



Project Title	Restaurant Management System
Technologies	MERN
Domain	Food
Project Level	Difficult

Table

## Contents

1. Problem Statement:	2
2. Features:	2
2.1. Report Generation:	3
3. Project Evaluation metrics:	3
3.2. Database:	3
3.3. API Details or User Interface:	4
3.4. Deployment:	4
3.5. Solutions Design:	4
3.6. System Architecture:	4
3.7. Optimization of solutions:	4
4. Submission requirements:	



.....

4

4.1. High-level Document: .....  
4

4.2. Low-level document: .....  
4

4.3. Architecture: .....  
4

4.4. Wireframe: .....  
4

4.5. Project code: .....  
5

4.6. Detail project report: .....  
5

4.7. Project demo video: .....  
5

## 1. Problem Statement:

.....

We suggest developing a software project that can effectively handle and manage different restaurant operations, all of which will take place under the administrator's supervision. Restaurant enterprises are currently expanding steadily. The necessity to manage its responsibilities and operations also arises at the same time. The expansion of the company's online presence is the greatest strategy to maximize these operations. The younger generation today favours high-tech services, particularly those provided online. As a result, the project is skilfully created to assist restaurant owners in automating their business processes. With this project, customers' information is maintained in the best possible way, and their needs are met.

## 2. Features:

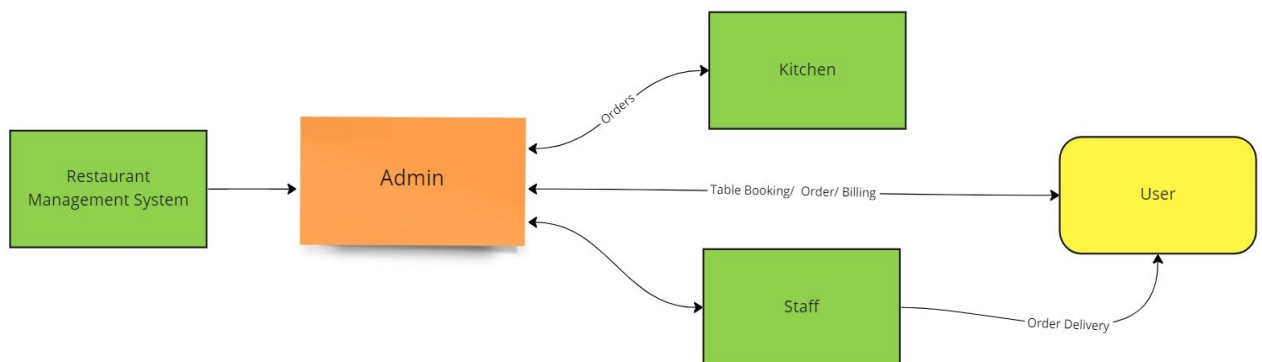
- Admin: The project features an admin login that manages all of the system's online operations. The admin can review and confirm different member details. Even requests to reserve tables can be approved or rejected by him.



- User: By registering themselves, users must first create an account on the system. Once logged in, he or she can access the system's many functions such as booking table, getting bill, check status of order etc.
- Restaurant table booking: The user can pick or deselect tables from the system's available slots using the system's graphical view, which is available to all users. The reservation system is set up so that reserved tables are indicated in red and those that are still available are indicated in their regular color. Admin can use his account to view these bookings and can decide whether to approve or deny the request.
- Kitchen: Kitchen will get orders from users table which is approved by admin.
- Staff: Staff will assign to table by admin. And he will be responsible to deliver food from kitchen to table.

## 2.1. Report Generation:

- Admin can generate all type of reports such as customers order report, kitchens report, billing reports.



miro

## 3. Project Evaluation metrics:

### 3.1. Code:



- You are supposed to write code in a modular fashion • Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works similarly in every environment (operating system).
- You have to maintain your code on GitHub.
- You must keep your GitHub repo public so anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

### **3.2. Database:**

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

### **3.3. API Details or User Interface:**

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

### **3.4. Deployment:**

Deploy the application on your preferred service.

### **3.5. Solutions Design:**

You have to submit complete solution design strategies in High-level Document (HLD), Low-level Document (LLD), and Wireframe documents.

### **3.6. System Architecture:**

You have to submit a system architecture design in your wireframe document and architecture document.

### **3.7. Optimization of solutions:**

Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

## **4. Submission requirements:**

---

### **4.1. High-level Document:**

You have to create a high-level document design for your project. You can reference the HLD form below the link.



Sample link: [HLD Document Link](#)

#### **4.2. Low-level document:**

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: [LLD Document Link](#)

#### **4.3. Architecture:**

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: [Architecture sample link](#)

#### **4.4. Wireframe:**

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: [Wireframe Document Link](#)

#### **4.5. Project code:**

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: [Project code sample link](#)

#### **4.6. Detail project report:**

You have to create a detailed project report and submit that document as per the given sample.

Demo link: [DPR sample link](#)

#### **4.7. Project demo video:**

You have to record a project demo video for at least 5 Minutes and submit that link .