Edoc-Online Doctor Appointment

and

Medical Record System

MINOR PROJECT REPORT

Submitted by

NITESH KUMAR (24MCI10231)

PRIYANSHU (24MCI10182)

DEV PATHAK (23MCI10155)

in partial fulfillment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

IN

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Chandigarh University, India.

CERTIFICATE OF ACCEPTANCE

Certified that this project report **"Edoc-Doctor appointment and report management System"** is the bonafide work of "**NITESH KUMAR, PRIYANSHU, and DEV PATHAK"** who carried out the project work under my/our supervision. This project work is hereby accepted by the Department of Computer Applications, University Institute of Computing, Chandigarh University, India, in partial fulfillment of the requirements for the award of Degree in Master of Computer Application.

SIGNATURE

Dr. Krishan Tuli

HEAD OF THE DEPARTMENT

SIGNATURE

Dr. Mandeep Singh

SUPERVISOR

ACKNOWLEDGEMENT

We would like to express my heartfelt gratitude to all those who assisted us in preparing and completing our Major Project report titled "Edoc-Doctor appointment and report management System" We extend my sincere appreciation and thanks to Dr. Mandeep Singh for inspiring, cooperating, and encouraging us to pursue my work. With his valuable suggestions and guidance, I was able to complete my work in a short time frame at this institution. We also convey our respect and gratitude to Dr. Krishan Tuli, (HOD) University Institute of Computing, and Chandigarh University, as well as all the faculty members, for their valuable suggestions and guidance towards completing the report. Furthermore, We are grateful to all our academic mentors who provided direct or indirect assistance in successfully completing our major project report. Last but not least, we are immensely grateful to our parents, who served as a constant source of encouragement and inspiration throughout the completion of this work at Chandigarh University, Mohali.

Nitesh Kumar (24MCI10231)

Priyanshu (24MCI10182)

Dev Pathak (24MCI10231)

TABLE OF CONTENTS

ABSTRACT

booking doctor appointments and managing medical records through an efficient and user-friendly digital platform. The system provides a smooth interface for patients to schedule appointments, upload medical reports, and receive prescriptions directly from doctors. It also enables doctors to manage their schedules, access patient details, and provide treatment recommendations in real time.

The platform offers **secure and reliable communication** between doctors and patients through authenticated login access, ensuring privacy and data protection. By integrating modern web technologies, EDOC enhances the efficiency of healthcare services, reducing waiting times and minimizing manual record handling.

This project aims to **digitize and streamline healthcare management**, making it easier for patients to access medical services and for doctors to

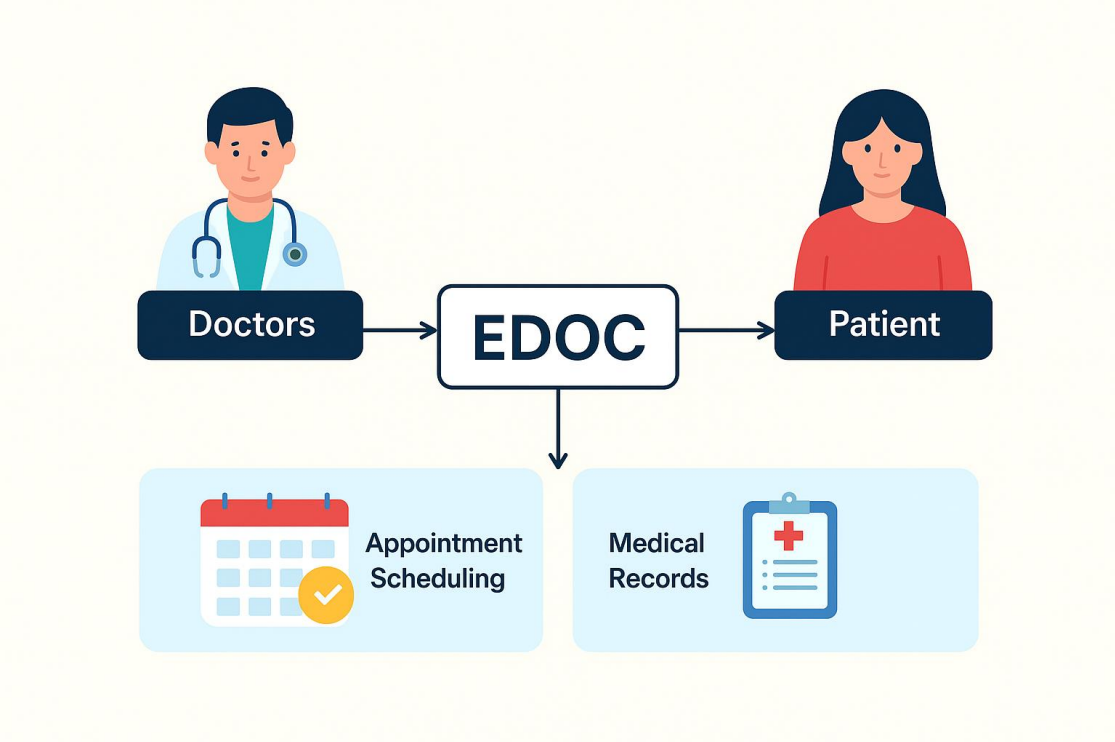
manage appointments and reports effectively. Through its innovative features, **EDOC transforms traditional healthcare interactions into a smart, organized, and accessible experience for both patients and doctors.**

CHAPTER 1

INTRODUCTION

1.1 Project Scope and Planning

In today's digital era, technology has become a driving force in improving healthcare efficiency, accessibility, and reliability. The traditional system of booking doctor appointments and managing medical records is often time-consuming and unorganized. To address these issues, we introduce "EDOC – Doctor Appointment and Report Management System", a web-based application that simplifies the process of booking appointments, managing patient reports, and improving communication between doctors and patients.
This system provides a user-friendly interface that allows patients to schedule appointments, upload reports, and receive prescriptions online, while doctors can efficiently manage their schedules, view patient records, and update reports. The platform's main objective is to create a smart healthcare management system that bridges the gap between doctors and patients using secure and efficient digital technology.



Key Highlights :

User-Centric Design :

EDOC features an intuitive and visually clean design that ensures ease of use for both doctors and

patients. The interface supports simple navigation for booking appointments, viewing prescriptions, and managing patient records, ensuring an effortless experience.
Efficient Appointment Management
The system enables patients to view available doctors, book appointments according to specialization and time slots, and receive confirmation in real-time. Doctors can manage their appointments, avoid scheduling conflicts, and provide online consultations.

Digital Record Maintenance

All patient records, including prescriptions, reports, and appointment histories, are stored securely in the database. This allows doctors and patients to access medical information anytime, reducing paperwork and ensuring data accuracy.
Secure Communication and Notifications
EDOC provides a secure platform for communication between doctors and patients. Real-time notifications are sent for appointment confirmations, updates, and prescription uploads.

Objectives

- Develop a web-based healthcare management system to simplify appointment scheduling and medical report handling.

- Provide secure login and authentication for both patients and doctors.
- Enable prescription upload and view feature for digital medical records.
- Implement real-time notification and status tracking for appointments.
- Ensure data security, reliability, and scalability using modern technologies.

Features

Patient Module:

- Patient registration and login
- Book doctor appointments
- Upload and view medical reports
- Receive prescriptions from doctors
- View appointment history and status

Doctor Module:

- Doctor login and dashboard
- Manage patient appointments
- Upload prescriptions and view medical records
- Track patient history and feedback

Admin Module:

- Manage users (patients/doctors)
- View system reports and logs
- Monitor data security and performance

Project Planning

Requirements Gathering:
Conducted meetings and discussions with healthcare professionals and patients to identify real-world problems in traditional appointment systems. Created use-case diagrams and user stories to document both functional and non-functional requirements.

Project Scheduling
Used MVC (Model-View-Controller) architecture for development. The project was divided into smaller modules such as user authentication, appointment management, and prescription handling. Each phase was assigned a clear deadline and testing period.

Risk Management
Identified potential risks such as server downtime, data breaches, and system errors. Implemented database backups, encryption techniques, and regular testing to ensure smooth operation and data safety throughout the project lifecycle.

