**Full Stack Development with MERN**

**API Development and Integration Report**

|  |  |
| --- | --- |
| **Date** | **18 July 2024** |
| **Team ID** | **SWTID1720077433** |
| **Project Name** | **Grocery App** |
| **Maximum Marks** |  |

**Project Title: All-Mart Grocery App**

Date: 19/07/2024

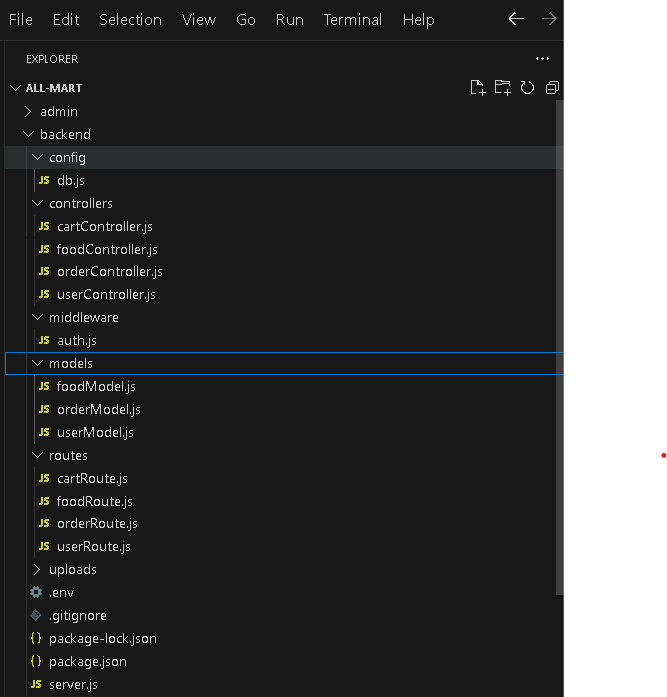
Prepared by: Mohammed Zaid, Naveen R, Mohammed Faisal

**Objective**  
The objective of this report is to document the API development progress and key aspects of the backend services implementation for the **All-Mart Grocery App** project.

**Technologies Used**

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

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



**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.

Cart Functionality

* **POST /api/cart /get**- gets user cart
* **POST /api/cart/remove** -remove food from user cart
* **POST /api/cart /add -** add to user 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.