**МIНIСТЕРСТВО ОСВIТИ I НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ОСТРОЗЬКА АКАДЕМІЯ”**

**Кафедра економіко-математичного моделювання та інформаційних технологій**

**ЗВІТ**

**Лабораторна робота №5**

**Створення скелетону проекту з Redux і React Router**

здобувачки вищої освіти

першого (бакалаврського) рівня

третього року навчання групи КН-31

спеціальності 122 Комп’ютерні науки

ОПП «Комп’ютерні науки»

Лайтер Ярини Семенівни

(прізвище, ім’я, по батькові)

База лабораторної - кафедра економіко-математичного моделювання  
 та інформаційних технологій

Керівник лабораторної від університету

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

(науковий ступінь, учене звання керівника)

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

(Власне ім’я та ПРІЗВИЩЕ)

Члени комісії:

\_\_\_\_\_

Острог - 2023

# Зміст

[**Зміст 2**](#_4l44eegb3r0h)

[**Опис 3**](#_faicydgoow6b)

[**Виконання 4**](#_wcwxvulfirkf)

[**1. Створення нового проекту React: 4**](#_4of8cbhax5pk)

[**2. Створюємо структуру каталогу: 4**](#_9efp1tfos1ow)

[**3. Створюємо компонент "App.jsx"та два простих маршрути "/home" і "/about": в 5**](#_53pszeb4r9z)

[**4. Створюємо редуктори (reducers) та дії (actions): 7**](#_8sk5794b74go)

[**5. Підключаємо Redux до додатку за допомогою react-redux: 9**](#_cz3556qtx5nl)

[**6. Виводимо дані з Redux-стору та створюємо компоненти для сторінок "/home","/about" та “/todo”: 9**](#_8npxsyfws0rt)

[**7. Результати з стилями та перевірками: 12**](#_ce8ckmfhq3j2)

[**Висновок: 13**](#_adesr1f640wd)

# 

# 

# 

# 

# 

# 

# **Опис**

**Тема**: Створення скелетону проекту з Redux та React Router

**Мета**: Ознайомити студентів зі створенням додатків на React з використанням Redux та редукторів (reducers)

**Завдання**

1. Створіть новий проект React за допомогою Vite або будь-якого іншого способу, який вам подобається та встановіть необхідні залежності:
2. redux
3. react-redux
4. react-router-dom
5. Створіть структуру каталогів для вашого проекту. Наприклад:

src/ components/ reducers/ actions/ containers/ App.jsx main.jsx

1. Створіть основний компонент "App.jsx", який буде кореневим компонентом вашого додатку. В цьому компоненті ви повинні використовувати React Router для налаштування маршрутів. Створіть два простих маршрути, наприклад, "/home" і "/about".
2. Створіть редуктори (reducers) та дії (actions) для вашого Redux-стору. Наприклад, створіть редуктор для збереження стану додатку та дію для оновлення цього стану.
3. Підключіть Redux до вашого додатку за допомогою react-redux та налаштуйте його так, щоб компонент "App.jsx" мав доступ до Redux-стору.
4. В компоненті "App.jsx" виводьте дані з Redux-стору, наприклад, виведіть поточний стан додатку на сторінці.

Створіть компоненти для сторінок "/home" та "/about", які будуть відображати різні дані на кожній сторінці.

Впевніться, що при переході між маршрутами дані в Redux-сторі залишаються незмінними, і вони правильно оновлюються та виводяться на відповідних сторінках.

Переконайтеся, що ваш додаток працює на локальному сервері та може бути запущений та перевірений в браузері.

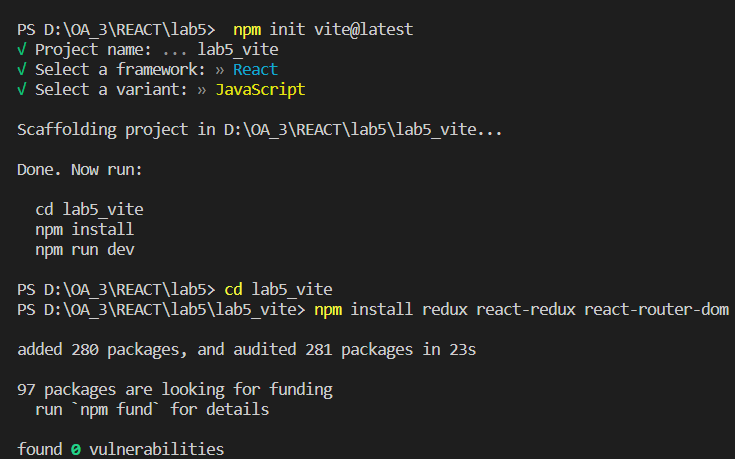
Додайте стилізацію до свого проекту за допомогою CSS або будь-якого іншого інструменту стилізації, який вам подобається.

