Canteen App Documentation

1. Landing Page

The landing page is the first interaction users will have with the canteen app. Its purpose is to welcome users, display featured food items, present different menu categories, and highlight special offers or promotions.

Features:

- **Featured Food Section:** A section showcasing popular or featured food items.
- **Menu Categories:** Categories like breakfast, lunch, snacks, drinks, etc. that help users easily navigate the food options.
- **Special Offers/Promotions:** Display ongoing discounts or special offers that users can avail of.
- **Navigation Bar:** Provides links to other sections such as the menu, order history, checkout, contact page, and user profile.

Backend Considerations:

- **Controllers:** LandingController.java – Handles requests to the landing page and fetches necessary data (featured items, special offers).
- **Services:** MenuService.java – Handles business logic for fetching food categories and featured items.
- **Models:** FoodItem.java, Offer.java – Represents food items and promotions.
- **Database:** Store food items, categories, and special offers in the database, queried by the backend.

Frontend:

- **HTML/CSS:** A simple, clean layout with featured images and offer highlights.
- **JavaScript:** To dynamically load featured items and offers from the backend.

---

2. Menu Page

The Menu page displays a list of categorized food items with details such as the name, image, price, and an option to add items to the cart.

Features:

- **Categories:** Food items grouped into categories like breakfast, lunch, snacks, beverages, etc.
- **Food Item Details:** Name, image, description, and price for each food item.
- **Add to Cart Button:** Users can select items and add them to their shopping cart.
- **Search/Filter Options:** Ability to search for food items and filter by category or price range.

Backend Considerations:

- **Controllers:** MenuController.java – Retrieves food items and categories from the database.
- **Services:** MenuService.java – Business logic for fetching, filtering, and categorizing food items.
- **Models:** FoodItem.java, Category.java – Represent the food items and categories.
- **Database:** Food items are stored with fields like name, description, price, image URL, and category.

Frontend:

- **HTML/CSS:** A card-based grid layout displaying food items with images, prices, and add-to-cart buttons.
- **JavaScript:** For filtering, searching, and dynamically loading data from the backend.

---

3. Order Page

The Order Page shows users their current orders and past orders, including the status of each order. Users can reorder from previous orders for convenience.

Features:

- **Current Orders:** A list of active orders with status (e.g., "Pending", "In Progress", "Completed").
- **Order History:** A list of completed orders with details such as items, total price, and order status.
- **Reorder Option:** Allows users to reorder items from their previous orders.

- **Order Details:** Users can view the details of past orders, including food items, quantities, and price.

Backend Considerations:

- **Controllers:** OrderController.java – Handles requests related to viewing current and past orders.
- **Services:** OrderService.java – Manages order statuses, order creation, and reordering.
- **Models:** Order.java, OrderItem.java, User.java – Represents orders and their associated items.
- **Database:** Orders and order items are stored in the database with user association, status, and item details.

Frontend:

- **HTML/CSS:** Display of orders in a list format with status indicators.
- **JavaScript:** Handles the dynamic updating of order statuses and reordering functionality.

---

4. Contact Page

The Contact Page provides customer support information, frequently asked questions (FAQs), and a feedback form for users to leave suggestions or complaints.

Features:

- **Customer Support Information:** Contact details such as phone numbers, emails, and office hours.
- **FAQs Section:** A list of common questions and answers to help users troubleshoot or find information quickly.
- **Feedback Form:** Users can submit feedback, report issues, or provide suggestions for improvements.

Backend Considerations:

- **Controllers:** ContactController.java – Handles requests related to contact information and form submissions.
- **Services:** FeedbackService.java – Processes and stores user feedback.

- **Models:** Feedback.java – Represents the user's feedback, with fields such as name, message, and rating.
- **Database:** Store feedback from users, queries, and FAQs in the database for future reference.

Frontend:

- **HTML/CSS:** A clean layout for displaying contact info, a list of FAQs, and a simple form for feedback.
- **JavaScript:** For form validation and AJAX-based submission of feedback.

---

5. Checkout Page

The Checkout Page is where users provide their delivery details, select payment methods, and confirm their order.

Features:

- **Delivery Details:** Users enter their address, contact details, and any special instructions.
- **Payment Information:** Users select a payment method (credit card, wallet, cash on delivery, etc.).
- **Order Confirmation:** A summary of the order is displayed with the total cost, including taxes and delivery charges.
- **Order Placement:** After confirming the details, users can place the order and receive an order confirmation.

Backend Considerations:

- **Controllers:** CheckoutController.java – Handles the checkout process, including payment processing and order finalization.
- **Services:** PaymentService.java – Manages payment gateway integration and order processing.
- **Models:** Order.java, Payment.java, DeliveryDetails.java – Represents the order, payment, and delivery information.
- **Database:** Store the checkout details including delivery and payment information.

Frontend:

- **HTML/CSS:** Forms for user inputs (delivery details, payment info) and an order summary table.
- **JavaScript:** Validation for payment methods and dynamic updates for order summary.

Backend Structure Overview (Java Full Stack)

1. **Controllers** – Java classes responsible for handling HTTP requests and returning appropriate responses to the frontend.
    - LandingController.java
    - MenuController.java
    - OrderController.java
    - ContactController.java
    - CheckoutController.java
2. **Services** – Java classes that handle the business logic and interactions with the database.
    - MenuService.java
    - OrderService.java
    - PaymentService.java
    - FeedbackService.java
3. **Models** – Java classes representing the data objects used in the application.
    - FoodItem.java
    - Order.java
    - Payment.java
    - User.java
    - Category.java
    - Feedback.java

4. **Database** – A relational database (e.g., MySQL, PostgreSQL) is used to store the data:
    - Tables for users, food items, orders, payments, feedback, etc.
5. **Authentication** – Implement user login and registration using JWT tokens or session-based authentication.

Frontend Structure (HTML, CSS, JavaScript)

1. **HTML Templates:** These will represent the structure of the pages (Landing, Menu, Order, Contact, Checkout, etc.).
2. **CSS/Bootstrap:** Styling of the frontend pages for responsive and aesthetically pleasing design.
3. **JavaScript (AJAX/Fetch API):** Used for making asynchronous requests to the backend and updating the UI dynamically (e.g., updating the cart or handling order status changes).