

# Курсовий проєкт “Книжковий інтернет-магазин”

Протягом курсу ви будете вивчати різні теми, що в сукупності дозволять **створити повноцінний вебдодаток “Книжковий інтернет-магазин” та розмістити його на віддаленому сервері**. Це і буде вашим курсовим проєктом, який ви зможете продемонструвати на співбесіді. Необхідну інформацію та рекомендований план виконання проєкту ви знайдете нижче.

В завданні використовуються прототипи сторінок застосунку, на яких лише описується загальний вигляд сторінок (місця розташування елементів). Стили сторінок залишаються на розсуд студента, можна використовувати або написані вручну стилі, або використати для стильового оформлення фреймворк (наприклад Bootstrap), або комбінований варіант (частина стилів написані вручну, частина - за допомогою фреймворка).

Результат виконання проєкту має бути такий або кращий :)

**Приклад реалізації проєкту “Книжковий інтернет-магазин”, написаного на React JS слухачем попередньої ітерації курсу**

**Author:** Ruslan Bezuhlyi

**Link:** [https://prometheus-platform.github.io/Example\\_of\\_course\\_project\\_2](https://prometheus-platform.github.io/Example_of_course_project_2)

## Прототип сторінок

### Сторінка “Авторизації”

The mockup shows a web browser window with the title 'JS Band Store'. The address bar displays 'http://localhost:3030/#/signin'. Below the browser window, there is a header bar with the text 'JS BAND STORE / Your full name'. The main content area features a placeholder for a user profile picture, a label 'Username' above a text input field with the placeholder text 'type Username', and a 'Sign-In' button below the input field.

## Сторінка “Список книг”

JS Band Store

← → ↺

http://localhost:3030/#/books

JS BAND STORE / Your full name

Sign-Out

Username

Search by book name

Price

**Book name, Book name, Bo...**  
Book author

Price

View

**Book name, Book name, Bo...**  
Book author

Price

View

**Book name, Book name, Bo...**  
Book author

Price

View

## Сторінка “Окремої книги”

JS Band Store

← → ↺

http://localhost:3030/#/books/:id

JS BAND STORE / Your full name

Sign-Out

Username

**Book name, Book name, Book name..**  
Book author  
Book level  
Book tags, book tags,..

Book description

Price, \$17

Count3

Total price51

Add to cart


## Сторінка “Кошик (порожній)”

JS Band Store

← → ↺

http://localhost:3030/#/cart

JS BAND STORE / Your full name



Sign-Out

Username

Purchase



Cart empty..


## Сторінка “Кошик (з книгами)”

JS Band Store

← → ↺

http://localhost:3030/#/cart

JS BAND STORE / Your full name



Sign-Out

Username

Purchase

Book name	Count	Total price
Book name	Count	Total price
Book name	Count	Total price

Total price, \$ 103

## Файли, які знадобляться для виконання курсового проєкту:

1. Зображення аватара (може бути обране інше зображення на ваш розсуд) - [avatar.png](#)
2. Зображення кошика (може бути обране інше зображення на ваш розсуд) - [cart.svg](#)
3. Зображення по замовчуванню при відсутності зображення для книги (може бути обране інше зображення на ваш розсуд) - [imageNotFound.png](#)

## Рекомендований план виконання проєкту

### Модуль 4 (HTML Advanced):

I. Створіть директорию з іменем "X-course-task". Весь проєкт буде розміщений в цій директорії.

II. Зверстайте сторінку "Авторизації". Ваше завдання полягає в створенні базової HTML-розмітки без стилів. Зовнішній вигляд (вирівнювання, кольори, ...) цієї сторінки надамо пізніше за допомогою стилів CSS.