Завершіть проект та переконайтеся, що він працює на вашому локальному сервері, та всі функції працюють коректно.

# **Виконання**

# **Створення нового проекту React:**

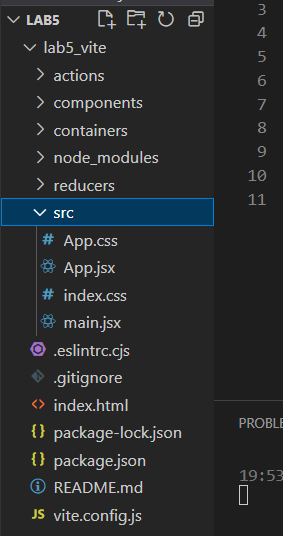
Створюємо проект. Вводимо в терміналі комаду npm init vite@latest та підключаємо залежності:

**

*Рис 1.1 Виконання команд*

# **Створюємо структуру каталогу**:

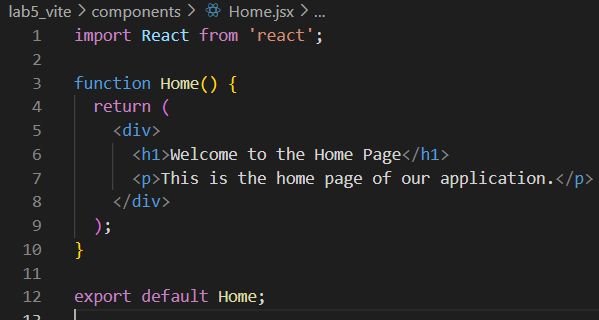
Створюємо нові папочки:



*Рис 2. 1 Створення структури проекту*

# **Створюємо компонент "App.jsx"та два простих маршрути "/home" і "/about": в**

Створюємо компоненти Home.jsx та About.jsx:



*Рис 3.1 Код сторінки Home.jsx*



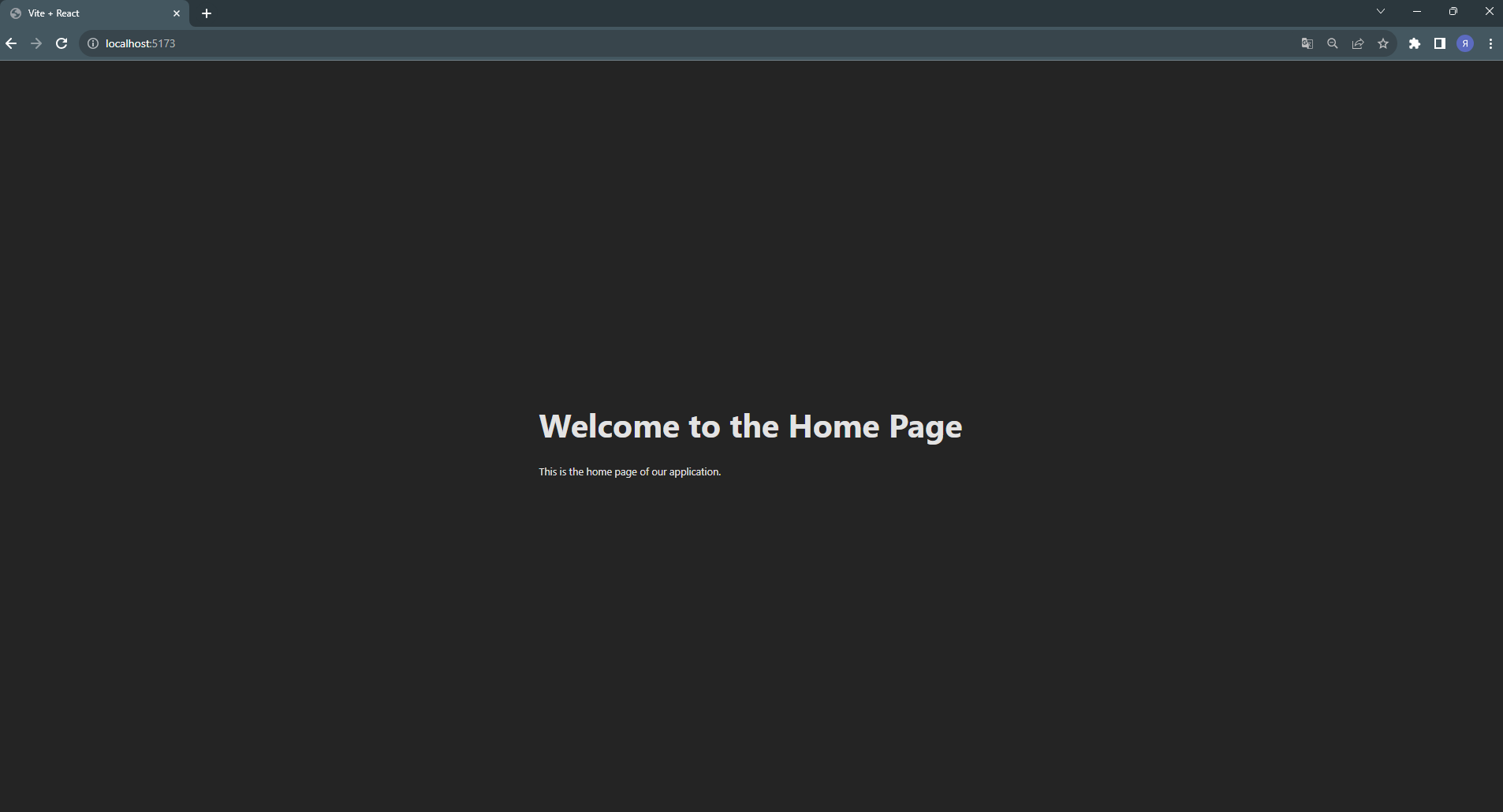
*Рис 3.2 Код сторінки About.jsx*

В компоненті App.jsx використовуємо React Router для налаштування маршрутів та створюємо два простих маршрути "/home" і "/about":

