**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет прикладної математики та інформатики

(повне найменування назва факультету)

Прикладна математика

(повна назва кафедри)

**КУРСОВА РОБОТА**

НАЗВА РОБОТИ

Виконав(ла): студент(ка) групи Пмп-32

спеціальності 113 – прикладна математика

(шифр і назва спеціальності)

Майстренко Андріан Віталійович

(підпис) (прізвище та ініціали)

Керівник

(підпис) (посада, прізвище та ініціали)

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

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали)

Львів – 2024

**Зміст**

Завдання на курсову роботу.........................................................3

Вступ..............................................................................................4

**Розділ 1.**Теоретична особливість реалізації...............................5  
**1.1.**Мета роботи та вимоги...........................................................5

**1.2.**Оцінка попиту..........................................................................6

**1.3.**Історична частина...................................................................7

**Розділ 2.**Практична реалізація....................................................8

**2.1.**Опис початку роботи над сайтом..........................................8

**2.2.**Створення файлів app.js та server.js......................................8

**2.3.**Створення моделей ...............................................................9

**2.4.**Створення контролерів..........................................................9

**2.5.**Створення маршрутизаторів.................................................9

**2.6**.Створення pug файлів............................................................10

**2.7.**Створення допоміжних js файлів.........................................10

**2.8.**Головна сторінка....................................................................11

**2.9.**Сторінка огляду продукту......................................................11

**2.10.**Особиста сторінка................................................................12

**2.10.1.**Сторінка для зміни даних.......................................12

**2.10.2.**Сторінка із купленими продуктами.......................12

**2.10.3.**Сторінка із залишеними відгуками на товари......13

**Розділ 3.**Демонстрація роботи....................................................13

Використані технології та пакети………………………………22

Висновок........................................................................................23

Джерела..........................................................................................24

**Завдання на курсову роботу**

Розробка веб-сайту складається з кількох ключових етапів:

1. Проєктування та дизайн:
   * Розробка шаблонів HTML-сторінок для відображення різних частин сайту (наприклад, домашня сторінка, сторінка товару тощо).
   * Розробка візуального дизайну.
   * Вибір колірної гами та шрифтів, що підкреслюють бренд магазину.
2. Розробка функціональності:
   * Розробка api
   * Інтеграція каталогу товарів з можливістю фільтрації та пошуку.
   * Впровадження системи реєстрації та авторизації користувачів, щоб вони могли створювати облікові записи, редагувати особисті дані та здійснювати покупки.
   * Налаштування системи оплати для забезпечення безпечних та зручних транзакцій.
   * Підключення бази даних MongoDB
3. Тестування та вдосконалення:
   * Проведення тестування для забезпечення коректної роботи сайту.
   * Завантаження проєкту на веб хостинг Heroky
   * Виправлення виявлених помилок та оптимізація продуктивності.
   * Збір зворотного зв'язку від потенційних користувачів та внесення необхідних покращень.

**Вступ**

Завданням моєї курсової роботи було створення сучасного веб-сайту для комп'ютерного магазину. Основною метою проєкту стало розроблення функціонального, зручного та естетично привабливого інтернет-ресурсу, який би задовольняв потреби як власників магазину, так і його клієнтів.

Перед початком роботи я визначив основні інструменти або технології для створення цього сайту JavaScript,Express,HTML,CSS,pug та інші.

Під час розробки, я визначив, що для досягнення максимальної ефективності та структурованості відповідно до сучасних стандартів, сайт буде реалізований на основі принципу Model-View-Controller (MVC). Цей підхід дозволив систематизувати логіку додатку та розділити його на три основні компоненти: модель, представлення та контролер. Такий підхід спрощує розробку, підтримку та розширення кодової бази, забезпечуючи більшу стабільність та продуктивність сайту.

**Розділ 1. Теоретична особливість реалізації**

**1.1 Мета роботи та вимоги**

Метою цієї дипломної роботи є розробка та реалізація інтернет-магазину з використанням JavaScript, HTML та CSS. Для досягнення цієї мети потрібно створити веб-сторінку, що забезпечить зручне та привабливе користувацьке інтерфейс для покупки товарів онлайн.

Сайт має містити наступні дані для забезпечення функціональності:

* Каталог товарів: Сайт повинен містити список доступних товарів, розділений за категоріями або мітками, що спрощує пошук і вибір товарів користувачами.
* Інформація про товар: Для кожного товару повинна бути надана детальна інформація, включаючи назву, зображення, опис, ціну та інші характеристики, що допоможуть користувачам при прийнятті рішення про покупку.
* Кошик: Сайт має мати функціональність додавання товарів до кошика, перегляду змісту кошика, видалення товарів та розрахунку загальної вартості замовлення.
* Оформлення замовлення: Користувачі повинні мати можливість оформити замовлення, вказати доставку та спосіб оплати.

Реалізація проекту вимагатиме використання розробних інструментів, таких як Visual Studio Code, знання мов програмування JavaScript, HTML та CSS, а також використання зовнішніх бібліотек та сервісів для поліпшення функціональності та безпеки сайту.

**1.2 Оцінка попиту**

Новизна роботи полягає в розробці та реалізації інтернет-магазину з фокусом на продаж комп'ютерів та комплектуючих. Інтернет-магазини з продажу ПК вже існують на ринку, проте дана робота включає деякі унікальні аспекти та переваги:

Концентрація на продажу комп'ютерів та комплектуючих: Робота фокусується на створенні інтернет-магазину спеціалізованого на продажі

комп'ютерної техніки. Це дозволяє зосередитись на потребах та вимогах цільової аудиторії, що сприяє збільшенню релевантності та задоволенню користувачів.

Вдосконалення користувацького досвіду: Розроблений інтернет-магазин надає зручний та інтуїтивно зрозумілий користувацький інтерфейс, спрощує процес пошуку та покупки товарів. Використання сучасних технологій та дизайну допомагає забезпечити позитивний досвід покупки для відвідувачів.

Детальна інформація про товари: Кожен продукт має докладний опис, включаючи технічні характеристики, фотографії та відгуки користувачів. Це допомагає покупцям зробити обґрунтований вибір та отримати повну інформацію про товар перед покупкою.

Розширений функціонал: Робота включає реалізацію таких функціональних можливостей, як можливість збереження товарів у списку бажань, порівняння різних моделей, розрахунок вартості з урахуванням додаткових послуг (доставка, монтаж тощо). Це покращує зручність та функціональність магазину для користувачів.

Застосування сучасних технологій: Робота використовує сучасні інструменти та технології, такі як JavaScript, HTML, CSS, для розробки інтерфейсу та функціональності сайту.

Використання Firebase дозволяє забезпечити хостинг та зручну базу даних для зберігання і обробки інформації.

Ці новаторські рішення та функціональність роботи сприяють поліпшенню користувацького досвіду покупки комп'ютерної техніки в інтернет-магазинах і відрізняють її від інших подібних проектів.

