BurgerStoreAPI Documentation

By Shubham Shejwal



# 1. Project Overview

The BurgerStoreAPI is a .NET 8 Web API project designed to handle the backend operations for a burger shop. This API allows users to interact with various entities such as menus, orders, users, and cart items. The project is structured to include separate controllers, services, and data models, ensuring a clean and maintainable codebase.

# 2. Installation and Setup

Requirements:

* - .NET 8 SDK
* - SQL Server (or any configured database)
* - Visual Studio 2022 (or any IDE supporting .NET 8)

Steps:

1. 1. Clone or download the repository.
2. 2. Open the solution file (BurgerStoreAPI.sln) in Visual Studio.
3. 3. Restore NuGet packages.
4. 4. Update the appsettings.json file with your database connection string.
5. 5. Run database migrations (if any) to set up the database.
6. 6. Build and run the project.

# 3. API Endpoints and Usage

Authentication:  
The API uses JWT tokens for authentication. Ensure to implement login and token generation in your application.

Endpoints:

* - Users: /api/users  
   - GET /api/users: Retrieves all users.  
   - POST /api/users: Registers a new user.
* - Menus: /api/menus  
   - GET /api/menus: Retrieves all menu items.
* - Orders: /api/orders  
   - GET /api/orders: Retrieves all orders.  
   - POST /api/orders: Places a new order.
* - Cart Items: /api/cartitems  
   - GET /api/cartitems: Retrieves cart items for a user.  
   - POST /api/cartitems: Adds an item to the cart.
* - Admins: /api/admins  
   - GET /api/admins: Retrieves admin details.

# 4. Authentication and Security

The API implements JWT-based authentication. Only authenticated users can access certain endpoints. The AdminsController is separated to handle admin-related tasks, which could involve higher privileges.

# 5. Data Models and Database Schema

Entities:

* - User: Represents a user in the system with properties like Id, Name, MobileNumber, and Email.
* - Menu: Represents a menu item with properties like Id, Name, Type, Price, and ImageUrl.
* - Order: Represents an order placed by a user, linked to the User entity.
* - CartItem: Represents items in a user's cart.

Database Context:  
The BurgerStoreContext class handles the database operations and is configured in the Program.cs file.

# 6. Code Structure

Controllers: Handle incoming HTTP requests and return appropriate responses.

* - UsersController.cs, MenusController.cs, OrdersController.cs, CartItemsController.cs, AdminsController.cs

Services: Business logic is encapsulated in service classes.

* - UserService.cs, MenuService.cs, OrderService.cs, CartItemService.cs, AdminService.cs

Models: Define the data structure.

* - User.cs, Menu.cs, Order.cs, CartItem.cs, Admin.cs

Data Layer: Handles database interactions.

* - BurgerStoreContext.cs, UserRepository.cs, OrderRepository.cs, etc.

# 7. Known Issues and Limitations

- The current version may not support extensive error handling for all endpoints.

- Performance optimizations are not yet implemented, especially for large datasets.

-DTO structure has not been used

# 8. Future Enhancements

- Implement detailed logging for better tracking and debugging.

- Add support for OAuth2.0 for improved security.

- Integrate with third-party payment services for order processing.

-Adding DTOs