1. PROJECT DESCRIPTION

A. Title
Restaurant Reservation and Management System

B. Team Info
• Seif Eldahrawy – ID: 13002024
• Farah Arafa – ID: 13001053
• Jana Elfarra – ID: 13005717
• Lujyne Selim – ID: 13002071

C. Problem

In many restaurants, managing table reservations, customer orders, and feedback is still done manually or through disconnected systems. This often leads to:
• Double bookings or lost reservations
• Long waiting times for customers
• Difficulty updating the menu or handling sudden changes
• Lack of organized feedback for service improvement.
Such issues affect customer satisfaction, staff productivity, and the restaurant's overall efficiency.

D. How to Solve It

To address these challenges, we propose developing a web-based Restaurant Reservation and Management System. The system will allow:

Customers to:
• Reserve tables online by date, time, and number of guests
• Browse the menu and pre-order items
• Submit feedback after their dining experience

Admins (Restaurant Staff) to:
• Manage and confirm bookings efficiently
• Update and organize menu items easily
• View and analyze customer feedback to improve services

This solution will streamline restaurant operations, reduce errors, and enhance the dining experience for both customers and staff.

PROJECT IDEA

Objectives:

- Automate reservation processes to eliminate manual errors and double bookings.
- Enhance customer convenience by allowing online table booking, menu browsing, and order placement in advance.
- Improve operational efficiency for restaurant staff through a centralized system for managing reservations, menu updates, and feedback.
- Enable better decision-making by collecting and analyzing customer feedback for service improvement.
- Provide a responsive and user-friendly interface accessible from both desktop and mobile devices.

Scope:

Customer Side:
- Create an account and log in securely.
- Reserve tables online by selecting date, time, and number of guests.
- Browse the restaurant's digital menu and optionally pre-order food.
- Submit feedback or ratings after dining.
- Receive booking confirmations and updates via email or dashboard.

Admin (Restaurant Staff) Side:
- View, manage, confirm, or cancel customer reservations.
- Add, remove, or edit menu items dynamically.
- Monitor and respond to customer feedback.
- Access analytics on table usage, popular menu items, and customer satisfaction.

Expected Outcomes:

- Reduced workload on restaurant staff due to automated reservation and order management.
- Improved customer satisfaction through faster service and accurate bookings.
- Higher restaurant efficiency, with fewer booking conflicts and better time management.
- Centralized data management for reservations, menu items, and customer feedback.
- Actionable insights from feedback reports to enhance overall restaurant performance.

Justification of the Idea:

The idea for a Restaurant Reservation and Management System is justified by the increasing need for digital transformation in the hospitality industry. Many small and mid-sized restaurants still rely on phone calls or manual logs, which are prone to human error and inefficiency. In contrast, a digital system:
- Saves time for both customers and staff,
- Ensures better resource utilization (tables, staff schedules, etc.),
- Enhances customer engagement through online interaction, and
- Provides valuable data for continuous improvement.

With the growing trend of online booking systems and customer self-service platforms, implementing such a system will modernize restaurant operations, making it more competitive, reliable, and customer-centric.

Functional Requirements (FRs):

FR1 User Registration & Login As a user, I want to create an account and log in securely so that I can access my reservations and personalize my experience.
FR2 Menu Browsing As a customer, I want to view the restaurant's menu with categories, prices, and images so that I can choose items before visiting.
FR3 Table Reservation As a customer, I want to select a date, time, and number of guests so that I can reserve a suitable table easily.
FR4 Pre-Ordering (Optional) As a customer, I want to pre-order food while booking my table so that my meal is prepared on time.
FR5 Feedback Submission As a customer, I want to rate my experience and leave comments so that the restaurant can improve its service.
FR6 Reservation Management (Admin) As an admin, I want to view, confirm, or cancel reservations so that I can manage seating arrangements efficiently.
FR7 Menu Management (Admin) As an admin, I want to add, edit, or remove menu items so that the displayed menu is always accurate and up to date.

Non-Functional Requirements (NFRs):

NFR1 Performance The system should respond to user actions (menu loading, reservation submission) within 2 seconds under normal load.
NFR2 Usability The interface should be intuitive, mobile-friendly, and allow customers to complete a reservation in three steps or fewer.
NFR3 Security Passwords and personal data must be encrypted using secure hashing , and admin routes should require authentication.
NFR4 Reliability The system must prevent double bookings by checking table availability in real time before confirming reservations.
NFR5 Scalability The MERN-based architecture should support future upgrades such as online payments or multi-branch integration without major redesign.

