**Baking Sisters Project Documentation**

**Introduction**

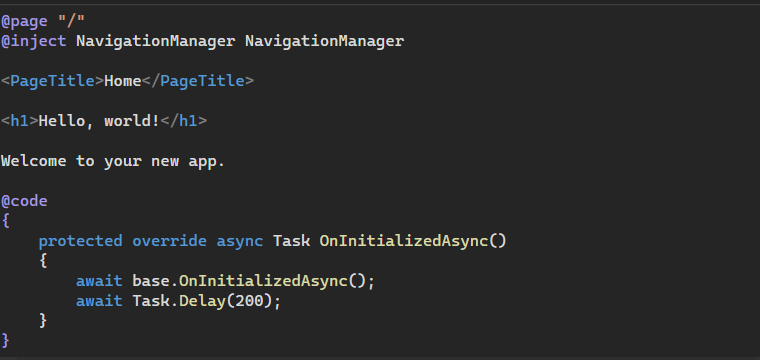
The Baking Sisters project is a web application designed to manage a bakery's operations, including product listings, user authentication, and order management. The project is built using Blazor for the front end and ASP.NET Core for the backend, targeting .NET 9.0. This document outlines the project's architecture, services, and test cases, focusing on the first milestone.

**Milestone 1: Project Setup and Basic Functionality**

**Architecture**

**Domain-Driven Design (DDD)**

The project follows the principles of Domain-Driven Design (DDD) to ensure a clear separation of concerns and a robust domain model. The key components of DDD architecture include:

* **Domain Layer**: Contains the core business logic and domain entities.
* **Application Layer**: Manages application-specific logic and orchestrates the use of domain services.
* **Infrastructure Layer**: Handles data access, external services, and other infrastructure concerns.
* Presentation Layer: Manages the user interface and user interactions.
* 

**Project Structure**

The project consists of the following main components:

* **BakingSisters.Web**: The Blazor frontend application.
* **BakingSisters.Api**: The ASP.NET Core backend API.
* **BakingSisters.Testing**: The test project for unit and integration tests.
* **BakingSisters.Mobile**: The mobile application built using .NET MAUI (to be developed in Milestone 3).
* **BakingSisters.Chatbot**: The chatbot for personalized user experience (to be developed in Milestone 5).

**Blazor Frontend**

The Blazor frontend is responsible for rendering the user interface and handling client-side logic. It includes components, pages, and services to interact with the backend API.

**ASP.NET Core Backend**

The ASP.NET Core backend provides RESTful API endpoints for managing data and business logic. It includes controllers, services, and data models.

**Services**

API Services

The API services are responsible for handling business logic and data access. The key services include:

1. **AuthService**: Manages user authentication and registration.
   * **LoginAsync**: Authenticates a user and returns a JWT token.
   * **RegisterAsync**: Registers a new user and returns the user details.
   * **GetUserByEmailAsync**: Retrieves user details by email.
2. **BaseService**: A generic service for CRUD operations.
   * **GetAllAsync**: Retrieves all entities.
   * **GetByIdAsync**: Retrieves an entity by its ID.
   * **CreateAsync**: Creates a new entity.
   * **UpdateAsync**: Updates an existing entity.
   * **DeleteAsync**: Deletes an entity by its ID.
   * **ExistsAsync**: Checks if an entity exists by its ID.
   * A computer screen with text and images

     AI-generated content may be incorrect.

