**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**ЛАБОРАТОРНА РОБОТА № 4**

Тема:” Реалізація фронтенд частини ”

Виконав студент І курсу

спеціальності «Інженерія

програмного забезпечення»

Пілгович Андрій Ярославович

**Ужгород-2025**

**Мета роботи:**

Набуття практичних навичок у розробці front-end частини веб-додатку з використанням бібліотеки React.

**Завдання до роботи:**

1. Студент повинен реалізувати необхідний мінімум сторінок (pages/views) обраного ним проєкту. Як було описано в попередніх завданнях - щонайменше 3 сторінки + авторизація (без бізнес логіки)  
2. Реалізувати всі компоненти необхідні для відтворення сторінок (без бізнес логіки)

3. Додати стилі (використовувати модульний **css** підхід або **styled-components**)

4. Створити пул-реквест

5. Оформити відповідного взірця [звіт](https://moodle.uzhnu.edu.ua/pluginfile.php/85713/mod_assign/intro/report_template.docx)

Хід роботи

1.Опис структури проєкту

Проєкт створено з використанням сучасного фронтенд-інструменту **Vite** у поєднанні з **React**. Така конфігурація забезпечує швидку збірку, високу продуктивність і зручність розробки завдяки автоматичному оновленню сторінок (Hot Module Replacement).

1.1.Загальна структура проєкту

Изображение выглядит как текст, снимок экрана, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.

#### 1.2.Компонентний підхід

Проєкт ефективно використовує компонентну декомпозицію, що видно зі структури підтек:

* **assets/**: Директорія для зберігання статичних ресурсів, що імпортуються у компоненти (зображення, шрифти, іконки).
* **components/**: Ключова тека, що містить багаторазові UI-компоненти, згруповані за функціональним призначенням (сторінками або сутностями):
  + navbar/: Компонент навігаційного меню.
  + home/, about-as/, contacts/, servics/ (ймовірно, services): Компоненти для відповідних сторінок сайту ("Головна", "Про нас", "Контакти", "Послуги").
  + catalog/: Компоненти, пов'язані зі сторінкою каталогу.
* **catalog-cars/**: Окрема тека, що, ймовірно, містить більш складні компоненти або бізнес-логіку, специфічну для функціоналу каталогу автомобілів.
* **registration/**: Містить компоненти, пов'язані з формою та логікою реєстрації користувача.
* **componetsCatalog/** (ймовірно, componentsCatalog): Ще одна група компонентів, що уточнює функціонал каталогу:
  + sidebar/: Компонент бічної панелі (ймовірно, для фільтрації).
  + unfavorite/: Компоненти для управління списком "Обраного" (додавання/видалення).
  + home\_page/: Можливо, специфічний компонент для відображення каталогу на головній сторінці.

Вся логіка застосунку побудована за **компонентною архітектурою**, що є основним принципом роботи з React. Це означає, що кожен елемент інтерфейсу розроблений як **окремий незалежний компонент**, який виконує свою чітку функцію.

Такий підхід має кілька переваг:

* **Модульність** — код поділений на невеликі частини, які легко тестувати, змінювати або перевикористовувати в інших частинах проєкту.
* **Зрозумілість структури** — кожен компонент відповідає лише за одну логічну частину інтерфейсу, що спрощує навігацію в коді.
* **Гнучкість і масштабованість** — у майбутньому можна без труднощів додавати нові сторінки або функції без порушення роботи всієї системи.
* **Ізольована стилізація** — кожен компонент має власний CSS, тому зміни стилів не впливають на інші частини застосунку.

## **2. Реалізація функціоналу у коді**

### **2.1. Маршрутизація сторінок**

У файлі **App.jsx** реалізовано систему маршрутизації за допомогою бібліотеки **react-router-dom**.  
Це дозволяє реалізувати **SPA (Single Page Application)** — тобто переходи між сторінками відбуваються без повного перезавантаження сайту.

