



**Graphic Era**  
**Hill University**  
 DEHRADUN • BHIMTAL • HALDWANI

## PROJECT AND TEAM INFORMATION

### Project Title

(Try to choose a catchy title. Max 20 words).

*HOSPITAL PATIENT RECORD SYSTEM*

### Student/Team Information

Team Name:	
<b>Team member 1 (Team Lead)</b> Name – Priya Aswal University Roll Number – 2418768 Student ID- 240111299 Email- paswal819@gmail.com	
<b>Team member 2</b> Name – Priyanshi Kimothi University Roll Number – 2418774 Student ID- 24012884 Email- priyanshikimothi111@gmail.com	

**Team member 3**

Name – Naman Kandpal

University Roll Number – 2418672

Student ID-240111611

Email- namankandpal1801@gmail.com

**Team member 4**

Name – Abhinav Singh

University Roll Number – 2418131

Student ID- 240112162

Email- abhinavrajput67814@gmail.com



## PROJECT PROGRESS DESCRIPTION

### Project Abstract

(Brief restatement of your project's main goal. Max 300 words).

The main goal of the Hospital Patient Record System project is to design and develop a secure, efficient, and user-friendly digital platform for managing hospital data. The system aims to replace traditional paper-based or scattered record-keeping methods with a centralized database that enables easy storage, retrieval and updating of patient information. Through this project, hospitals and clinics will be able to register patients, maintain in medical histories, manage appointments, and provide doctors and staff with real-time access to critical information. The system will ensure data privacy through role-based access control, allowing only authorized personnel to view or modify specific information.

### Updated Project Approach and Architecture

(Describe your current approach, including system design, communication protocols, libraries used, etc. Max 300 words).

The project follows a modular and structured approach to ensure systematic development and integration. The frontend is designed using HTML, CSS, and JavaScript, providing responsive forms for patient registration, login, and appointment scheduling. JavaScript handles data validation and enhances user interactivity. The backend is being implemented using Python Flask, which facilitates communication between the frontend and the database through RESTful APIs. Flask manages user authentication, session handling, and CRUD (Create, Read, Update, Delete) operations securely. The database layer uses MySQL, integrated via Flask's connector libraries to store patient information, appointments, and staff details. Data security is maintained through SQL injection prevention and password hashing techniques. For system design, the architecture follows a three-tier model — presentation (UI), application (logic), and data (database) layers — ensuring scalability and maintainability. HTTP/HTTPS protocols are used for communication between client and server. The development environment utilizes VS Code as the IDE and XAMPP/WAMP for local server testing.

## Tasks Completed

(Describe the main tasks that have been assigned and already completed. Max 250 words).

Task Completed	Team Member
<b>Frontend (UI/UX):</b> All primary user interface components, layout designs, and navigation structures are complete. The application is fully navigable and displays static mock data. <b>Backend (API Logic):</b> All necessary business logic and endpoint structures have been defined and coded (e.g., user authentication skeletons, resource processing functions).	Naman Kandpal Priyanshi Kimothi  Abhinav Singh Priya Aswal

## Challenges/Roadblocks

(Describe the challenges that you have faced or are facing so far and how you plan to solve them. Max 300 words).

During the development of the *Hospital Patient Record System*, the team encountered several technical and operational challenges that temporarily affected progress.

One of the primary challenges was backend integration between Flask and MySQL. Establishing a stable database connection and managing data consistency during CRUD operations required additional debugging and configuration. To resolve this, the team reviewed Flask–MySQL documentation and implemented parameterized queries to prevent connection errors and data duplication.

Another challenge involved frontend–backend synchronization. Ensuring smooth communication between the user interface and server-side APIs, especially during form submissions and data retrieval, caused minor delays. This was addressed by testing API endpoints with sample data and refining JavaScript functions for asynchronous data handling. User authentication and role-based access also posed complexity. Designing a secure yet simple login system that distinguishes between admins, doctors, and staff required careful planning. The team adopted password hashing techniques and session-based authentication to ensure data privacy and authorized access.

In terms of project management, coordinating work among multiple team members with varying technical expertise led to occasional delays. Regular virtual meetings, version control practices, and documentation reviews helped maintain consistency.

Lastly, UI responsiveness and cross-browser compatibility required multiple iterations to ensure the system performed well on both desktop and mobile devices. CSS optimization and layout adjustments solved these issues.

Moving forward, the team plans to mitigate such roadblocks by conducting early integration testing, maintaining continuous communication, and documenting each module's functionality. These measures will ensure smoother final deployment and enhance the overall reliability of the system.

## Tasks Pending

(Describe the main tasks that you still need to complete. Max 250 words).

Task Pending	Team Member (to complete the task)
<b>1.Data Persistence Layer (15%)</b> The backend connection with the MySQL database has been established, but final optimization and validation for all CRUD operations (Create, Read, Update, Delete) across user and patient modules are still pending.	Abhinav and Priya
<b>2.Backend-Frontend Integration (15%)</b> Integration of the completed frontend components with the backend REST API endpoints is ongoing. The team needs to finalize asynchronous data handling, error messages, and response validation to ensure smooth functionality.	Naman and Priyanshi (frontend), Abhinav and Priya (backend support)
<b>3.Authentication Flow (5%)</b> The sign-up, login, and session management system is developed but requires final end-to-end testing to confirm role-based access (admin, doctor, staff).	Abhinav and Priya
<b>4.End-to-End Testing (5%)</b> Full system testing, including validation of data flow between modules, will be conducted to ensure stability and eliminate integration bugs.	Entire Team

## Project Outcome/Deliverables

(Describe what are the key outcomes / deliverables of the project. Max 200 words).

