# Restaurant Management System

**Project Title:** Restaurant Management System.
**Course:** CSC 478 – Software Engineering Capstone
**Team Name:** Error 404
**Team Members:** Haripriya Koneru, Krishna Kumar Nimmagadda, Venkata Sai Gowtham Panyam, Vansh Parihar, Brijeshkumar Mansukhbhai Raiyani, Pratham Patel, Vijaya Venkateswara Rao Rowthu
**Document:** Project Plan and Umbrella Activities

# 1. Project Plan and Umbrella Activities

## 1.1 Scope Statement

The Restaurant Management System (RMS) is a full-stack web system built using React (Vite) on the frontend and Node.js with MongoDB on the backend. The system assists restaurants in managing their operations digitally by replacing manual processes with structured dashboards.

In Scope:

• Menu management

• Order management

• Role-based authentication

• MongoDB persistent storage

• Deployment on AWS (backend) and Render (frontend)

## 1.2 Tools, Technologies, and Standards

**Frontend:** React, Vite, JavaScript, HTML, CSS

**Backend:** Node.js, Express.js, MongoDB

**Deployment:** Render (Frontend), AWS (Backend), ngrok (Routing)

**Version Control:** Git & GitHub (Repo: Error_404)

**Editors & Tools:** VS Code, Chrome/Edge, Postman

**Standards:** Modular components, meaningful names, requirement IDs (FR-xx, NFR-xx)

## 1.3 Organizational Chart & Roles

**Project Lead:** Pratham Patel

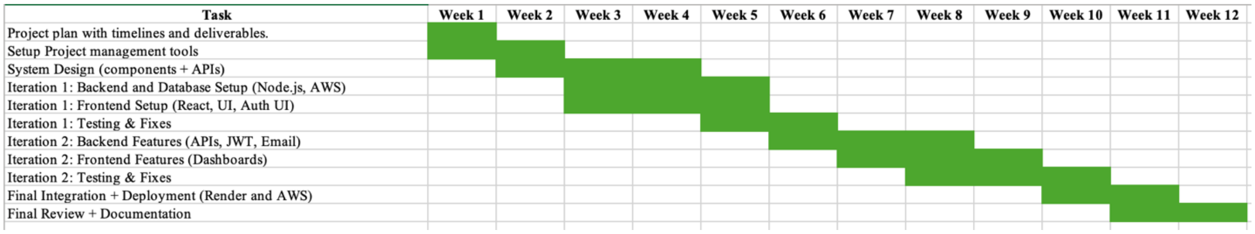
**Developers:** Brijeshkumar Mansukhbhai Raiyani & Venkata Sai Gowtham Panyam

**Deployment:** Hari Priya Koneru

**Documentation:** Vansh Parihar

**Testing:** Krishna Kumar Nimmagadda & Vijaya Venkateswara Rao Rowthu

## 1.4 Gnatt Chart



| Task | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Project plan with timelines and deliverables. | 🟩 | | | | | | | | | | | |
| Setup Project management tools | 🟩 | 🟩 | | | | | | | | | | |
| System Design (components + APIs) | | 🟩 | 🟩 | 🟩 | | | | | | | | |
| Iteration 1: Backend and Database Setup (Node.js, AWS) | | | 🟩 | 🟩 | 🟩 | | | | | | | |
| Iteration 1: Frontend Setup (React, UI, Auth UI) | | | 🟩 | 🟩 | 🟩 | | | | | | | |
| Iteration 1: Testing & Fixes | | | | | 🟩 | 🟩 | | | | | | |
| Iteration 2: Backend Features (APIs, JWT, Email) | | | | | | 🟩 | 🟩 | 🟩 | | | | |
| Iteration 2: Frontend Features (Dashboards) | | | | | | | 🟩 | 🟩 | 🟩 | | | |
| Iteration 2: Testing & Fixes | | | | | | | | 🟩 | 🟩 | 🟩 | | |
| Final Integration + Deployment (Render and AWS) | | | | | | | | | | 🟩 | 🟩 | |
| Final Review + Documentation | | | | | | | | | | | 🟩 | 🟩 |

**Week 1 – Week 2**

- Define project goals
- Identify deliverables
- Establish deadlines and milestones
- Assign tasks and responsibilities to the team

**Week 1 – Week 2**

- Set up GitHub repository
- Create Kanban board / project tracker
- Configure communication tools
- Prepare folder structures and documentation templates

**Week 2 – Week 4**

- Break down UI screens into React components
- Design backend API endpoints
- Define MongoDB schemas (Menu, Orders, Users, Roles)
- Decide interaction flow between frontend and backend

**Week 3 – Week 6**

- Initialize Node.js backend
- Set up Express server
- Configure MongoDB connection
- Deploy backend skeleton to AWS
- Create basic API routes (menu, orders)

**Week 4 – Week 7**

- Initialize React project (Vite)
- Build main layout (navigation, components)
- Build login screen UI
- Build initial pages: menu view, order view
- Connect frontend UI to backend endpoints

**Week 6 – Week 7**

- Test API responses
- Test UI rendering and routing
- Fix bugs in login flow, menu pages, order display
- Refine UI layout and make adjustments

**Week 7 – Week 10**

- Add role-based authentication using JWT
- Expand API functionality: orders, status updates
- Implement email notifications (optional)
- Improve security and validation

**Week 8 – Week 10**

- Manager dashboard (CRUD menu items, assign roles)
- Chef dashboard (view orders, update status)
- Waiter dashboard (place orders, modify orders)
- Improved visuals and navigation

**Week 9 – Week 11**

- Regression testing of all modules
- Functional testing of dashboards
- API testing using Postman
- UI bug fixes and usability improvements

**Week 10 – Week 12**

- Connect frontend (Render) with backend (AWS)
- Validate CORS configuration
- Resolve deployment bugs
- Setup production environment variables

**Week 11 – Week 12**

- Prepare final project report
- Finalize diagrams, documentation, and user manual
- Perform final code cleanup
- Prepare for presentation and submission

## 1.5 Configuration Management Plan

The project is configured primarily by means of Git and GitHub:

- The single main branch is used in the final version of the code in the project.
- Features like manager dash, chef dash and others get created on feature branches and subsequently merged later after review.
- The messages in the commit have a simple yet understandable format e.g.
  adding chef dashboard layout
  managing dropdown manager dashboard
- GitHub repository will serve as a backup point and the source of truth.
- Only proven and successful changes are commit to main source such that the project can be proved at any point in time.

In case a change is bringing issues, Git history enables the team to revert to a stable commit.