Project Title: Hospital Management System (HMS)

Group Name: **Geeky Techs**

Group Members:

1. Yaramala Anusha
2. Rangineni Vishwitha
3. Dinesh Reddy Puthalapattu
4. Chenna Reddy Deepak Kumar Reddy
5. Sai Satya Pavan Pandu Vanka
6. Tejaswi Ganjinaboina
7. Chandrupatla Divya Anusha
8. Jaswanth Makala

Project Description:

Hospital Management System (HMS) project aims to revolutionize patient management and healthcare administration. Our initiative aims to provide a smooth, integrated platform that improves patient care while simultaneously streamlining operational complexity. Essentially, the HMS will function as the digital spine of hospitals, facilitating the effective administration of patient data, scheduling, invoicing, stock, and additional aspects.

Environment for Development:

Programming Languages: Python, which is well-known for being straightforward and reliable, serves as the foundation for our HMS. Python is a great choice for backend programming because of its versatility, which enables the construction of sophisticated algorithms for data administration and analysis.

Web Frameworks: A high-level Python web framework that promotes quick development and simple, practical design, Flask is what we will use. The "light-weight" design of Flasks gives you the tools you need right out of the box, which expedites the development process.

Frontend Development: ReactJS will be used to create a dynamic and responsive user interface. The user experience can be improved by creating huge web apps using this JavaScript framework, where data can update without requiring the page to reload.

Database Management: We will use MySQL as our database system because of its sophisticated features and track record of reliably managing a wide range of data kinds and substantial data volumes.

Hosting and Deployment: To ensure scalability, dependability, and security, our system will be hosted in the cloud on AWS and other similar platforms. Additionally, cloud hosting will enable remote access, enabling anywhere access to the system.

API Integration: We will integrate multiple APIs to offer extra features like calendar management and external data synchronization, guaranteeing smooth communication with other healthcare services and systems.

Features of the Hospital Management System (HMS):

Important attributes and features:

1. **Patient Record Management:** An all-inclusive module for safely keeping and retrieving medical histories, treatment records, and patient data.

2. **Appointment Scheduling:** This feature cuts down on wait times and enhances service delivery by providing a simple interface for patients and physicians to plan, monitor, and handle appointments.

3**. Inventory and Billing Management:** To guarantee accuracy and transparency in financial operations, an integrated system for managing inventory, insurance claims, and billing is used.

4. **Electronic Health Records (EHR):** By implementing EHR, authorized staff can access patient data in real-time, enhancing the standard of treatment.

5. **Telemedicine Services:** By including telemedicine elements that enable remote consultations, healthcare services can now reach a wider audience.

6. **Data Security and Compliance:** To safeguard patient confidentiality, the highest levels of data security and compliance with healthcare laws such as HIPAA must be maintained.

7. **Analytics and Reporting:** State-of-the-art analytics tools to help with informed decision-making by providing insights into patient care, hospital operations, and financial management.

Risk Management Plan:

Top three Risks of our project:

1**. Technical Proficiency:** If team members lack the knowledge of the varied technological stack (Python, Django, ReactJS, and MySQL) needed for the HMS project, there could be a danger. Delays, shoddy implementations, or security flaws could result from this gap.

2**. Integration Complexities:** It might be difficult to integrate different modules, including appointment scheduling, patient information, and billing systems, with external APIs for services like calendar management. Data silos, operational inefficiencies, and a weakened user experience can all be consequences of inadequate integration.

3**. Scope Creep:** Because the HMS project is so ambitious, there is a chance that it will experience scope creep, which is the unwarranted addition of features and requirements that results in resource strain, budget overruns, and project delays.

Strategies for Monitoring and Re-Evaluation:

Technical Expertise:

• **Observation:** Establish code reviews and frequent check-ins to evaluate each team member's proficiency with the required technologies. Promote the use of pair programming as a means of enhancing information exchange.

• **Re-evaluation:** Hold biweekly skill assessments to pinpoint areas that need work and modify training materials appropriately.