**1.3 Історична частина**

Якщо робити оцінку online-ринку для покупця, це зазвичай базується на таких критеріях:

1. Асортимент товарів: Оцінюється розмаїття та наявність необхідних моделей комп'ютерів та комплектуючих.
2. Цінова політика: Оцінюється конкурентоспроможність цін, наявність акцій та знижок.
3. Якість обслуговування: Оцінюється якість консультації, швидкість обробки замовлень та доставки, можливість повернення товару
4. Відгуки користувачів: Оцінюються відгуки та рейтинги, залишені покупцями, що допомагають сформувати уявлення про надійність та якість обслуговування магазину.

Ці оцінки допомагають користувачам зробити вибір між різними інтернет-магазинами та забезпечити оптимальний досвід покупки.

**Розділ 2.Практична реалізація**

**2.1.Опис початку роботи на сайтом**

Для початку нам потрібно визначити задачу та мету нашого сайту,його тематику та проблему,яку він буде вирішувати.Коли будуть відповіді на ці питання,потрібно визначитися із середовищем розробки,у моєму випадку - Visual Studio Code (VS Code).Також при розробці нашого сайту нам стане в допомозі Postman. Postman - це популярний інструмент для тестування та розробки API (Application Programming Interface), який використовується розробниками програмного забезпечення.

Також я визначився із базою даних,де будуть зберігатися усі потрібні дані,а саме MongoDB, адже є дуже популярною та розвиненою для роботи.Також для повноцінної реалізації вебсайту нам потрібно буде завантажити цей проєкт на веб-хостинг,щоб зробити його доступним для користувачів. Для вирішення цього я використав Heroku.

Heroku - це популярна платформа як послуга (PaaS), яка дозволяє розробникам програмного забезпечення легко розгортати, масштабувати та управляти своїми веб-застосунками. Вона пропонує безліч функцій, які полегшують процес розробки та розгортання, роблячи її популярним вибором для багатьох команд.

**2.2.Створення файлів app.js та server.js**

Файл server.js у вашій курсовій роботі виконує дві ключові функції:

1. **Налаштування та запуск веб-застосунку Node.js:**
   * Завантажує змінні середовища з .env файлу для безпечного зберігання конфігурацій.
   * Обробляє непередбачені помилки та забезпечує елегантне завершення роботи сервера.
   * Підключається до бази даних MongoDB для зберігання та отримання даних.
   * Налаштовує та запускає сервер Express, який обробляє HTTP-запити до вашого веб-застосунку.
2. **Інтеграція з Express-застосунком:**
   * Імпортує модуль Express із файлу app.js.
   * Цей модуль app ймовірно, містить код, який визначає:
     + Маршрути API для обробки запитів до веб-застосунку.
     + Логіку обробки даних для отримання, створення, оновлення та видалення даних з бази даних.
     + Будь-яке проміжне програмне забезпечення для авторизації, валідації або інших завдань.

Файл app.js виконує обробку власних помилок, маршрутизацію запитів та обробку загальні помилки.

**2.3.Створення моделей**

В нас є основні чотири моделі.Модель для продуктів,покупок,відгуків та користувачів.

Моделі є основою для зберігання та роботи з даними у node.js/express. Вони визначають структуру даних,що дозволяє організувати та керувати інформацією ефективно.

Моделюємо інформацію за допомогою mongoose для MongoDB,mongoose.Schema містить всі необхідні,прописані поля моделі даних.MongoDB на основі цієї схеми автоматично створює колекції.

Моделі забезпечують чітку структуру для даних,що полегшує їх розуміння та керування.Одночасно ми можемо виконувати певну валідацію даних,що буде гарантувати правильність даних перед зберіганням їх у базі даних.

**2.4.Створення контролерів**

Контролери відіграють ключову роль в обробці запитів користувачів у додатку Node.js/Express. Вони діють як посередники між маршрутизацією (яка співставляє URL-адреси з функціями обробки) та моделями даних (які зберігають і отримують дані).

У даному вебсайті реалізовано багато контролерів,котрі відповідають за автентифікацію,помилки,продуктів,відгуків,користувачів та виглядів,тобто pug файлів.

Кожен із цих файлів містить функції,котрі забезпечують комфортну та коректну роботу із моделями даних та маршрутизацією.

**2.5.Створення маршрутизаторів**

Маршрутизатори - це ключовий компонент у додатку Node.js/Express, який відповідає за співставлення вхідних HTTP-запитів (URL-адрес і методів) з відповідними функціями обробки. Вони допомагають організувати та керувати маршрутами застосунку, роблячи його більш структурованим і масштабованим.

Структура цього файлу полягає із:

* Імпорту необхідних модулів(потрібних контролерів та інше)
* Створення екземпляру роутера(використовується для визначення маршрутів)
* Визначення маршрутів
* Вкладені маршрути("/:productId/reviews")
* Експорт маршрутизатора

**2.6.Створення pug файлів**

PUG (раніше відомий як Jade) - це простий і потужний шаблонізатор для Node.js. Він дозволяє створювати HTML-подібні структури з лаконічним і читабельним синтаксисом.

**Переваги використання PUG:**

* **Простота:** Синтаксис PUG схожий на HTML, що робить його легким для вивчення та використання.
* **Лаконічність:** PUG дозволяє писати менше коду, ніж HTML, роблячи шаблони більш читабельними та підтримуваними.
* **Гнучкість:** PUG підтримує динамічні елементи, умовні блоки та інші функції, що робить його потужним інструментом для створення складних шаблонів.
* **Розширюваність:** PUG має широкий спектр плагінів, які розширюють його функціональні можливості.

PUG є популярним вибором для розробки Node.js, оскільки він пропонує простий, лаконічний та гнучкий спосіб створення HTML-шаблонів.

**2.7.Створення допоміжних js файлів**

Немаловажливу роль відіграють окремі js файли,котрі дозволяти розширити функціонал нашого веб застосунку.

В нашому випадку це :

* alerts.js - використовується для створення власних повідомлень
* login.js - виконує вхід до нашого облікового запису
* logout.js - виконує вихід з нашого облікового запису
* signup.js - виконує реєстрацію нового облікового запису
* stripe.js - файл,котрий допомагає реалізовувати оплату за продукти за допомогою популярного інструменту stripe
* updateSettings.js. - використовуємо для оновлення даних користувача самим користувачем на сторінці користувачу

**2.8.Головна сторінка**

Головна сторінка у нашому застосунку досягається за покликанням   
<https://computer-shop-andrian-cec202636687.herokuapp.com/>

при доступі до цього посилання, app.js буде викликати спеціальний маршрутизатор,в нашому випадку viewRouter .Відповідно до цього покликання буде викликатися get метод,котрий буде викликати спеціальні функції,котрі мають назву middleware functions.В нашому випадку він викличе функцію із viewController котра має назву getOverview, котрий буде діставати із бази даних дані про продукти,генерувати відповідний pug шаблон із назвою ‘overview’ та передасть туди інформацію про продукти.Після генерації цього шаблону ми отримаємо нашу основну сторінку перегляду каталогу.

**2.9.Сторінка огляду продукту**

Сторінка огляду конкретного продукту викликається натиском на кнопку відповідного продукту,котрий містить кнопку ‘Details’.Цей натиск буде доступатися подібного покликання  
<https://computer-shop-andrian-cec202636687.herokuapp.com/product/><product-slug>, де <product-slug> залежить від назви конкретного продукту.