The *Hospital Patient Record System* will deliver a fully functional, secure, and user-friendly web-based application designed to simplify healthcare data management. The primary outcome is an integrated digital platform that enables hospitals and clinics to efficiently store, retrieve, and manage patient information, replacing traditional paper-based systems.

Key deliverables include:

- A **responsive web interface** allowing doctors, nurses, and administrative staff to register patients, view medical history, and schedule appointments from any device.
- A **secure authentication system** with role-based access control to ensure that only authorized users can access or modify records.
- A **backend database (MySQL)** for structured storage of patient demographics, prescriptions, test results, and appointment details.
- **CRUD functionality (Create, Read, Update, Delete)** to manage records dynamically.
- **System documentation**, including database schema, deployment guide, and user manual to facilitate easy setup and usage.

## Progress Overview

(Summarize how much of the project is done, what's behind schedule, what's ahead of schedule. Max 200 words.)

The *Hospital Patient Record System* project is currently progressing as per the planned milestones. The **planning and design phase** has been successfully completed, including requirement analysis, feature selection, and database schema design. The **frontend development** using HTML, CSS, and JavaScript is nearly complete, with responsive forms for patient registration, login, and appointment scheduling already functional.

The **backend development** using Python Flask is underway. Basic routes for user authentication and CRUD operations have been implemented, and integration with the **MySQL database** has been partially completed. Data storage and retrieval functionalities are operational, while work on advanced modules like role-based access control and appointment tracking is ongoing.

Testing of initial modules (login and record management) has begun to ensure stable connectivity between frontend, backend, and database layers.

Although the integration phase is slightly behind schedule due to debugging and configuration adjustments, the overall project remains on track. The **documentation and deployment** phases are expected to be completed ahead of schedule. The team is maintaining steady progress, ensuring that the final system will be ready for testing and presentation within the designated timeframe.

## Codebase Information

(Repository link, branch, and information about important commits.)

### Repository Link:

<https://github.com/namankandpal801-svg/Hospital-Patient-Record-System>

The project's source code is maintained on GitHub for version control, collaboration, and continuous updates. The repository hosts all modules, including frontend files, backend scripts, and the database schema.

### Branch Structure:

- **main branch:** Contains the stable, tested, and deployable version of the system.
- **development branch:** Used for integrating new features, performing module-level testing, and resolving bugs before merging into the main branch.

### Key Commits:

- **Initial Commit:** Created the project structure and added basic HTML/CSS templates for the user interface.
- **Commit #2:** Integrated the Flask backend with MySQL database connectivity.
- **Commit #3:** Implemented user authentication and CRUD functionality for patient records.
- **Commit #4:** Added appointment scheduling and enhanced input validation using JavaScript.
- **Commit #5:** Completed testing, fixed security vulnerabilities, and finalized deployment setup.

The project's source code is maintained on GitHub for version control, collaboration, and continuous updates. The repository hosts all modules, including frontend files, backend scripts, and the database schema.

Testing and Validation Status

(Provide information about any tests conducted)

Test Type	Status (Pass/Fail)	Notes
Unit Testing	Pass	Individual modules such as login, registration, and CRUD operations were tested successfully.
Integration Testing	In Progress	Minor issues found during frontend-backend data synchronization; fixes underway.
System Testing	Pass	End-to-end workflow validated; data flows correctly between database, backend, and UI.
Security Testing	Pass	Password hashing and SQL injection prevention verified for data protection.

Deliverables Progress

(Summarize the current status of all key project deliverables mentioned earlier. Indicate whether each deliverable is completed, in progress, or pending.)

<p>The development of the <i>Hospital Patient Record System</i> is progressing systematically, with most core deliverables nearing completion.</p> <ul style="list-style-type: none"><li><b>Frontend Interface: Completed.</b> The responsive web interface using HTML, CSS, and JavaScript is fully developed, featuring patient registration, login, and appointment scheduling forms.</li><li><b>Backend Development: In Progress.</b> The Flask-based backend is partially completed. User authentication, CRUD operations, and database connectivity have been implemented; integration of role-based access and error handling is ongoing.</li><li><b>Database Integration (MySQL): Completed.</b> The database schema is finalized with structured tables for patients, appointments, and staff. Connectivity with the backend has been tested successfully.</li><li><b>System Security Features: In Progress.</b> Password hashing and SQL injection prevention have been implemented. Session management and access control mechanisms are under testing.</li><li><b>Testing and Debugging: In Progress.</b> Unit testing for individual modules has begun; end-to-end integration testing is scheduled next.</li><li><b>Documentation and Deployment Guide: Pending.</b> The documentation structure has been drafted and will be finalized after testing is complete.</li></ul>
--

