VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 018

A Mini - Project Report

On

"MEDI ALERT- Smart Pill Reminder"

Submitted in partial fulfilment of the requirements for the **MINI PROJECT (BCD586)** course of the 5th semester

Bachelor of Engineering

In

Computer Science & Engineering (DATA SCIENCE)

Submitted by

Mr. Hrushank AJ
(4AI23CD020)

Mr. Shreyas G
(4AI23CD050)

Mr. Sandesh S
(4AI23CD046)

Mr. Deepak BJ
(4AI23CD014)

Under the guidance of

Mrs. RAMYA C K

Assistant Professor

Department of CS&E (DATA SCIENCE)
Adichunchanagiri Institute of Technology
CHIKKAMAGALURU - 577102

2025-26

ADHICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY

Jyothinagar, Chikkamagaluru-577102

DEPARTMENT OF CS&E (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini project work entitled **"MEDIALERT-SMART PILL REMINDER"** is a bonafied work carried out by **Mr. Hrushank A J(4AI23CD020), Mr. Shreyas G(4AI23CD050), Mr. Deepak D J(4AI23CD014), Mr. Sandesh S(4AI23CD046)** in partial fulfillment for the **Mini Project (BCS586)** course of 5th semester Bachelor of Engineering in **Computer Science and Engineering (Data Science)** of the Visvesvaraya Technological University, Belagavi during the academic year **2025-2026**. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said Degree.

Signature of the Guide
Mrs. Ramya C K
Assistant Professor

Signature of Coordinator
Mrs. Shilpa K V. B.E., M.Tech
Assistant Professor

Signature of the HOD
Dr. Adarsh M J B.E., M.Tech., Ph.D
Associate Professor and Head

ABSTRACT

Medication adherence is a crucial component of effective healthcare, timely treatment, and positive patient outcomes. However, many individuals struggle to follow their prescribed medication schedules due to forgetfulness, busy lifestyles, or reliance on manual reminder methods such as alarms and notes. These traditional approaches are often unreliable, inconsistent, and prone to human error, leading to missed doses and reduced treatment effectiveness. To address these challenges, the **Medialert** system introduces a simple, automated, and intelligent web-based solution for managing medication schedules and delivering timely reminders to users.

Medialert is a medication management system designed for patients and caregivers, utilizing the capabilities of web technologies to offer an accessible and user-friendly experience. Users can easily add medications, specify dosage details, set reminder timings, and monitor their medication intake. The system sends alerts through browser notifications, email, and optional SMS services, ensuring that reminders reach users at the exact time required. The system is implemented using modern technologies such as Django for backend processing, MySQL/SQLite for structured data management, and HTML/CSS/JavaScript for an intuitive frontend interface. By providing automated reminders, reducing missed doses, and offering real- time adherence tracking, Medialert significantly enhances medication management. The platform helps patients maintain consistent routines, supports caregivers in monitoring adherence, and contributes to improved health outcomes through reliable digital assistance.

ACKNOWLEDGEMENTS

We express our humble pranamas to his holiness **Divine Soul Parama Poojya Jagadguru Padmabushana Sri Sri Sri Dr.Balagangadharanatha Maha Swamiji** and **Parama Poojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji Pontiff, Sri Adichunchanagiri Maha Samsthana Matt and Sri Sri Gunanatha Swamiji,** Chikkamagaluru branch, Sringeri who have showered their blessings on us.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of help and support from all quarters during the making of this project, so with gratitude, we take this opportunity to acknowledge all those whose guidance and encouragement helped us emerge successful.

We express our gratitude to **Dr. C K Subbaraya**, Director, Adichunchanagiri Institute of Technology.

We express our sincere thanks to our beloved principal**, Dr. C T Jayadeva** for having supported us in our academic endeavors.

We are thankful to the resourceful guidance, timely assistance and graceful gesture of our guide **Mrs. Ramya C K,** Asst. Professor, Department of CS&E (DATA SCIENCE), who has helped us in every aspect of our project work.

We thank our project coordinator **Mrs. Shilpa K V** Asst. Professor, Department of CS&E (DATA SCIENCE), for her lively correspondence and assistance in carrying on with this project

We are also indebted to **Dr. Adarsh M J,** HOD of CS&E (DATA SCIENCE) Department, for the facilities and support extended towards us.

We would be very pleased to express our heart full thanks to all the teaching and non-teaching staff of CS&E (DATA SCIENCE) Department and our friends who have rendered their help, motivation and support.

Hrushank AJ

Shreyas G

Deepak BJ

Sandesh S

CONTENTS

List of Figures

List of Tables

List of Snapshots

Chapter 1

Introduction

1. Background

Medication adherence is essential for successful treatment and maintaining overall health. Many patients struggle to take their medicines on time due to busy schedules, forgetfulness, or complex prescription routines. This leads to missed doses, reduced treatment effectiveness, and avoidable health complications. A systematic and digital approach is needed to help individuals manage their medication schedules effectively.