Структура MVC (model view controller), дозволяє нам спрости процес генерування та відображення сторінок.За допомогою цієї структури можна спростити процес відображення сторінок і дотримуватися логіки,наведеної у відображенні головної сторінки.  
 Логіка залишається тою самою,тобто app.js направляє для спеціального маршрутизатора,котрий у свою чергу залежно від покликання буде викликати відповідні функції із відповідного контролера із відповідним генеруванням потрібного шаблону.

**2.10.Особиста сторінка**

Дана сторінка залежить від користувача,проте вони мають спільні функції:

* зміна даних(ім’я,електронна адреса,фотографія)
* зміна паролю
* можливість доступу до покупок,здійснених покупок
* можливість доступу до відгуків, зроблених користувачем

За допомогою вищевказаних допоміжних js файлів,таких як updateSettings.js, viewsController.getMyProducts та viewsController.getMyReviews, можемо виконувати такі операції.

**2.10.1.Сторінка для зміни даних**

Дана сторінка водночас є особистою сторінкою користувача.

Заповнивши потрібні поля та вибравши фотографію, ми можемо змінити дані користувача.Важливо зауважити,що ми можемо оновлювати лише окремі поля,тобто ми можемо оновити лише ім’я або лише фотографію.

Окремим блоком є блок змінення паролю.Цей блок є окремим,адже для цього використовується інше покликання та окремий js файл.

**2.10.2.Сторінка із купленими продуктами**

Ця сторінка реалізована цікавим чином,адже нам потрібно колекції покупок вибрати id продуктів,де id користувача буде співпадати із id поточного користувача.Далі ми вибираємо товари із колекції Products ті товари чи продукти, id котрих будуть співпадати із id, котрі ми вибрали із Purchases.

Для цієї сторінки був використаний такий самий шаблон,як і для головної сторінки,адже і там,і там відображаються товари.

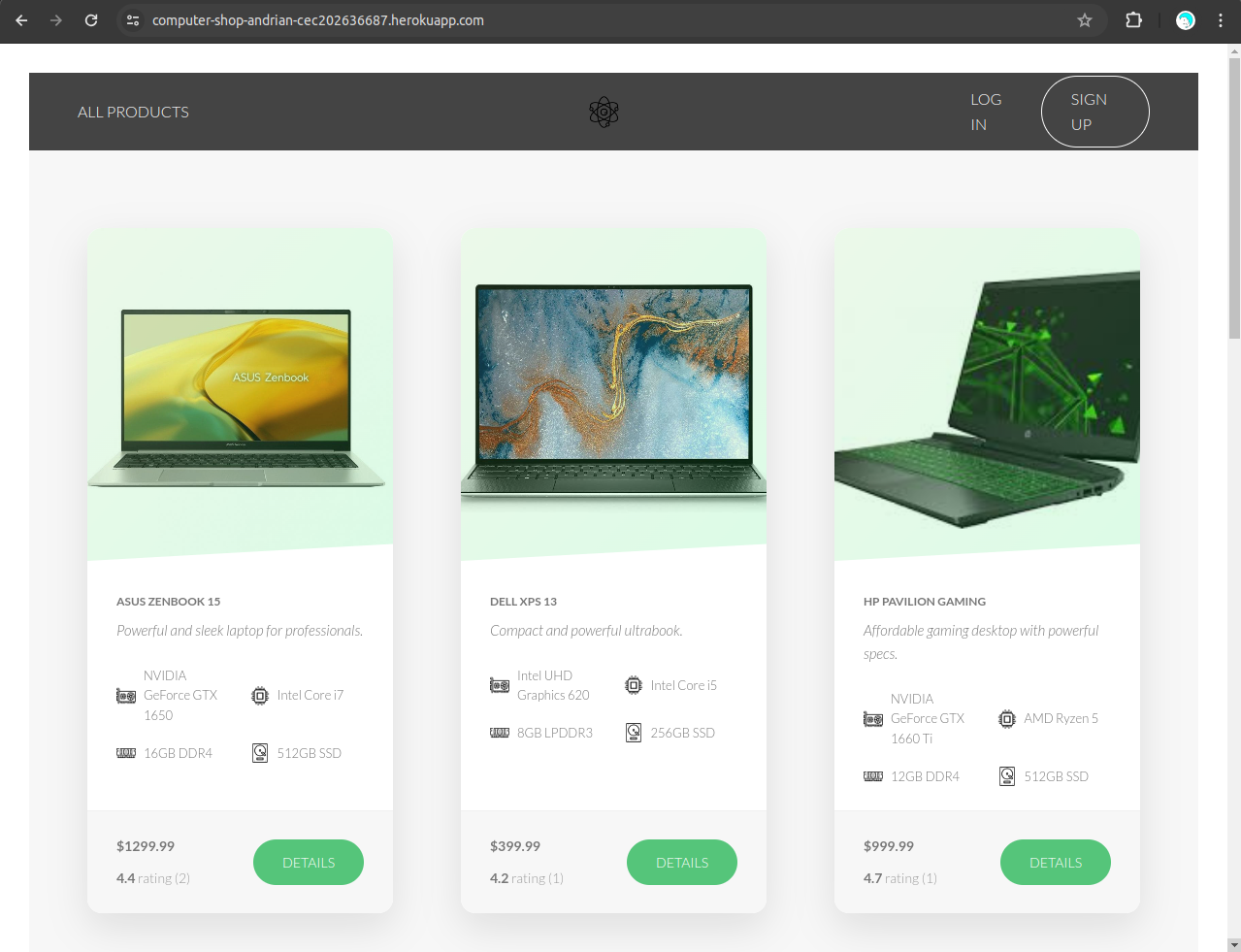
**2.10.3.Сторінка із залишеними відгуками на товари**

