**Project Roadmap: AfyaFlow – Digital Queue Management System for hospitals.**

**Week 1: Project Planning and Requirements Gathering**

* **Objective**: Define the project's scope, requirements, and goals.
* **Tasks**:
  + Identify key features (e.g., patient registration, queue management, real-time updates, admin dashboard).
  + Define user roles (e.g., patients, doctors, receptionists, admins).
  + Research existing solutions and identify gaps.
  + Create a project plan, and timeline, and assign roles to team members.
  + Set up project management tools (e.g., Jira, Trello, Asana).
* **Deliverables**:
  + Requirements specification document.
  + Market Research report.
  + System Architecture diagram.

**Week 2: Design and Prototyping**

* **Objective**: Create wireframes and design the user interface.
* **Tasks**:
  + Design wireframes for key pages (e.g., patient registration, queue display, admin dashboard).
  + Create a UI/UX design prototype using Figma
  + Finalize the design and create a style guide (colors, fonts, etc.).
* **Deliverables**:
  + Wireframes and UI/UX prototype.
  + Approved design and style guide.

**Week 3: Technology Stack Selection and Setup**

* **Objective**: Choose the tech stack and set up the development environment.
* **Tasks**:
  + Select frontend framework (e.g., React, Angular, or Vue.js).
  + Choose backend framework (e.g., Node.js, Django, or Spring Boot).
  + Decide on a database (e.g., PostgreSQL, MySQL, or MongoDB).
  + Set up version control (e.g., GitHub/GitLab).
  + Set up development environments for the team.
  + Create a basic project structure.
* **Deliverables**:
  + Finalized tech stack.
  + Development environment ready.
  + Basic project structure.

**Week 4: Backend Development (Core Functionality)**

* **Objective**: Build the core backend functionality.
* **Tasks**:
  + Set up the database schema (e.g., tables for patients, queues, doctors).
  + Implement APIs for:
    - Patient registration.
    - Queue management (adding, removing, updating patients).
    - Real-time updates (using WebSockets or similar).
  + Implement user authentication and role-based access control.
  + Write unit tests for backend logic.
* **Deliverables**:
  + Functional backend APIs.
  + Database schema and initial data models.
  + Basic authentication and authorization.

**Week 5: Frontend Development (Core Functionality)**

* **Objective**: Build the core frontend functionality.
* **Tasks**:
  + Develop the patient registration interface.
  + Create a real-time queue display interface.
  + Build the admin dashboard for managing queues and users.
  + Integrate frontend with backend APIs.
  + Implement real-time updates (e.g., using WebSockets or polling).
* **Deliverables**:
  + Functional frontend interfaces.
  + Integration with backend APIs.
  + Real-time updates working.

**Week 6: Testing and Integration**

* **Objective**: Test the application and integrate all components.
* **Tasks**:
  + Perform end-to-end testing of the application.
  + Fix bugs and optimize performance.
  + Integrate third-party tools (e.g., SMS/email notifications for queue updates).
  + Ensure cross-browser and device compatibility.
* **Deliverables**:
  + Tested and bug-free application.
  + UAT feedback and improvements.

**Week 7: Advanced Features and Enhancements**

* **Objective**: Add advanced features and polish the application.
* **Tasks**:
  + Implement additional features (e.g., analytics, reporting, priority queues).
  + Add notifications (SMS, email, or in-app) for queue updates.
  + Optimize the UI/UX based on feedback.
  + Improve security (e.g., data encryption, secure APIs).
* **Deliverables**:
  + Advanced features implemented.
  + Polished and secure application.

**Week 8: Deployment Preparation**

* **Objective**: Prepare the application for deployment.
* **Tasks**:
  + Set up a production environment (e.g., AWS, Azure, or Heroku).
  + Configure the database and server for production.
  + Perform load testing and optimize performance.
  + Set up CI/CD pipelines for automated deployment.
  + Prepare documentation (user manuals, admin guides).
* **Deliverables**:
  + Production-ready application.
  + CI/CD pipelines set up.
  + Documentation completed.

**Week 9: Deployment and Launch**

* **Objective**: Deploy the application and launch it.
* **Tasks**:
  + Deploy the application to the production environment.
  + Conduct final testing in the live environment.
  + Train hospital staff on how to use the system.
  + Monitor the system for any issues post-launch.
  + Gather feedback and plan for future iterations.
* **Deliverables**:
  + Live demo of AfyaFlow platform.
  + Final project report.
  + Presentation slides and pitch deck.

**Post-Launch (Ongoing)**

* Monitor system performance and user feedback.
* Fix any bugs or issues.
* Plan and implement additional features based on feedback.