REPORT

GitHub Repository: https://github.com/Aishascreations/novel-store-project

Name: Aishat Fadekemi Gbemisola Onakoya

Student ID: 22040102216

Signature:

Project Overview:

The Novel Store Project is a full-stack e-commerce platform developed to bridge the gap between modern frontend aesthetics and robust backend logic. Built using Spring Boot, React, and PostgreSQL, the application allows users to interact with a library of 72 unique novel records. My goal was to create a clean, minimalist storefront where the user experience is driven by data-heavy backend services, mimicking a real-world production environment.

Research, Learning Effort, and Challenges:

My research focused heavily on asynchronous state management—specifically, how to use Axios to fetch data from a Spring Boot API without blocking the React user interface. I spent a considerable amount of time learning the Material UI (MUI) ecosystem. Moving away from standard CSS to a component-based library required understanding "Theming" and "Grid systems" to ensure the site looked professional on all screen sizes. Additionally, managing a database of 72 records required learning how to write efficient SQL export scripts to ensure my data remained "self-contained" for instant portability.

Distinctive Features:

Data-Rich Environment: I populated this system with 72 diverse novel records. This tested the efficiency of my React mapping and the backend’s ability to serve large JSON payloads smoothly.

Strict Material UI Compliance: To meet professional standards, every interactive element (from the "Add to Cart" buttons to the navigation bar) is a specialized MUI component. This ensures a unified "Material Design" language across the entire app.

Modular "SourceCode" Architecture: I followed a strict separation of concerns. By isolating the frontend, backend, and database into a singular SourceCode folder, I ensured that any developer can easily navigate and launch the components independently.

Technical Specifications Table

JS Framework	React.js (v18) using Hooks (useState, useEffect) for dynamic data rendering.
Backend Service	Spring Boot with Maven; REST controllers handle GET/POST requests for novel data.
Component Library	Material UI (MUI); utilized for Grid systems, Buttons, and Layout containers.
Database System	PostgreSQL; structured via schema.sql and populated via data.sql (72 entries).
Folder Hierarchy	Root project contains a SourceCode folder with three distinct sub-folders.
API Integration	Axios library handles the connection between the React UI and the Java API.
Run Command	Frontend is executable via npm start as per the specific project requirements.

TECHNOLOGIES USED:

Frontend: React.js, Material UI (MUI), Axios, React Router.

Backend: Java, Spring Boot, Spring Data JPA, Hibernate, Maven.

Database: PostgreSQL (SQL Scripting).

Tools: IntelliJ IDEA, VS Code, Git/GitHub.

SUMMARY:

Building this project was a great journey into the world of full-stack development. It was more than just a coding exercise; it was about making a React frontend and a Spring Boot backend actually "talk" to each other reliably. I put a lot of focus into the organization of the SourceCode folder to make sure the project is professional and easy to navigate. By including a full dataset of 72 novels, I wanted to show that this isn't just a simple demo, but a solid foundation for a real-world e-commerce site that I am proud to present.