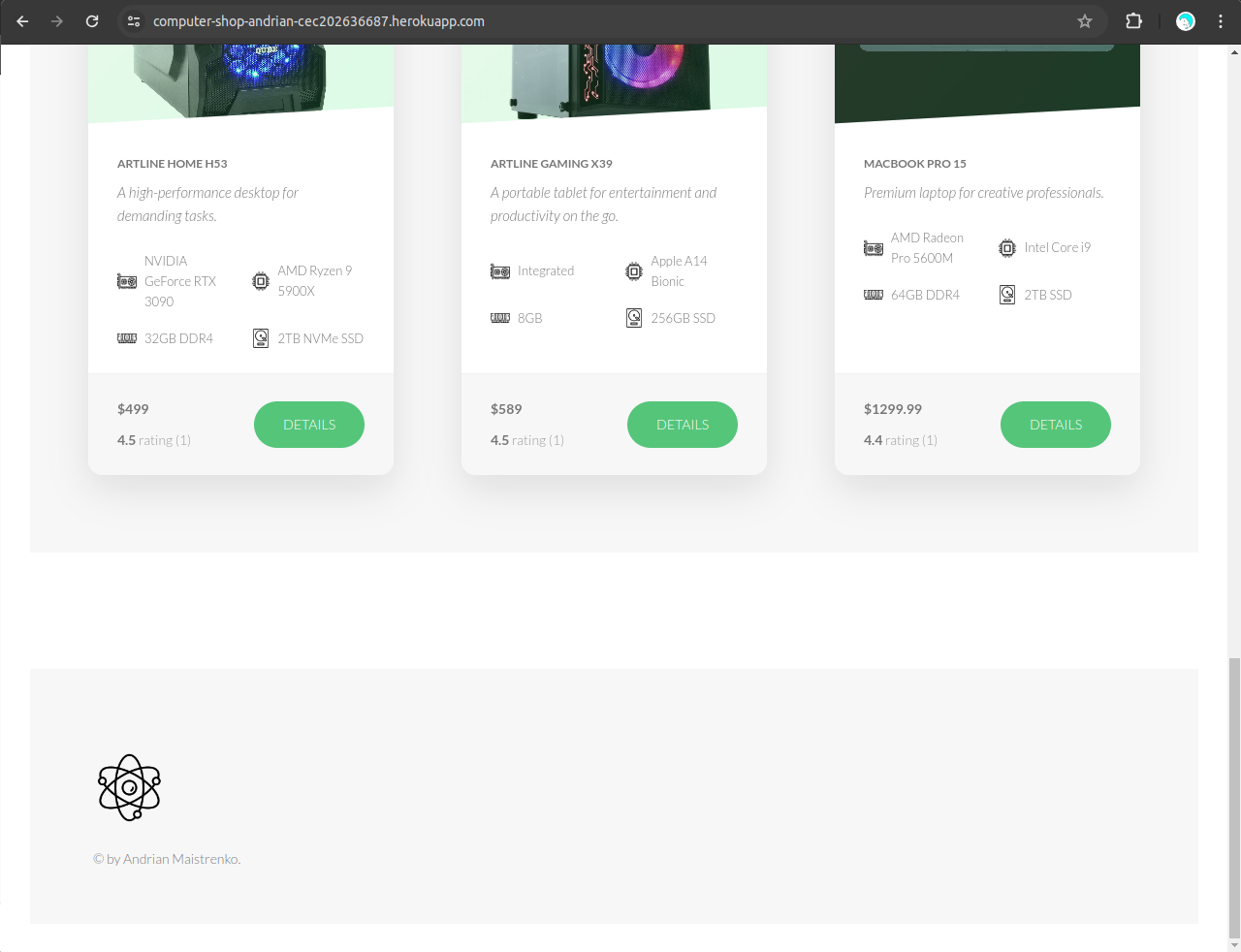
Реалізація даної сторінки відбувалася простіше та менш цікаво,адже все що нам потрібно зробити - це знайти відгуки із моделі Reviews,де id користувача буде співпадати. Далі відсилаємо як результат новостворений шаблон pug із назвою reviews, куди як параметри передалися знайдені відгуки.

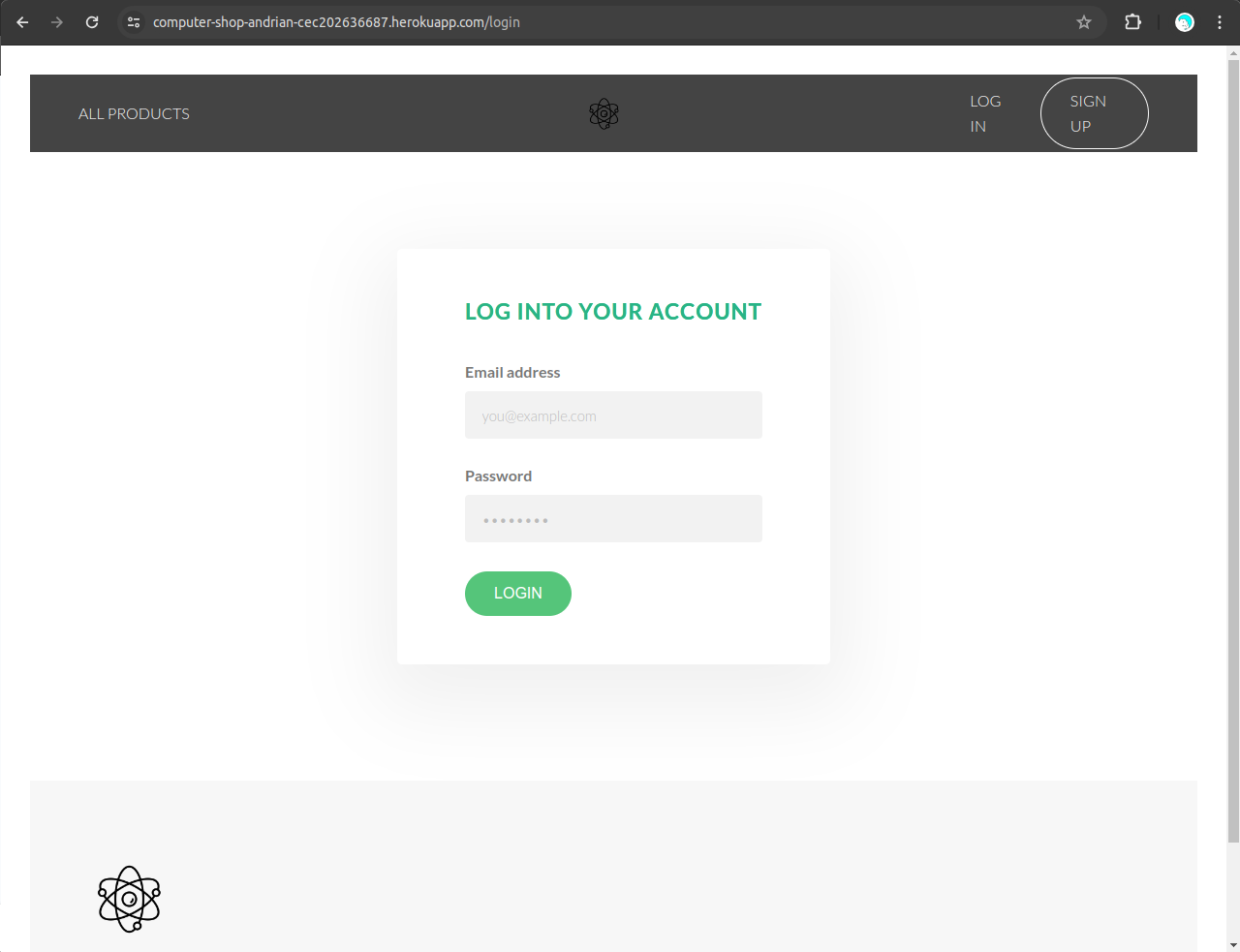
**Розділ 3. Демонстрація результату**

**Cторінки складаються із трьох частин:**

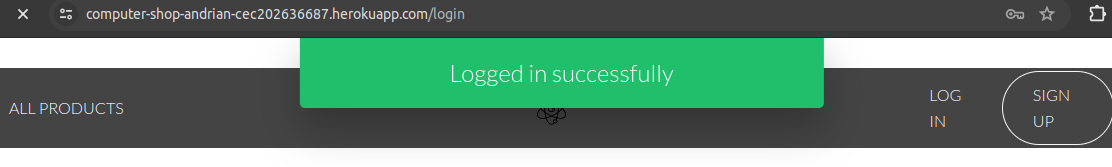
1. header - частина,котра містить у собі основні кнопки керування та логотип,перехід до основної сторінки з каталогом товарів,авторизація та логування
2. content - наповнення каталогу
3. footer - міститься логотип та авторське право