**Test Cases**

The test project, **BakingSisters.Testing**, includes unit and integration tests to ensure the reliability and correctness of the application. The key components of the test project are:

1. **TestBase**: A base class for setting up the test environment.
   * **BakeryDbContext**: An in-memory database context for testing.
   * **Configuration**: Configuration settings for testing, including JWT keys.
2. **AuthServiceTests**: Tests for the AuthService.
   * **LoginAsync\_ValidCredentials\_ReturnsToken**: Tests successful login.
   * **LoginAsync\_InvalidCredentials\_ReturnsUnauthorized**: Tests login with invalid credentials.
   * **RegisterAsync\_ValidUser\_ReturnsUser**: Tests successful user registration.
   * **RegisterAsync\_DuplicateEmail\_ReturnsConflict**: Tests registration with a duplicate email.
3. **BaseServiceTests**: Tests for the BaseService.
   * **GetAllAsync\_ReturnsAllEntities**: Tests retrieval of all entities.
   * **GetByIdAsync\_ValidId\_ReturnsEntity**: Tests retrieval of an entity by a valid ID.
   * **CreateAsync\_ValidEntity\_ReturnsEntity**: Tests creation of a new entity.
   * **UpdateAsync\_ValidEntity\_ReturnsUpdatedEntity**: Tests updating an existing entity.
   * **DeleteAsync\_ValidId\_DeletesEntity**: Tests deletion of an entity by a valid ID.
   * **ExistsAsync\_ValidId\_ReturnsTrue**: Tests existence check for a valid ID.

**Future Milestones**

**Milestone 2: User Management and Authentication**

* **User Registration and Login**: Implement user registration and login functionality with JWT authentication.
* **Role-Based Access Control**: Implement role-based access control (RBAC) to manage different user roles (e.g., Admin, Customer, Guest).
* **Profile Management**: Allow users to update their profile information.

**Milestone 3: Mobile Application Development**

* **Mobile App with .NET MAUI**: Develop a mobile application using .NET MAUI to provide a seamless experience across iOS and Android devices.
* **Synchronization**: Ensure data synchronization between the mobile app and the web application.
* **Push Notifications**: Implement push notifications for order updates and promotions.

**Milestone 4: Product Management**

* **Product Catalog**: Implement a product catalog to display available bakery items.
* **Product Categories**: Add support for product categories to organize items.
* **Product Search and Filtering**: Implement search and filtering functionality for products.

**Milestone 5: Chatbot Integration**

* **Chatbot for Personalized Experience**: Integrate a chatbot to provide personalized assistance to users, including product recommendations and order support.
* **Natural Language Processing (NLP)**: Utilize NLP to understand and respond to user queries effectively.
* **Chatbot Analytics**: Implement analytics to track chatbot interactions and improve its performance.

**Milestone 6: Order Management**

* **Shopping Cart**: Implement a shopping cart for users to add and manage items before checkout.
* **Order Placement**: Allow users to place orders and process payments.
* **Order History**: Provide users with access to their order history.

**Milestone 7: Inventory Management**

* **Stock Management**: Implement inventory management to track product stock levels.
* **Notifications**: Add notifications for low stock levels and out-of-stock items.
* **Supplier Management**: Manage suppliers and restock products.

**Milestone 8: Reporting and Analytics**

* **Sales Reports**: Generate sales reports to analyze revenue and performance.
* **Customer Insights**: Provide insights into customer behavior and preferences.
* **Inventory Reports**: Generate reports on inventory levels and restocking needs.

**Milestone 9: User Experience Enhancements**

* **Responsive Design**: Ensure the application is fully responsive and works well on different devices.
* **Accessibility**: Improve accessibility to make the application usable for all users.
* **Performance Optimization**: Optimize the application for better performance and faster load times.

**Milestone 10: Testing and Quality Assurance**

* **Unit Testing**: Expand unit tests to cover all critical functionalities.
* **Integration Testing**: Implement integration tests to ensure different components work together seamlessly.
* **End-to-End Testing**: Conduct end-to-end testing to simulate real user scenarios and ensure the application works as expected.

**Conclusion**

This document provides an overview of the Baking Sisters project, focusing on the initial milestone and future enhancements. It includes the project structure, key services, and test cases. Future milestones will build upon this foundation, adding more features and functionality to the application, including a mobile application, chatbot integration, and comprehensive reporting and analytics.

From below you can find the link of the GitHub repository for the current code  
<https://github.com/NooriaNasir92/BakingSisters.Web>