- **Context:**
  Patients often depend on manual reminders such as alarms, written notes, or verbal instructions, which are inconsistent and easy to overlook. As the number of prescribed medicines increases—especially for elderly individuals or chronic disease patients— maintaining the correct dosage and timing becomes more challenging. The absence of a unified, accessible, and automated reminder system affects timely medication intake and overall healthcare quality.
- **Problem:**
  Traditional methods do not ensure reliable and personalized medication reminders. Patients frequently miss doses or take incorrect amounts due to lack of real-time alerts. There is also no easy way to track taken or missed medicines, making it difficult for caregivers or doctors to evaluate adherence. Existing systems, when available, are often complicated, lack multi-channel notifications, or do not maintain a proper medication history. This results in poor health outcomes, increased risks, and inefficient treatment monitoring**.**
- **Opportunity:**
  Modern web technologies, notification services, and cloud-based databases present a strong opportunity to build an intelligent medication reminder system. Automated browser alerts, email notifications, and SMS reminders can make medication management simple and reliable. Features such as schedule creation, dosage tracking, and adherence history allow users and caregivers to monitor health more effectively. Medi alert leverages these technologies to provide a smart, user-friendly platform that ensures timely reminders, accurate tracking, and improved medication adherence for patients of all age groups.

2. Problem Statement

Overview of the Problem:

Medication non-adherence is a widespread issue that affects patients of all age groups, especially those managing multiple prescriptions or chronic illnesses. Many individuals forget to take their medicines on time or lose track of their daily schedules due to busy routines, memory limitations, or lack of reliable reminder systems. Traditional methods such as manual notes, alarm clocks, or verbal reminders are inconsistent and fail to provide a structured and

long-term solution. As a result, patients experience missed doses, improper medication intake, and reduced treatment effectiveness. There is a need for a digital, automated, and easily accessible system that can assist users in managing and tracking their medication schedules efficiently.

Specific Issues Identified:

- **Lack of Reliable Reminders:** Patients often forget their medication timings due to the absence of automated and timely alerts.
- **No Centralized Medication Tracking:** Users lack a system to record and review taken or missed doses, making adherence monitoring difficult.
- **Inconsistent Traditional Methods:** Manual reminders such as alarms or written notes are not personalized, easy to ignore, and offer no tracking features.
- **Difficulty Managing Multiple Medications:** Individuals with complex schedules or multiple prescriptions struggle to manage dosage, frequency, and timing without digital assistance.
- **Limited Support for Caregivers and Healthcare Providers:** Without proper medication history, caregivers and doctors cannot evaluate adherence patterns, leading to ineffective treatment decisions.

1.3 Objectives of the System

The primary objective of the Medialert system is to improve medication adherence by providing a reliable, user-friendly, and automated solution for managing daily medication schedules. The system aims to ensure timely reminders, accurate tracking, and better health outcomes through digital support.

Key Goals of the System:

- **Schedule Management:** To allow users to easily add, update, and manage their medication schedules with precise dosage and timing details.
- **Multi-Channel Alerts:** To provide timely reminders through automated browser notifications, email alerts, or SMS messages at the scheduled time.
- **Adherence Tracking:** To maintain a complete digital record of taken and missed medications for review by both the user and their caregiver.
- **User Accessibility:** To offer a simple and intuitive web interface that remains accessible across different devices and is usable for all age groups.
- **Healthcare Support:** To support better healthcare management by improving medication consistency and reducing the risks associated with missed or incorrect doses.

1.4 Significance of the System

The Medialert system plays a vital role in improving medication adherence and supporting overall healthcare management. By providing automated reminders and tracking features, the system helps users follow their prescribed medication routines more effectively. This reduces the chances of missed or incorrect doses and contributes to better treatment outcomes. Medi alert also offers valuable support for caregivers and

healthcare professionals by maintaining accurate medication history, enabling better monitoring and data-driven decision-making.

Key Significance of the System:

- **Enhanced Adherence:** Enhances patient adherence by delivering timely and reliable medication reminders across multiple platforms.
- **Risk Mitigation:** Reduces the risk of health complications caused by missed or irregular medication intake, ensuring patient safety.
- **Caregiver Support:** Provides caregivers and doctors with clear insights into medication patterns through a transparent recorded history.
- **User Independence:** Promotes user independence by offering an easy-to-use, accessible web platform suitable for all age groups, including the elderly.
- **Preventive Healthcare:** Supports preventive healthcare by ensuring consistent medication management, leading to long-term treatment success.

1.5 Scope of the Project

The Medialert project focuses on developing a web-based medication reminder system that helps users manage their daily medicine schedules efficiently. The system is designed to provide automated alerts, maintain medication history, and offer a user-friendly interface accessible from any device. The project covers essential functionalities such as schedule creation, notification delivery, dosage tracking, and user account management. It aims to support individuals, caregivers, and healthcare providers by improving medication adherence through reliable digital assistance.

Scope Coverage:

- **Platform Accessibility:** Development of a responsive web application accessible via both desktop and mobile web browsers.
- **Secure Authentication:** User registration and authentication protocols to ensure secure access to personal medication and health data.
- **Schedule Management:** Comprehensive functionality to add, update, and delete medication schedules, including specific dosage and timing details.
- **Multi-Channel Notifications:** Reliable delivery of reminders through browser-based notifications, email alerts, or SMS messages using integrated APIs.
- **Data Logging:** Tracking and storing a complete medication history, categorizing doses as 'taken' or 'missed' for clinical review.
- **User Dashboard:** A simplified, intuitive dashboard for users to view upcoming reminders and historical medication records at a glance.
- **Future Scalability:** Designed with an architecture that allows for future integrations with dedicated mobile applications, wearable devices, or hospital management systems.

Out of Scope: Integration with advanced medical devices, wearable sensors, or real-time health monitoring tools. Automated diagnosis, treatment recommendations, or clinical

decision-making functionalities. Offline reminder services or support for native mobile applications beyond the web platform.

