для збору проєктів. Для забезпечення швидкого завантаження сторінки важливо звертати увагу на оптимізацію зображень і використання CDN для статичних файлів

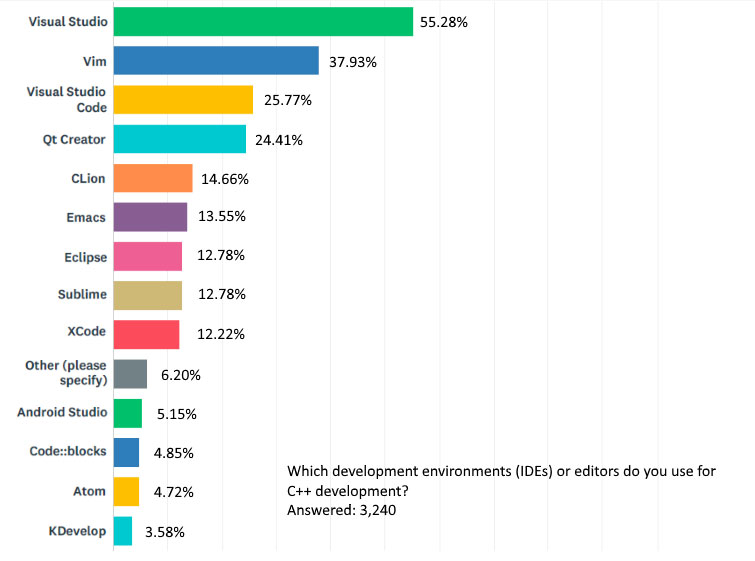
**

Рисунок 1.3 – Топ кращих IDE для фронтенду [3]

Таким чином, використання сучасних технологій дозволяє створювати ефективні, масштабовані та зручні веб-додатки, які відповідають сучасним вимогам ринку.

## 1.2 Постановка завдання на курсову роботу

Тема цієї курсової роботи – "Розробка лендінгової сторінки, використовуючи технології React". Метою роботи є дослідження та практична реалізація лендінгової сторінки для сучасного веб-додатку з використанням популярних технологій фронтенд-розробки, зокрема бібліотеки React. Реалізація лендінгової сторінки є важливим етапом у створенні будь-якого веб-сервісу, оскільки саме ця сторінка відіграє ключову роль у залученні користувачів та формуванні першого враження від продукту.

Для досягнення цієї мети необхідно визначити і реалізувати кілька конкретних завдань, кожне з яких вимагає ґрунтовного аналізу та застосування сучасних інструментів розробки:

– аналіз вимог до лендінгової сторінки та її функціоналу.;

– вибір та впровадження технологій для створення лендінгової сторінки;

– розробка та оптимізація інтерфейсу користувача;

– інтеграція з зовнішніми сервісами та бібліотеками;

– тестування та деплоймент лендінгової сторінки.

# РОЗДІЛ 2 РОЗРОБКА ЛЕНДІНГОВОЇ СТОРІНКИ

## 2.1 Обґрунтування вибору технологій і засобів вирішення поставленого завдання

У рамках курсової роботи на тему "Розробка лендінгової сторінки, використовуючи технології React" було обрано набір сучасних технологій, що відповідають вимогам високої продуктивності, зручності розробки та можливості масштабування проекту. У цьому розділі буде детально розглянуто кожен з інструментів, який планується використовувати для виконання завдання. Вибір React для розробки лендінгової сторінки обґрунтований його популярністю та здатністю ефективно справлятися з розробкою динамічних веб-інтерфейсів. React дозволяє створювати компоненти — автономні одиниці інтерфейсу, що сприяє повторному використанню коду і полегшує його підтримку. Основною перевагою є віртуальний DOM, що дозволяє мінімізувати операції з реальним DOM, підвищуючи продуктивність і забезпечуючи швидку реакцію інтерфейсу на зміни даних. Завдяки цьому, React є ідеальним вибором для створення швидких, інтерактивних лендінгових сторінок, де кожна деталь повинна миттєво реагувати на зміну стану.

Для стилізації лендінгової сторінки було обрано Tailwind CSS, оскільки він є утилітарним CSS-фреймворком, який дозволяє швидко і ефективно створювати адаптивні та гнучкі дизайни. Tailwind забезпечує високу гнучкість, дозволяючи визначати стильові властивості без написання громіздких CSS-класів. Це дозволяє зберігати чистоту коду та забезпечувати зручну роботу з різними компонентами. Однією з головних переваг є можливість швидко змінювати стиль за допомогою класів, що значно прискорює процес розробки.

Завдяки підтримці адаптивного дизайну в Tailwind, створення мобільної версії лендінгової сторінки стає простішим завданням. Важливим аспектом є також препроцесор SASS, який дозволяє розширити можливості Tailwind і значно полегшує роботу з великими проектами, додаючи змінні, вкладені стилі та інші переваги.