****

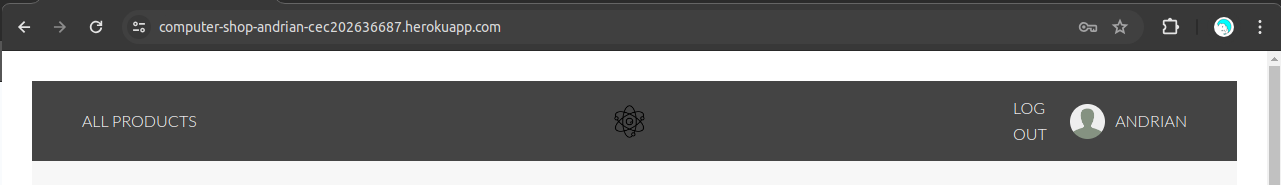


Сторінка логування

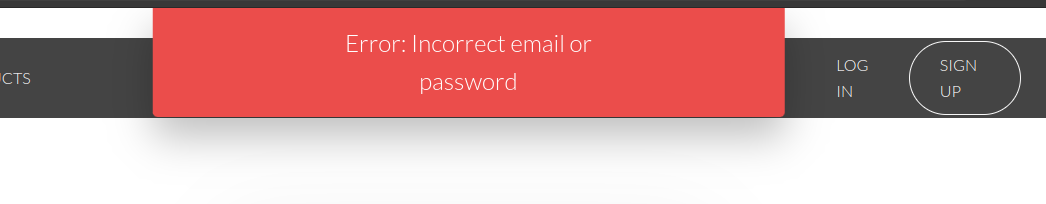
При успішному вході ми побачимо інформування про це



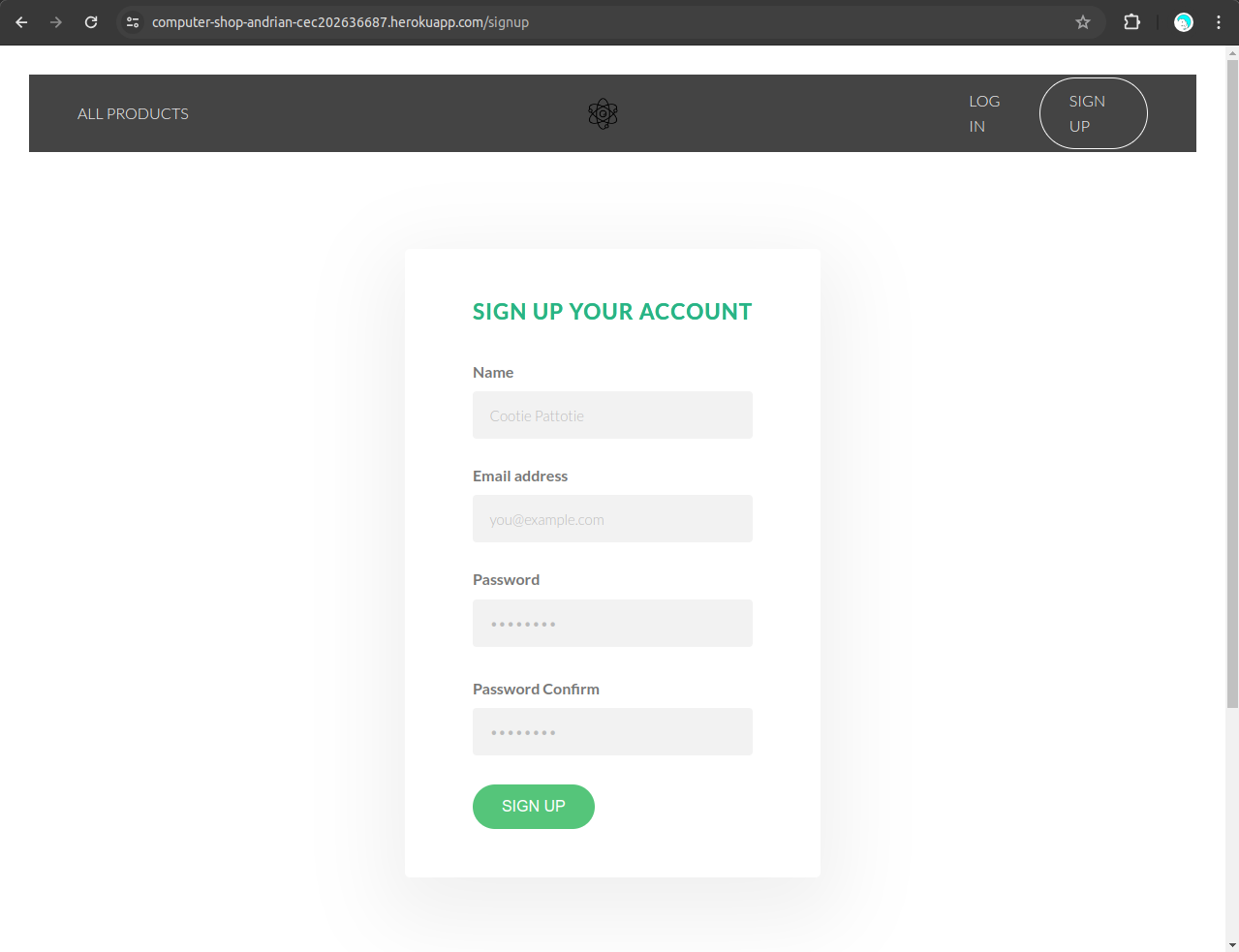
Відповідно наш header зміниться із входом відповідно до нашого ім’я та фотографії



