**Project Design Phase-II**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 26 June 2025 |
| Team ID | LTVIP2025TMID59601 |
| Project Name | orderonthego: your on-demand food ordering solution |
| Maximum Marks | 4 Marks |

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form (Email and Password)  Registration for Customers  Registration for Restaurant Owners  Registration for Admins |
| FR-2 | User Confirmation | User Login (Email and Password)  Role-based access control (Admin, Restaurant, Customer) |
| FR-3 | Profile Management | View User Profile (Name, Email, Address, Phone)  Update User Profile (Name, Email, Address, Phone) |
| FR-4 | Restaurant Management | Restaurant Owner Registration and Approval (Admin approval)  View List of Restaurants (Admin, Customer)  Add/Edit Restaurant Details (Owner, Admin)  Delete Restaurant (Admin) |
| FR-5 | Menu Management | View Menu Items by Restaurant  Add/Edit Food Items (Restaurant Owner)  Delete Food Items (Restaurant Owner) |
| FR-6 | Order Management | Create New Order (Customer)  View Order History (Customer, Restaurant Owner, Admin)  Update Order Status (e.g., pending, preparing, delivered, cancelled) (Restaurant Owner) |
| FR-7 | Cart Management | Add Food Item to Cart  Update Quantity in Cart  Remove Item from Cart  Clear Cart |
| FR-8 | Admin Functions | Approve/Reject User Accounts (Restaurant Owners)  Manage Categories (Add, Edit, Delete global food categories)  View Dashboard Counts (Total Users, Restaurants, Orders, Food Items, Pending Approvals, Feedback)  Manage Promoted Restaurants  View/Manage Feedback (Customer to Restaurant, Restaurant to Admin, User to Admin)  Send Announcements/Feedback from Admin to Users/Restaurants  Generate Reports (Metrics, Order Trend, Top Restaurants, Category Popularity, Rating Distribution) |
| FR-9 | Feedback System | Submit Feedback/Ratings (Customer to Restaurant/Food Item)  Submit Feedback (Restaurant to Admin)  Submit Feedback (Customer/User to Admin) |

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The application should have an intuitive and easy-to-navigate user interface. Responsive design for various devices (desktop, mobile). Clear feedback for user actions. |
| NFR-2 | **Security** | All user data (passwords, personal info) must be securely stored (hashed passwords). Authentication and authorization mechanisms (JWT) must prevent unauthorized access. Protection against common web vulnerabilities (e.g., XSS, SQL Injection - via Mongoose). Secure communication (HTTPS in production). |
| NFR-3 | **Reliability** | The system should be robust and handle errors gracefully without crashing. Data integrity must be maintained during all operations (e.g., orders, inventory). Appropriate logging for troubleshooting. |
| NFR-4 | **Performance** | The application should respond to user requests promptly. API response times should be optimized. Database queries should be efficient to handle a reasonable load. |
| NFR-5 | **Availability** | The application should be accessible and operational for users during expected hours. Minimized downtime. (e.g., 99% uptime target). |
| NFR-6 | **Scalability** | The system should be able to handle an increasing number of users, restaurants, and orders without significant degradation in performance. (e.g., support for horizontal scaling of backend, database sharding capability). |