1.2 Client Identification and Recognition of Need

The EDOC System primarily targets patients, doctors, and healthcare administrators who require a convenient and secure platform to manage medical appointments and records.
Client Identification

1. **Patients:** Individuals who seek an easy way to book doctor appointments, upload medical reports, and receive prescriptions digitally.
2. **Doctors:** Healthcare professionals who want to manage appointments efficiently and access patient data quickly.
3. **Hospitals/Clinics:** Institutions that aim to digitalize their appointment systems and maintain an organized database for better patient management.

Collaboration and Feedback

Continuous communication with end-users and stakeholders was maintained throughout the project lifecycle. Regular feedback sessions ensured that the application evolved according to user needs and expectations.

1.3 Recognition and Knowledge of Relevant Contemporary Issues

Developing the EDOC System required addressing several contemporary issues in healthcare

technology:

1. **Data Privacy and Security**
   - *Issue:* Handling sensitive patient information requires strict data protection measures.
   - *Solution:* Implemented encrypted data storage, secure authentication, and compliance with healthcare privacy standards.
2. **System Reliability and Availability**
   - *Issue:* Healthcare systems must be accessible 24/7 without technical interruptions.
   - *Solution:* Ensured server reliability, load balancing, and regular maintenance.

1.4 Task Identification

The project development was divided into several key tasks:

1. Project Planning and Setup: Defined scope, objectives, and timelines.
2. Requirement Analysis: Collected user and system requirements from stakeholders.
3. Database Design: Designed relational schemas for patient, doctor, and appointment data.
4. Front-end Development: Built user interfaces for login, dashboard, and appointment management.

1.5 Technologies Used

Front-end Technologies
- HTML5, CSS3, JavaScript
- Bootstrap / Tailwind CSS for responsive design

Back-end Technologies
- PHP / Java for server-side scripting
- MySQL Database for data storage and management

Additional Tools
- XAMPP Server
- Visual Studio Code / NetBeans IDE

1.6 Timeline of the Project

Phase	Description	Duration
Phase 1	Requirement Gathering and Analysis	2 Weeks
Phase 2	Database and UI Design	3 Weeks
Phase 3	Back-end Development	4 Weeks
Phase 4	Integration and Testing	1 Week
Phase 5	Deployment and Final Evaluation	1 Week

CHAPTER 2

DESIGN FLOW / PROCESS

2.1 Evaluation & Selection of Features

The EDOC – Doctor Appointment and Report Management System is designed to provide a smooth, secure, and efficient digital healthcare experience. Its features are carefully selected to ensure a balance between usability, performance, and data security. The goal is to make healthcare services easily accessible to patients and help doctors manage appointments and records efficiently.

1. User-Centric Feature Selection

- **User Profiles & Authentication:** The system allows patients and doctors to create and manage profiles securely. Patients can store personal and medical details, while doctors can maintain professional profiles including specialization, qualifications, and available timings. User authentication and role-based access control ensure privacy and authorized access.
- **Appointment Booking System:** Patients can search for doctors by specialization, availability, and location, then book appointments conveniently. The system prevents scheduling conflicts and sends automatic confirmations and reminders.
- **Digital Prescription Management:** Doctors can create, upload, and share prescriptions directly with patients through their dashboard. Patients can view, download, or print prescriptions anytime.
- **Medical Report Handling:** Patients can upload test results and medical reports for doctor review, making follow-ups and diagnoses more efficient. Doctors can also attach reports with prescriptions for record-keeping.

2. Real-Time Updates and Notifications

- The system uses automated notifications to inform patients about appointment confirmations, prescription uploads, and upcoming visits.
- Doctors receive alerts about new bookings, cancellations, and report submissions.
- Notifications ensure timely communication and reduce manual follow-ups.

3.Security and Data Protection Features

- Secure Login & Role-Based Access: Only verified users (patients/doctors) can log in, ensuring confidentiality.
- Encrypted Data Transmission: All medical data, prescriptions, and reports are encrypted

to maintain privacy.
- Backup and Recovery: Regular database backups protect against data loss or system failures.

4. Admin Monitoring and Analytics

- The admin module monitors system activities, manages user records, and generates analytical reports on appointments, doctor performance, and patient interactions.
- Admins can view platform usage statistics and ensure system reliability.

2.2 Navigation Menu

The EDOC System includes a clear and intuitive navigation structure to ensure ease of use for every user type.

For Patients:
- Dashboard: Overview of booked appointments, notifications, and reports.
- Book Appointment: View doctors by specialization, select date/time, and confirm booking.
- My Reports: Upload and view medical reports.
- Prescriptions: Access prescriptions provided by doctors.
- Profile: Manage personal information and medical history.
- Logout: End session securely.

For Doctors:
- Dashboard: Overview of daily appointments and patient requests.

For Admin:
- User Management: View, edit, or remove patient and doctor accounts.
- Appointments Overview: Track ongoing and completed appointments.
- Reports Monitoring: View uploaded medical records.

2.3 Backend Design

The backend of the EDOC System handles data processing, authentication, scheduling, and secure communication. It follows an MVC (Model-View-Controller) architecture to ensure modularity and maintainability.

1. User Authentication and Authorization

- Secure registration and login for patients, doctors, and admin.
- Role-based access ensures users only access relevant modules.
- Uses hashed passwords and session management for security.

2. User Profile Management

- Patients and doctors can update their information through controlled access.
- Profile data is validated to ensure accuracy and integrity.
- Doctors' specialization and patient records are linked for efficient retrieval.

3. Appointment Scheduling and Management

- Allows real-time scheduling, updates, and cancellation of appointments.
- Prevents overlapping appointments using conflict-checking logic.
- Tracks appointment status (pending, confirmed, completed).

4. Prescription and Report Handling

- Doctors can upload prescriptions after consultations.
- Patients can view and download prescriptions anytime.
- Uploaded reports are linked to patient IDs for easy access.

5. Notifications and Alerts

- Automated email/SMS notifications for appointment status and prescription updates.
- Admin and doctors get system alerts for new activities or issues.

6. Data Security and Backup

- Implements SSL/TLS encryption for data exchange.
- Daily backups ensure recovery in case of database corruption.
- Uses access logs to track all user actions for security auditing.

2.4 Technologies Used in Design

Front-End:
- HTML5, CSS3, JavaScript – For interactive and responsive UI
- Bootstrap/Tailwind – For styling and layout responsiveness

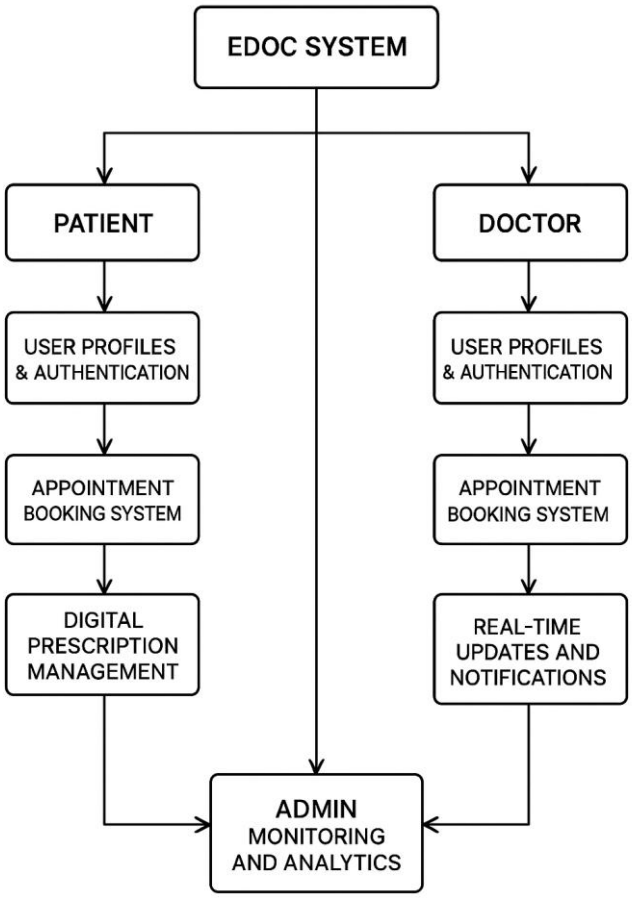
Back-End:
- PHP – Server-side scripting
- MySQL – Database for storing all user and appointment data