Оскільки лендінгова сторінка повинна бути не тільки функціональною, але й привабливою для користувачів, було обрано Framer Motion для реалізації анімацій. Цей інструмент дозволяє створювати плавні та продуктивні анімації для компонентів, що забезпечує високу взаємодію з користувачем. З Framer Motion можна легко реалізовувати анімації для елементів, таких як кнопки, форми або слайдери, що значно покращує досвід користувача і підвищує залученість.

Для зручності і забезпечення чистоти коду було обрано react-icons — бібліотеку, що дозволяє використовувати векторні іконки без необхідності завантаження важких зображень. Це забезпечує зниження часу завантаження сторінки і покращує загальну продуктивність. Бібліотека підтримує понад 15 000 різних іконок, що дає можливість вибрати найкращі варіанти для будь-якої задачі.

Vercel був обраний для деплойменту лендінгової сторінки через свою простоту та високу продуктивність у роботі з статичними сайтами. Платформа автоматично оптимізує код, мінімізує час завантаження та пропонує потужні інструменти для автоматичного тестування і CI/CD (неперервна інтеграція та розгортання). Завдяки інтеграції з GitHub, процес деплойменту стає ще більш зручним та автоматизованим.

Основною мовою програмування для реалізації логіки лендінгової сторінки є JavaScript. Це потужна і універсальна мова, яка використовується на всіх етапах веб-розробки. У поєднанні з React, JavaScript дозволяє створювати складні веб-додатки з високим рівнем взаємодії та ефективною обробкою подій. JavaScript також підтримує асинхронне програмування за допомогою Promise та async/await, що дозволяє покращити швидкість виконання операцій, таких як запити до серверів чи інтеграція з API.

Для ефективної організації процесу розробки та забезпечення високої якості коду планується використання ESLint для перевірки стилю коду та Prettier для автоматичного форматування. Для управління залежностями та пакунками буде використовуватися npm — стандартний інструмент для JavaScript-проектів, який забезпечує швидке й ефективне управління бібліотеками.

Вибір технологій для розробки лендінгової сторінки з використанням React обґрунтовано їхньою здатністю забезпечити високу продуктивність, інтерактивність та зручність у підтримці. Інструменти, такі як Tailwind CSS, Framer Motion та react-icons, допоможуть реалізувати адаптивний та естетичний інтерфейс з мінімальними затратами часу на розробку. Платформа Vercel надасть можливість легко та швидко деплоїти готовий проект. Усе це забезпечить ефективну та сучасну розробку лендінгової сторінки, яка буде відповідати вимогам користувачів і забезпечить високий рівень залученості.

## 2.2 Етапи розробки лендінгової сторінки

Розробка лендінгової сторінки є складним і багатоступінчатим процесом, що включає кілька етапів, від початкового аналізу вимог до фінального розгортання готового продукту на хостингу. Кожен з етапів має велике значення для досягнення високої якості кінцевого результату. У цьому розділі буде детально описано процес розробки лендінгової сторінки з використанням технологій React.

Першим етапом розробки є детальне розуміння цілей і вимог до лендінгової сторінки. Важливо визначити, яка інформація має бути представлена на сторінці, хто є цільовою аудиторією, і які функції має виконувати сторінка (наприклад, форма зворотного зв’язку, підписка на розсилку, або інтеграція з соціальними мережами). Крім того, на цьому етапі обговорюються такі аспекти, як необхідність адаптивного дизайну, вимоги до швидкості завантаження та продуктивності сторінки. На основі цих даних створюється технічне завдання, яке визначає основні принципи розробки.

Після аналізу вимог переходимо до проектування макета сторінки. Для цього було обрано Figma, популярний інструмент для створення інтерактивних прототипів і макетів. Figma дозволяє розробникам і дизайнерам працювати спільно в реальному часі, що робить процес створення макета більш ефективним. На цьому етапі визначаються розміри елементів, колірна схема, шрифти, структура сторінки та її адаптивність до різних пристроїв.

Макет, створений в Figma, є важливим етапом, оскільки він забезпечує чітке розуміння того, як сторінка виглядатиме на різних етапах розробки. Дизайнер створює кілька варіантів розміщення елементів, враховуючи UX/UI принципи та забезпечуючи зручність користування. Розробник отримує доступ до макету для подальшого втілення в коді (рис. 2.1).

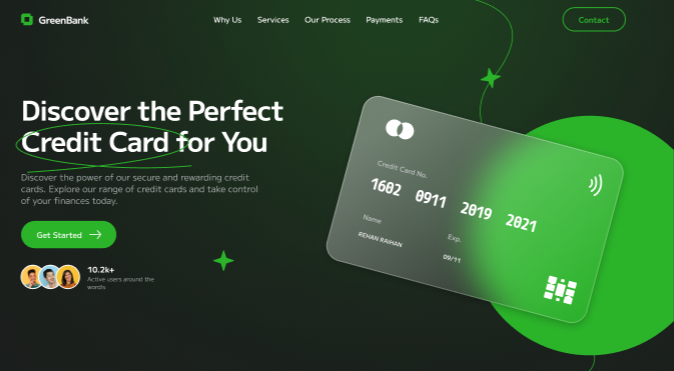


Рисунок 2.1 – Макет лендінгово сайту

Наступним етапом є реалізація інтерфейсу користувача (UI) на основі макета. Для цього використовуються React та Tailwind CSS. React дозволяє створювати повторно використовувані компоненти інтерфейсу, що значно підвищує ефективність розробки і знижує кількість помилок. Компоненти можна структурувати, щоб забезпечити їх незалежність та повторне використання в різних частинах сторінки. У той же час, Tailwind CSS забезпечує швидку стилізацію сторінки завдяки своїм утилітарним класам, що дозволяє уникнути написання громіздких CSS-правил.