4. System Architecture

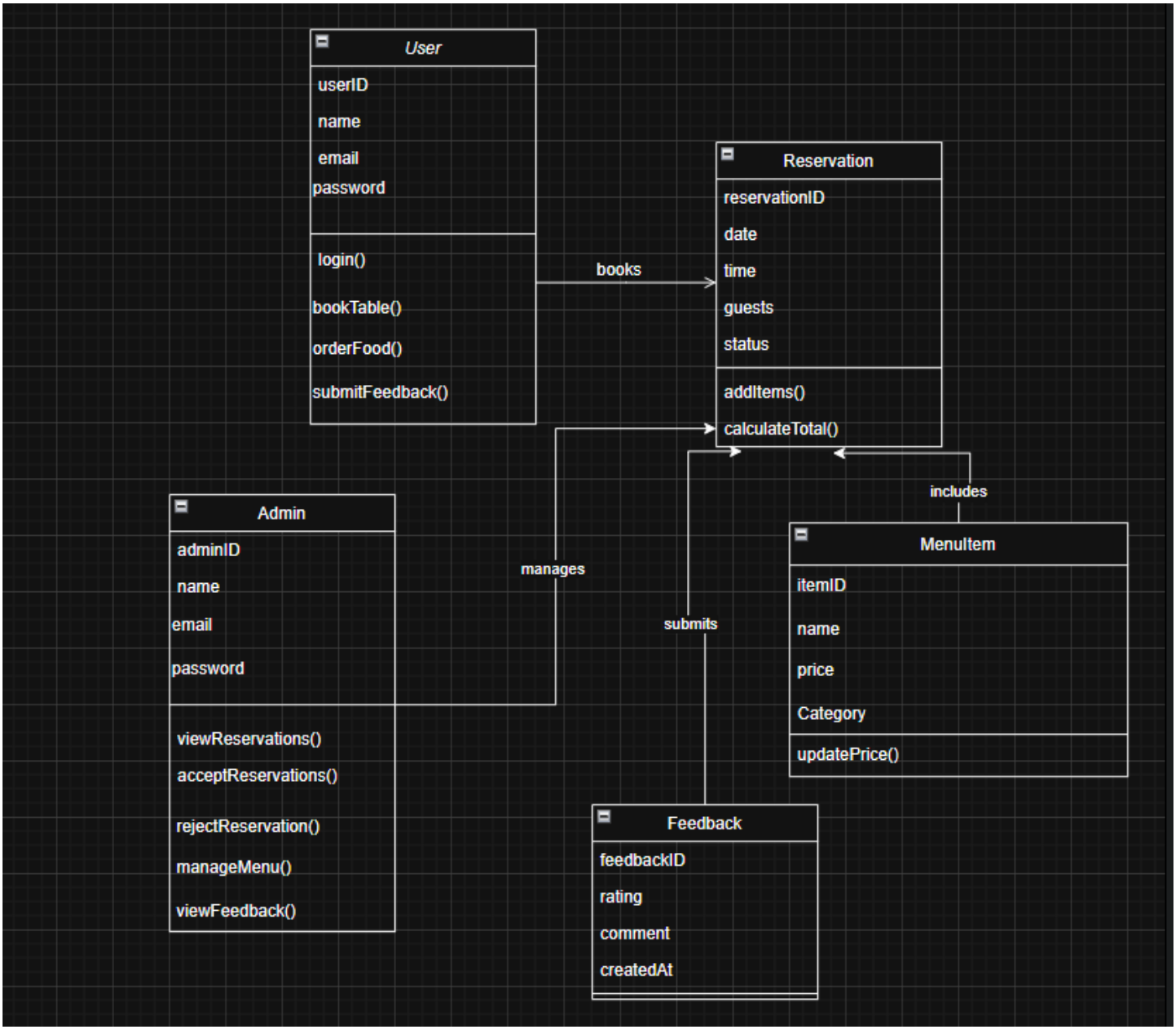
Selected Architecture: Layered Architecture