Additional Tools:
- XAMPP Server – Local development and testing
- VS Code / NetBeans IDE – Code development

2.5 System Flow Diagram

The system flow consists of the following steps:

1. **User Registration/Login → Secure authentication for all users.**
2. **Patient Dashboard → Appointment booking and report upload.**
3. **Doctor Dashboard → View appointments, upload prescriptions.**
4. **Admin Dashboard → System management and monitoring.**
5. **Database Interaction → Store, retrieve, and update all records.**
6. **Notification Service → Send alerts and confirmations to users.**



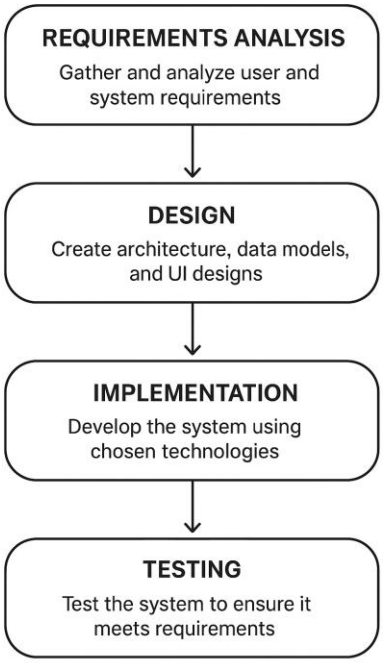
CHAPTER-3

THEORY AND DESIGN METHODOLOGY

3.1 Consistency and Standards

Consistency and standards are vital in ensuring a smooth, reliable, and efficient user experience for EDOC – Doctor Appointment and Report Management System. By following well-established design principles, developers can create a cohesive, intuitive, and trustworthy platform that simplifies healthcare interactions for both doctors and patients.

In EDOC, maintaining consistent user flows for common actions—such as booking an appointment, viewing prescriptions, or uploading medical reports—is key to ensuring a user-friendly experience. When users know what to expect from the interface, they can focus on managing their healthcare needs without confusion or delay.



Visual Design

A consistent visual design is essential for building clarity and trust. EDOC uses a clean, professional color scheme—typically blue and white—to symbolize healthcare reliability and calmness. Clear typography, consistent button styles, and recognizable icons contribute to a polished and accessible interface.

Adhering to standard UI/UX guidelines like Material Design for Android and Bootstrap standards for the web ensures the platform looks and feels uniform across devices. These design standards improve

usability and accessibility, ensuring a consistent experience for every user.

Brand Identity

EDOC maintains a strong and professional brand identity that reflects trust, care, and efficiency in the healthcare domain. The logo, color palette, and consistent tone of communication reinforce EDOC's mission of providing digital healthcare solutions that are reliable and secure.
All visuals, messages, and notifications maintain this tone—creating a sense of dependability and professionalism that aligns with healthcare service expectations.

User Experience (UX)

EDOC's UX design focuses on simplicity, clarity, and accessibility. Consistent layouts and intuitive navigation paths reduce user confusion and cognitive load.
For example:

- Patients can easily navigate from the Dashboard → Appointments → Reports → Prescriptions with minimal clicks.
- Buttons and icons maintain uniform meanings across modules.
- Error messages and confirmations are presented clearly with descriptive feedback.

By standardizing these UX components, users quickly become familiar with the system, leading to smoother operation and increased satisfaction.

Content and Data Architecture

In EDOC, data architecture is organized logically to simplify navigation and ensure secure handling of sensitive medical data.
Key sections such as Appointments, Reports, Patients, Doctors, and Prescriptions are structured with clear hierarchies. For example:

- Patients can view their past and upcoming appointments under one tab.
- Doctors can access patient medical histories before consultations.
- Administrators can generate summary reports with one click.

This logical categorization allows users to find what they need quickly while maintaining system clarity and efficiency.

Functionality and Features

EDOC standardizes functionality to deliver consistent performance across all modules.
Key features include:

- Doctor and Patient Registration: Secure signup and authentication.
- Appointment Scheduling: Real-time availability updates and confirmation notifications.
- Prescription Management: Doctors can upload prescriptions that appear instantly on the patient's dashboard.
- Medical Report Uploads: Secure cloud-based storage for patient reports.
- Notifications: Automated reminders for upcoming appointments or new prescriptions.

Each feature behaves consistently across pages to ensure users can interact confidently and efficiently.

Performance and Reliability

Performance and reliability are central to EDOC's usability. The system ensures that each function responds quickly and accurately.

- Pages load efficiently, even during peak usage.
- Data synchronization between doctor and patient portals is handled in real time.
- Secure APIs ensure instant communication between frontend and backend.

Robust error handling ensures stability—for example, if a network issue interrupts an appointment booking, users receive a clear notification and can retry without losing their data.

Accessibility

Accessibility is a core priority in EDOC's design. The interface adheres to WCAG (Web Content Accessibility Guidelines) to ensure that users with visual, auditory, or mobility challenges can use the system comfortably.

Features such as clear font contrast, keyboard navigation, and descriptive tooltips make EDOC inclusive and usable by a broad range of users.

3.2 Simplicity and Clarity

In healthcare applications like EDOC, simplicity and clarity are essential. Patients and doctors must interact quickly and efficiently without facing unnecessary complexity. The system's goal is to make appointment booking, report management, and prescription viewing simple, intuitive, and reliable.

Understanding User Needs

Before designing EDOC, user needs were carefully analyzed:

- Patients need quick access to available doctors, easy appointment scheduling, and a secure area to view medical reports.
- Doctors require organized patient data, prescription tools, and appointment management systems.
- Admins must manage records, users, and system data efficiently.

By understanding these requirements, EDOC ensures that each user type has a clear, dedicated workflow aligned with their purpose.

Simplicity in Design

Simplicity means reducing unnecessary complexity. EDOC follows a minimalist interface with only the most essential options visible at each stage.

Key principles applied:

- Clear Navigation: Direct menus for Dashboard, Appointments, Patients, Reports, and Prescriptions.
- Minimalistic Layout: Clean white backgrounds, uniform typography, and clear spacing.
- Consistent Visuals: Icons, buttons, and colors follow uniform design rules.

- Focused Content: Only relevant information (doctor name, specialization, date, time) is shown per screen.
- Streamlined Interactions: Booking and uploading actions require minimal steps.

This ensures users complete their tasks efficiently without confusion or visual overload.

Clarity in Communication

Clarity is vital for ensuring that users understand every system response.

- Clear labels and error messages (e.g., “Invalid Email Address” instead of generic “Error”).
- Feedback indicators confirm successful actions such as “Appointment Booked Successfully.”
- Progress indicators show when data is being processed.
- Help Section and FAQs assist users with step-by-step guides.

This transparent communication fosters confidence and reduces user frustration.

Usability Testing and Iteration

After development, EDOC undergoes regular usability testing with real users (patients and doctors). Testing helps identify:

- Confusing layouts or unclear labels.
- Performance or navigation delays.
- Suggestions for additional improvements.

Through iterative design, feedback is incorporated to refine workflows and enhance efficiency. Continuous testing ensures the app remains user-friendly and up-to-date with real-world needs.

Conclusion

In conclusion, simplicity and clarity are fundamental to the success of EDOC. By focusing on clean design, easy navigation, and consistent communication, EDOC delivers a seamless experience that simplifies healthcare management.

The combination of strong UX standards, robust security, and reliable performance ensures that EDOC – Doctor Appointment and Report Management System stands out as a trustworthy and user-friendly healthcare solution.

