Full Stack Development with MERN

API Development and Integration Report

Date	18 July 2024
Team ID	SWTID1720091047
Project Name	Project – E-commerce Platform
Maximum Marks	

Project Title: E-commerce Platform
Date: 18 July 2024
Prepared by: Thulasi Ram Reddy V T

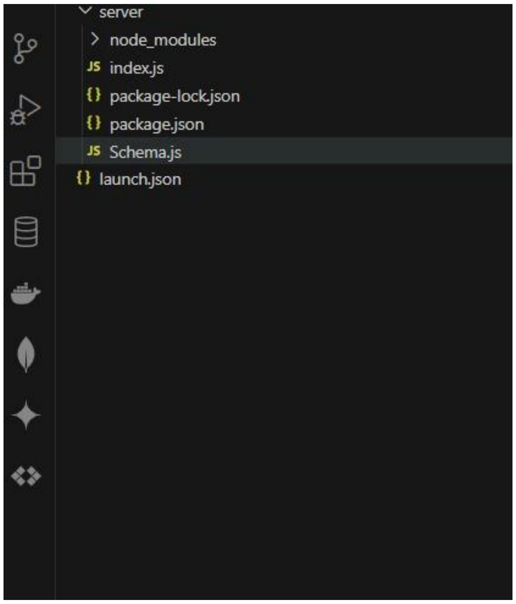
Objective

The objective of this report is to document the API development progress and key aspects of the backend services implementation for the E-commerce Platform project.

Technologies Used

- **Backend Framework:** Node.js with Express.js
- **Database:** MongoDB
- **Authentication:** [e.g., JWT, OAuth]

Project Structure



```
∨ server
  > node_modules
  JS index.js
  {} package-lock.json
  {} package.json
  JS Schema.js
{} launch.json
```

Key Directories and Files

1. **/controllers**
   - Contains functions to handle requests and responses.
2. **/models**
   - Includes Mongoose schemas and models for MongoDB collections.
3. **/routes**
   - Defines the API endpoints and links them to controller functions.
4. **/middlewares**
   - Custom middleware functions for request processing.
5. **/config**
   - Configuration files for database connections, environment variables, etc.

API Endpoints

A summary of the main API endpoints and their purposes:

User Authentication

- **POST /api/user/register** - Registers a new user.
- **POST /api/user/login** - Authenticates a user and returns a token.

User Management

- **GET /api/user/-** Retrieves user information by ID.
- **PUT /api/user/-** Updates user information by ID.

Product Management

- **GET /api/products** - Retrieves all products.
- **POST /api/products** - Creates a new product.
- **GET /api/products/-** Retrieves a product by ID.
- **PUT /api/products/-** Updates a product by ID.
- **DELETE /api/products/-** Deletes a product by ID.

Order Management

- **GET /api/orders** - Retrieves all orders.
- **POST /api/orders** - Creates a new order.
- **GET /api/orders/-** Retrieves an order by ID.
- **PUT /api/orders/-** Updates an order by ID.
- **DELETE /api/orders/-** Cancels an order by ID.

Cart Management

- **GET /api/cart** - Retrieves the current user's cart.
- **POST /api/cart** - Adds an item to the cart.
- **PUT /api/cart/**- Updates a cart item by ID.
- **DELETE /api/cart/**- Removes an item from the cart by ID.

Integration with Frontend
The backend communicates with the frontend via RESTful APIs. Key points of integration include:

- **User Authentication:** Tokens are passed between frontend and backend to handle authentication.
- **Data Fetching:** Frontend components make API calls to fetch necessary data for display and interaction.

Error Handling and Validation
Describe the error handling strategy and validation mechanisms:

- **Error Handling:** Centralized error handling using middleware.
- **Validation:** Input validation using libraries like Joi or express-validator.

Security Considerations
Outline the security measures implemented:

- **Authentication:** Secure token-based authentication.
- **Data Encryption:** Encrypt sensitive data at rest and in transit.