****

**BSc. (Hons) In Software Engineering**

**Faculty of Computing**

**Name of the Student: A.D.S. Nadeeshan**

**Student Registration Number:** **IT\_IFLS\_001/B003/0032**

**Module: Advanced Web Technology**

**Submission Date: 6th May 2025**

Table of Contents

[1 System Architecture Diagram 1](#_Toc197463780)

[1.1 Components 1](#_Toc197463781)

[1.2 Architecture Flow 1](#_Toc197463782)

[1.3 API Documentation 1](#_Toc197463783)

[1.4 Database Schema 2](#_Toc197463784)

[1.5 Setup and Deployment Guide 2](#_Toc197463785)

[1.6 Backend Setup 3](#_Toc197463786)

[1.7 Frontend Setup 3](#_Toc197463787)

[1.8 Deployment Tips 3](#_Toc197463788)

# System Architecture Diagram

## Components

* **Frontend (React.js)**: Provides UI for users to browse, add to cart, log in, and checkout.
* **Backend (Node.js + Express.js)**: Handles API requests, user authentication, cart management, and communication with the database.
* **Database (MySQL)**: Stores user data, book details, cart data, and order information.
* **External API**: Google Books API fetches book details by book ID.

## Architecture Flow

1. User interacts with the React frontend.
2. Frontend sends authenticated requests to the Node.js backend.
3. Backend queries MySQL DB or Google Books API based on the request.
4. Responses are sent back to the frontend for rendering.

## **API Documentation**

**Base URL**: http://localhost:5000/api

| **Method** | **Endpoint** | **Description** | **Auth Required** | **Body / Params** |
| --- | --- | --- | --- | --- |
| GET | /books/:id | Fetch a single book by Google Books ID | No | : id (string - Google Books ID) |
| GET | /cart/getcart | Get the current user's cart | Yes | - |
| PUT | /cart/update | Update item quantity in cart | Yes | { product\_id, quantity } |
| DELETE | /cart/remove/:productId | Remove item from cart | Yes | : productId (Google Books ID) |
| POST | /auth/login | Log the user in | No | { email, password } |
| POST | /auth/register | Register user | No | { name, email, password , phoneNo} |

## Database Schema

**Tables**

**Users**

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255),

email VARCHAR(255) UNIQUE,

password VARCHAR(255),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**cart\_items**

sql

CopyEdit

CREATE TABLE cart\_items (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_email VARCHAR(60),

product\_id VARCHAR(255), -- Google Books ID

quantity INT,

FOREIGN KEY (user\_id) REFERENCES users(email)

);

**Note**

Book metadata is fetched from the Google Books API and not stored in the DB.

## Setup and Deployment Guide

**Prerequisites**

1. Node.js and npm installed
2. MySQL server running
3. React installed (optional: create-react-app)

## Backend Setup

cd backend

npm install

* Create .env file:

DB\_HOST=localhost

DB\_USER=root

DB\_PASSWORD=yourpassword

DB\_NAME=bookshop

JWT\_SECRET=your\_jwt\_secret

* Start server:

npm run dev

## Frontend Setup

cd frontend

npm install

npm start

## Deployment Tips

* Use **Vercel** or **Netlify** for React frontend.
* Deploy backend on **Render**, **Railway**, or **Heroku**.
* Use **ClearDB** (MySQL) for remote DB hosting if needed.