Створення компонентів починається з базових елементів, таких як кнопки, навігаційне меню, форма підписки. Кожен компонент реалізується із використанням станів (state) і ефектів (useEffect) у React для забезпечення інтерактивності сторінки. Наприклад, кнопки можуть змінювати свій вигляд при наведенні курсору, а форма підписки може перевіряти правильність введених даних перед відправленням.

Тестування є невід’ємною частиною процесу розробки. Тому, основним етапом є інтеграційне тестування, яке перевіряє, як компоненти взаємодіють один з одним. Важливо також провести тестування адаптивності, щоб переконатися, що сторінка коректно відображається на різних пристроях.

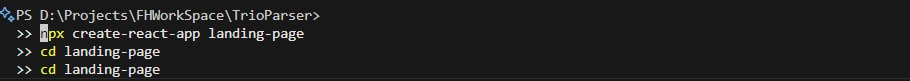
Для підвищення надійності розробленої сторінки, тестування повинно проводитись на різних браузерах та в різних мережевих умовах, щоб упевнитися в її високій продуктивності.

Після успішного завершення тестування наступає етап деплойменту лендінгової сторінки. Для цього був обраний Vercel, оскільки ця платформа спеціалізується на швидкому розгортанні статичних сайтів і інтегрується безпосередньо з GitHub, що забезпечує автоматичне деплоймент з кожним оновленням коду.

## 2.3 Практична реалізація лендінгової сторінки

У цьому підрозділі детально описано процес розробки лендінгової сторінки з використанням технологій React. Завдання полягало в тому, щоб створити сучасний і функціональний інтерфейс, який би відповідав вимогам користувачів та мав високу продуктивність. Для цього використовувалися такі технології, як React, Tailwind CSS, framer-motion, react-icons та Vercel для хостингу. Оскільки проект не передбачав використання бекенд-технологій, ми зосередилися на фронтенд-частині, забезпечивши інтерактивність і ефективність лендінгової сторінки.

Спочатку розпочнемо з налаштування середовища розробки. Для початку було створено проект за допомогою Create React App, що дозволило швидко налаштувати середовище для розробки (рис. 2.2)

Рисунок 2.2 – Створення реакт-проєкту

Далі був підключений Tailwind CSS, що є потужним інструментом для швидкої стилізації компонентів (рис. 2.3).

Рисунок 2.3 – Налаштування Tailwind CSS

Після встановлення потрібно імпортувати Tailwind у CSS файл index.css, реалізацію можна побачити на лістингу 2.1.

Лістинг 2.1 – Імпорт Tailwind у CSS файл

@tailwind base;

@tailwind components;

@tailwind utilities;

кінець лістингу 2.1

Одним із основних завдань було створення динамічної секції, яка має взаємодію з користувачем. Для цього було використано React, а також framer-motion для анімацій. Це дозволило додати анімаційні ефекти при завантаженні елементів та їх взаємодії. Приклад реалізації цієї функції на лістингу 2.2.

Лістинг 2.2 – Приклад реалізації framer-motion у коді

<motion.button

whileHover={{ scale: 1.05 }}

whileTap={{ scale: 0.95 }}

type="submit"

className="px-4 py-2 rounded-md text-sm font-medium text-white bg-[#2BB32A] hover:bg-[#2BB32A]/90 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-[#2BB32A] transition duration-200 shadow-lg"

>

Send

</motion.button>

кінець лістингу 2.2

Іншою важливою частиною проекту була реалізація форми зворотного зв'язку. Оскільки в проекті не використовувався бекенд, для обробки даних форми не передбачалася серверна частина, але було реалізовано місце для подальшої інтеграції, якщо в майбутньому це стане необхідним. Приклад створеної форми можна побачити на лістингу 2.3.