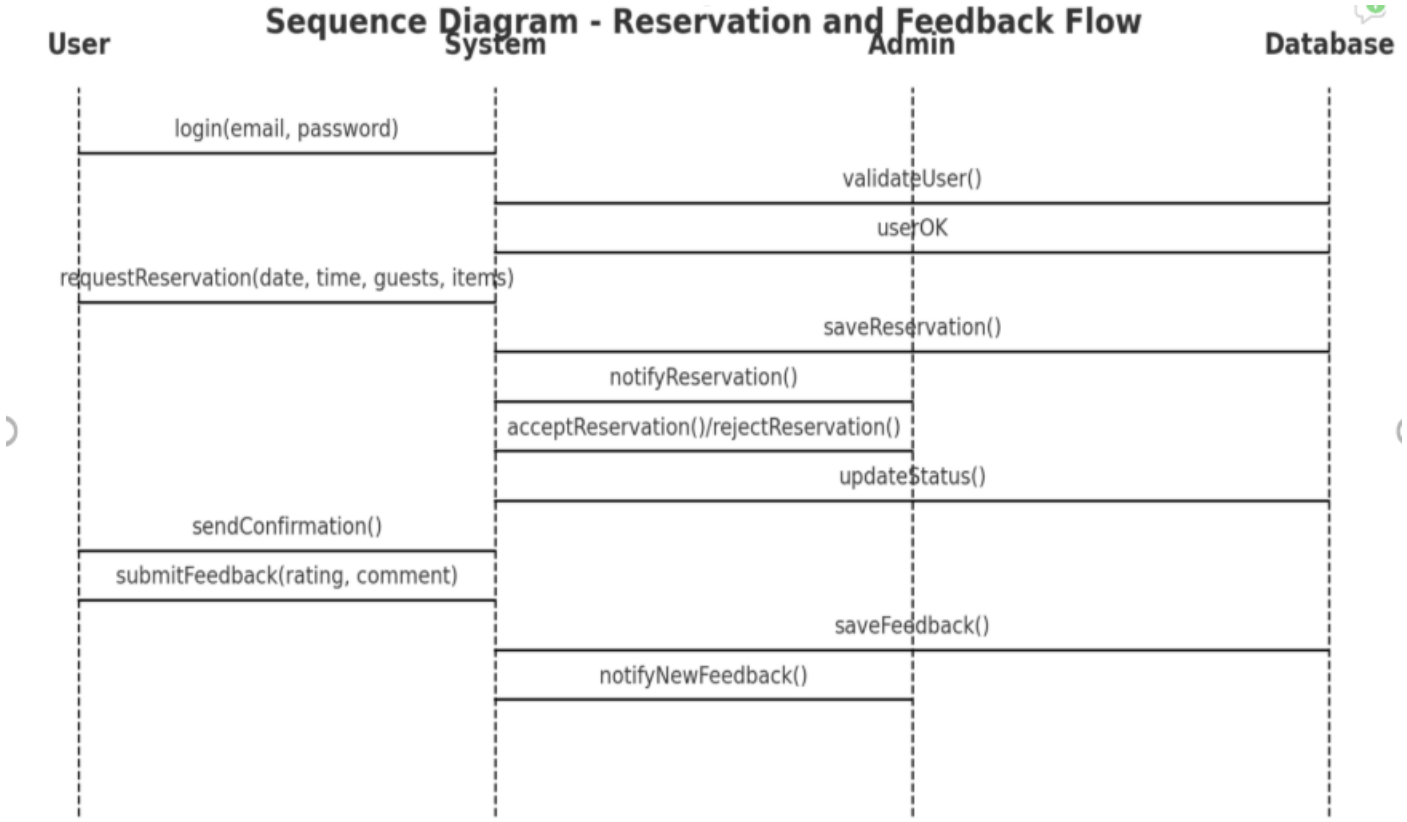
Justification:
The Layered Architecture (also known as the n-tier architecture) is the most suitable choice for our Restaurant Reservation and Management System because it offers clear separation of concerns, scalability, and ease of maintenance, all of which are important for a web-based system built using the MERN stack (MongoDB, Express.js, React.js, Node.js).

