

# Full Stack Development with MERN

## API Development and Integration Report

Date	13-07-2024
Team ID	SWTID1720199985
Project Name	Grocery Webapp
Maximum Marks	6 marks

**Project Title:** Grocery Webapp

**Date:** 13-07-2024

**Prepared by:** Surepalli Pujitha & Hima Sruthi Sappidi

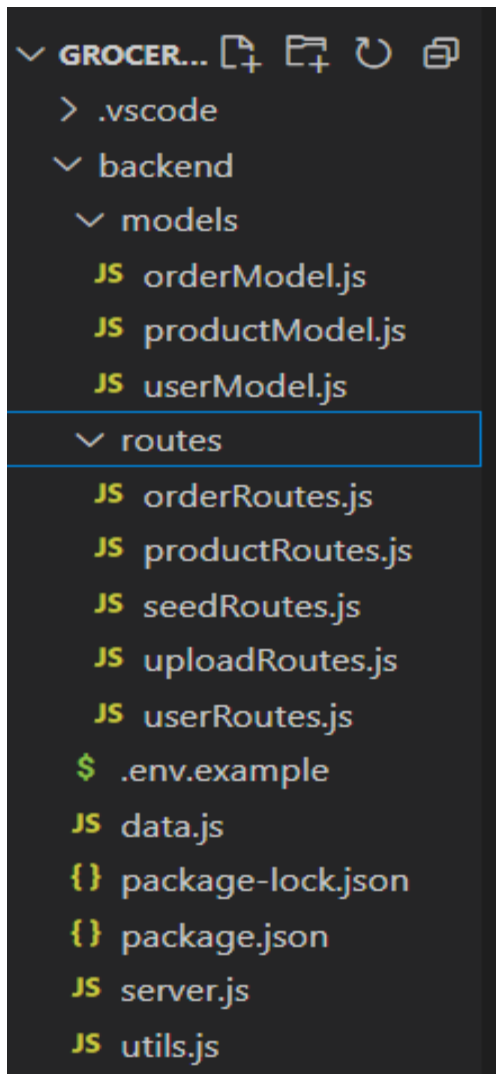
### Objective

The objective of this report is to document the API development progress and key aspects of the backend services implementation for the “Grocery Webapp” project.

### Technologies Used

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

### Project Structure:



## Key Directories and Files

### 1. /models

- **orderModel.js:** Schema and model for orders.
- **productModell.js:** Schema and model for products.
- **userModel.js:** Schema and model for users.

### 2. /routes

- **orderRoutes.js:** API routes for orders.
- **productRoutes.js:** API routes for products.
- **seedRoutes.js:** API routes for seeding initial data.
- **uploadRoutes.js:** API routes for file uploads.
- **userRoutes.js:** API routes for users.

### 3. /config

- **.env.example:** Example environment variables file.

### 4. /additional files

- **server.js:** Entry point for the backend server.
- **utils.js:** Utility functions used across the backend.

## API Endpoints

A summary of the main API endpoints and their purposes:

### User Authentication

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

### User Management

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

### Product Management

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

### Order Management

- **GET /api/orders:** Retrieves all orders.
- **POST /api/orders:** Creates a new order.
- **GET /api/orders/:** Retrieves order information by ID.
- **PUT /api/orders/:** Updates order information by ID.
- **GET /api/orders/mine:** Retrieves orders of the authenticated user.
- **PUT /api/orders//pay:** Updates order payment status.

### Seed Data

- **POST /api/seed:** Seeds initial data.

### File Uploads

- **POST /api/upload:** Uploads a file.

## 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

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

## Security Considerations

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