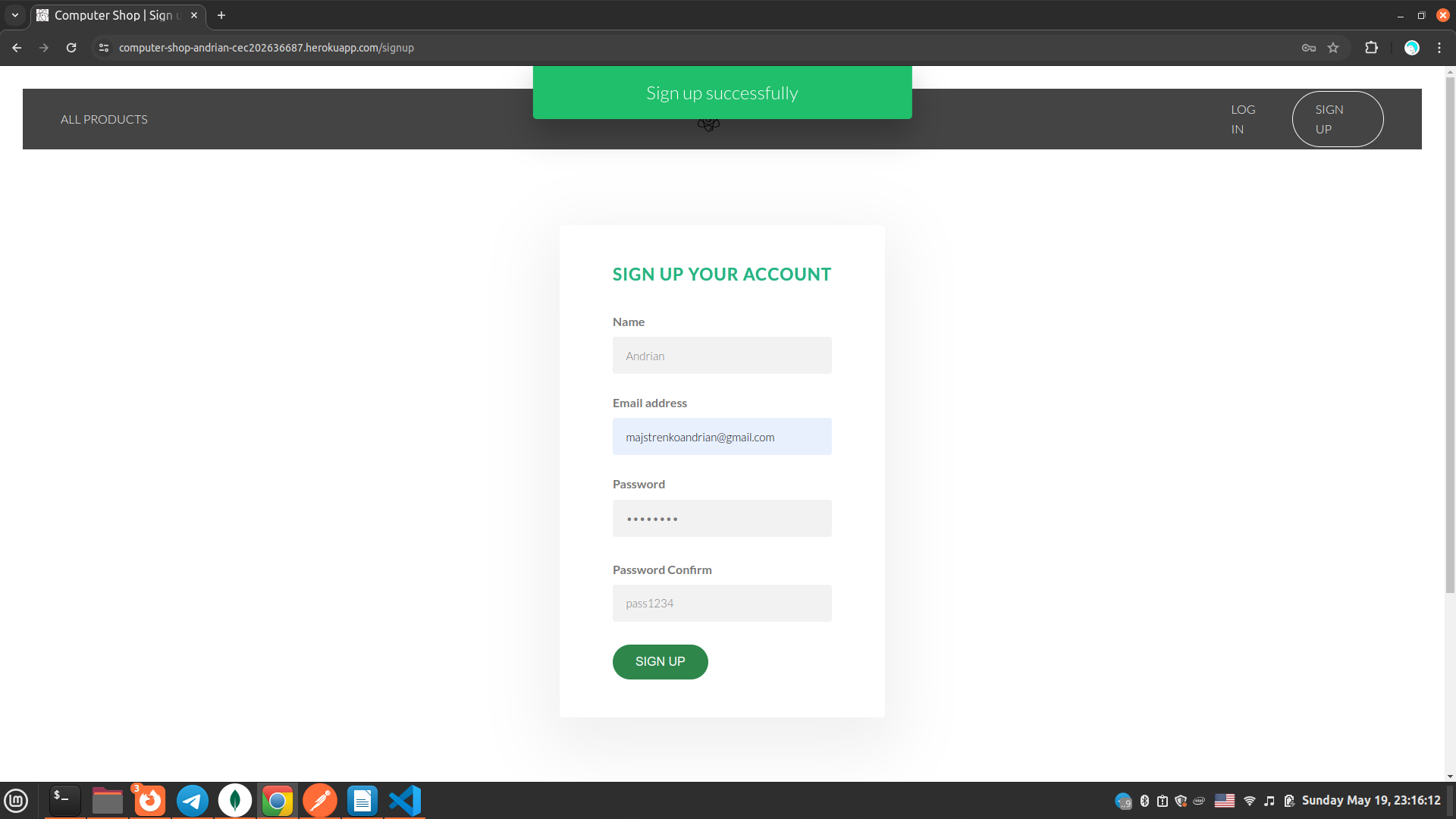
Якщо ж з певних причин ми не можемо успішно зайти,ми отримуємо повідомлення

