**Full Stack Development with MERN**

**API Development and Integration Report**

|  |  |
| --- | --- |
| Date | 21-07-2024 |
| Team ID | SWTID1719938064 |
| Project Name | ShopEZ: E-Commerce Application |
| Maximum Marks | 12 |

**Project Title:** ShopEZ: E-Commerce Application  
**Date:** 21-07-2024  
**Prepared by:** Mokshagna & Team

**Objective**  
The objective of this report is to document the API development progress and key aspects of the backend services implementation for the ShopEZ: E-Commerce Application project.

**Technologies Used**

* **Backend Framework:** Node.js with Express.js
* **Database:** MongoDB
* **Authentication:** non

**Project Structure**  
Provide a screenshot of the backend project structure with explanations for key directories and files.

**Key Directories and Files**

1. **/index**
   * Contains functions to handle requests and responses.
2. **/schema**
   * Includes Mongoose schemas and models for MongoDB collections.

**API Endpoints**  
A summary of the main API endpoints and their purposes:

**User Authentication**

* **POST /api/user/register** - Registers a new user.

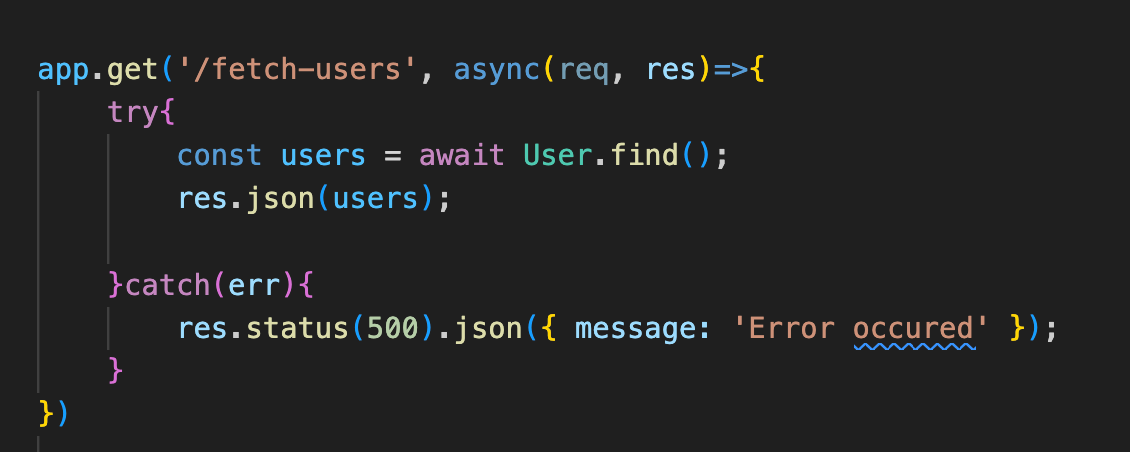


* **POST /api/user/login** - Authenticates a user and returns a token.