1. У директорії "X-course-task" створіть директорию з іменем "signin"
2. У директорії "signin" створіть файл "index.html"
3. У файлі "index.html" додайте розмітку відповідно до зображення (зображення, поле для вводу та кнопку);
4. Примітки:
  - 4.1. Файл прототипа сторінки "Авторизації" - [signin\\_screen.png](#).
  - 4.2. Файл сторінки "Авторизації" без стилів - [signin-without-styles.png](#)

III. Зверстайте сторінку "Окремої книги". Ваше завдання полягає в створенні базової HTML-розмітки без стилів. Зовнішній вигляд цієї сторінки надамо пізніше за допомогою стилів CSS

1. У директорії "X-course-task" створіть директорию з іменем "specific-book"
2. У директорії "specific-book" створіть файл "index.html"
3. У файлі "index.html" додайте семантичну розмітку відповідно до прототипу (зображення, інформація про книгу, поля інформації про вартість книги та загальної вартості, поле для вводу тільки чисел та кнопку);
4. Примітки:
  - 4.1. Файл прототипа сторінки "Окремої книги" - [specific\\_book\\_screen.png](#).
  - 4.2. Файл сторінки "Окремої книги" без стилів - [specific-book-without-styles.png](#)

### Модуль 5 (CSS Basics):

Використовуючи створену базову HTML-розмітку оформіть сторінку "Окремої книги" відповідно до прототипу за допомогою CSS (стилі можуть бути написані вручну, або використовуючи фреймворк, або комбінованим варіантом). Стильове оформлення (кольори, ефекти, шрифти, ...) оберіть самостійно.

1. В директорії "specific-book" створіть файл "styles.css" та підключіть його до файлу "index.html"
2. Пропишіть css стилі так, щоб сторінка виглядала максимально подібною до прототипу.
3. Примітки:
  - 3.1. Файл прототипа сторінки "Окремої книги" - [specific\\_book\\_screen.png](#).

## **Модуль 6 (CSS Advanced):**

Використовуючи створену базову HTML-розмітку оформіть сторінку "Авторизації" з використанням адаптивного дизайну за допомогою CSS (стилі можуть бути написані вручну, або використовуючи фреймворк, або комбінованим варіантом). Стильове оформлення (кольори, ефекти, шрифти, ...) оберіть самостійно.

1. В директорії "signin" створіть файл "styles.css" та підключіть його до файлу "index.html"
2. Пропишіть css стилі так, щоб сторінка виглядала максимально подібною до прототипу (зображення висотою - 160px та шириною - 160px, поле для вводу та кнопка шириною - 200px).
3. Пропишіть css стилі так, щоб при ширині екрану "320px":
  - 3.1 Ширина кнопки та поля для вводу розтягувалася на весь екран з відступами по бокам рівними 8px.
  - 3.2 Розміри зображення збільшувалися: ширина і висота до 200px.
4. Примітки:
  - 4.1. Файл прототипа сторінки "Авторизації" - [signin\\_screen.png](#).

## **Модуль 8 (JS Basics):**

Використовуючи створену розмітку зі стилями для сторінки "Окрема книга" (specific-book) додайте інтерактивності за допомогою JavaScript так, щоб при введенні вручну в поле вводу та натисканні на кнопки збільшення/зменшення загальна вартість позиції змінювалася (збільшувалася/зменшувалася).

1. В директорії "specific-book" створіть файл "main.js" та підключіть його до файлу "index.html"
2. Використовуючи JavaScript оживіть поле для вводу так, щоб при збільшенні значення кількості книжок загальна вартість також збільшувалася. Також при зменшенні кількості книжок - загальна вартість зменшувалася.
3. Передбачити випадки, щоб користувач не міг ввести значення 0 (нуль) чи від'ємне значення' та не більше 42 (42 - значення, яке ще дозволяється обрати користувачу (поле "amount" в books.json)).
4. Функціонал збільшення/зменшення повинен працювати, як через натискання кнопок (збільшити/зменшити) так і через введення з клавіатури.

## **Модуль 9 (JS Advanced):**

Використовуючи HTML та CSS створіть сторінку "Список книг" (book-list) та за допомогою JavaScript виконайте завантаження та відображення списку книг з файлу "books.json".