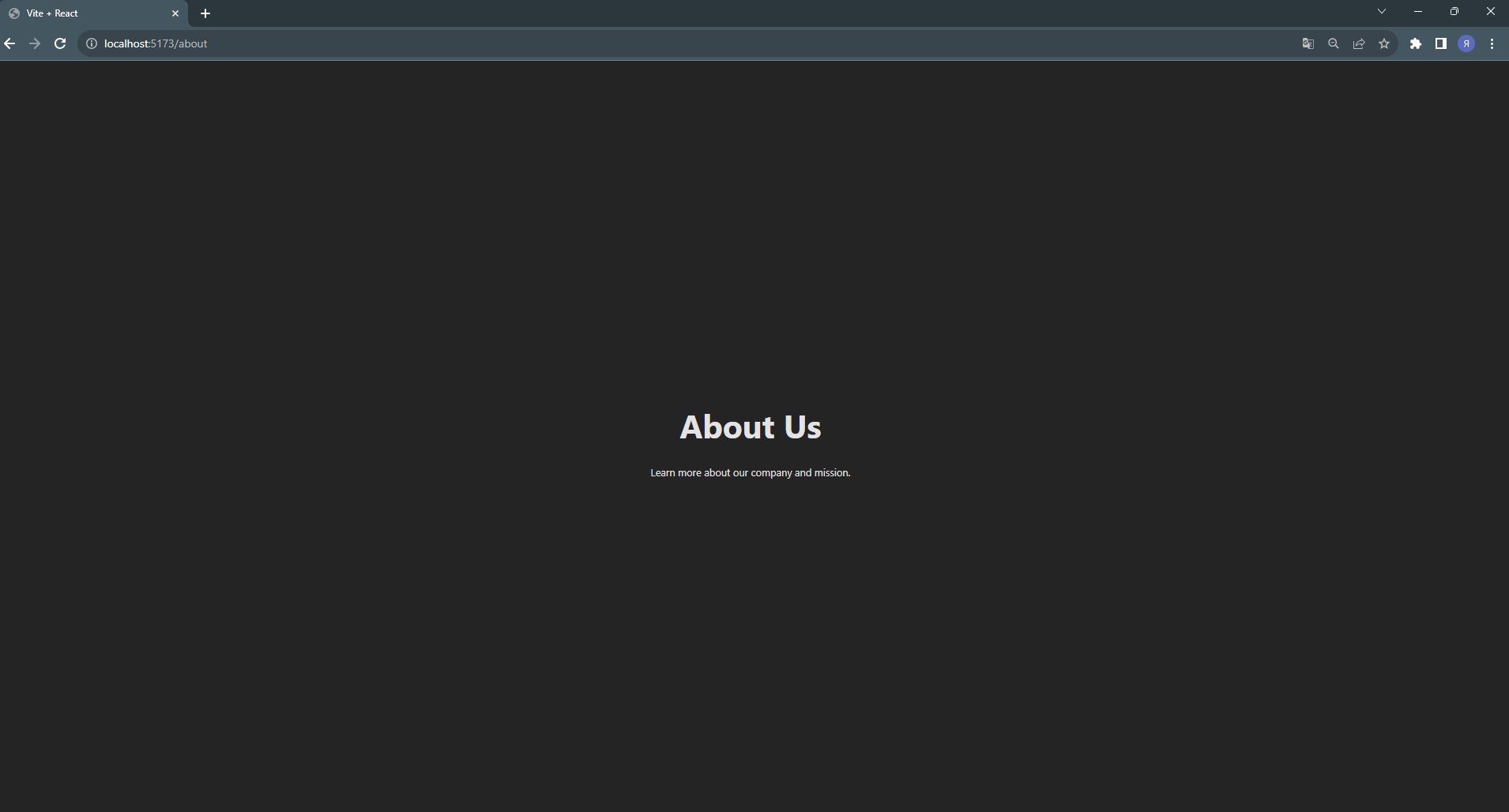


*Рис 3.3 Код сторінки App.jsx*

В результаті отримуємо:



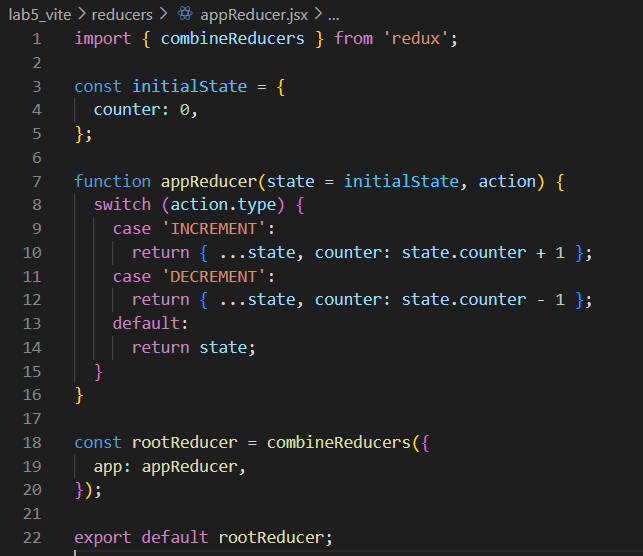
*Рис 3.4. Результат програми*



*Рис 3.5. Результат програми*

# **Створюємо редуктори (reducers) та дії (actions):**

Створюємо редуктор appReducer.jsx, який відповідає за лічильник counter. Коли відбувається дія INCREMENT або DECREMENT, appReducer змінює відповідну частину стану додатку:



*Рис 4.1 Код сторінки appReducer.jsx*

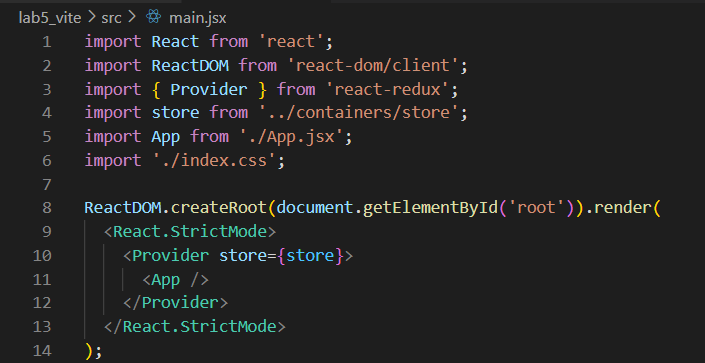
Та створюємо *appActions.jsx*, який відповідає дії, які використовуються в компонентах або діяльностях для відправки запитів до Redux-стору. Коли дія відправляється, відповідний редуктор обробляє цю дію і вносить відповідні зміни в стан додатку*:*



