**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**ЛАБОРАТОРНА РОБОТА № 3**

Тема:” Каркас (структура) проєкту”

Виконав студент І курсу

спеціальності «Інженерія

програмного забезпечення»

Пілгович Андрій Ярославович

**Ужгород-2025**

**Мета роботи:**

Ознайомитися з принципами розробки вебзастосунку на основі компонентної архітектури та реалізувати проєкт **EuroCars** з використанням сучасного фронтенд-стеку.

**Завдання до роботи:**

1. Детальний опис ідеї з відповідними ілюстраціями (необхідно, щоб зрозуміти структуру папок/файлів)
2. Представлення структури проєкту, директорій...
3. Обгрунтувати обране рішення (по архітектурі)
4. Представити стек проєкту.
5. Детальне обгрунтування у випадку відхилення від запропонованої на заняттях архітектури.
6. Опублікувати всі зміни на GitHub
7. Створити PullRequest (нас додати як переглядачів (reviewers))
8. ВИСНОВКИ

Хід роботи

### **1. Детальний опис ідеї проєкту**

Проєкт **EuroCars** — це вебзастосунок, створений для демонстрації автомобілів європейського виробництва. Метою проєкту є представлення каталогу автомобілів із можливістю перегляду інформації про кожну модель, її характеристики, зображення та ціну.

Ідея полягає в тому, щоб створити зручний і сучасний інтерфейс, який дає змогу користувачу легко переглядати різні бренди та моделі, отримувати детальні відомості та візуальне уявлення про автомобіль.

У застосунку реалізовано компонентну архітектуру, що спрощує підтримку та масштабування проєкту.

### **2. Структура проєкту**

eurocars-project/

index.html

vite.config.js

README.md

.gitignore

package.json

package-lock.json

eslint.config.js

/node\_modules

/src

/components

/assets

/pages

/styles

**Опис основних елементів:**

* **index.html** — головний HTML-файл, у який вставляється React-додаток.
* **vite.config.js** — конфігураційний файл Vite, що визначає параметри збирання проєкту.
* **package.json** — містить інформацію про залежності, скрипти запуску та метадані проєкту.
* **src/** — головна директорія з вихідним кодом застосунку.
* **components/** — окремі React-компоненти (наприклад, карточки авто, навігаційна панель тощо).
* **pages/** — сторінки додатку (наприклад, головна, про компанію, контакти).
* **assets/** — зображення автомобілів та інші медіа-файли.
* **styles/** — CSS або файли з використанням Tailwind для стилізації елементів.

### **3. Обґрунтування архітектурного рішення**

Для реалізації проєкту обрано **React** із використанням **Vite** як бандлера, що забезпечує високу швидкість розробки, миттєве оновлення (HMR) і легке налаштування.

Архітектура компонентна — кожна частина інтерфейсу винесена в окремий компонент, що спрощує:

* повторне використання коду;
* незалежне оновлення частин застосунку;
* швидке тестування та налагодження.

Така архітектура дає змогу у майбутньому легко додавати нові функціональні можливості (наприклад, фільтри за брендами або ціною).

### **4. Використаний стек технологій**

**Основний стек:**

* **HTML5, CSS3, JavaScript (ES6)**
* **React.js** — бібліотека для побудови інтерфейсу користувача
* **Vite** — інструмент для швидкої збірки застосунку
* **Node.js та npm** — для управління залежностями
* **ESLint** — для перевірки якості коду
* **TailwindCSS або звичайний CSS** (для оформлення інтерфейсу)

### **5. Відхилення від базової архітектури**

Проєкт частково відхиляється від базової архітектури, запропонованої на заняттях, у таких аспектах:

* Використано **Vite** замість Webpack, щоб прискорити процес розробки.
* Можливо, використано додаткові бібліотеки для візуальних ефектів чи анімацій (за потреби).
* Структура компонентів організована у власний спосіб для зручності підтримки.

Такі відхилення обґрунтовані тим, що Vite забезпечує меншу складність конфігурації та швидше оновлення інтерфейсу під час розробки.

Висновок

У ході виконання лабораторної роботи було розроблено каркас вебзастосунку **EuroCars**, який реалізовано з використанням компонентної архітектури на базі **React** та інструмента збірки **Vite**. У процесі роботи було створено логічну структуру проєкту, налагоджено основні конфігураційні файли, визначено залежності та організовано директорії для компонентів, сторінок і стилів.

Отримано практичні навички зі створення структури сучасного фронтенд-проєкту, роботи з інструментами керування залежностями (**npm**) і системами контролю версій (**Git/GitHub**). Проєкт продемонстрував розуміння принципів модульності, масштабованості та гнучкості вебзастосунків.