**User Management**

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



**Products**

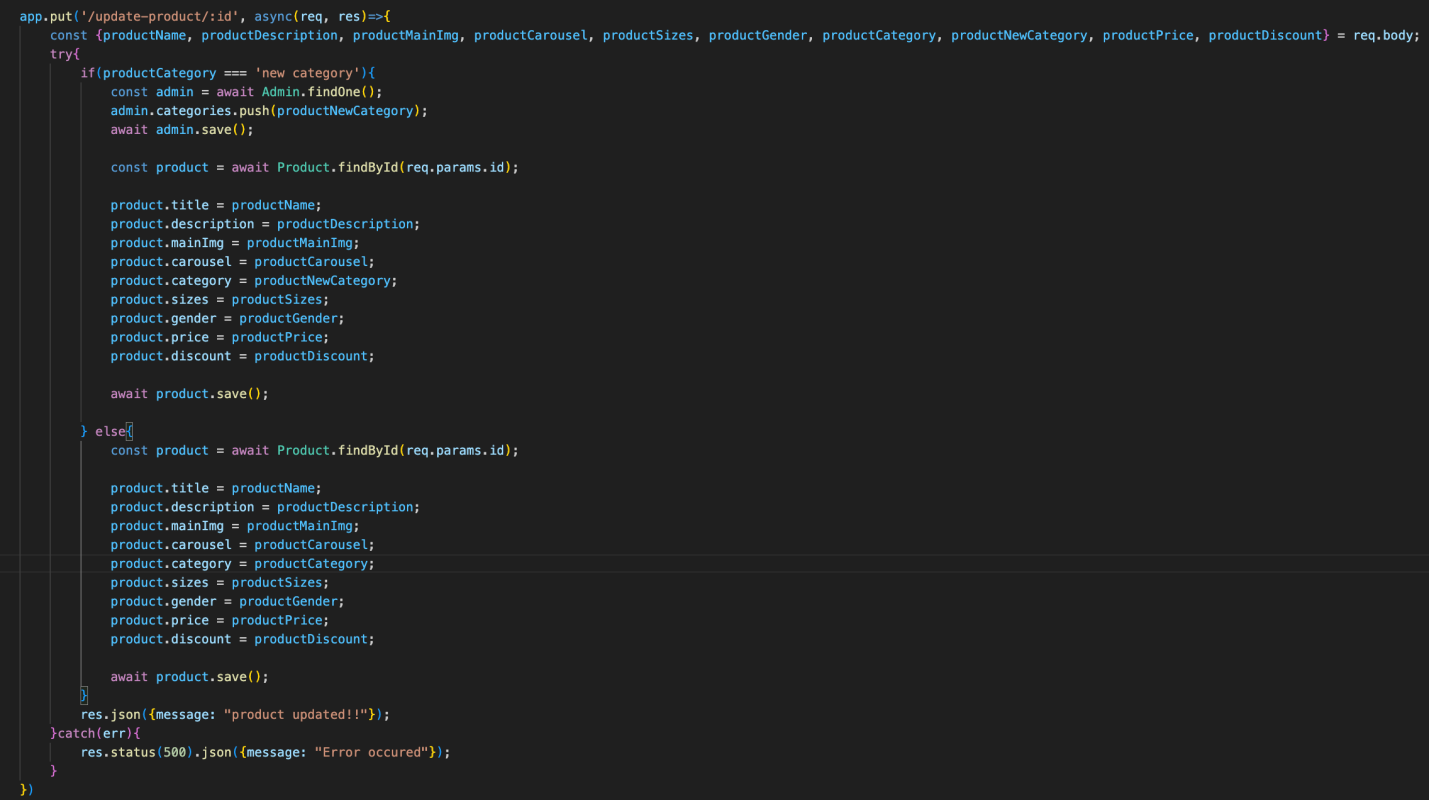
* **GET /api/product-details** - Retrieves individual product information



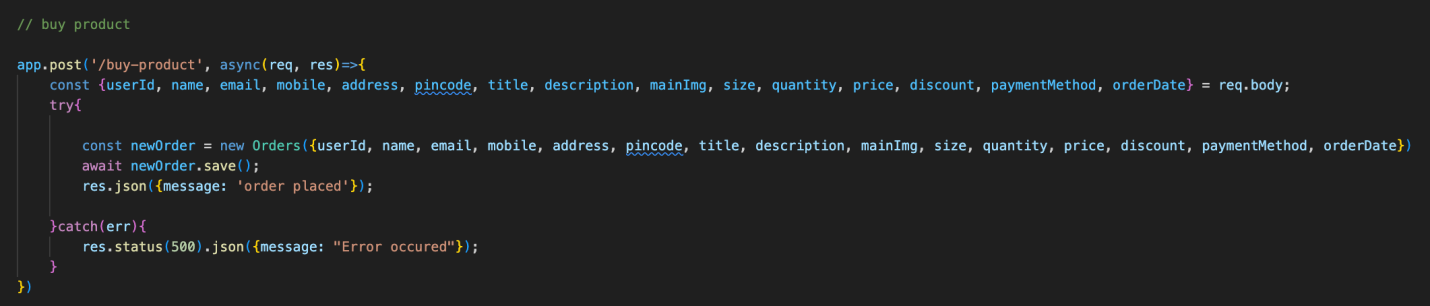
* **POST /api/add-new-product** - Add new product