*Рис 4.2 Код сторінки appActions.jsx*

# **Підключаємо Redux до додатку за допомогою react-redux:**

Підключаємо відповідні бібліотеки у файлі main.jsx, щоб додаток функціонував:



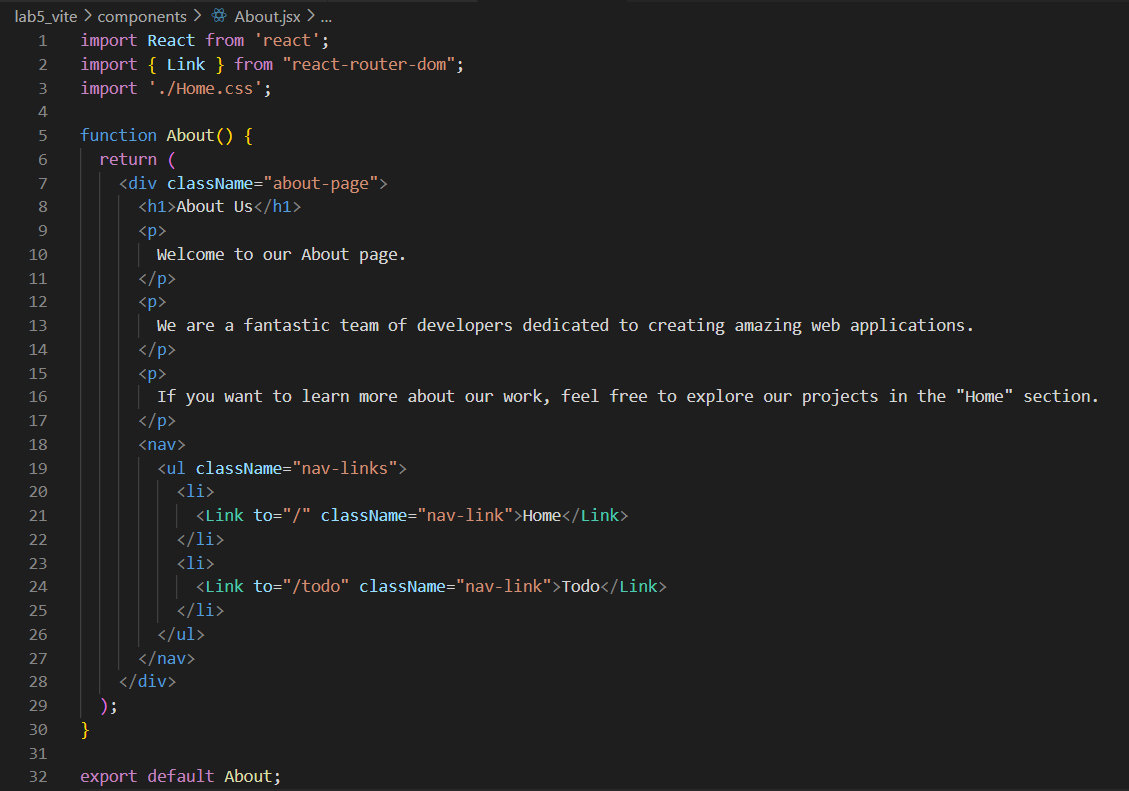
*Рис 5.1 Код компонента main.jsx*

# **Виводимо дані з Redux-стору та створюємо компоненти для сторінок "/home","/about" та “/todo”:**

Спочатку створимо компоненти з різними даними в них:



*Рис 6.1 Код компонента Home.jsx*



*Рис 6.2 Код компонента About.jsx*



*Рис 6.3 Код компонента Todo.jsx*

Далі викликаємо всі компоненти в *App.jsx*:



*Рис 6.4 Код компонента App.jsx*

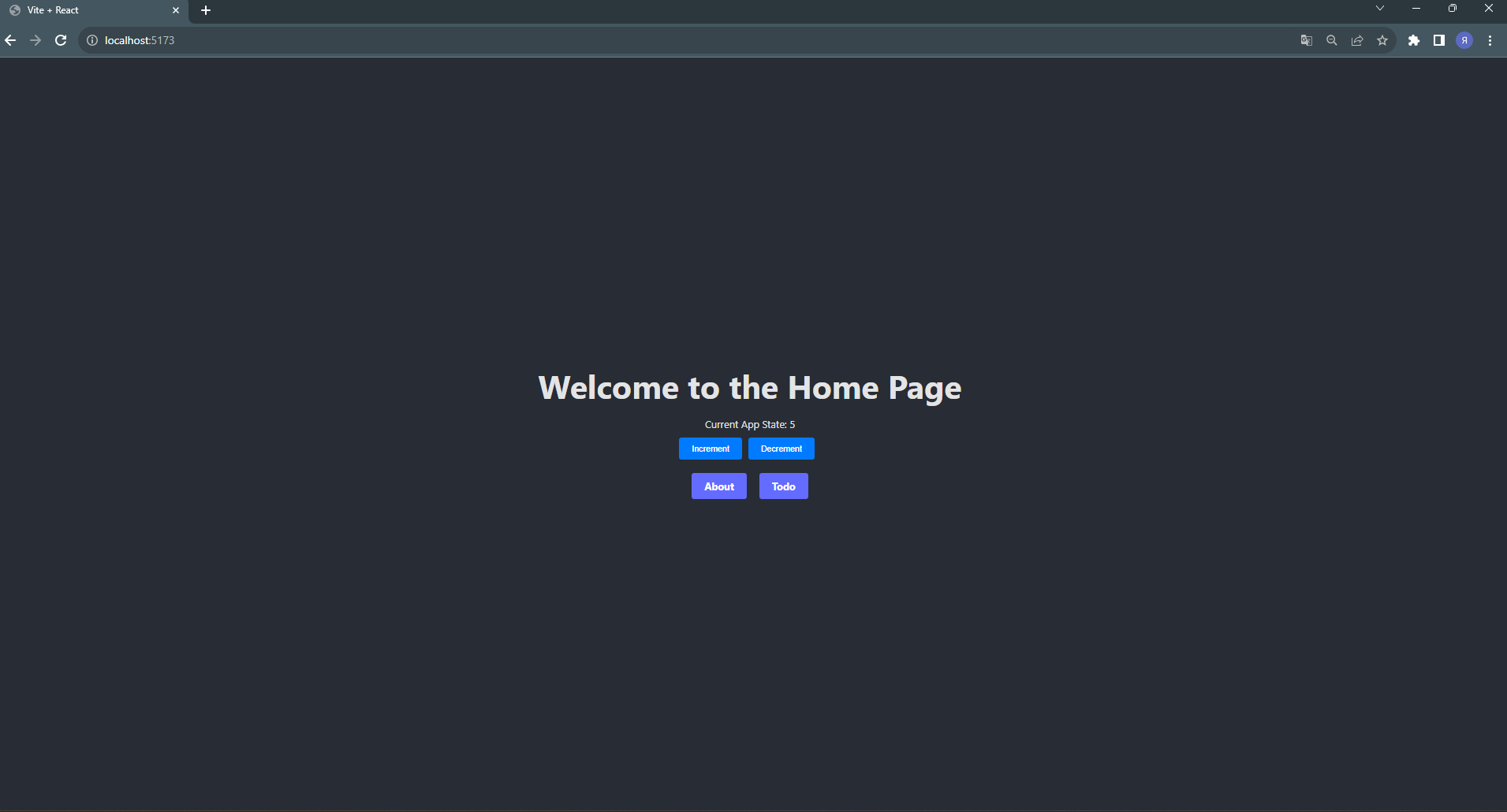
При переході між маршрутами дані в Redux-сторі залишаються незмінними, і вони правильно оновлюються та виводяться на відповідних сторінках (а зокрема на сторінці Home, About). На сторінці Todo дані не будуть зберігатися і при кожному вході на сторінку все буде “перезагружатися”\повертатися в початковий стан.