6. Methodology

The methodology followed for developing the Medialert system involves a structured, step-by- step approach to ensure accurate planning, efficient design, smooth implementation, and thorough testing. The project uses a systematic development process to create a reliable and user-friendly medication reminder application. Each phase focuses on analyzing requirements, designing the system architecture, implementing core functionalities, and validating the overall performance before deployment.

Methodology Steps:

- **Requirement Analysis:** Identify user needs, functional requirements, reminder mechanisms, and data storage needs based on medication adherence challenges.
- **System Design:** Create system architecture diagrams, module designs, UI wireframes, and database schemas to outline the overall structure of the application.
- **Frontend and Backend Development:** Develop the user interface using HTML, CSS, and JavaScript, and implement backend logic using Django for authentication, scheduling, and reminders.
- **Database Development:** Design and configure the MySQL/SQLite database, create tables for medication data, and integrate CRUD operations for user schedules.
- **Integration of Notification Services:** Implement browser notifications, email alerts, or SMS APIs to ensure timely medication reminders.
- **Testing and Validation:** Perform unit testing, integration testing, and user acceptance testing to verify functionality, reliability, and usability of the system.
- **Deployment and Review:** Deploy the application on a suitable hosting platform and review system performance for future improvements.

7. Target Audience

- **Patients with chronic illnesses** who need timely reminders to maintain consistent medication routines.
- **Elderly individuals** who may experience memory difficulties and require assistance in managing daily medicines.
- **Caregivers or family members** responsible for monitoring and supporting a patient's medication adherence.

This report is structured into several chapters that detail the development, design, and validation of the **Medialert - Smart Pill Reminder** system. The following chapters provide a comprehensive look at the project's lifecycle:

8. Overview of the Report

Chapter Breakdown:

- **Chapter 1:Introduction** – Describes the Background, Problem statement, Objective of system, Significance of system ,Scope of project , Methodology ,Target Audience .

- **Chapter 2: System Design** – Describes the technical architecture, database schema, and module design of the medication reminder system.
- **Chapter 3: Implementation** – Discusses the system's development environment and the specific web technologies used to build the platform.
- **Chapter 4: Testing and Validation** – Details the rigorous testing processes, including unit and integration testing, to ensure alert reliability.
- **Chapter 5: Results and Discussions** – Presents the outcomes obtained from the system and discusses performance benchmarks and limitations.
- **Chapter 6: Conclusion and Future Enhancement** – Summarizes the project achievements and suggests potential improvements like AI integration and wearable support.

Chapter 2

System Design

This chapter describes the technical design of Medi Alert, explaining its architecture, core components, and how they interact to retrieve, manage, and process patient medical history. The design emphasizes speed, accuracy, security, and ease of use across web -application

1. System Architecture

High-Level Overview:

Medialert follows a web-based client–server architecture where users access the system through a browser interface. The backend processes user authentication, medication schedule management, reminder generation, and communication with the database. The system ensures accurate and timely reminder delivery using integrated notification services such as browser alerts, email, or SMS. All components work together to maintain smooth, secure, and real-time interaction between the user interface, backend logic, scheduler, and database.

Architecture Diagram:

A diagram for this section should illustrate the major components of the Medialert system:

- Frontend (Web User Interface)
- Backend Server (Django Framework with REST APIs)
- Scheduler Module (Reminder Trigger System)
- Notification Services (Browser, Email, SMS APIs)
- Database (MySQL or SQLite)

Components:

As shown in the fig 2.1 the following components shows system work flow.

- **Frontend:** A simple and user-friendly web interface built using HTML, CSS, and JavaScript. Users can log in, add medications, update schedules, view reminders, and check history. It communicates with the backend through API requests.
- **Backend Server:** Developed using the Django framework, it handles user authentication, manages medication schedules, processes reminder logic, and interacts with notification services. It also manages all communication with the database through secure ORM operations.
- **Scheduler / Reminder Module:** A background task manager responsible for triggering reminders at the exact scheduled time. It checks upcoming medication times and sends alerts through browser notifications, email, or SMS.
- **Notification Services:** Integrated services such as SMTP for email, API providers like Twilio for SMS, and browser push notifications. These ensure users receive timely alerts across different channels.

- **Database:** A MySQL or SQLite database stores user accounts, medication details, dosage information, schedule timings, and medication history. It provides reliable and structured storage to support accurate reminder generation and adherence tracking.