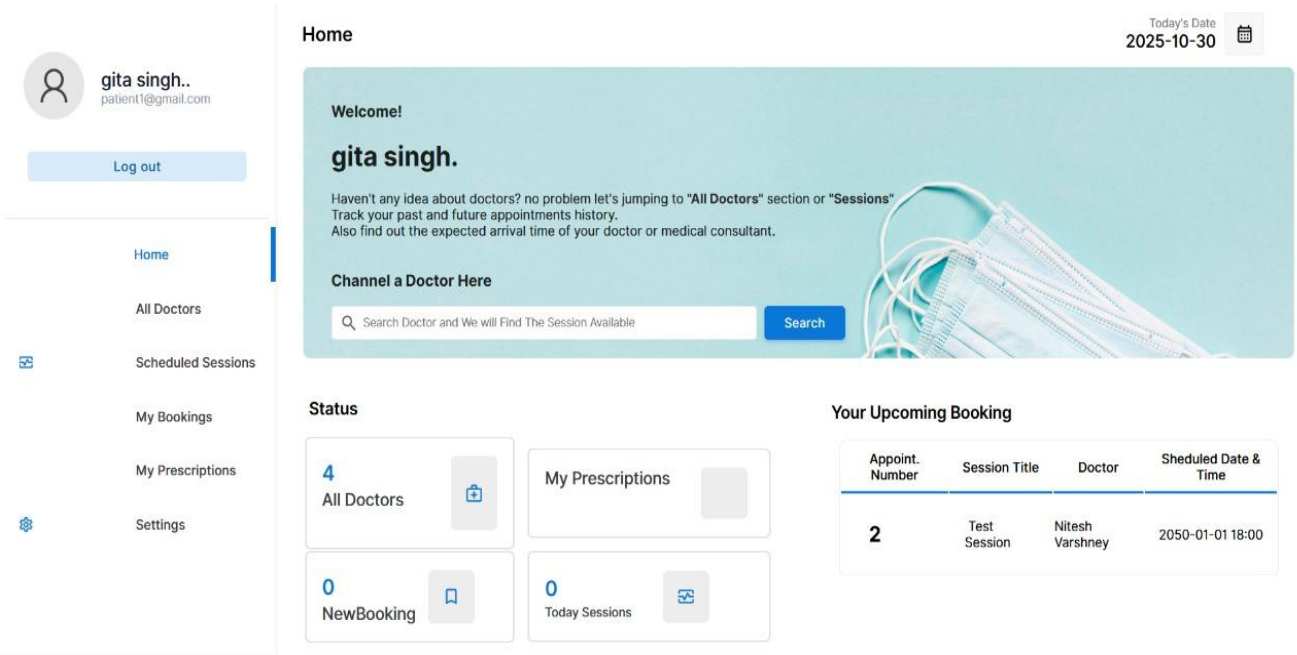
CHAPTER 4

RESULTS AND PERFORMANCE ANALYSIS

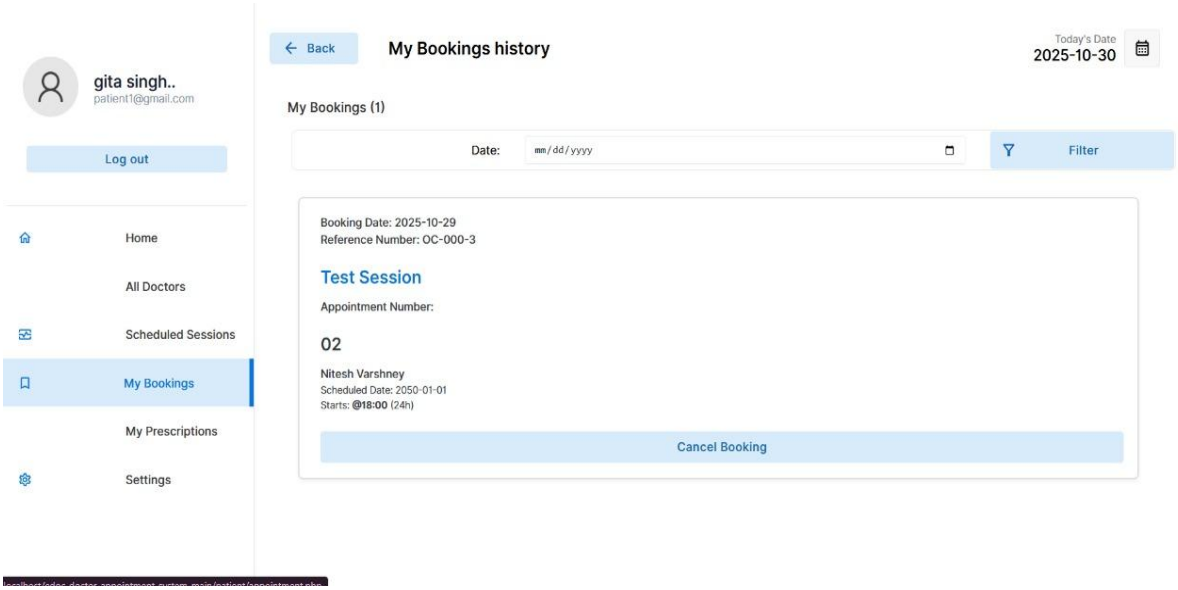
User Interface Layout

This layout provides an overview of the user dashboards—including the Patient Dashboard, Doctor Dashboard, and Admin Panel. Each dashboard is designed to offer a user-friendly experience with clear navigation and accessibility features.