Complexities of Integration:

• **Monitoring**: To keep an eye on the smooth operation of various system components, use integration testing and continuous integration/continuous deployment (CI/CD) pipelines.

• **Re-evaluation**: Arrange integration review sessions on a regular basis to assess the state of integration and quickly resolve any problems.

Scope Creep:

• **Monitoring**: Implement a change management procedure whereby each modification request is assessed for how it will affect the project's resources, schedule, and scope.

• **Re-evaluation**: Schedule frequent project review meetings to reevaluate priorities and make sure they are in line with the initial project goals and limitations.

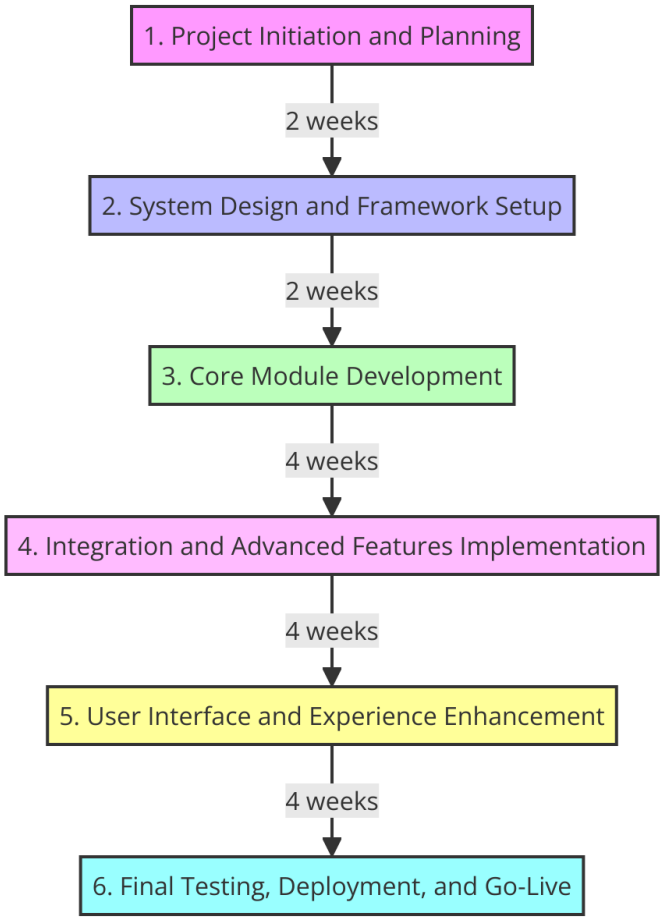
Backup Plans:

1. **Technical Proficiency**: Set aside time for focused training sessions to help team members fill up identified skill gaps.

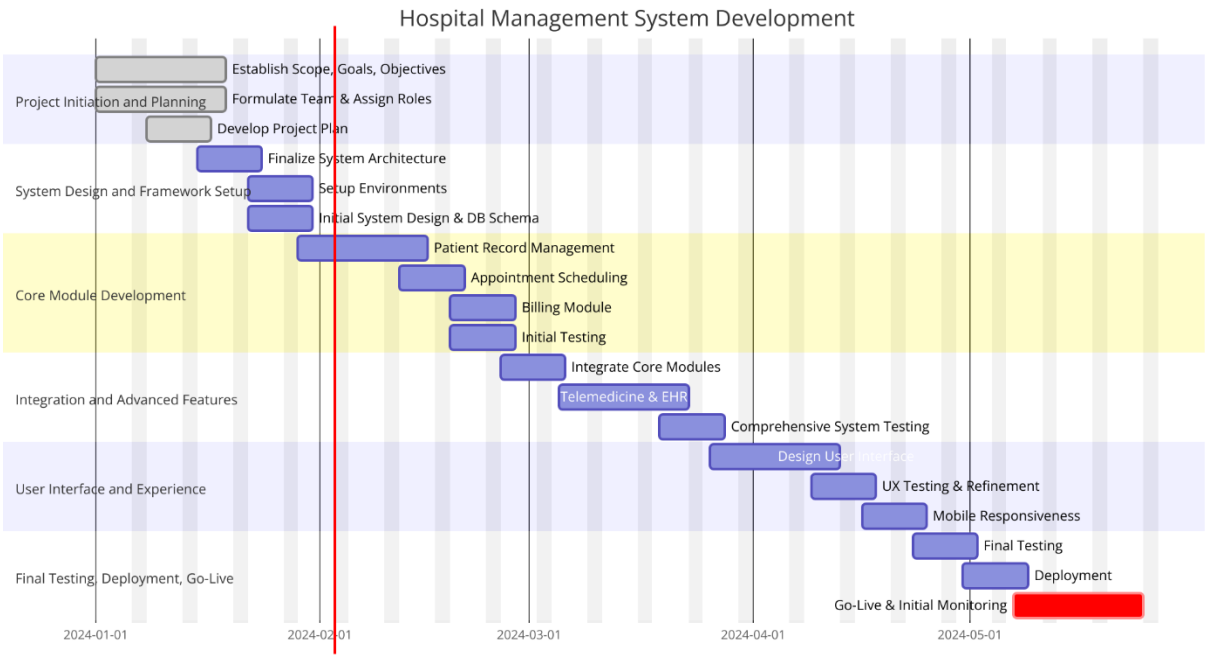
2. **Integration Complexities**: To help with the integration process, create a strong framework that includes established procedures and well-documented information.