1. Створіть директорію з іменем "book-list"
  2. Створіть файли "index.html", "styles.css" та "main.js"
  3. Підключіть файли "styles.css" та "main.js" до файлу "index.html"
  4. Використовуючи JavaScript отримайте інформацію з файлу "books.json" та використайте її для побудови каталогу товарів.
  5. Пропишіть css стилі так, щоб сторінка виглядала максимально подібною до прототипу.
- Стильове оформлення (кольори, ефекти, шрифти, ...) оберіть самостійно.
6. Примітки:
    - 6.1. Файл прототипа сторінки "Списку книг" - [list\\_screen.png](#).
    - 6.2. Файл зі списком книг - [books.json](#).
    - 6.3. Кожний елемент файлу "books.json" має такі поля:
      - "id" - унікальний ідентифікатор (тип - число)

- "author" - автор книги (тип - рядок)
- "price" - ціна книги (тип - число)
- "image" - посилання на зображення книги (тип - рядок)
- "title" - назва книги (тип - рядок)
- "shortDescription" - короткий опис вмісту книги (тип - рядок)
- "description" - розгорнутий опис вмісту книги (тип - рядок)

## Модуль 12 (React Basics):

Використовуючи "create-react-app" створіть React-застосунок та перенесіть в React-компоненти створені сторінки "Авторизації"(signin), "Окремої книги"(specific-book) та "Список книг" (book-list) та створіть новий React-компонент для сторінки "Кошик"(cart).

1. За допомогою "create-react-app" створіть застосунок з іменем "x-course-task" в директорії "X-course-task".
2. Створіть React-компоненти для сторінок "signin", "specific-book" та "book-list" (перенесіть HTML та CSS без JavaScript).
3. Створіть React-компонент для сторінки "Кошик" (порожній та з бажаними до придбання книгами)
4. Створіть React-компонент для футера, який повинен повторюватися на кожній сторінці, з текстом "Виконано в Prometheus © 2022" (текст "Prometheus" повинен бути посиланням на сайт "<https://prometheus.org.ua/>", який відкривається на новій вкладці) розташованим по центру сторінки.
5. Створіть React-компонент для хедера, який повинен повторюватися на кожній сторінці, з текстом "X-course task / Прізвище Ім'я".
6. Створіть React-компонент для сторінки "404" (Сторінки може містити довільний контент, головне, щоб на ній був текст, який інформує користувача, що це сторінки 404. Наприклад: "Oops, something went wrong. 404 error").
7. Примітки:
  - 7.1. Файл прототипа сторінки "Порожній кошик" - [purchase complete screen.png](#).
  - 7.2. Файл прототипа сторінки "Кошик з бажаними до придбання книжками" - [cart screen.png](#).

## Модуль 13 (React Advanced):

Використовуючи написану поведінку на JavaScript, реалізовану в розділах 8 та 9, переписати її за допомогою React-hooks.

Використовуючи "Reactrouter v6" реалізувати переміщення по сторінкам застосунку.

Для сторінки "Авторизація" реалізувати поведінку при якій при введенні допустимої кількості символів для імені користувача та натисненні на кнопку "Sign-In" перенаправляло на сторінку "Список книг" після чого виконувалося завантаження та відображення книжок, а при кліку на кнопку "Sign-out" перенаправляло на сторінку "Авторизація".

Для сторінки "Список книжок" реалізувати поведінку фільтрації відображених книжок по назві книги та ціні книги, а також при натисненні на кнопку "View", для конкретної книги, перенаправляло на сторінку "Окрема книга" з наступним відображенням всієї інформації про дану книгу.

Для сторінки "Окрема книга" реалізувати додавання книги до кошика.

Для сторінки "Кошика" реалізувати відображення обраних книжок та при кліку на кнопку "Purchase" список обраних книг повинен видалятися.

1. Використовуючи Хуки оживіть компоненти, щоб їхня поведінка була, як в розділах 8 та 9.
2. Встановіть бібліотеку "Reactrouter v6"
3. Додайте наступну поведінку:

### 3.1. Сторінка "SignIn".

3.1.1. При введенні символів менше 4 або більше 16 кнопка "Sign-In" повинна бути задієйблена.

3.1.2. Якщо введено від 4 до 16 символів користувач може клікнути на кнопку "Sign-In" і після цього буде перенаправлений на каталог з книгами. "Username" повинно бути збережене в локальний стейт для використання на інших сторінках.

3.1.3. Якщо користувач не авторизований, то він не може відвідати інші сторінки, крім "SignIn". Наприклад, при намаганні через URL перейти на сторінку "Каталог книг" перенаправляється на сторінку "SignIn".