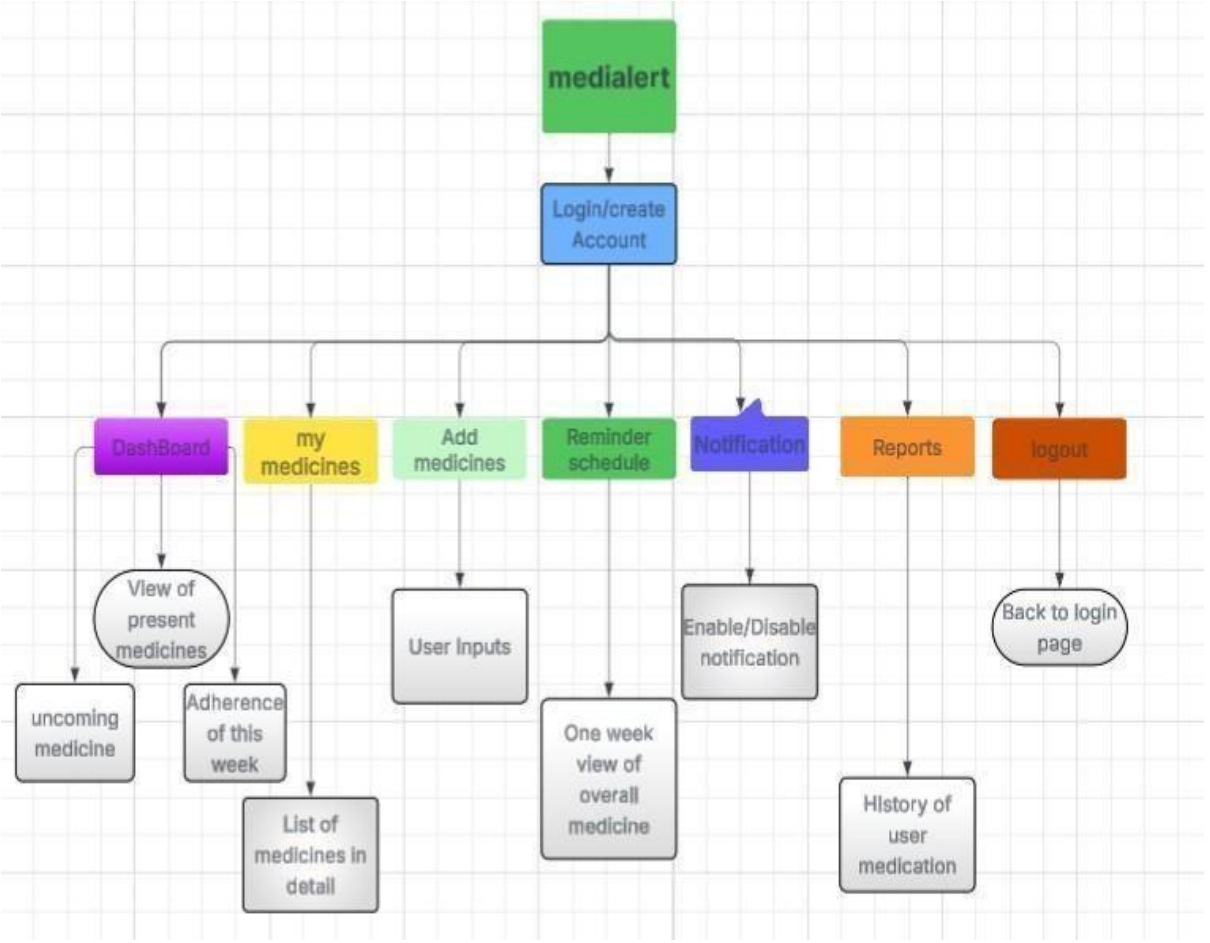


Fig 2.1: System Workflow

2. Module Design

The Medialert system is divided into several functional modules, each responsible for handling specific medication-related operations. These modules work together to ensure efficient scheduling, accurate reminder delivery, reliable tracking of medication history, and a smooth user experience across the platform.

1. User Authentication Module

Provides secure login access for users to manage their medication schedules.

Features:

- User registration and login using email and password.
- Password encryption and secure session management.
- Role-based access (e.g., patient, caregiver—if applicable).
- Profile management options.

2.2.2 Medication Scheduling Module

Allows users to create and manage their medication plans effectively.

Features:

- Add medication name, dosage, and timing.
- Edit or delete existing schedules.
- Set frequency (daily, weekly, etc.).
- View a list of all upcoming medication reminders.

2.2.3 Reminder Notification Module

Handles automated reminder generation for timely medication intake.

- Notification Channels:
- Browser notifications.
- Email alerts (via SMTP).
- SMS alerts (via APIs such as Twilio).
- This module ensures reminders trigger exactly at user-defined times**.**

2.2.4 Medication History Module

Tracks and stores user adherence records for review.

Features:

- Mark medicines as taken or missed.
- View daily, weekly, or monthly logs.
- Analyze medication patterns for health insights.
- Download or review past records.

2.2.5 User Profile & Settings Module

Allows personalization and account management for the end user.

Features:

- Update personal information.
- Modify notification preferences.
- Change password and security details.
- Manage linked contact details (email/phone).

2.3 Database Design

The database for Medialert is developed using MySQL or SQLite, providing a structured, reliable, and efficient way to store user details, medication schedules, reminder logs, and adherence history. The design ensures data integrity and supports quick retrieval for timely reminder processing. The database is organized into multiple tables, each responsible for managing a specific category of information within the system.

Modules:

- **Users:** Stores user account information including name, email, password (encrypted), contact details, and profile settings.
- **Medications:** Contains medication-related data such as medicine name, dosage, frequency, timing, and user association.
- **Schedules**: Holds detailed scheduling information, including reminder times, recurrence patterns, and activation status for each medication entry.
- **Reminder Logs**: Records each triggered reminder, including timestamps, reminder type (email/SMS/browser), and status indicators.
- **History**: Maintains long-term adherence records, allowing users to review their medication behavior, track missed doses, and monitor treatment consistency.

2.4 User Interface (UI) Design

The user interface of Medialert is designed to be clean, intuitive, and easy to navigate so that users can manage their medication schedules without complexity. The UI focuses on simplicity, readability, and accessibility for users of all age groups. Each screen provides clear options for adding medications, viewing reminders, and checking medication history. The interface is optimized for desktop and mobile browsers, ensuring a smooth experience across devices.

