NAME : Gowsik B

PROJECT TITLE : Catering Reservation and Ordering System

UNID : UMIP282039

1. 🎯 Purpose

The purpose of this Low-Level Design (LLD) is to provide a detailed technical implementation plan for the Catering Reservation and Ordering System. It is intended to guide developers on how each module interacts with Firebase and how front-end pages communicate through JavaScript.

2. ⚙️ Technology Stack

Layer	**Technology Used**
Frontend	HTML5, CSS3, JavaScript (ES6)
Backend	Firebase Authentication, Firestore, Storage
Hosting	GitHub Pages / Firebase Hosting

3. 🧩 Module Overview

🔐 Authentication Module

- Handles user registration and login using Firebase Authentication.
- Stores user data (name, role) in Firestore.

🧑‍🍳 Admin Module

- Upload new catering products (with image).
- View and manage all orders (filter by status).
- Confirm or delete orders.

👥 User Module

- Browse products.
- Add to cart.
- Place order (with delivery address).
- View and track past orders.
- View/Edit profile.

4. 🔄 Workflow Diagram

```
[Login/Register] → [Redirect to Dashboard (based on role)]
 ↓
 └── [User]
     ├── View Products
     ├── Add to Cart
     ├── View Cart → Place Order → View Orders
     └── View/Edit Profile

 └── [Admin]
     ├── Upload Product
     ├── View All Orders
     └── Confirm/Delete Order
```

5. 📁 Page Mapping

Page	Description
index.html	Home page with hero, how it works
login.html, register.html	Auth pages
dashboard.html (User/Admin)	Role-based landing
cart.html	Cart UI for users
place-order.html	Confirm order with address
orders.html	Past orders (user/admin version)
upload-product.html (Admin)	Add new catering item
profile.html	User details and edit
occasion-*.html	Static info on special occasions

6. 🔥 Firebase Firestore Schema

users (Collection)

```
{
  "userId": "abc123",
  "name": "John Doe",
  "email": "john@example.com",
  "role": "user"
}
```

users/{userId}/cart (Subcollection)

```
{
  "productId": "xyz789",
  "name": "Paneer Butter Masala",
  "price": 150,
  "description": "Delicious creamy paneer",
  "quantity": 2
}
```

products (Collection)

```
{
  "productId": "xyz789",
  "name": "Sambar Rice",
  "price": 80,
  "description": "South Indian favorite",
  "imageURL": "https://firebasestorage.googleapis.com/..."
}
```

orders (Collection)

```
{
  "orderId": "ord456",
  "userId": "abc123",
  "userEmail": "john@example.com",
  "items": [
    { "name": "Paneer Butter Masala", "price": 150, "quantity": 2 },
    { "name": "Sambar Rice", "price": 80, "quantity": 1 }
  ],
  "total": 380,
  "status": "pending",
  "address": "123 Anna Nagar, Coimbatore",
  "createdAt": Timestamp
}
```

7. 🔧 Functional Specifications

User Functions

Feature	Description
Register/Login	Email + password using Firebase Auth
View Products	Fetch from Firestore products
Cart Management	Add/delete/update quantity in cart
Place Order	Store in orders collection
View Orders	Filter by current user's userId
Edit Profile	Update fields in Firestore users

Admin Functions

Feature	Description
Upload Product	Add product with image to Firestore
View Orders	Fetch all orders, filter by status
Confirm Order	Update status to "confirmed"
Delete Order	Remove order document from Firestore

8. 🔐 Security Rules (Example - Firebase)

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/{userId} {
      allow read, write: if request.auth.uid == userId;
      match /cart/{cartId} {
        allow read, write: if request.auth.uid == userId;
      }
    }
    match /orders/{orderId} {
      allow read, write: if request.auth != null;
    }
    match /products/{productId} {
      allow read: if true;
      allow write: if request.auth != null && get(/databases/$(database)/documents/users/$(request.auth.uid)).data.role == "admin";
    }
  }
}
```

9. 🎨 UI/UX Notes

- Mobile-responsive layout.
- Product cards and cart UI are styled with CSS.
- Each occasion has its own static HTML file styled with images and service details.
- Navigation dynamically changes by user rol

10. 🧪 Testing Checklist

Module	Status
User Login/Register	☑
Add to Cart	☑
Place Order	☑
Admin Upload	☑
Admin Order View	☑
Role Redirection	☑
Profile View/Edit	☑
Occasion Pages	☑

11. 📦 Future Enhancements

- Payment Gateway Integration
- Vendor dashboard (multi-vendor support)
- OTP-based mobile login
- Order history download as PDF
- Rating & Reviews