### 3.2. Сторінки "Каталог книг", "Окрема книга" та "Кошик".

3.2.1. Можна потрапити на ці сторінки тільки, якщо користувач зареєстрований.

3.2.2. Інформація про ім'я користувача (Username) повинно братися з локального стейта (п. 3.1.2).

3.2.3. При кліку на іконку "Кошик", користувач повинен бути переміщений до сторінки "Кошик".

### 3.3. Сторінка "Каталог книг".

3.3.1. Додайте поле для вводу тексту, яке буде фільтрувати книжки в каталозі по назві книг, при введенні.

3.3.2. Додайте випадаюче меню з обиранням вартості книжок. Повинно бути 4 варіанти (Всі,  $0 < \text{ціна} < 15$ ,  $15 < \text{ціна} < 30$ ,  $\text{ціна} > 30$ ).

3.3.3. При кліку на кнопку "View" користувач перенаправляється на сторінку окремої книжки за допомогою бібліотеки "Reactrouter".

3.3.4. Якщо довжина назви книги більша за 24-и символи, то все, що йде після 24-го символу, замінюється трикрапкою ("...").

3.3.5. При відсутності зображення для книги застосувати зображення по замовчуванню.

### 3.4. Кнопка "Sign-Out".

3.4.1. При кліку на кнопку перенаправляється на сторінку "SignIn".

3.4.2. Видаляється інформація про Username з локального стейта.

### 3.5. Сторінка "Окремої книги".

3.5.1. При кліку на кнопку "Add to cart" обрана кількість книжок повинна бути додана до кошика.

3.5.2. Додані книжки до кошика повинні також бути додані до локального стейта, щоб при повторному вході на сторінку книжки показувалися кількість, яка вже додана до кошика, якщо такі є.

3.5.3. При відсутності зображення для книги застосувати зображення по замовчуванню.

### 3.6. Сторінка "Кошик".

3.6.1. Повинна містити інформацію про всі обрані книжки, вартість за одиницю, загальну вартість позиції та загальну вартість всіх книжок у кошику (відповідно до прототипу).

3.6.2. Якщо в кошику відсутні обрані до покупки книжки, кнопка "Purchase" повинна бути заблокована.

3.6.3. Якщо в кошику є одна або більше позицій, кнопка "Purchase" повинна бути дотупна.

3.6.4. При кліку на кнопку "Purchase" повинен очиститися кошик та локальний стейт з інформацією про обрані книжки.

### 3.7 Сторінка "404".

3.7.1. Сторінка "404" повинна з'являтися, коли користувач намагається відвідати сторінку застосунку, яка не існує (Наприклад, ви створили сторінку, в якій URL `"/books"`, а користувач намагається відвідати сторінку `"/boocs"` або `"/bookc"`, в такому випадку ми повинні відобразити сторінку "404").

## Модуль 14 (Context API):

Використовуючи можливості "React-Context" збережіть в ньому всі завантажені книжки та обрані книжки, щоб використовувати ці дані у вашому застосунку.

1. Створіть Context для всіх книжок (завантажених через JSON файл) та обраних книжок (доданих до кошика).

2. Перепишіть функціонал, щоб сторінки "Каталог книг" та "Окремої книги" використовували Context для потрібної інформації про книжки.

## Модуль 15 (Testing):

Використовуючи "Jest" напишіть тести, які будуть перевіряти коректну поведінку функціоналу збільшення та зменшення кількості книжок на сторінці "Окрема книга".

1. Покрийте тестами сторінку "Окремої книги".

2. Повинна бути покрита наступна поведінка:

2.1. При кліку збільшення кількості - кількість повинна збільшуватися.

2.2. При кліку зменшення кількості - кількість повинна зменшуватися.

2.3. При зміні кількості - загальна вартість повинна змінюватися.

## Модуль 16 (Deployment):

Використовуючи власний GitHub акаунт та вивчені скрипти, розгорніть застосунок на "GitHub Pages" з іменем проєкту "prometheus-x-course-task".

1. Розгорніть свій проєкт на "GitHub Pages".