# **Результати з стилями та перевірками:**

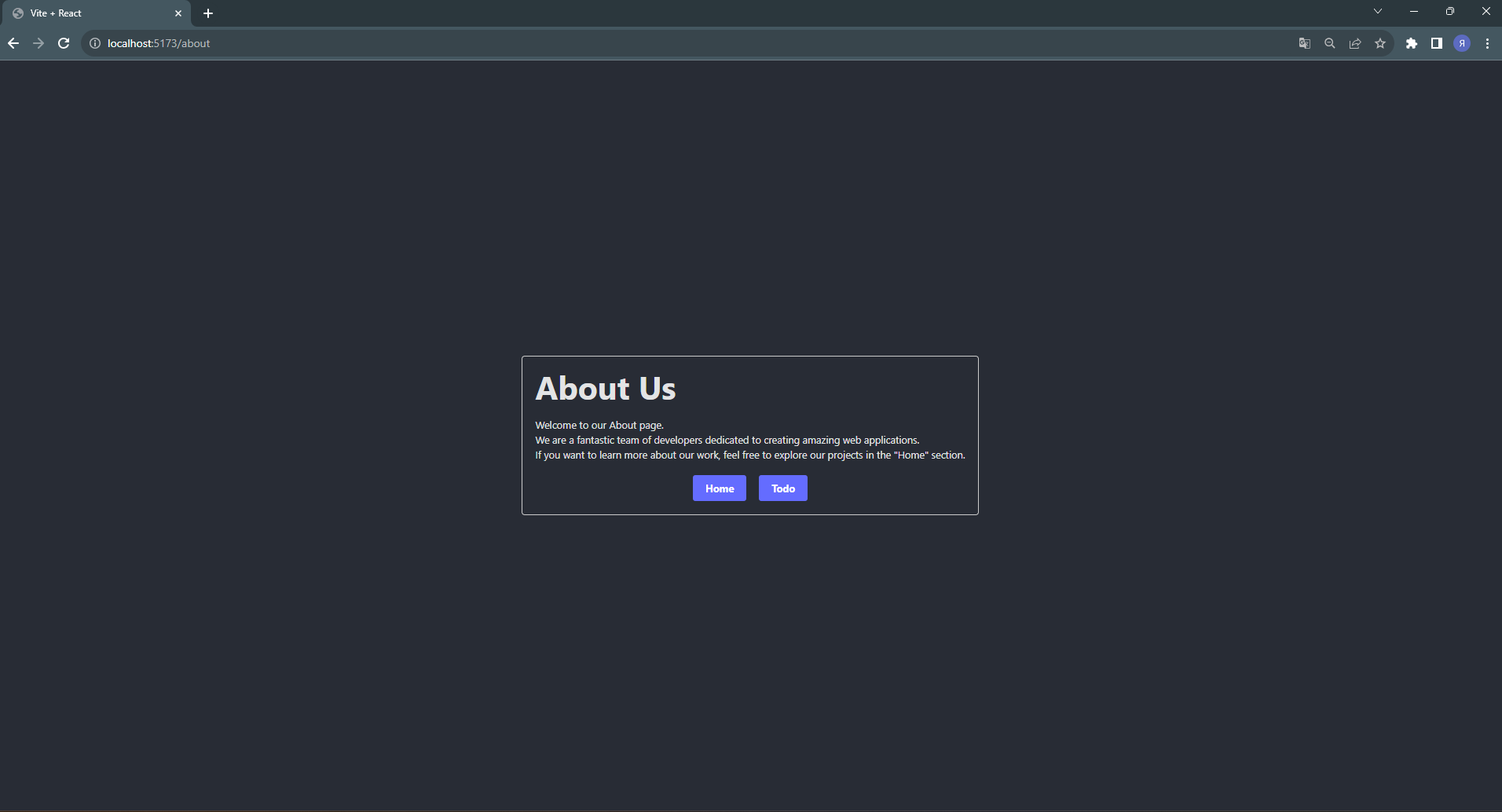
Додаток працює на локальному сервері та був запущений та перевірений в браузері.

Додано стилізацію до свого проекту за допомогою CSS.