****

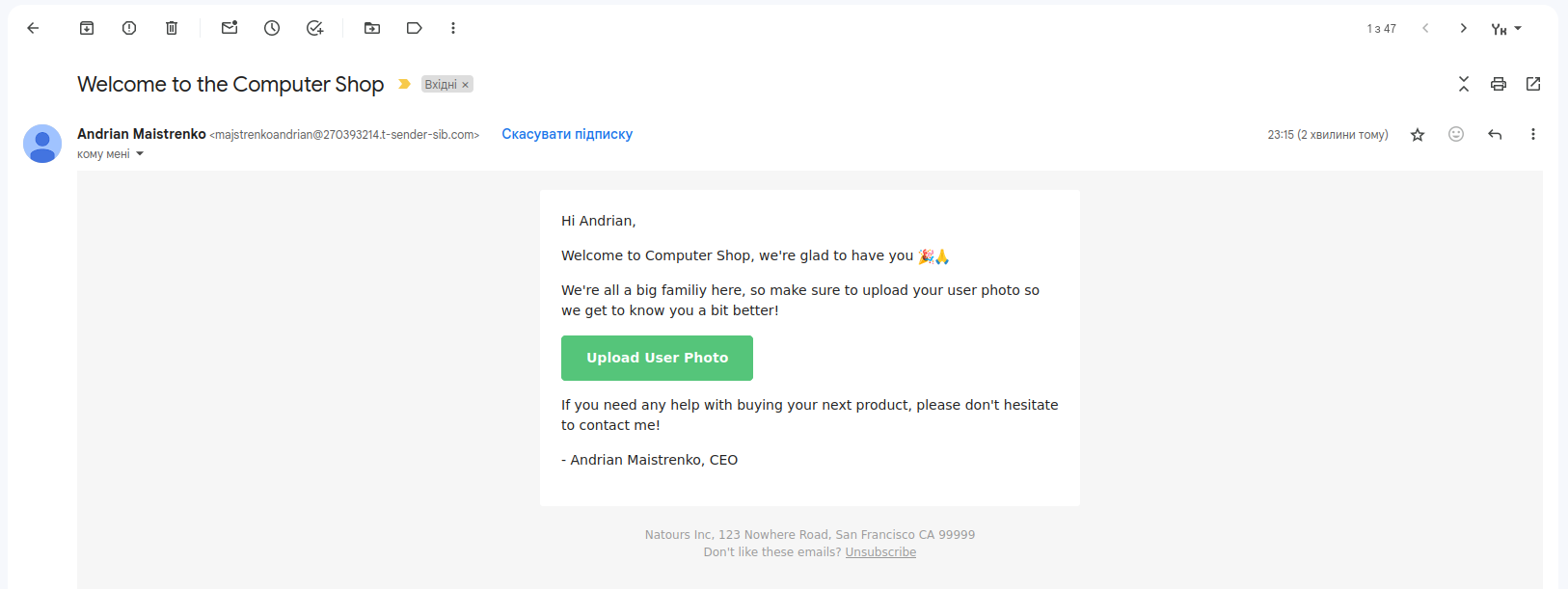
Сторінка реєстрації

****

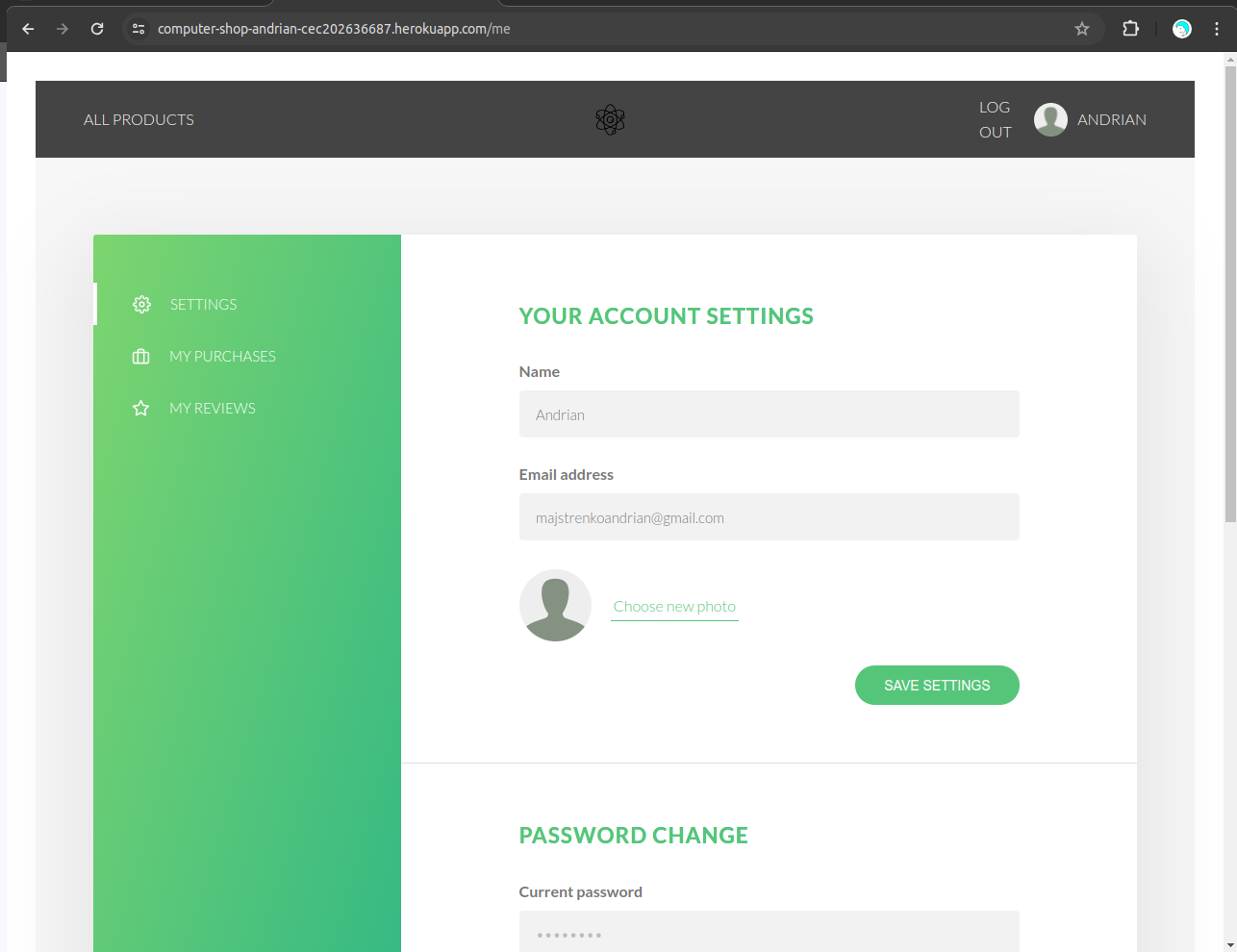
При успішній реєстрації ми отримаємо таке повідомлення

****

Також після реєстрації нам надійде електронний лист із привітанням і запрошенням додати фотографію для власного профілю