Project Planning and Milestones:

The project to implement the Hospital Management System is divided into discrete stages, each with its own deliverables and milestones. The timeline is broken down into multiple months, with important tasks and milestones arranged to make sure the project is moving forward according to plan and in line with its overall goals.


1. Project Initiation and Planning
2 weeks
2. System Design and Framework Setup
2 weeks
3. Core Module Development
4 weeks
4. Integration and Advanced Features Implementation
4 weeks
5. User Interface and Experience Enhancement
4 weeks
6. Final Testing, Deployment, and Go-Live

Pert chart



Gantt chart

Schedule and Stages:

1. Project Planning and Initiation (Weeks 1-2):

• Define the project's objectives, goals, and scope.

• Assign roles to the project team and model it.

• Create the project plan, which should include schedule, and resources.

2. Framework setup and system design (Weeks 3–4):

• Complete the technological stack and system architecture.

• Configure environments for testing, development, and production.

• Produce preliminary database schemas and system design documentation.

3. Development of the Core Module (Weeks 5-8):

• Create modules for billing, scheduling appointments, and managing patient records.

• Put database integration and fundamental functions into practice.

• Carry out preliminary testing for the main modules.

4. Integration and Implementation of Advanced Features (Weeks 9–12):

• Ensure smooth data flow and integrate key elements.

• Create and incorporate cutting-edge features including electronic health records and telemedicine services.

• Carry out thorough system testing, which includes stress and integration testing.

5. Enhancing the User Interface and Experience (Weeks 13–16):

The user interface should be designed and implemented with accessibility and usability in mind.

• Use user experience (UX) testing comments to improve the interface.

• Verify cross-browser compatibility.

6. Deployment, and Final Testing (Weeks 17–20):

• Complete the last phase of testing, which should include performance and security evaluations.

• Set up the system in the operational setting.

• Launch the system and keep an eye on its early use.

Team Roles and Responsibilities

Each member brings unique skills and expertise to the project, contributing to various aspects of development, planning, and execution. The delineation of roles ensures accountability and efficiency, facilitating a structured approach to achieving project milestones.

Name	**Role**
Pavan	Team Lead and helps in all the aspects of the project
Deepak	System Administrative Lead
Anusha	Front end and configuration
Vishwitha	Requirements
Dinesh	Configuration
Tejaswi	Demo, Documentation and Presentation
Jaswanth	Testing
Deepak	Backend

Member Contribution Table:

Name	**Contribution Description**	**Overall Contribution (%)**
Dinesh and Deepak	Project Description	100
Vishwitha	Features	100
Anusha and Divya	Risk Management Plan and Strategies	100
Pavan and Tejaswi	Project Planning and Gantt Chart	100
Jaswanth	Schedule and Stages	100