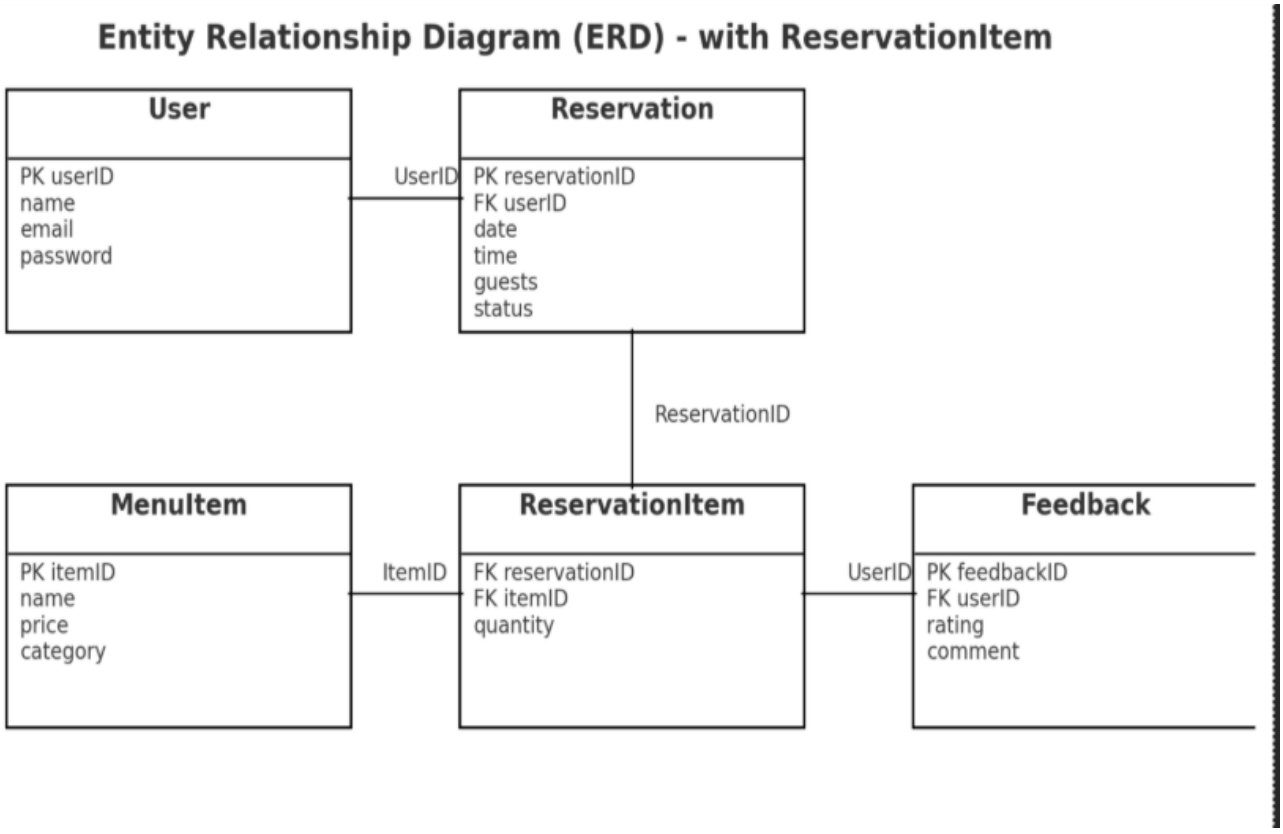
Our system handles multiple responsibilities—user interactions, business logic, and database operations—which can be efficiently organized into distinct layers.

Architecture Overview:
The system will be divided into four main layers:
1. Presentation Layer (Frontend)
2. Application Layer (Backend Logic)
3. Data Access Layer
4. Database Layer

Why we chose Layered Architecture:
- Separation of Concerns: Each layer handles a specific function, making the codebase organized and easier to maintain.
- Scalability: New features (like online payments or multi-branch support) can be added to the application layer without affecting the others.
- Maintainability: Developers can modify or debug one layer independently without disrupting the entire system.
- Security: Sensitive data like user credentials are only handled in backend layers, reducing exposure.
- Reusability: Common services (e.g., authentication, database connections) can be reused across modules.
- Ease of Testing: Each layer can be tested individually using unit and integration tests.


User
userID
name
email
password
login()
bookTable()
orderFood()
submitFeedback()
books
Reservation
reservationID
date
time
guests
status
addItems()
calculateTotal()
includes
Admin
adminID
name
email
password
viewReservations()
acceptReservations()
rejectReservation()
manageMenu()
viewFeedback()
manages
submits
MenuItem
itemID
name
price
Category
updatePrice()
Feedback
feedbackID
rating
comment
createdAt


Sequence Diagram - Reservation and Feedback Flow
User
System
Admin
Database
login(email, password)
validateUser()
userOK
requestReservation(date, time, guests, items)
saveReservation()
notifyReservation()
acceptReservation()/rejectReservation()
updateStatus()
sendConfirmation()
submitFeedback(rating, comment)
saveFeedback()
notifyNewFeedback()


Entity Relationship Diagram (ERD) - with ReservationItem
User
PK userID
name
email
password
UserID
Reservation
PK reservationID
FK userID
date
time
guests
status
ReservationID
MenuItem
PK itemID
name
price
category
ItemID
ReservationItem
FK reservationID
FK itemID
quantity
UserID
Feedback
PK feedbackID
FK userID
rating
comment