**PUT /api/update-product** - Update the product

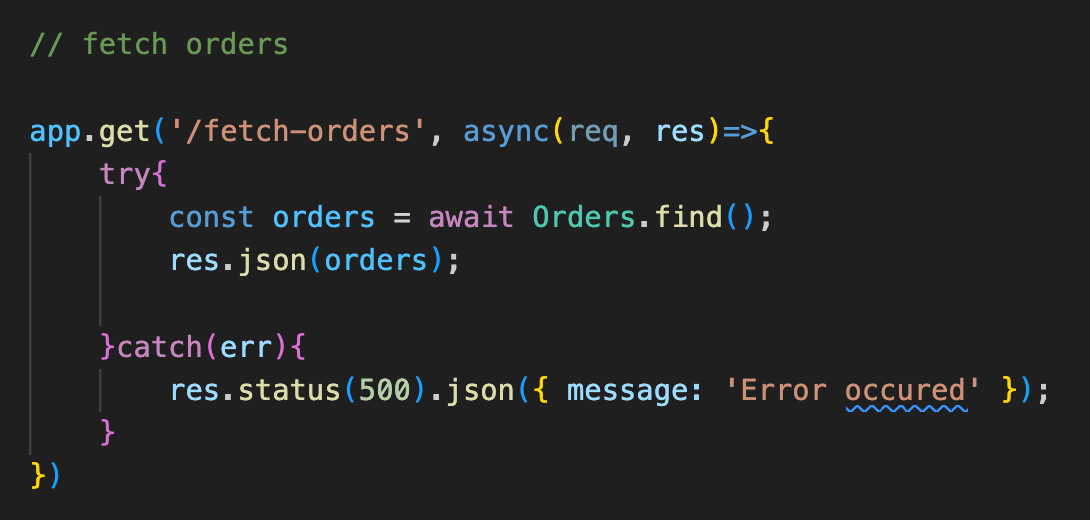


* **POST/api/buy-product** - Buy a product