Лістинг 2.3 – Реалізація фронтенду форми зворотнього зв’язку

import React, { useState } from "react";

import { motion, AnimatePresence } from "framer-motion";

const ContactForm = ({ setIsContactModalOpen }) => {

const [name, setName] = useState("");

const [email, setEmail] = useState("");

const [message, setMessage] = useState("");

return (

<AnimatePresence>

<motion.div

className="fixed inset-0 px-4 bg-[#002b1f] bg-opacity-70 flex items-center justify-center z-50"

initial={{ opacity: 0 }}

animate={{ opacity: 1 }}

exit={{ opacity: 0 }}

transition={{ duration: 0.3 }}

>

<motion.div

className="bg-gradient-to-br from-[#003726] to-[#001d12] text-white p-8 rounded-xl max-w-md w-full shadow-xl border border-[#2BB32A]/30"

initial={{ scale: 0.8, opacity: 0 }}

animate={{ scale: 1, opacity: 1 }}

exit={{ scale: 0.8, opacity: 0 }}

transition={{ duration: 0.3 }}

>

<h2 className="text-3xl font-extrabold mb-4 text-[#2BB32A] text-center">

Contact Us

</h2>

<p className="text-sm text-gray-300 mb-6 text-center">

Have questions? Let us know and we'll get back to you!

</p>

<form className="space-y-6">

<div>

<label

htmlFor="name"

className="block text-sm font-medium text-gray-200"

>

Name

</label>

<div className="relative">

<input

type="text"

id="name"

value={name}

onChange={(e) => setName(e.target.value)}

name="name"

className="mt-1 block w-full rounded-lg bg-transparent border border-gray-500 text-white px-4 py-3 shadow-sm transition duration-200 ease-in-out focus:border-[#2BB32A] focus:ring focus:ring-[#2BB32A]/50 focus:outline-none placeholder-gray-400"

placeholder="Enter your name"

/>

<div className="absolute inset-y-0 right-4 flex items-center pointer-events-none text-gray-400">

<i className="fas fa-user"></i>

</div>

</div>

</div>

<div> <label

htmlFor="email"

className="block text-sm font-medium text-gray-200"

>

Email </label>

<div className="relative">

<input

type="email"

value={email}

onChange={(e) => setEmail(e.target.value)}

id="email"

name="email"

className="mt-1 block w-full rounded-lg bg-transparent border border-gray-500 text-white px-4 py-3 shadow-sm transition duration-200 ease-in-out focus:border-[#2BB32A] focus:ring focus:ring-[#2BB32A]/50 focus:outline-none placeholder-gray-400"

placeholder="Enter your email"

/>

<div className="absolute inset-y-0 right-4 flex items-center pointer-events-none text-gray-400">

<i className="fas fa-envelope"></i> </div>

</div>

</div><div> <label

htmlFor="message"

className="block text-sm font-medium text-gray-200"

> Message</label>

<textarea

id="message"

value={message}

onChange={(e) => setMessage(e.target.value)}

name="message"

rows={4}

className="mt-1 block w-full rounded-lg bg-transparent border border-gray-500 text-white px-4 py-3 shadow-sm transition duration-200 ease-in-out focus:border-[#2BB32A] focus:ring focus:ring-[#2BB32A]/50 focus:outline-none placeholder-gray-400"

placeholder="Write your message here..."

></textarea>

</div>

<div className="flex justify-end space-x-4">

<motion.button

whileHover={{ scale: 1.05 }}

whileTap={{ scale: 0.95 }}

type="button"

onClick={() => setIsContactModalOpen(false)}

className="px-4 py-2 rounded-md border border-[#2BB32A] text-sm font-medium text-[#2BB32A] bg-transparent hover:bg-[#2BB32A]/10 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-[#2BB32A] transition duration-200"

>

Cancel

</motion.button>

<motion.button

type="submit"

whileHover={{ scale: 1.05 }}

whileTap={{ scale: 0.95 }}

className="px-4 py-2 rounded-md text-sm font-medium text-white bg-[#2BB32A] hover:bg-[#2BB32A]/90 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-[#2BB32A] transition duration-200 shadow-lg"

> Send

</motion.button>

</div>

</form>

</motion.div>

</motion.div>

</AnimatePresence>

);

};

export default ContactForm;

кінець лістингу 2.3

Для покращення зовнішнього вигляду сторінки, ми використовували бібліотеку react-icons, яка надає велику кількість іконок для різних цілей. Наприклад, було додано іконку у футері сторінки, код зображено на лістингу 2.4.

