# Restaurant Management System Project Timeline

This document outlines the full timeline for the restaurant management system project using the MERN stack. The timeline includes UI/UX design and other key development phases, distributed over a 12-week period.

## Week 1-2: Planning, UI/UX Design, and Setup

### 1. Requirements Gathering & Finalizing Scope

- Define detailed requirements for each role (user, chef, admin).  
- Establish the key features and functionalities required for each role.

### 2. UI/UX Design

- Wireframing and Mockups:  
 - Use design tools like Figma, Sketch, or Adobe XD.  
 - Design wireframes for:  
 - User (Customer) Flow: QR code scanning, menu viewing, cart, checkout, and payment screens.  
 - Chef Flow: Dashboard for viewing orders, setting timers, and managing order preparation.  
 - Admin Flow: Dashboard for managing orders, viewing reports, and editing menu items.  
- Ensure a seamless user experience with intuitive navigation and accessible design.

### 3. Tech Stack Setup

- Set up project repository using Git/GitHub.  
- Initialize a MERN stack project:  
 - Backend: Set up an Express server.  
 - Frontend: Initialize a React project.  
 - Database: Set up MongoDB and connect to the backend.  
 - Install necessary libraries: `bcryptjs`, `jsonwebtoken`, `cors`, `dotenv`, `mongoose`, etc.

## Week 3-4: User Role Development

### 1. QR Code Scanning and Menu Display

- Implement functionality to scan QR codes using the user's device.  
- Create an API endpoint to fetch menu data from MongoDB.  
- Develop a React component to display the menu items based on QR code scans.

### 2. Cart Functionality

- Build a cart component to allow users to add items from the menu.  
- Include input fields for name, count number, table number, and special requests.  
- Implement state management to handle the cart data.

### 3. Checkout and Payment Integration

- Develop a checkout page to review order details.  
- Integrate a payment gateway (e.g., Stripe) to process payments.  
- Redirect to a confirmation page and display order details after successful payment.

## Week 5-6: Chef Role Development

### 1. Chef Login and Authentication

- Create a login page for chefs.  
- Implement JWT-based authentication and role-based access control.

### 2. Order Management for Chefs

- Build a chef dashboard to display new and ongoing orders.  
- Display order details: menu items, quantity, table number, customer name, and special requests.  
- Implement a feature for chefs to set a preparation timer for each order.  
- Use real-time updates to show countdown timers for ongoing orders.

## Week 7-8: Admin Role Development

### 1. Admin Dashboard

- Create an admin login page with proper authentication.  
- Develop a dashboard to view all orders with filtering options (by day, week, month).

### 2. Data Management

- Implement CRUD functionalities for menu management (adding, updating, deleting menu items).  
- Build features to generate and view billing reports and order history.

## Week 9: Testing and QA

### 1. Unit Testing

- Write unit tests for key components and functionalities using Jest or Mocha.  
- Test API endpoints to ensure data consistency and proper error handling.

### 2. Integration Testing

- Perform end-to-end testing for all user flows (user, chef, admin).  
- Use automated testing tools like Cypress or Selenium for integration tests.

## Week 10: Debugging and Refinement

### 1. Debugging

- Identify and fix bugs discovered during the testing phase.  
- Optimize performance for database queries and API responses.

### 2. User Experience Improvements

- Refine UI/UX based on feedback and testing results.  
- Ensure mobile responsiveness and cross-browser compatibility for all components.

## Week 11: Deployment and Launch

### 1. Prepare for Deployment

- Set up the production environment (using cloud services like AWS, Heroku, or DigitalOcean).  
- Configure environment variables for production.

### 2. Launch

- Deploy the application to the production server.  
- Monitor the application for any post-launch issues and fix them promptly.

## Week 12: Documentation and Maintenance

### 1. Documentation

- Write comprehensive documentation for code, API endpoints, and database schema.  
- Create user manuals for admin and chef roles to guide them in using the system.

### 2. Maintenance and Feature Roadmap

- Plan for regular updates and maintenance schedules.  
- Gather user feedback for potential future enhancements and new features.