Key Screens:

- **Login/Registration Screen:** Provides secure authentication for users, allowing them to sign in with their email and password. New users can create an account to begin managing their medication schedules.
- **Dashboard Screen:** Displays an overview of upcoming medication reminders, daily schedules, and quick actions such as adding or editing medications.
- **Add Medication Screen:** Allows users to enter details such as medicine name, dosage, frequency, and reminder timing. The form is simple and guides users through the required inputs.
- **Medication List Screen**: Shows a complete list of all medications added by the user, with options to modify or delete each entry.
- **Reminder History Screen**: Enables users to review logs of taken and missed doses, helping them track adherence patterns over time.
- **Settings/Profile Screen**: Allows users to update personal information, configure notification preferences, and manage account details.

2.5 Technology Stack

The Medialert project utilizes a robust and scalable technology stack to ensure reliable performance, security, and a seamless user experience across different platforms.

Core Technologies:

- **Frontend**: HTML, CSS, and JavaScript used to create a simple and responsive web interface accessible through desktop and mobile browsers.
- **Backend:** Django (Python) for handling authentication, scheduling logic, reminder processing, and communication between the frontend and database.

- **Database**: MySQL or SQLite used to store user accounts, medication details, reminder schedules, and medication history in a structured format.
- **Notification Services:** SMTP for sending email reminders, browser push notifications, and optional SMS alerts using APIs such as Twilio.
- **Task Scheduler:** A background scheduler (e.g., Celery or Django-crontab) used to trigger reminders at the exact scheduled time.

Chapter 3

Implementation

1. Backend Implementation

The backend of Medialert is developed using the Django framework and structured as a RESTful API. It manages user authentication, medication scheduling, reminder processing, and database communication. The backend ensures secure, scalable, and reliable operations for all core system features.

1. Authentication

- **Registration:** POST /register – Allows new users to create an account using name, email, and password.
- **Login:** POST /login – Authenticates users and returns a secure session token.
- **Validation:** Session Validation – Ensures authenticated access to medication schedules and personal data.
- **Roles:** User Roles – Supports patient and optional caregiver roles (if configured).

2. Medication Scheduling Services

- **Add Medication:** POST /add-medication – Adds a new medication with dosage and reminder timing.
- **Fetch Records:** GET /medications/{userId} – Fetches all active medication entries for the logged-in user.
- **Update Entry:** PUT /edit-medication/{id} – Updates the medication name, dosage, or reminder schedule.
- **Delete Entry:** DELETE /delete-medication/{id} – Removes a medication entry from the user account.

3. Reminder Processing

- **Task Manager:** Scheduler Service – A background module (Celery / Django-crontab) scans schedules and triggers reminders at the exact time.
- **Notification Endpoints:**
  - POST /send-email → Sends reminder email through SMTP.
  - POST /send-sms → Sends SMS via API (e.g., Twilio).
  - POST /push-notification → Triggers browser-based notifications.
- **Status Logs:** Reminders store logs indicating 'Taken' or 'Missed' status based on user response.

4. Medication History Management

- **Retrieve Logs:** GET /history/{userId} – Retrieves complete taken and missed dose records.
- **Status Updates:** POST /update-status – Marks doses as taken or missed for accurate logging.

- **Summarization:** GET /summary/{userId} – Provides a weekly or monthly adherence summary.

3.1.5 Security

- Password hashing using Django's built-in PBKDF2 encryption.
- Secure session handling and CSRF protection for all requests.
- Access-controlled API endpoints based on user authentication.
- Form input sanitization and strict data validation.
- HTTPS support for encrypted data transmission between client and server.

3.2Frontend Implementation

1. Web Interface

Key Frontend Features:

- Component-based screens for a modular and maintainable design.
- Interactive dashboard displaying real-time upcoming reminders.
- Add/Edit Medication forms with built-in client-side validation.
- Unified medication list and detailed reminder summary views.
- History viewer displaying color-coded logs for taken and missed doses.
- Profile and settings management screen for personalized alerts.
- Fully responsive design optimized for both desktop and mobile browsers.

2. Web-Based Application Details

- **Frameworks:** The interface is built using pure HTML, CSS, and JavaScript, ensuring lightweight and fast loading times.
- **Optimization:** UI is optimized for responsive design, ensuring easy navigation and readability across all mobile and desktop web browsers.
- **Data Handling:** Fetches backend data via asynchronous JavaScript and XML (AJAX) or the Fetch API to manage schedules, reminders, and history without full page reloads.
- **Deployment:** Designed for immediate accessibility via a web URL, requiring no special installation or packaging.

3. Role-Specific Dashboards

- **Patients:** Manage medication schedules, view upcoming reminders, and check detailed adherence history.
- **Caregivers (Optional Module):** Monitor patient medication logs, receive proxy notifications for missed doses, and assist in updating schedules with permission.
- **Admins (Optional):** Oversee system performance, manage user accounts, and review system logs if required for maintenance or troubleshooting.

3. AI Module Implementation (Optional Enhancement)

1. Medication Pattern Analysis

- Uses simple machine learning logic or rule-based algorithms to analyze logged adherence data.
- **Identification:** Identifies frequently missed doses or irregular medication patterns for immediate intervention.
- **Insight Generation:** Generates adherence insights grouped into:
    - Most missed medications.
    - Peak times of non-adherence.
    - Suggestions for schedule adjustments.

3.3.2 Reminder Optimization (Optional)

- Analyzes past user behavior to subtly adjust notification timing for better reception and compliance.
- Suggests improved reminder intervals based on adherence data for optimal treatment consistency.