Лістинг 2.4 – Реалізація бібліотеки react-icons

<p className="flex gap-2 items-center justify-center">

<BiCopyright className="relative top-[1px]" />

{new Date().getFullYear()} DoraDesign All Rights Reserved

</p>

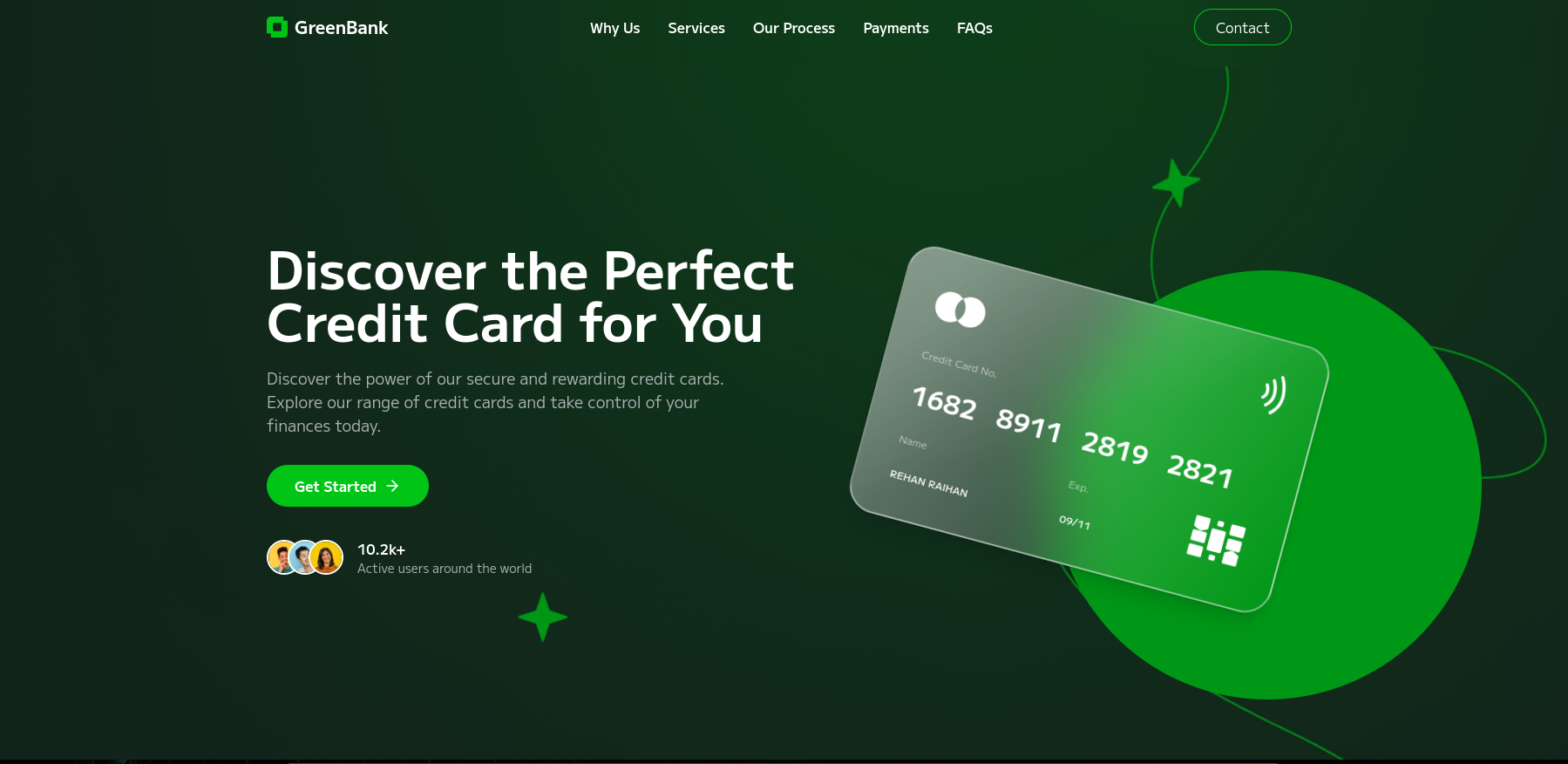
кінець лістингу 2.4

Оскільки проект не містив серверної частини, розгортання було виконано за допомогою Vercel, що є зручним інструментом для деплоя фронтенд-додатків. Всі зміни, внесені в проект, автоматично деплояться після пушу в репозиторій на GitHub. Щоб розгорнути проект, необхідно було:

– Підключити репозиторій до Vercel через GitHub.

– Вибрати правильні налаштування (платформа — React).

– Натискати "Deploy", після чого проект автоматично розгортався на безкоштовному хостингу.

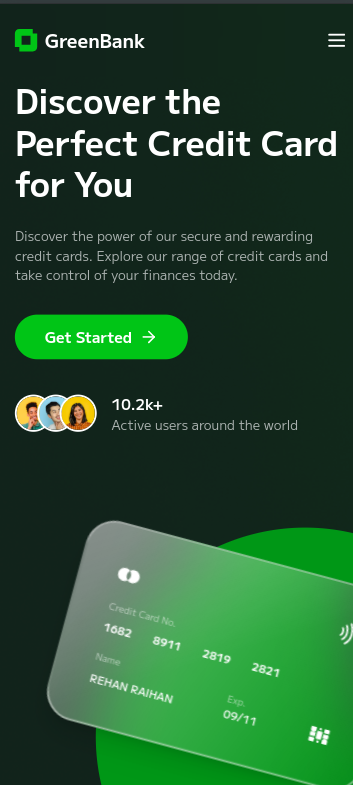


￼Рисунок 2.4 – Реалізована лендінгова сторінка

## 2.4 Тестування лендінгової сторінки

Тестування лендінгової сторінки, розробленої за допомогою React, є необхідною частиною процесу розробки, яка дозволяє впевнитися в коректній роботі всіх компонентів та функціональних елементів. Основною метою тестування було забезпечити, щоб сторінка коректно відображалася на різних пристроях і розмірах екранів, а також перевірити правильність роботи інтерактивних елементів та забезпечення продуктивності сторінки.