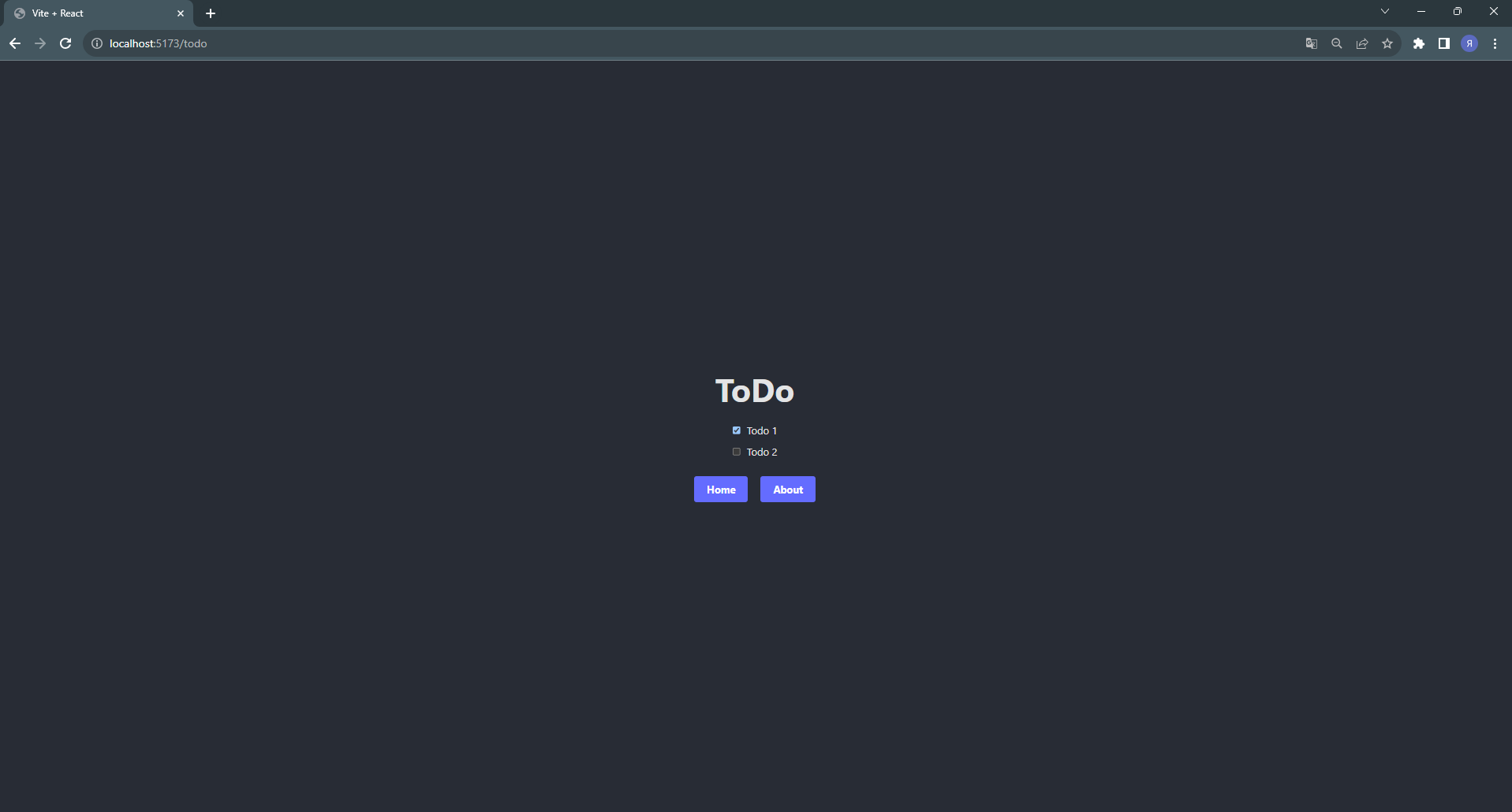
В результаті отримуємо:



*Рис 7.1 Результат запуску програми*



*Рис 7.2 Результат запуску програми*



*Рис 7.3 Результат запуску програми*

# 

# **Висновок:**

На цій лабораторній роботі ми створили проект React з використанням Vite, налаштували структуру каталогів, підключили Redux та React Router. Окрім того, створили компоненти для сторінок "/home", "/about" та "/todo", які відображають різні дані на кожній сторінці.

Ми переконалися, що дані в Redux-сторі залишаються незмінними при переході між маршрутами, і вони правильно оновлюються та виводяться на відповідних сторінках.

Також ми додали стилізація до проекту для поліпшення зовнішнього вигляду та завершено проект, переконавшись у його працездатності на локальному сервері.