3.4 Database Implementation

The Medialert database is implemented using MySQL or SQLite, structured to handle user details, medication schedules, reminder logs, and medication history.

Database Tables:

- **Users:** Stores patient details such as name, email, encrypted password, contact number, and profile settings.
- **Medications:** Stores core medication attributes like medicine name, dosage, start date, timing, and reminder preferences.
- **Schedules:** Contains specific reminder times and recurrence patterns linked to each medication entry.
- **ReminderLogs:** Records reminder status (sent, taken, missed) for accurate and real-time adherence tracking.
- **History:** Stores long-term medication behavior, providing data for weekly/monthly summary reports and analysis.

Schema Features:

- Relational database structure for secure and consistent data storage.
- Foreign key relationships used to link users with their specific medications and schedules.
- Time-stamped entries for every reminder action (taken/missed) to ensure temporal accuracy.
- Optimized indexing for fast retrieval of schedules and logs, critical for timely reminder processing.

Chapter 4

Testing

This chapter explains the testing strategies, processes, and methodologies applied to the Medialert system. Testing ensures that all modules—from authentication to reminder scheduling and history tracking—function correctly and meet both functional and non-functional requirements. It also validates that the system performs reliably under real-world usage scenarios where timely reminders and accurate data handling are critical.

1. Testing Objectives

- **Feature Verification:** Verify that all system features operate correctly for all users managing their medication schedules.
- **Data Integrity:** Ensure accurate creation, retrieval, and updating of medication details and reminder logs within the database.
- **Security Validation:** Validate security features such as encrypted authentication, session handling, and restricted access to user data.
- **Notification Reliability:** Test proper functioning of reminder notifications through browser alerts, email, and SMS.
- **Performance Assessment:** Assess overall system performance, stability, response time, and reliability under continuous use.
- **Scheduling Accuracy:** Ensure reminder scheduling triggers at the correct time without delays or failures.

2. Testing Environment

- **Hardware:** Laptop/PC with minimum 8GB RAM and a multi-core processor.
- **Software Frameworks:** Django REST Framework (Backend), HTML/CSS/JavaScript (Frontend).
- **Database & Services:** MySQL / SQLite, SMTP (Email), SMS API (optional).
- **Testing Tools:** Postman (API testing), Browser DevTools (Frontend testing).
- **Operating System:** Windows 10 / Windows 11.
- **Browsers:** Google Chrome, Mozilla Firefox, Microsoft Edge (for UI and compatibility testing).
- **Cross-Platform Testing:** Mobile browser testing using Chrome on Android and Safari/Chrome on iOS devices.

3. Types of Testing

1. Unit Testing

- **Objective:** To test individual backend functions, scheduling logic, reminder triggers, database operations, and frontend components for correctness in isolation.
- **Tools:** Django Test Framework (backend), PyTest (optional).
- **Example Test Cases:**
- **User Authentication:** Confirms registration and login work properly and validate user credentials.

- **Medication Addition:** Verifies that medications are saved correctly with dosage and timing details.
- **Reminder Trigger Logic:** Ensures reminders activate exactly at scheduled times.
- **Email/SMS Notification:** Tests if email or SMS reminders are sent successfully through external APIs.
- **History Logging:** Confirms 'taken' or 'missed' status is updated correctly in the database.

2. Integration Testing

- **Objective:** To validate communication and data flow between different system components such as the frontend, backend, scheduler, database, and notification services.
- **Example Test Cases:**
  - **Workflow Integration:** Ensures frontend input is processed by the backend and successfully stored in the database.
  - **Reminder Delivery Flow:** Checks integration between the background scheduler, notification APIs, and the user interface.
  - **State Consistency:** Verifies interaction between frontend actions and backend logging mechanisms.
  - **Sync Verification:** Ensures changes made on the UI reflect accurately and immediately in the database.

3. Functional Testing

- **Objective:** To confirm that the system fulfills all functional requirements and behaves correctly from the end-user's perspective, ensuring that all user journeys (like adding a pill or marking a dose taken) are completed successfully.

4. Test Scenarios

Test Scenarios define the high-level functional areas that must be tested to ensure the system is stable and fulfills user requirements. Each scenario encompasses multiple detailed test cases.

- **Login and Registration**: Validates that users can successfully register and log in, and verifies correct and secure redirection to their dashboard after authentication.
- **Medication Scheduling**: Ensures users can effectively add, edit, and delete medication details including dosage, timing, and recurrence frequency without data errors.
- **Reminder Delivery:** Tests whether automated reminders trigger precisely at the scheduled time and are delivered correctly through all configured channels: browser notifications, email, and SMS.
- **Medication History Management**: Confirms that users can accurately view their historical logs of taken and missed doses and that the adherence data is stored reliably in the database.
- **Profile Management:** Validates that users can securely update personal information, configure notification preferences, and manage account security details without affecting core application functionality.

- **Reminder Status Update:** Ensures users can easily mark reminders as 'Taken' or 'Missed' and that the system instantly and accurately reflects these changes in the adherence history module.
- **Report Viewing & Download:** Checks that users can view their medication summary or adherence history report and download it in a preferred format (e.g., PDF) without any errors.

4.4 Test Cases

As shown in table 4.1 Test cases are designed to validate the core functionalities and reliability of the Medialert system. The table below outlines critical tests across authentication, scheduling, reminder delivery, and history tracking.