Першим етапом тестування було перевірка візуального відображення компонентів на різних пристроях. Отже, спочатку для перевірки адаптивності сайту було проведено тестування на мобільних пристроях (рис. 2.5).



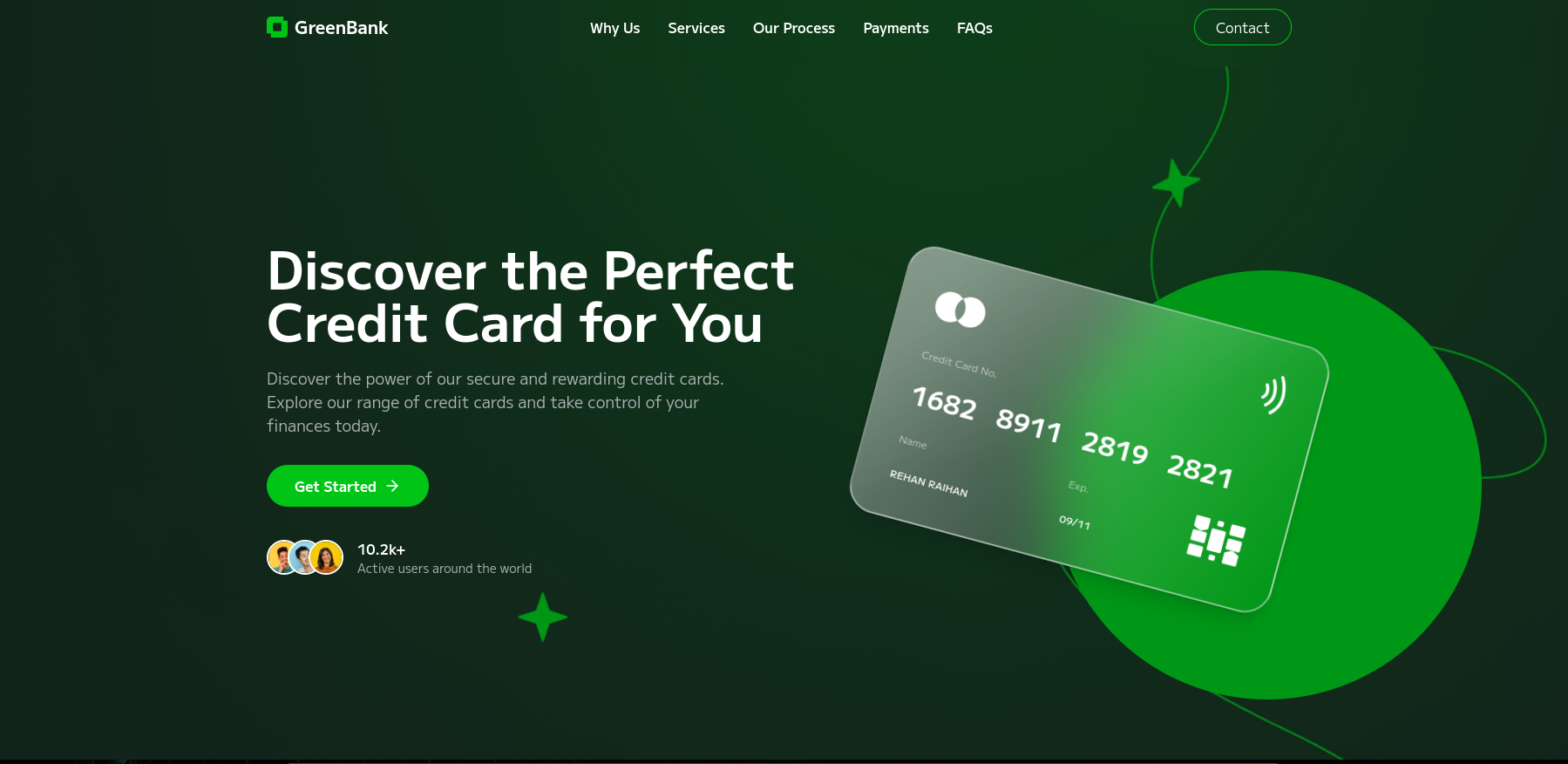
￼Рисунок 2.5 – Сайт на мобільних пристроях

На планшетах (рис. 2.6) основна увага під час тестування була зосереджена на коректності масштабування елементів та правильному розташуванні блоків дизайну. Використання адаптивних утиліт Tailwind CSS дозволило зберегти чіткість структури та естетичний вигляд сайту, незважаючи на зміну орієнтації екрану.