Така структура дозволяє користувачеві зручно перемикатися між головною сторінкою, каталогом автомобілів та формою реєстрації.

### **2.2. Головна сторінка (HomePage.jsx)**

Компонент **HomePage** є головною частиною сайту, що відображається при переході на кореневий маршрут /.  
Він поєднує кілька внутрішніх компонентів:

1. **Слайдер автомобілів** — реалізовано з використанням хуків useState та useEffect.  
   Автоматичне перемикання зображень відбувається через встановлений інтервал часу.  
   Додано можливість ручного перемикання кнопками «вперед» / «назад».
2. **Секція послуг (Services)** — містить картки з основними послугами компанії (наприклад: оренда, продаж, обслуговування).  
   Для візуального оформлення використано бібліотеку **lucide-react** (іконки).
3. **Блок “Про нас” (AboutAs)** — текстовий блок з інформацією про компанію, перевагами та статистикою.
4. **Контактна секція (Contacts)** — включає футер із контактами, соціальними мережами та юридичною інформацією.

Таким чином, HomePage.jsx є **контейнером**, який збирає ключові частини головної сторінки в єдину структуру.

### **2.3. Каталог автомобілів (Catalog.jsx)**

Компонент **Catalog** відповідає за відображення списку автомобілів.  
У його основі — дочірні компоненти:

* Catalogs — візуалізація карток авто;
* Sidebar — бічна панель фільтрації або сортування;
* Unfavorite — функціонал для управління обраними авто.

Кожна картка автомобіля містить:

* зображення;
* назву марки та моделі;
* рік випуску, тип палива, пробіг;
* кнопку “обране”.

Верстка базується на **Flexbox**, для анімацій застосовано ефекти **hover** (зміна тіні, масштабу).  
Кожна картка реагує на дії користувача, підсвічується при наведенні миші.

### **2.4. Реєстрація користувача (Registration.jsx)**

Компонент **Registration** реалізує просту форму реєстрації / авторизації користувача.  
Форма містить:

* поле логіну (username або email);
* поле паролю;
* кнопку **“Увійти”**.

На даному етапі логіка роботи з сервером не реалізована.

3.Фото сайту

3.1.Home

Изображение выглядит как текст, транспортное средство, Наземный транспорт, колесо

Содержимое, созданное искусственным интеллектом, может быть неверным.

3.2.Catalog

Изображение выглядит как текст, машина, снимок экрана, транспортное средство

Содержимое, созданное искусственным интеллектом, может быть неверным.

3.3.Registration – невстиг доробити

Висновок

Під час виконання лабораторної роботи № 4 я здобув практичні навички у розробці front-end частини веб-додатку з використанням бібліотеки React, що відповідає меті роботи.Відповідно до поставлених завдань, було реалізовано ключові сторінки проєкту:

1. **Головна сторінка (HomePage)**: Створено компонент, що об'єднує слайдер (реалізований з useState та useEffect), секції послуг, "Про нас" та контакти.
2. **Каталог (Catalog)**: Реалізовано сторінку з картками автомобілів, бічною панеллю (Sidebar) та функціоналом "обраного" (Unfavorite).
3. **Реєстрація (Registration)**: Створено базовий компонент форми, що містить поля логіну, паролю та кнопку входу, хоча бізнес-логіка не реалізовувалася.

Для забезпечення переходів між сторінками без перезавантаження (SPA) була інтегрована бібліотека react-router-dom. Уся структура проєкту побудована на компонентній архітектурі , що забезпечує модульність, гнучкість та ізольованість стилів. Хоча основний функціонал сторінок "Home" та "Catalog" було успішно реалізовано, компонент реєстрації не вдалося завершити в повному обсязі, зокрема його стилізацію ("невстиг доробити") Таким чином, завдання роботи в основному виконані. Я набув практичного досвіду у створенні структури React-проєкту за допомогою Vite, реалізації компонентів, налаштуванні маршрутизації та застосуванні стилів.