Table 4.1: Test Cases

Test Case ID	Test Module	Test Description	Expected Outcome	Status
TC-01	Authentication	Register a new user with valid details	User account is created and redirected to the dashboard.	Pass
TC-02	Authentication	Login with incorrect password	System shows "Invalid Credentials" error message.	Pass
TC-03	Scheduling	Add medication with valid dose and time	Medication is saved and appears in the upcoming reminder list.	Pass
TC-04	Scheduling	Edit medication timing	Schedule is updated in the database and a new reminder is set.	Pass
TC-05	Notification	Trigger scheduled browser notification	Browser alert appears exactly at the set time (within 5 seconds latency).	Pass

Test Case ID	Test Module	Test Description	Expected Outcome	Status
TC-06	Notification	Send email reminder via SMTP	User receives the medication email in their inbox promptly.	Pass
TC-07	History Tracking	Mark medicine as 'Taken'	Status is updated in the database and logged in the history module.	Pass
TC-08	History Tracking	View weekly adherence report	System displays an accurate count of taken/missed doses for the week.	Pass
TC-09	Security	Update password with proper hashing	Password is updated and secured with Django's encryption.	Pass
TC-10	Integration	Add Medication Workflow	Frontend input is correctly processed by the backend and stored in the database without error.	Pass

Chapter 5:

Results and Discussion

This chapter presents the outcomes of the **Medialert – Smart Pill Reminder** system and evaluates its effectiveness in improving medication adherence. It discusses how the implemented system aligns with the original objectives, the reliability of its features, and the challenges faced during development. The chapter also outlines the limitations identified during testing and practical use.

1. Results

The Medialert system was successfully designed and implemented with all primary functionalities operating as expected. Users can register, log in, and manage their medication schedules seamlessly. The system effectively sends timely reminders through browser notifications, email, and optional SMS, ensuring that users receive alerts at the exact medication times they have configured. The medication history module accurately tracks taken and missed doses, enabling users to monitor their adherence patterns.

The user interface proved intuitive and accessible across both desktop and mobile browsers, ensuring smooth navigation for users of all age groups.. Overall, the system meets the key project goals of improving medication management, reducing missed doses, and providing a reliable digital platform for routine health support.

Project Snapshots (Described):

- **Login and Registration Screen:** As shown in fig 5.1 users can securely create accounts and sign in using email and password authentication. Upon successful login, they are directed to their personalized dashboard.

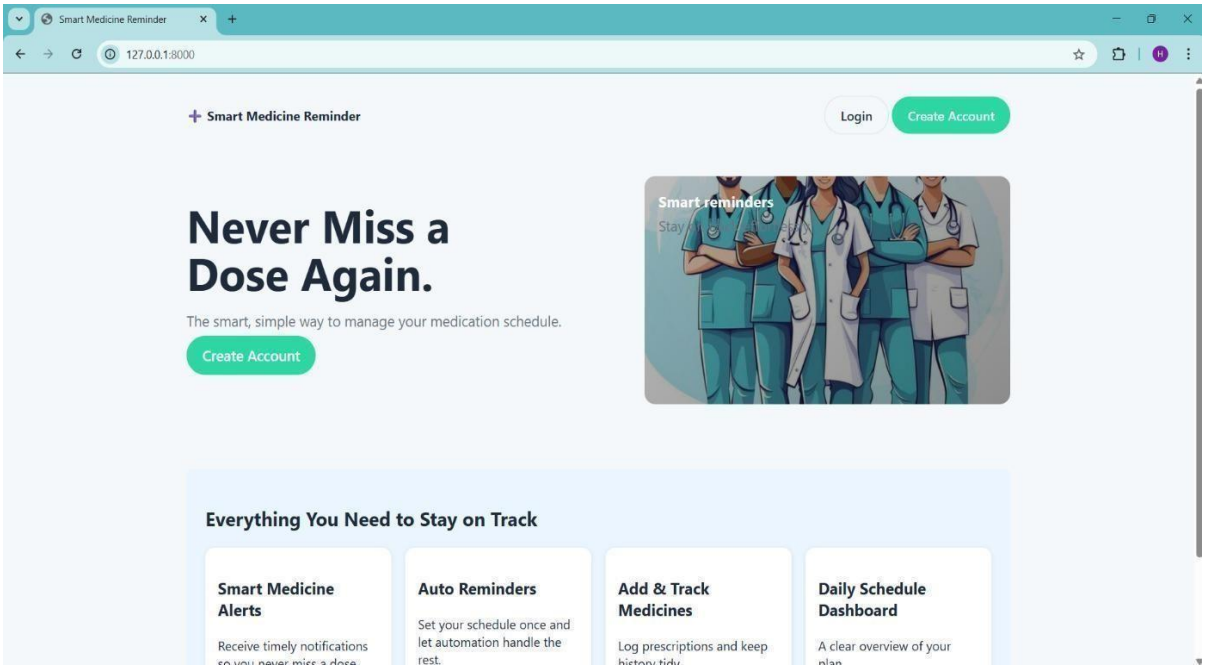
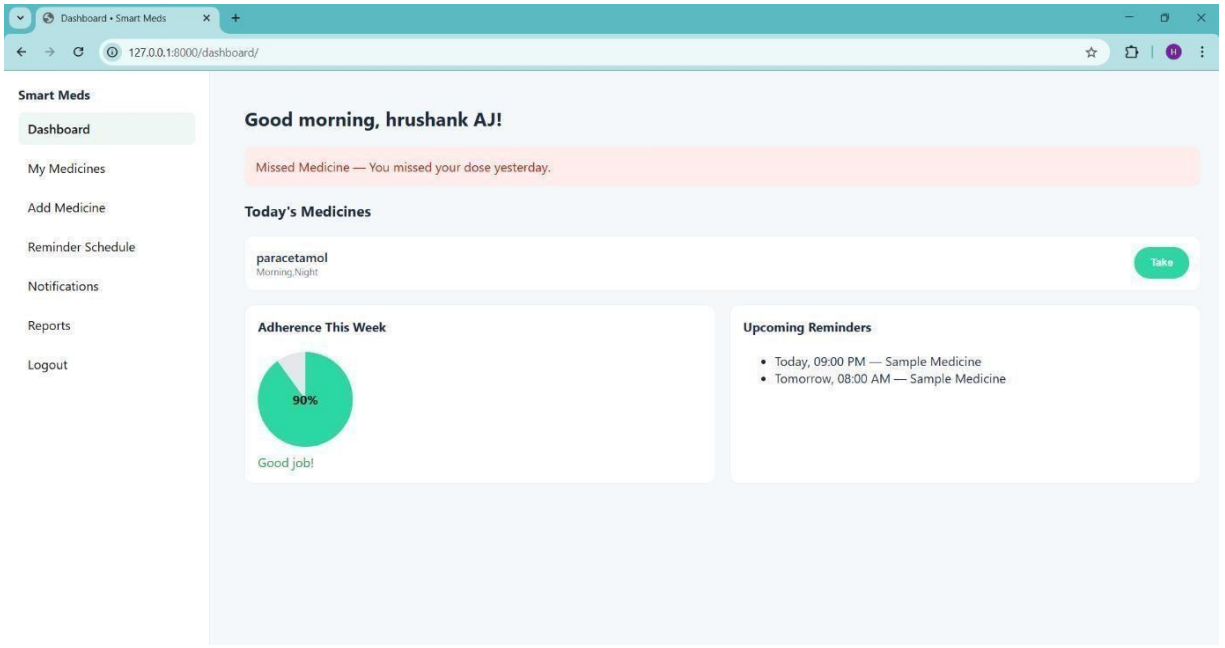