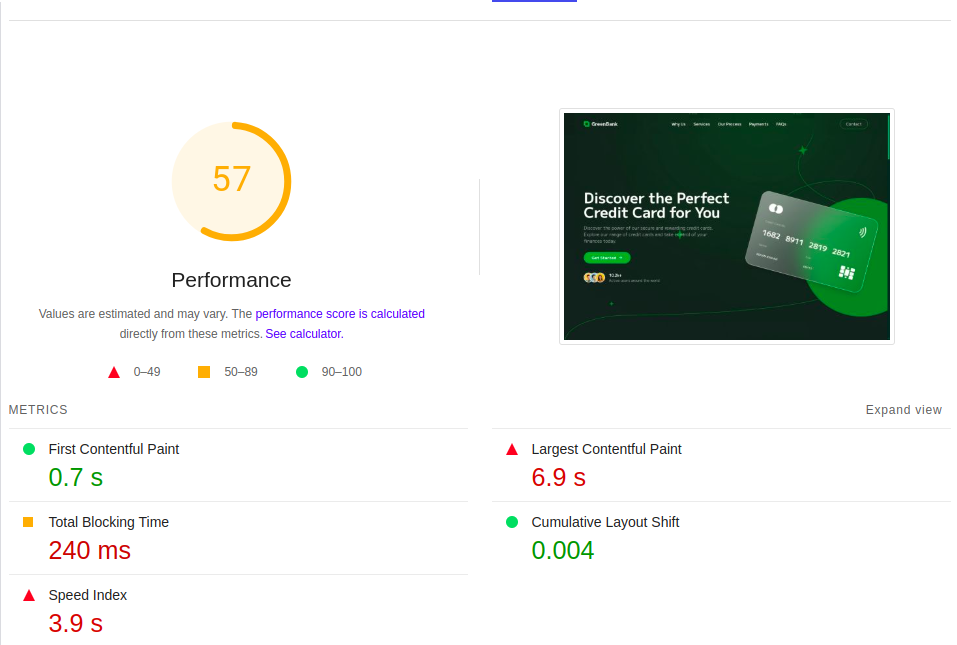
￼Рисунок 2.6 – Сайт на планшетних пристроях

Тестування адаптивності на персональних комп’ютерах (рис. 2.7)



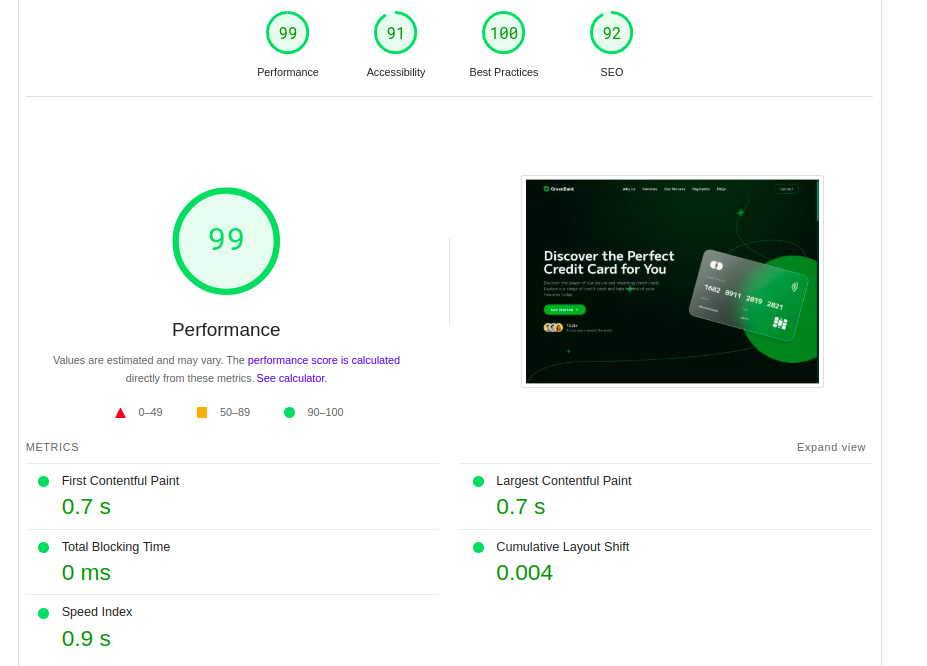
￼Рисунок 2.7 – Лендінгова сторінка на ПК

Останнім етапом тестування стало перевірка продуктивності сторінки. Це важливо, оскільки лендінгові сторінки мають бути максимально швидкими, щоб користувачі не залишали їх через довге завантаження. Для цього використовувався інструмент Lighthouse, який дозволяє оцінити швидкість завантаження сторінки та виявити можливі проблеми з продуктивністю. Результати тестування показали, що сторінка завантажується досить повільно, тому це потрібно пофіксити (рис. 2.8).



￼Рисунок 2.8 – Результати тестування Lighthouse

Вже повторний тест показав зміни, які можна рахувати задовільними (рис. 2.9). Для досягнення таких цілей було використано squoosh.app для зменшення розміру картинок і таких видалили непотрібний код.