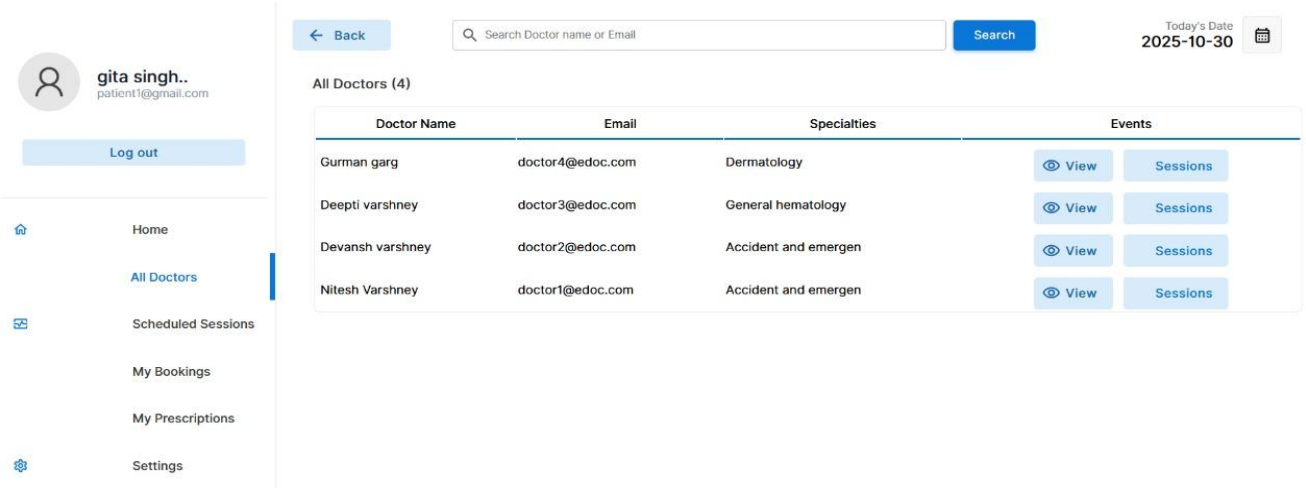
Patient UI



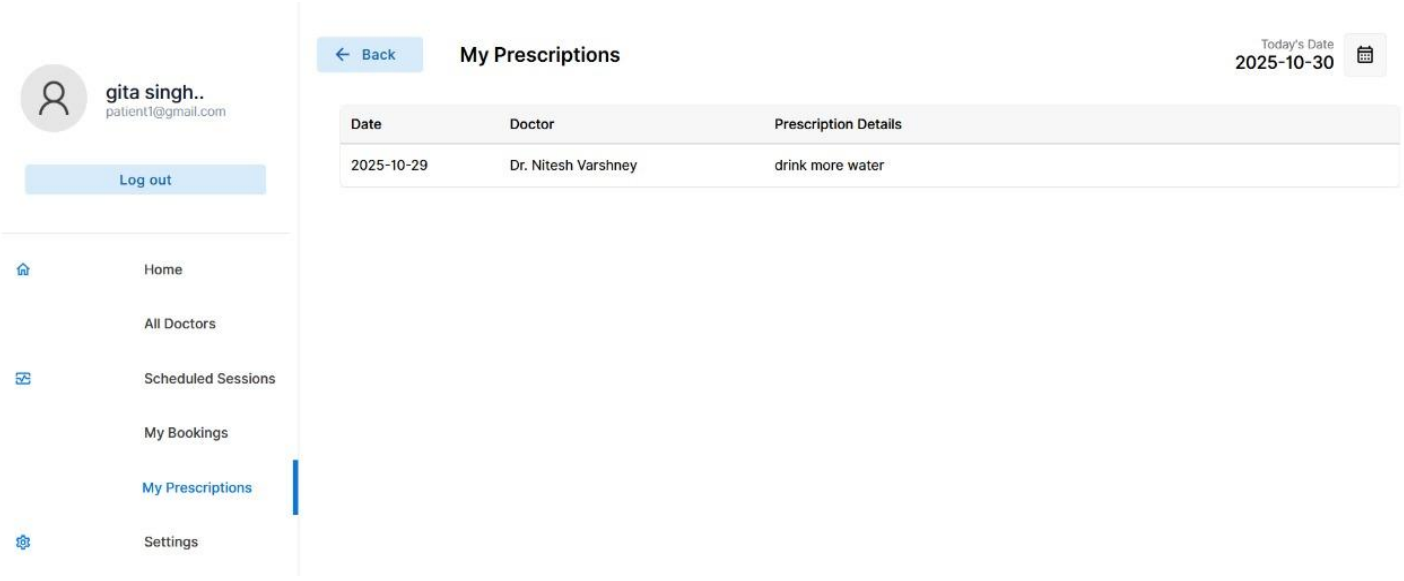
Patient Booking



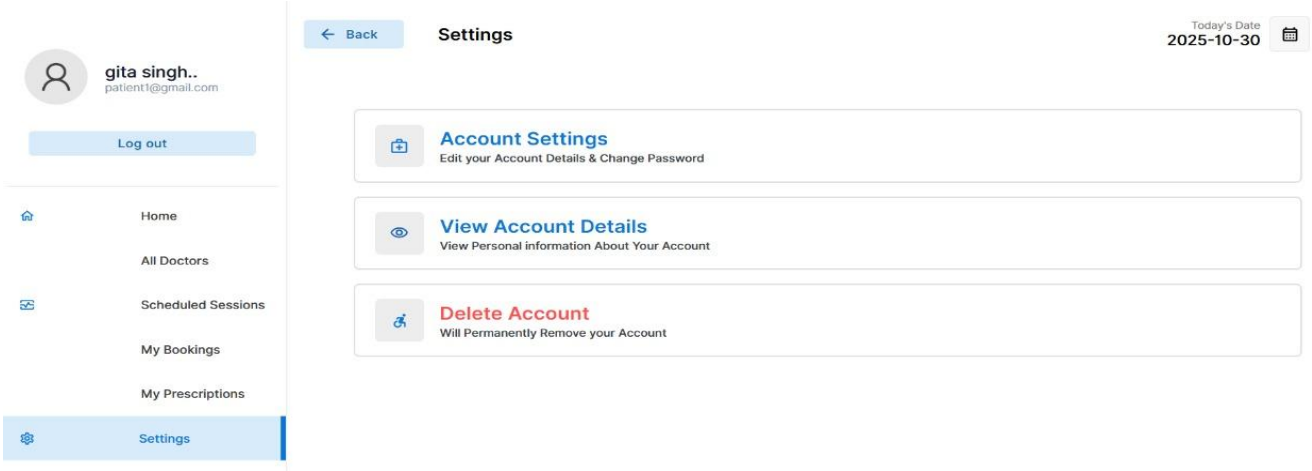
Doctor’s List



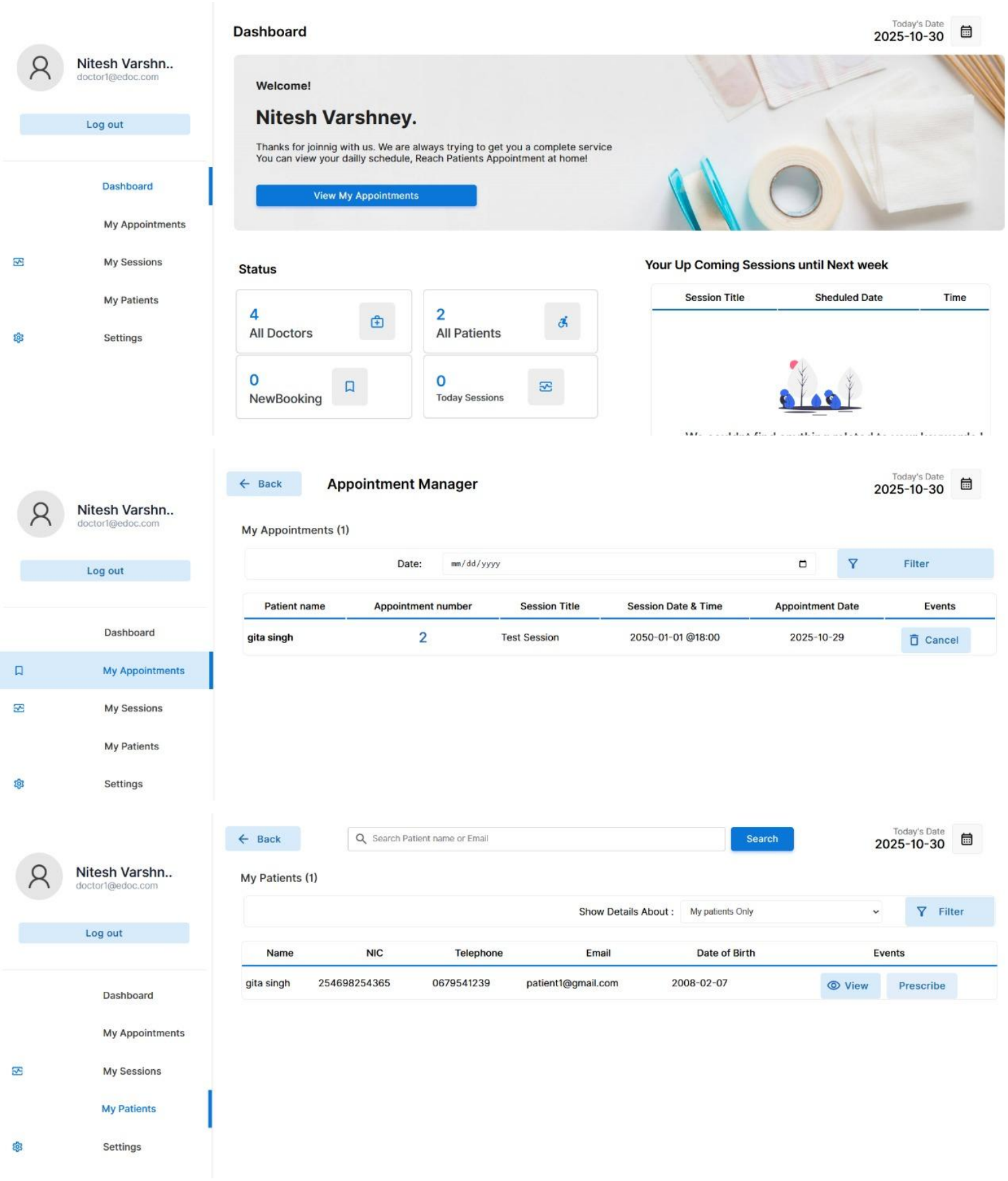
Prescriptions



Accound Settings



Doctor's UI


Nitesh Varshn..
doctor1@edoc.com
Log out
Dashboard
My Appointments
My Sessions
My Patients
Settings
Dashboard
Today's Date
2025-10-30
Welcome!
Nitesh Varshney.
Thanks for joinnig with us. We are always trying to get you a complete service
You can view your dailly schedule, Reach Patients Appointment at home!
View My Appointments
Status
4
All Doctors
2
All Patients
0
NewBooking
0
Today Sessions
Your Up Coming Sessions until Next week
Session Title
Sheduled Date
Time
Nitesh Varshn..
doctor1@edoc.com
Log out
Dashboard
My Appointments
My Sessions
My Patients
Settings
Back
Appointment Manager
Today's Date
2025-10-30
My Appointments (1)
Date:
mm/dd/yyyy
Filter
Patient name
Appointment number
Session Title
Session Date & Time
Appointment Date
Events
gita singh
2
Test Session
2050-01-01 @18:00
2025-10-29
Cancel
Nitesh Varshn..
doctor1@edoc.com
Log out
Dashboard
My Appointments
My Sessions
My Patients
Settings
Back
Search Patient name or Email
Search
Today's Date
2025-10-30
My Patients (1)
Show Details About :
My patients Only
Filter
Name
NIC
Telephone
Email
Date of Birth
Events
gita singh
254698254365
0679541239
patient1@gmail.com
2008-02-07
View
Prescribe