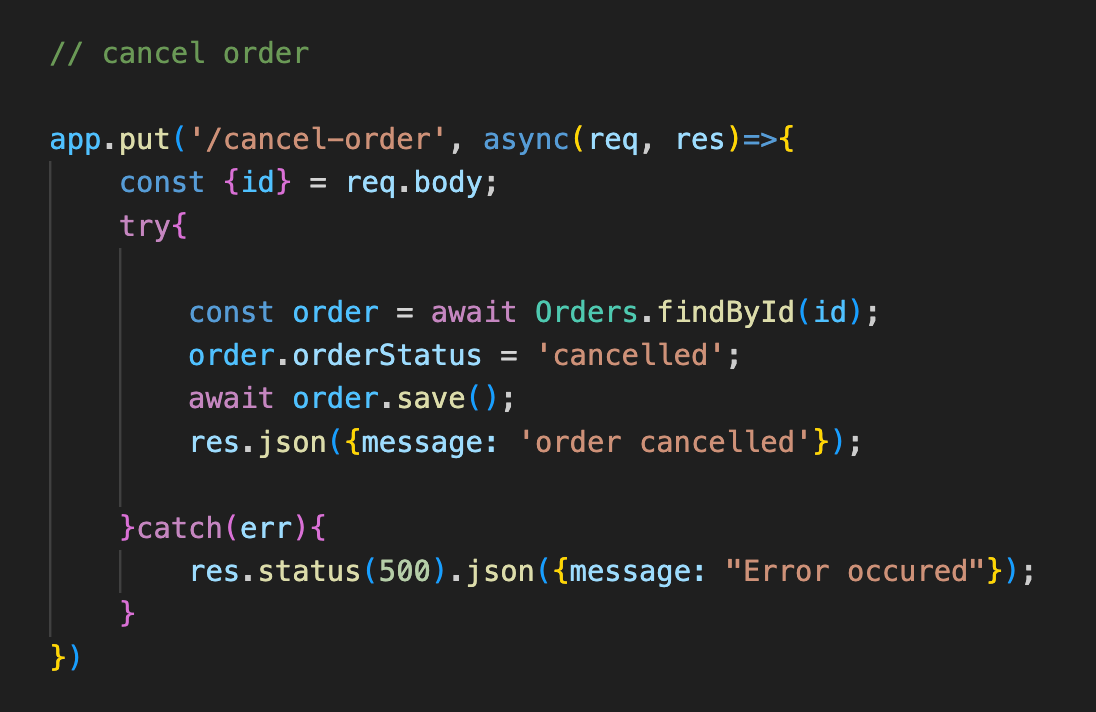


**Orders**

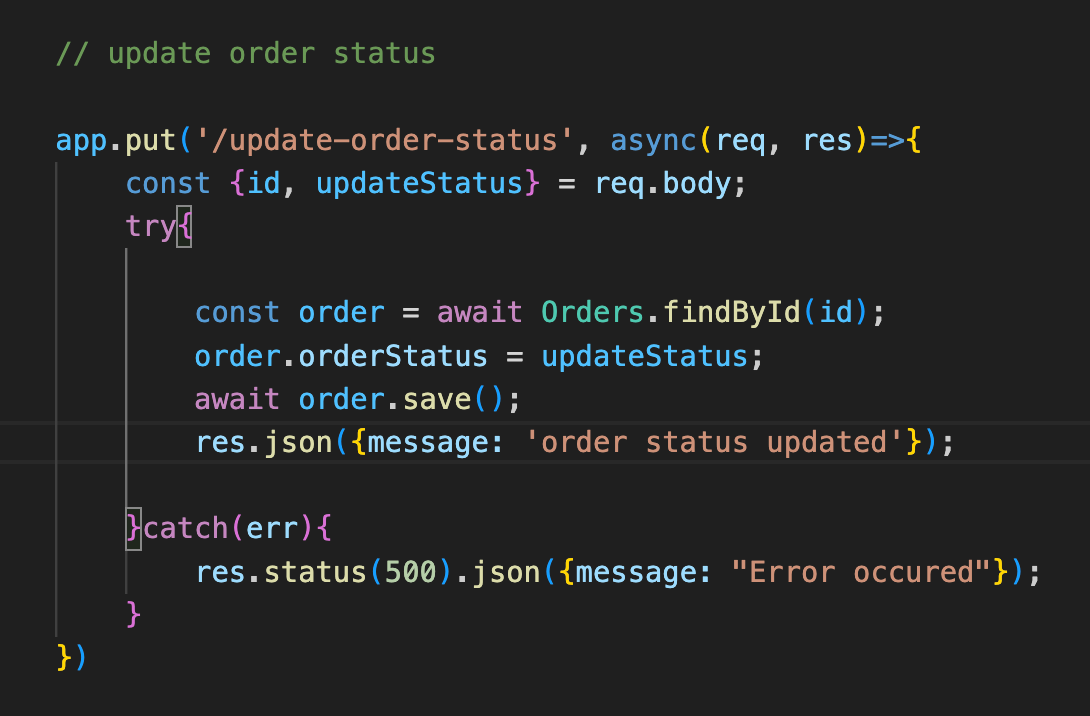
* **GET /api/fetch-orders** - Retrieves the orders