￼Рисунок 2.9 – Повторний результат тестування Lighthouse

# ВИСНОВКИ

У процесі виконання курсової роботи на тему "Розробка лендінгової сторінки, використовуючи технології React", було досягнуто чітко визначених цілей, що стосуються створення ефективної, функціональної та адаптивної лендінгової сторінки з використанням сучасних інструментів фронтенд-розробки. Розробка включала застосування таких технологій, як React, Tailwind CSS, framer-motion, react-icons і хостинг на платформі Vercel.

Робота була виконана в повному обсязі відповідно до завдань, визначених у вступній частині курсової роботи. Було успішно створено лендінгову сторінку, що включала основні функціональні елементи, такі як інтерактивні блоки, адаптивний дизайн, анімації для покращення взаємодії з користувачем. Реалізація дизайну з використанням Tailwind CSS дозволила швидко та ефективно створити привабливий та адаптивний інтерфейс, а також мінімізувати час розробки завдяки готовим класам для стилів. Компонентний підхід у React дозволив легко організувати структуру проекту та забезпечити високу реюзабельність коду.

Усі завдання, передбачені технічним завданням, були виконані в межах наданого часу. Вдалося створити лендінгову сторінку з інтерактивними елементами, що включає основну частину сторінки, контактну форму зворотного зв'язку та анімації для покращення візуального сприйняття. Було також здійснено налаштування середовища розробки, що включає інтеграцію з платформою Vercel для хостингу проекту. Проте, реалізація функцій, пов'язаних з бекендом або інтеграцією бази даних, в рамках цієї роботи не передбачалася.

Вдосконалення розробленої лендінгової сторінки може включати кілька напрямків. По-перше, доцільно реалізувати інтеграцію з бекендом для забезпечення функціональності, як-от збереження даних з форми зворотного зв'язку або обробка запитів користувачів. По-друге, варто додати оптимізацію для покращення показників ефективності, таких як швидкість завантаження сторінки, застосовуючи техніки lazy loading для зображень та скриптів, а також використання більш ефективних форматів зображень, таких як WebP. Крім того, можна поліпшити доступність сторінки шляхом додавання додаткових елементів для покращення взаємодії з користувачами, що мають обмежені можливості.

Тестування лендінгової сторінки передбачало оцінку її продуктивності, зокрема, часу завантаження сторінки, а також взаємодії з елементами на сторінці. Для вимірювання продуктивності використовувались інструменти, такі як Google PageSpeed Insights та Lighthouse, що дозволили виявити основні проблеми з завантаженням ресурсів та оптимізацією. Однією з основних проблем була велика кількість ресурсів, які блокують рендеринг, зокрема шрифти та зовнішні стилі. Для покращення результатів рекомендується виконати попереднє завантаження шрифтів і відкласти завантаження не критичних ресурсів до завершення рендерингу основного контенту. Тестування також охоплювало перевірку адаптивності сторінки на різних пристроях та браузерах, що дозволило забезпечити зручний користувацький досвід на мобільних і десктопних пристроях.

У підсумку, розробка лендінгової сторінки з використанням технологій React виявилась успішною і дозволила досягти поставлених цілей, зокрема, створення ефективної, швидкодійної та адаптивної сторінки, готової до подальших вдосконалень і розширення.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Найкращі фреймворки JavaScript у 2024 році для швидкої розробки ПЗ | Блог на Stfalcon.com. *Custom Software Development Company | Stfalcon.com*. URL: <https://stfalcon.com/uk/blog/post/javascript-frameworks-for-software-development> (дата звернення: 15.11.2024).

2. Найкращі фреймворки JavaScript у 2023| Блог на Stfalcon.com. *Custom Software Development Company | Stfalcon.com*. URL: <https://stfalcon.com/uk/blog/post/javascript-frameworks-for-software-development> (дата звернення: 15.11.2024).

3. ТОП-7 популярных IDE для программирования на С++. *ITVDN*. URL: <https://itvdn.com/ru/blog/article/cplspls-top7> (дата звернення: 07.12.2024).

4. Getting Started – React. *React*. URL: <https://reactjs.org/docs/getting-started.html> (дата звернення: 15.11.2024).

5. Motion - A modern animation library for JavaScript and React. *Framer: The web builder for stunning sites.* URL: <https://www.framer.com/motion/> (дата звернення: 15.11.2024).

6. Vercel Documentation. *Vercel Documentation*. URL: <https://vercel.com/docs> (дата звернення: 15.11.2024).

7. Browse Fonts - Google Fonts. *Google Fonts*. URL: <https://fonts.google.com/> (дата звернення: 15.11.2024).

8. MDN Web Docs. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/> (дата звернення: 15.11.2024).

9. Учасники проектів Вікімедіа. Front end та back end – Вікіпедія. *Вікіпедія*. URL: [https://uk.wikipedia.org/wiki/Front\_end\_та\_back\_end](https://uk.wikipedia.org/wiki/Front_end_%D1%82%D0%B0_back_end) (дата звернення: 15.11.2024).

10. PageSpeed Insights. *Google for Developers - from AI and Cloud to Mobile and Web*. URL: <https://developers.google.com/speed/pagespeed/insights/> (дата звернення: 15.11.2024).