4.1 Technology Roadmap

Developing an innovation roadmap for EDOC – Doctor Appointment and Report Management System involves outlining the essential stages, technologies, and methodologies required to design, develop, deploy, and maintain the system effectively. The roadmap ensures a clear direction from concept to completion while maintaining performance, reliability, and security standards expected from a healthcare-oriented system.

Define Goals and Objectives

1. **Primary Goals:**
    - To create a seamless digital healthcare platform for appointment scheduling, medical report management, and prescription viewing.
    - To enhance doctor–patient communication through secure and efficient digital interactions.
    - To digitize the manual healthcare record process, improving accessibility and reducing errors.
2. **Key Performance Indicators (KPIs):**
    - Appointment booking success rate.
    - System uptime and response time.
    - User satisfaction and engagement levels.
    - Data security compliance (privacy, encryption, and access control).

Market Research and Analysis

1. Platform Selection:
   The system is designed as a web-based application for universal access via desktop or mobile browsers, ensuring cross-platform compatibility.
2. Technology Stack:

- Frontend: HTML5, CSS3, JavaScript, Bootstrap (for responsive UI design).
- Backend: Java (Jakarta EE) with MySQL for database management.
- Server: Apache Tomcat for deployment and RESTful API integration.
- Security: AES encryption for data confidentiality and SSL for secure communication.
- Cloud Storage (optional): Integration with Google Cloud or AWS S3 for storing reports and prescriptions.

Feature Prioritization

The core features of EDOC are prioritized based on user requirements and implementation feasibility for the MVP (Minimum Viable Product):

- Doctor and Patient Registration/Login
- Appointment Scheduling and Management
- Digital Prescription Upload and Viewing
- Medical Report Upload and Download
- Notification System for Appointment Updates and Prescriptions
- Admin Dashboard for Record Management

Future releases may include features like AI-based health recommendations, teleconsultation, and EHR (Electronic Health Record) integration**.**

User Interface (UI) and User Experience (UX) Design

The UI/UX design of EDOC emphasizes simplicity, accessibility, and trustworthiness.

- Consistent Layouts: Common design elements such as buttons, icons, and color themes provide visual harmony.
- Color Scheme: A medical-themed palette (white and blue) promotes a professional and calm environment.
- Usability Testing: Conducted iteratively to ensure smooth navigation for patients, doctors, and administrators.
- Accessibility Features: Designed to accommodate users with limited technical skills or

disabilities.

Integration of Core and Third-party Services

1. Secure Communication:
   Implement AES encryption for secure data exchange between doctors and patients.
2. Database Management:
   MySQL is used for structured data storage, ensuring data integrity and relational consistency.
3. Notifications:
   Email or in-app notifications inform users about appointment confirmations, prescription updates, and report availability.
4. Cloud Integration:
   Future versions may use cloud APIs for report backup, scalability, and reliability.

Testing and Quality Assurance

Quality assurance ensures that EDOC meets its functional and non-functional requirements.

- Unit Testing: Each module (login, booking, reports) tested individually.
- Integration Testing: Ensures modules interact correctly.
- System Testing: Verifies complete workflow (from registration to report management).
- User Acceptance Testing (UAT): Conducted with sample users (patients/doctors) to validate usability and efficiency.

Automated testing tools and manual review cycles guarantee the platform remains error-free and stable before deployment.

Deployment and Launch

After rigorous testing, EDOC is deployed on Apache Tomcat and made accessible through web browsers.
Deployment Steps:

1. Configure Tomcat server and JDBC connections.
2. Deploy .war file to the server.
3. Perform post-deployment verification.
4. Release the system for use within the healthcare environment or pilot group.

Marketing or internal awareness initiatives may be implemented to introduce the platform to hospitals or clinics.

Post-launch Support and Maintenance

Ongoing maintenance includes:

- Bug fixes and performance enhancements.
- Periodic data backup and database optimization.
- Updating features based on user feedback.
- Ensuring compatibility with evolving healthcare regulations and technologies.

A continuous improvement approach ensures long-term sustainability of the system.

Scaling Strategies

To accommodate future growth and high user traffic, scalability is integrated into the design:

- Modular Architecture: Each component (appointment, reports, user management) is independent and scalable.
- Database Optimization: Indexing, normalization, and caching enhance query performance.
- Cloud Readiness: System designed to migrate easily to cloud infrastructure when scaling is needed.
- API Design: RESTful APIs facilitate integration with external healthcare services and future mobile applications.

Data Analytics and Business Intelligence

Analytics tools will help administrators track and analyze system usage, appointment

trends, and patient feedback.

- Metrics: Number of active users, appointment frequencies, and doctor performance.
- Tools: Google Analytics or custom dashboards integrated with MySQL reports. Insights gathered are used for data-driven decision-making and service improvement.

Customer Support and Feedback Mechanisms

A built-in support module allows users to raise queries or report issues directly from their dashboard.

- Support Channels: Email and in-app ticket system.
- Feedback Collection: Rating and comment forms after appointments.
- Improvement Cycle: Regular analysis of feedback to identify and fix recurring usability or technical problems.

Offline and Accessibility Features

While EDOC primarily functions online, limited offline features may include:

- Local caching of recent prescriptions and appointment history.
- Offline form filling, which syncs automatically once the internet connection is restored.

Accessibility standards (WCAG) ensure EDOC is usable by users with different abilities.

Future Enhancements

- AI-based Health Insights: Suggest preventive health check-ups based on patient history.
- Chatbot Support: For instant help and appointment guidance.
- Telemedicine Integration: Allow patients to consult doctors through video conferencing.
- Mobile App Extension: A dedicated Android/iOS app using Flutter or React Native.

4.3 Scalability Planning for Future Growth

Scalability ensures that EDOC can accommodate increased users, data volume, and functionality over time without compromising system performance or reliability**.**

System Architecture Design

EDOC follows a modular and layered architecture:

- Presentation Layer: Handles UI (web-based interface).
- Business Logic Layer: Contains appointment, report, and prescription services.
- Data Access Layer: Manages database interactions with MySQL.

This structure allows independent scaling of components, enhancing flexibility and maintainability.

Cloud Infrastructure

Although initially hosted locally or on-premises, EDOC is designed for cloud deployment using services like AWS, Azure, or Google Cloud Platform (GCP). Cloud hosting offers:

- Auto-scaling for increasing workloads.
- Load balancing for high traffic.
- Continuous uptime with managed services.

Database Scalability

To handle growing healthcare data:

- MySQL Partitioning and Indexing: Enhances query speed and performance.
- Read Replicas: Improve load distribution for simultaneous access.
- Backup Strategies: Scheduled backups prevent data loss.

Caching and Content Delivery

Static assets (CSS, JS, images, reports) can be cached via a Content Delivery Network (CDN) to reduce load times and server overhead.

In-memory caching mechanisms (e.g., Redis) may be introduced to store frequently accessed data like appointment lists**.**

Monitoring and Performance Optimization

Real-time monitoring tools (such as Prometheus or Firebase Performance Monitoring) will track:

- Server performance (CPU/memory usage).
- Database query efficiency.
- Response latency and uptime.

Regular optimization of code and queries will maintain high-speed performance even as the system expands.