* **PUT /api/cancel-order** - Cancels an order

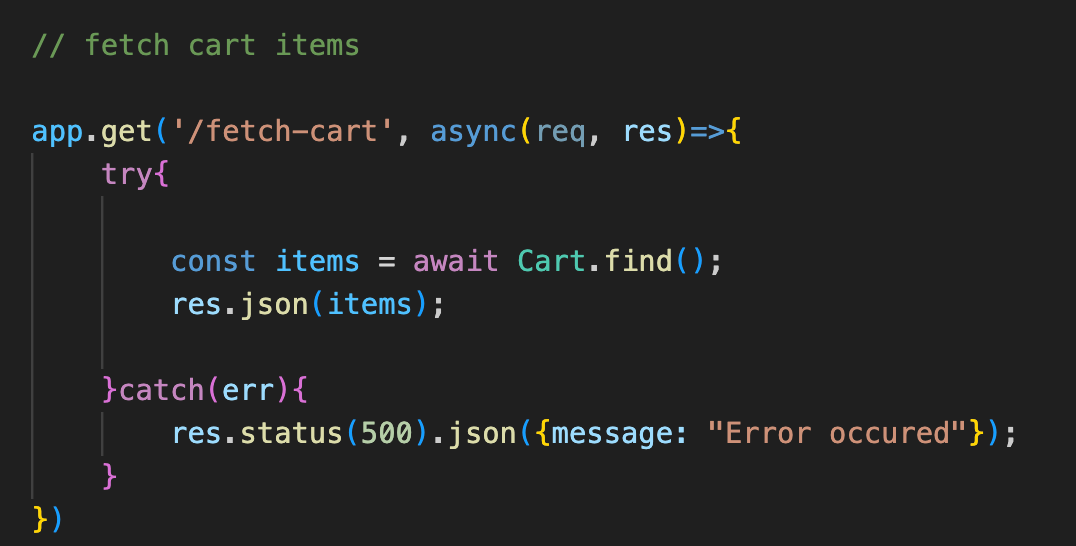


**PUT /api/update-order-status** - Updates an order’s status



**Cart**

* **GET /api/fetch-cart** - Retrieves all the items in the cart.



**POST /api/add-to-cart** - Adds an item to the cart.



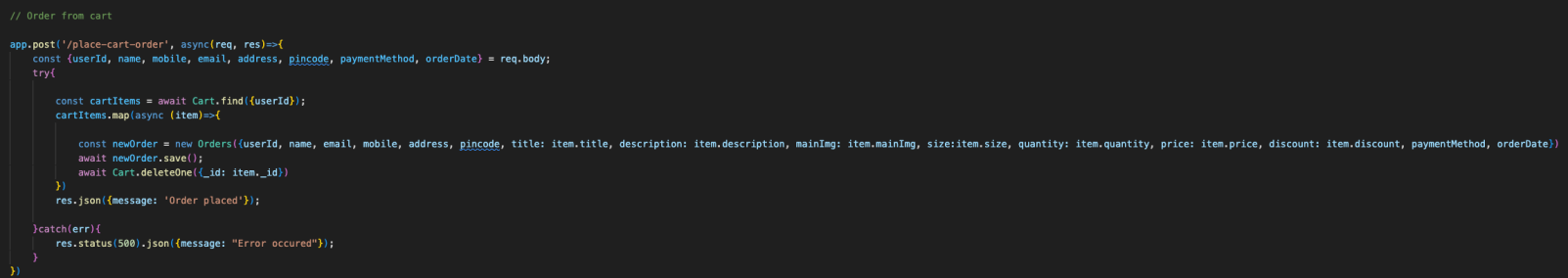
* **POST /api/add-to-cart** - Adds an item to the cart.



* **PUT /api/remove-item**- Removes an item from the cart.



* **POST /api/place-cart-order**- Places an order from the cart.



**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.