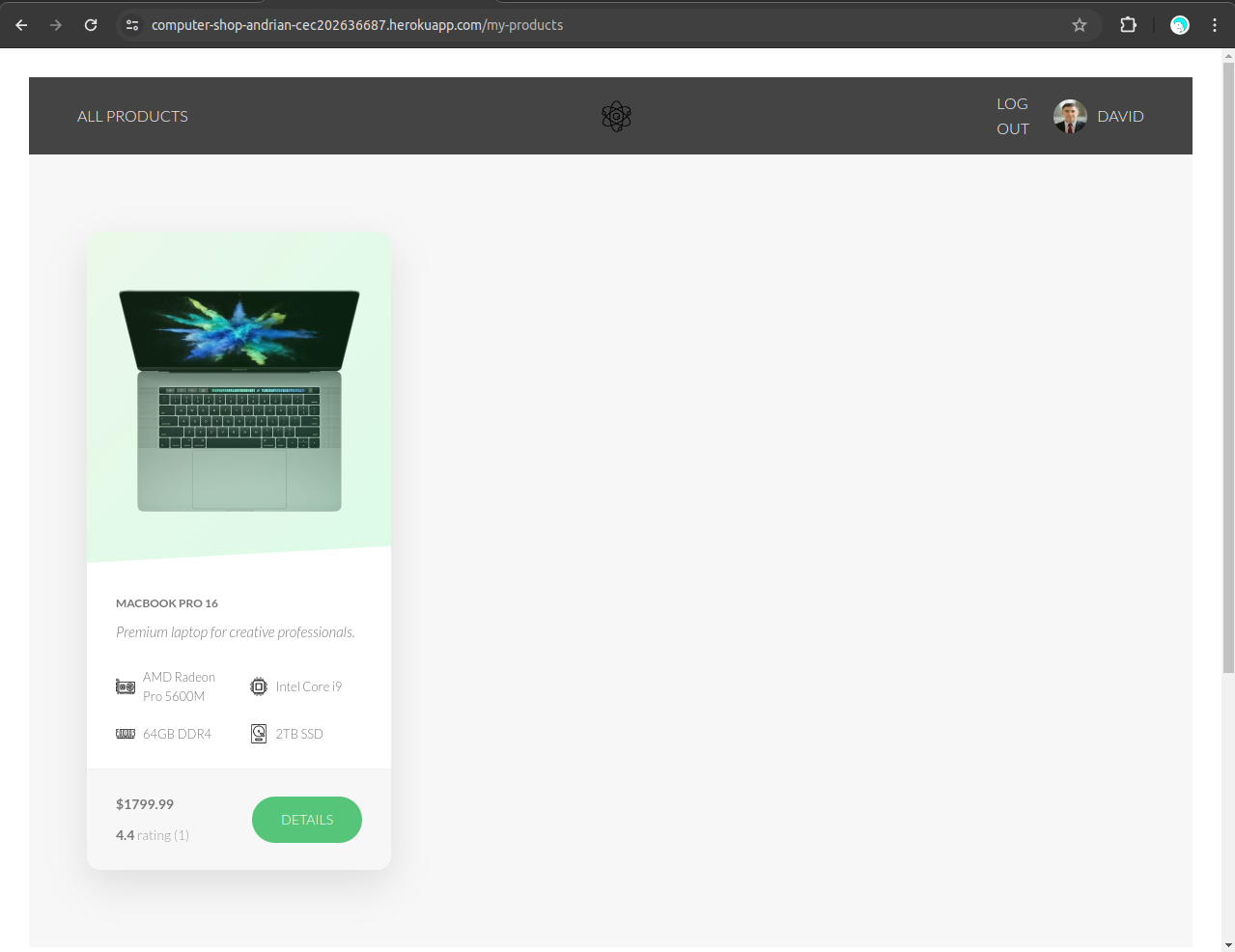
****

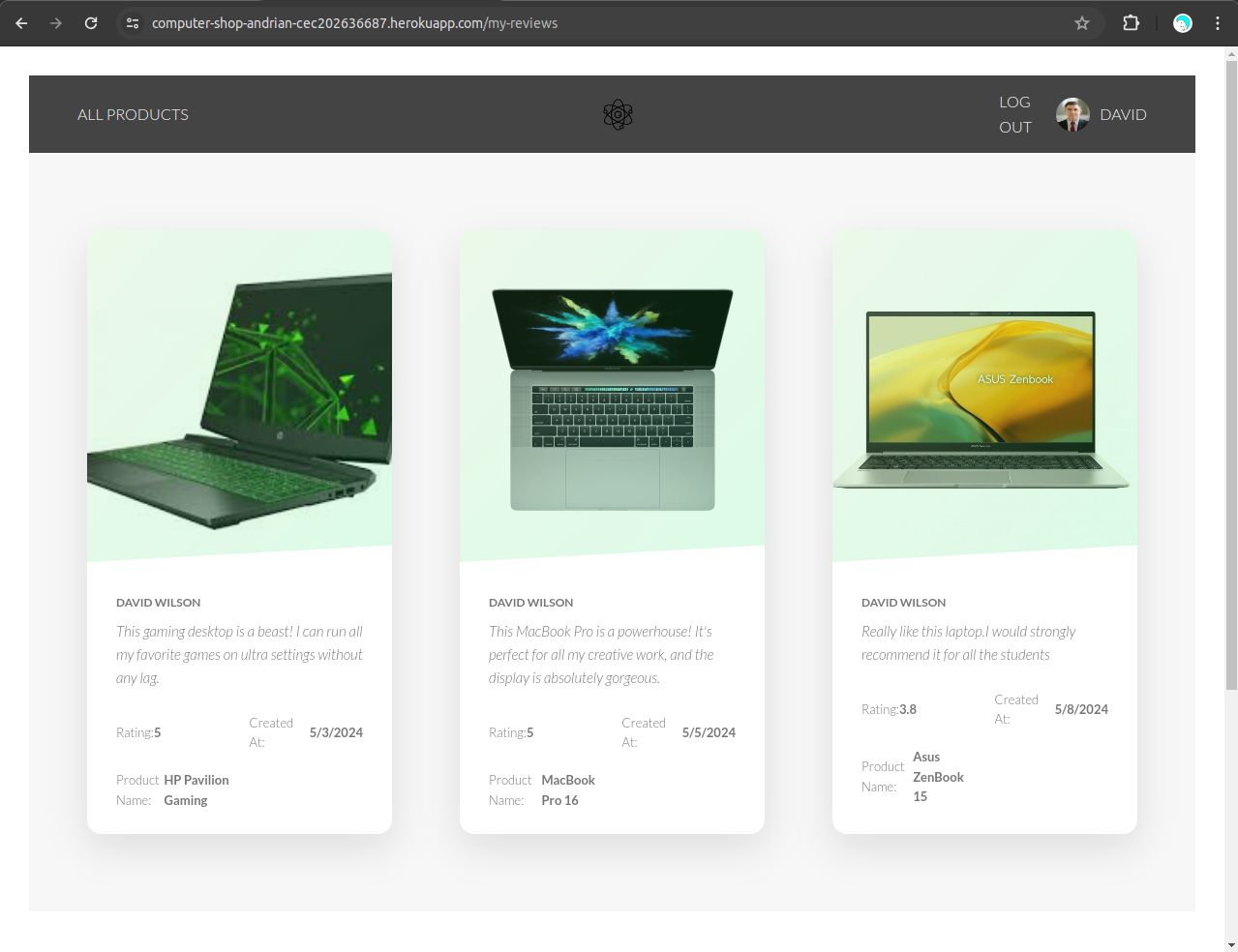
При натисненні на дану кнопку нас перенаправить на сторінку нашого профілю

****

На даній сторінці ми можемо змінити наші дані:ім’я,електронну адресу, фотографію та змінити наш пароль.

Також є можливість переглядати наші куплені продукти та відгуки,написані нами, відповідно натиснувши на My Purchases та My Reviews

****

****

**Використані технології та пакети**

Дані пакети та технології кратно спрощують процес розробки та покращує функціонал сайту,дозволяючи впроваджувати різноманітні корисні функції.Внизу наданий список та коротке пояснення ролі використаних пакетів.

1. frisbee: Бібліотека для здійснення HTTP-запитів на основі обіцянок (promises).
2. bcryptjs: Бібліотека для шифрування паролів у форматі bcrypt.
3. compression: Middleware для стискання вихідних HTTP-відповідей на сервері.
4. cookie-parser: Middleware для обробки cookies в запитах HTTP.
5. dotenv: Пакет для завантаження змінних середовища з файлу .env в проекті.
6. express: Фреймворк для створення веб-серверів та API на Node.js.
7. express-mongo-sanitize: Middleware для захисту від вторгнень через MongoDB запити.
8. express-rate-limit: Middleware для обмеження частоти HTTP-запитів на сервер.
9. helmet: Middleware для захисту від різноманітних атак на сервер, таких як XSS та інші.
10. hpp: Middleware для запобігання HTTP параметрів polution атак.
11. html-to-text: Бібліотека для перетворення HTML у текст.
12. jsonwebtoken: Бібліотека для створення та перевірки JWT токенів для аутентифікації.
13. leaflet: Бібліотека для роботи з інтерактивними картах у веб-додатках.
14. mongoose: Бібліотека для зручної роботи з MongoDB у Node.js додатках.
15. morgan: Middleware для реєстрації HTTP запитів у форматі логів.
16. multer: Middleware для обробки файлових завантажень у веб-додатках.
17. ndb: CLI інструмент для відладки Node.js додатків.
18. nodemailer: Бібліотека для відправки електронних листів з Node.js.
19. nodemailer-brevo-transport: Транспорт для відправки електронних листів через сервіс Brevo.
20. nodemailer-sendgrid: Транспорт для відправки електронних листів через сервіс SendGrid.
21. pug: Шаблонний двіжок для генерації HTML у веб-додатках.
22. sharp: Бібліотека для обробки та маніпулювання зображеннями.
23. slugify: Бібліотека для перетворення тексту у "slug" формат (URL-friendly).
24. stripe: Бібліотека для інтеграції платіжної системи Stripe у веб-додатки.
25. validator: Бібліотека для валідації та фільтрації різних типів даних.
26. xss-clean: Middleware для фільтрації вхідних даних для запобігання XSS атак.
27. Ці технології та пакети допомагають забезпечити різноманітні функціональність, безпеку та ефективність вашого веб-додатка.

**Висновок**

Створення веб-сайтів — це захоплюючий та багатогранний процес, що вимагає комбінації технічних знань, креативності та розуміння потреб користувачів. Кожен етап розробки, від проєктування до впровадження функціоналу та дизайну, є викликом, який мотивує до постійного самовдосконалення та пошуку інноваційних рішень. У процесі роботи над сайтом, я освоїв нові технології та підходи, що дозволило покращити якість та ефективність кінцевого продукту.

Крім того, створення веб-сайтів відіграє важливу роль у сучасному бізнес-середовищі. Інтернет став основним каналом для комунікації та торгівлі, тому наявність привабливого та функціонального веб-ресурсу є критично важливою для будь-якої компанії. Завдяки розробленому сайту ми змогли забезпечити нашим користувачам зручний доступ до інформації та можливість здійснювати покупки в зручний для них спосіб.

Підсумовуючи, я успішно створив та вдосконалив веб-сайт, додаючи нові функції та покращуючи його візуальний вигляд. Кожна сторінка отримала логічну структуру, стильний дизайн та необхідну функціональність, що дозволяє користувачам легко знаходити потрібну інформацію та замовляти товари. Результатом став функціональний, інтерактивний та естетично привабливий веб-сайт, який повністю задовольняє потреби наших користувачів.

**Список використаних джерел**

1.Сайт для вивчення html та css <https://w3schoolsua.github.io/html/index.html#google_vignette>

2.Книга Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics

3.

4.Курс з вивчення Node.js,Express,MongoDb,pug та інших технологій

<https://ua.udemy.com/course/nodejs-express-mongodb-bootcamp/>

5.Реалізований сайт  
<https://computer-shop-andrian-cec202636687.herokuapp.com/>