Auto-scaling and Load Balancing

If deployed on cloud platforms:

- Auto-scaling policies automatically add or remove server instances based on traffic load.
- Load balancers distribute user requests evenly to prevent downtime and ensure reliability.

Disaster Recovery and Backup

To ensure high availability and reliability:

- Multiple server backups are maintained.
- Automated disaster recovery ensures minimum downtime in case of server or network failures.
- Encrypted data backups are stored securely on cloud storage.

Performance Testing and Benchmarking

Performance testing will simulate real-world loads to identify bottlenecks and ensure stability.

Tools such as JMeter or Loader.io can be used to perform:

- Load testing
- Stress testing
- Response time measurement

This ensures the system remains responsive under high traffic conditions.

DevOps and Continuous Integration (CI/CD)

Adopting DevOps practices ensures faster and more reliable updates.

- Version Control: Git for managing source code changes.
- CI/CD Pipelines: Automated deployment via Jenkins or GitHub Actions.
- Testing Automation: Continuous testing before every deployment cycle.

These practices enhance productivity and ensure consistent software delivery.

.

CHAPTER 5

CONCLUSION

5.1. Developing Edoc – Doctor Appointment and Report Management System has been a transformative experience aimed at simplifying and digitalizing healthcare management. Our primary goal was not just to create a platform for online appointments but to build a comprehensive healthcare ecosystem that connects doctors, patients, and medical records seamlessly.

Throughout the development journey, our focus remained on user-centric design, reliability, and data security, ensuring that both patients and doctors can interact effortlessly. By integrating modern technologies and secure database management, Edoc offers patients an easy way to book appointments, upload medical reports, and access prescriptions, while doctors can manage their schedules, update patient information, and issue digital prescriptions with minimal effort.

From the beginning, our vision was to create a bridge between patients and healthcare professionals—a system that saves time, reduces paperwork, and promotes efficient communication. The system is designed to handle various real-world challenges such as appointment conflicts, medical record organization, and secure prescription sharing, ensuring that healthcare becomes more accessible and efficient for everyone.

Security and privacy were key priorities during development. We implemented robust authentication mechanisms, role-based access control, and encrypted data storage to protect sensitive medical information. The system's scalable architecture also ensures that it can handle increasing user demands and future expansions without compromising performance.

Team collaboration played a vital role in this project. Each member—Nitesh Kumar, Priyanshu, and Dev Pathak—contributed their technical expertise, creativity, and dedication, resulting in a solution that aligns with the needs of modern digital healthcare.

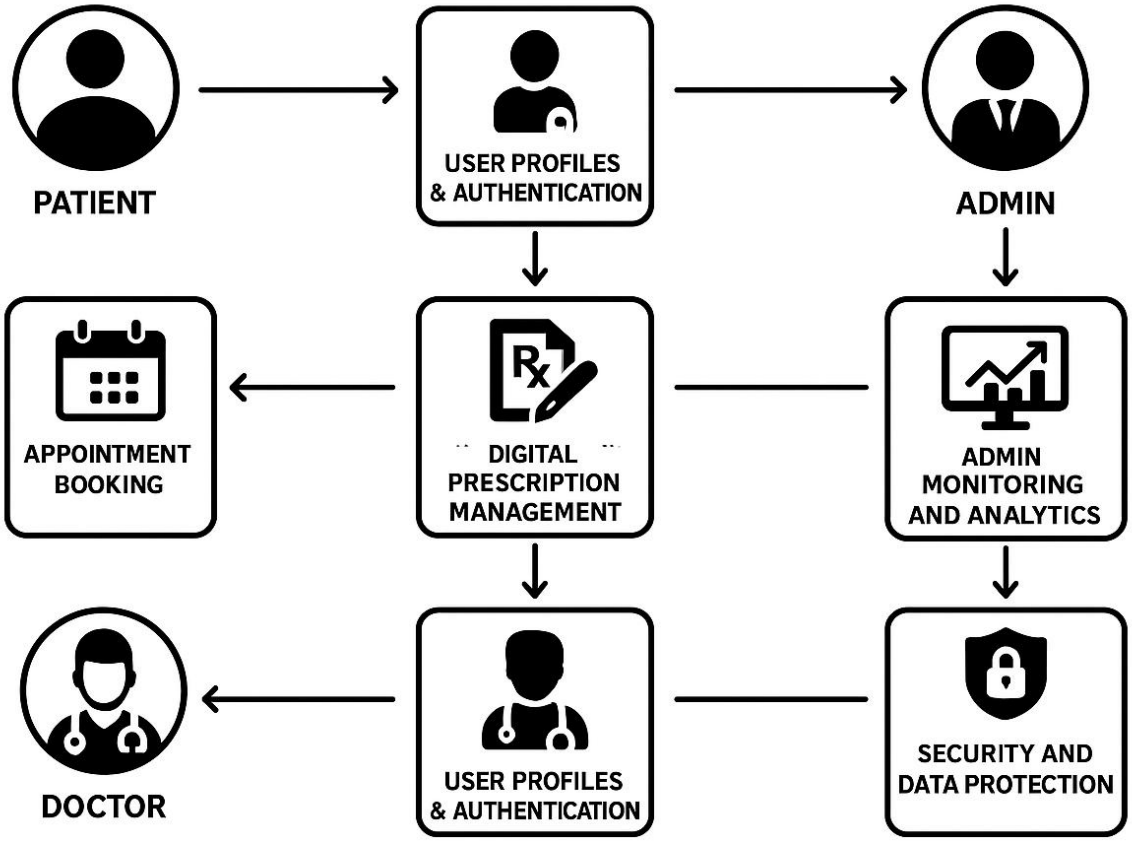
In conclusion, Edoc stands as a step forward in transforming traditional healthcare systems into smart, secure, and efficient digital solutions. It demonstrates how technology can improve doctor-patient interactions, streamline medical record management, and promote a healthier, more connected future. As we continue to refine and enhance the system, Edoc will evolve into a powerful healthcare management platform that redefines convenience and reliability in the medical field.

5.3Future Planning

To further enhance and expand the capabilities of Edoc, our team has outlined several future developments and improvements:

1. Mobile Application Development:
   Build a dedicated Android and iOS app to make appointment booking, report access, and prescription viewing easier for patients and doctors on the go.
2. AI-Based Health Prediction:
   Integrate AI models to analyze patient health records and provide early disease predictions, personalized health tips, and preventive care suggestions.
3. Video Consultation Feature:
   Implement a secure video calling module for online consultations, allowing doctors to attend patients remotely, especially in rural or emergency scenarios.
4. Smart Prescription and Medicine Reminders:
   Add an automated medicine reminder system that notifies patients about their medication schedule and upcoming appointments via SMS or app notifications.
5. Integration with Pharmacies and Labs:
   Enable direct integration with pharmacies for prescription orders and diagnostic labs for report uploads, creating a connected healthcare ecosystem.
6. Advanced Analytics Dashboard:
   Develop data visualization tools for doctors and admins to monitor appointment trends, patient feedback, and treatment outcomes for better decision-making.
7. Blockchain for Medical Data Security:
   Explore blockchain technology for secure, transparent, and tamper-proof storage of patient medical histories and transaction records.
8. Multilingual and Accessibility Support:
   Introduce multiple language options and accessibility features to ensure ease of use for patients from diverse backgrounds and abilities.
9. Machine Learning for Appointment Optimization:
   Use ML algorithms to predict peak booking times, suggest available slots, and reduce wait times through intelligent scheduling.

By pursuing these enhancements, Edoc aims to become a complete digital healthcare assistant—providing secure, smart, and user-friendly healthcare management for both doctors and patients. With continuous innovation and user feedback, our team envisions Edoc evolving into a nationwide healthcare platform that simplifies access to quality medical servic



Code of the project:

Login.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/animations.css">
    <link rel="stylesheet" href="css/main.css">
    <link rel="stylesheet" href="css/login.css">
```

```
    <title>Login</title>



</head>
<body>
    <?php

    //learn from w3schools.com
    //Unset all the server side variables

    session_start();

    $_SESSION["user"]="";
    $_SESSION["usertype"]="";

    // Set the new timezone
    date_default_timezone_set('Asia/Kolkata');
    $date = date('Y-m-d');

    $_SESSION["date"]=$date;


    //import database
    include("connection.php");




    if($_POST){

        $email=$_POST['useremail'];
        $password=$_POST['userpassword'];

        $error='<label for="promter" class="form-label"></label>';

        $result= $database->query("select * from webuser where email='$email'");
        if($result->num_rows==1){
            $utype=$result->fetch_assoc()['usertype'];
            if ($utype=='p'){
                //TODO
                $checker = $database->query("select * from patient where pemail='$email' and
ppassword='$password'");
                if ($checker->num_rows==1){


                    //   Patient dashbord
                    $_SESSION['user']=$email;
```

```
                    $_SESSION['usertype']='p';

                    header('location: patient/index.php');

                }else{
                    $error='<label for="promter" class="form-label" style="color:rgb(255, 62,
62);text-align:center;">Wrong credentials: Invalid email or password</label>';
                }

            }elseif($utype=='a'){
                //TODO
                $checker = $database->query("select * from admin where aemail='$email' and
apassword='$password'");
                if ($checker->num_rows==1){


                    //   Admin dashbord
                    $_SESSION['user']=$email;
                    $_SESSION['usertype']='a';

                    header('location: admin/index.php');

                }else{
                    $error='<label for="promter" class="form-label" style="color:rgb(255, 62,
62);text-align:center;">Wrong credentials: Invalid email or password</label>';
                }


            }elseif($utype=='d'){
                //TODO
                $checker = $database->query("select * from doctor where docemail='$email' and
docpassword='$password'");
                if ($checker->num_rows==1){


                    //   doctor dashbord
                    $_SESSION['user']=$email;
                    $_SESSION['usertype']='d';
                    header('location: doctor/index.php');

                }else{
                    $error='<label for="promter" class="form-label" style="color:rgb(255, 62,
62);text-align:center;">Wrong credentials: Invalid email or password</label>';
                }

            }

        }else{
            $error='<label for="promter" class="form-label" style="color:rgb(255, 62, 62);text-
align:center;">We cant found any acount for this email.</label>';
```

```
        }









    }else{
        $error='<label for="promter" class="form-label"> </label>';
    }

    ?>




    <center>
    <div class="container">
        <table border="0" style="margin: 0;padding: 0;width: 60%;">
            <tr>
                <td>
                    <p class="header-text">Welcome Back!</p>
                </td>
            </tr>
        <div class="form-body">
            <tr>
                <td>
                    <p class="sub-text">Login with your details to continue</p>
                </td>
            </tr>
            <tr>
                <form action="" method="POST" >
                <td class="label-td">
                    <label for="useremail" class="form-label">Email: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td">
                    <input type="email" name="useremail" class="input-text" placeholder="Email
Address" required>
                </td>
            </tr>
            <tr>
                <td class="label-td">
                    <label for="userpassword" class="form-label">Password: </label>
                </td>
            </tr>
```

```
            <tr>
                <td class="label-td">
                    <input type="Password" name="userpassword" class="input-text"
placeholder="Password" required>
                </td>
            </tr>


            <tr>
                <td><br>
                <?php echo $error ?>
                </td>
            </tr>

            <tr>
                <td>
                    <input type="submit" value="Login" class="login-btn btn-primary btn">
                </td>
            </tr>
        </div>
            <tr>
                <td>
                    <br>
                    <label for="" class="sub-text" style="font-weight: 280;">Don't have an
account&#63; </label>
                    <a href="signup.php" class="hover-link1 non-style-link">Sign Up</a>
                    <br><br><br>
                </td>
            </tr>




                    </form>
        </table>

    </div>
</center>
</body>
</html>
```

Signup.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <link rel="stylesheet" href="css/animations.css">
    <link rel="stylesheet" href="css/main.css">
    <link rel="stylesheet" href="css/signup.css">

    <title>Sign Up</title>

</head>
<body>
<?php

//learn from w3schools.com
//Unset all the server side variables

session_start();

$_SESSION["user"]="";
$_SESSION["usertype"]="";

// Set the new timezone
date_default_timezone_set('Asia/Kolkata');
$date = date('Y-m-d');

$_SESSION["date"]=$date;



if($_POST){



    $_SESSION["personal"]=array(
        'fname'=>$_POST['fname'],
        'lname'=>$_POST['lname'],
        'address'=>$_POST['address'],
        'nic'=>$_POST['nic'],
        'dob'=>$_POST['dob']
    );


    print_r($_SESSION["personal"]);
    header("location: create-account.php");




}

?>
```

```
    <center>
    <div class="container">
        <table border="0">
            <tr>
                <td colspan="2">
                    <p class="header-text">Let's Get Started</p>
                    <p class="sub-text">Add Your Personal Details to Continue</p>
                </td>
            </tr>
            <tr>
                <form action="" method="POST" >
                <td class="label-td" colspan="2">
                    <label for="name" class="form-label">Name: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td">
                    <input type="text" name="fname" class="input-text" placeholder="First Name"
required>
                </td>
                <td class="label-td">
                    <input type="text" name="lname" class="input-text" placeholder="Last Name"
required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <label for="address" class="form-label">Address: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <input type="text" name="address" class="input-text" placeholder="Address"
required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <label for="nic" class="form-label">NIC: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <input type="text" name="nic" class="input-text" placeholder="NIC Number"
required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
```

```
                    <label for="dob" class="form-label">Date of Birth: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <input type="date" name="dob" class="input-text" required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                </td>
            </tr>

            <tr>
                <td>
                    <input type="reset" value="Reset" class="login-btn btn-primary-soft btn" >
                </td>
                <td>
                    <input type="submit" value="Next" class="login-btn btn-primary btn">
                </td>

            </tr>
            <tr>
                <td colspan="2">
                    <br>
                    <label for="" class="sub-text" style="font-weight: 280;">Already have an
account&#63; </label>
                    <a href="login.php" class="hover-link1 non-style-link">Login</a>
                    <br><br><br>
                </td>
            </tr>

                    </form>
            </tr>
        </table>

    </div>
</center>
</body>
</html>
```

Logout.php

```
<?php

    session_start();

    $_SESSION = array();
```

```php
    if (isset($_COOKIE[session_name()])) {
        setcookie(session_name(), '', time()-86400, '/');
    }

    session_destroy();

    // redirecting the user to the login page
    header('Location: login.php?action=logout');

?>
```

REFERENCES:

1) **Android Developers.** (2025). *Android API Reference.* Retrieved from https://developer.android.com/reference

2) **Firebase Developers.** (2025). *Firebase Authentication and Realtime Database Documentation.* Retrieved from https://firebase.google.com/docs

3) **Google Cloud Developers.** (2025). *Firebase Cloud Messaging for Android Apps.* Retrieved from https://firebase.google.com/docs/cloud-messaging.

4) **Material Design Guidelines.** (2025). *Material Design Principles for Android Healthcare Applications.* Retrieved from https://material.io/design

5) **Flutter Documentation.** (2025). *Flutter Framework for Cross-Platform App Development.* Retrieved from https://docs.flutter.dev

6) **Jetpack Compose Documentation.** (2025). *Modern UI Toolkit for Android Application Development.* Retrieved from https://developer.android.com/jetpack/compose

7) **WHO Digital Health.** (2024). *Digital Health and Telemedicine Technologies Overview.* Retrieved from https://www.who.int/health-topics/digital-health

8) **ResearchGate.** (2024). *A Study on Online Doctor Appointment and Health Record Management Systems.* Retrieved from https://www.researchgate.net

9) **GeeksforGeeks.** (2024). *How to Build a Doctor Appointment App using Firebase and Flutter.* Retrieved from https://www.geeksforgeeks.org

10) **Appinventiv Blog.** (2025). *Healthcare App Development: Features, Costs, and Technology Stack.* Retrieved from https://appinventiv.com/blog/healthcare-app-development

11) **GitHub Repository – Flutter Doctor App.** (2025). *Doctor Appointment and Report Management System using Flutter and Firebase.* Retrieved from https://github.com