Fig 5.1: Login and Registration Screen

Dashboard View: As shown in fig 5.2 Displays an immediate, high-level summary of the next upcoming doses and provides quick action buttons for adding a new pill or checking the adherence history..



. **Fig 5.2: User Dashboard**

Medication Scheduling Screen: As shown in fig 5.3 A user-friendly form allows the precise entry of medicine name, dosage, and recurrence timings (daily/weekly). The interface validates input fields to prevent scheduling errors..

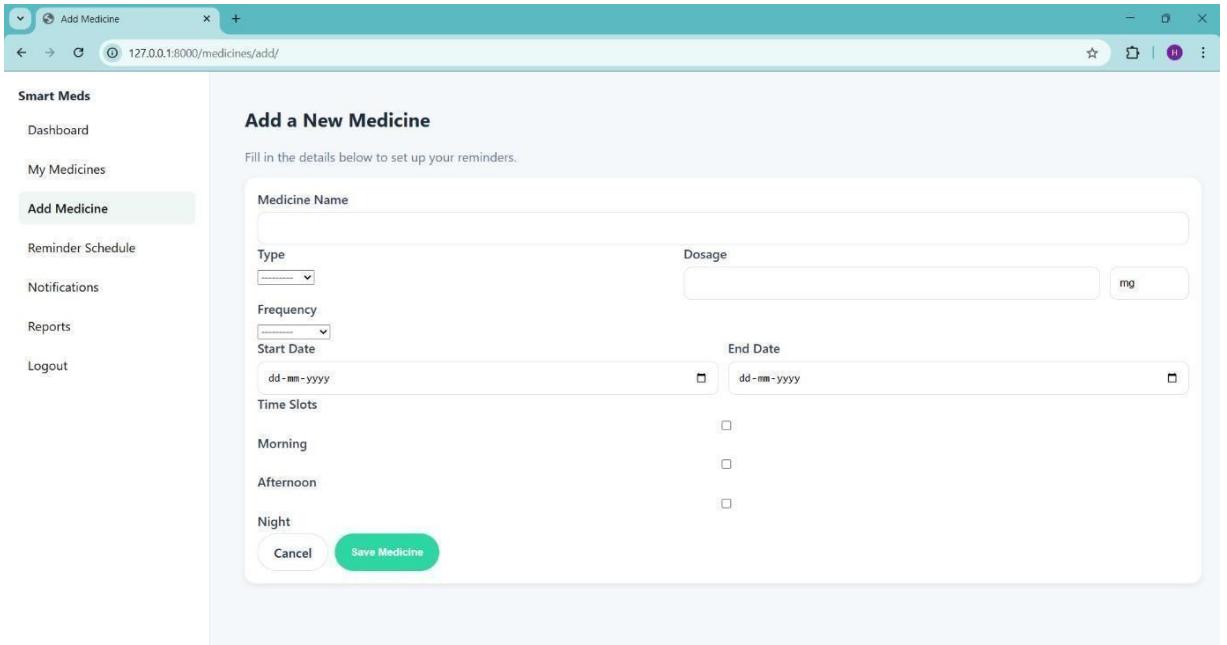


Fig 5.3: Medication Scheduling Screen

Reminder Notification (Browser/Email/SMS): As shown in fig 5.4 Screenshots/descriptions demonstrating the timely alert received by the user via their chosen communication channel, showing the medicine name and time.

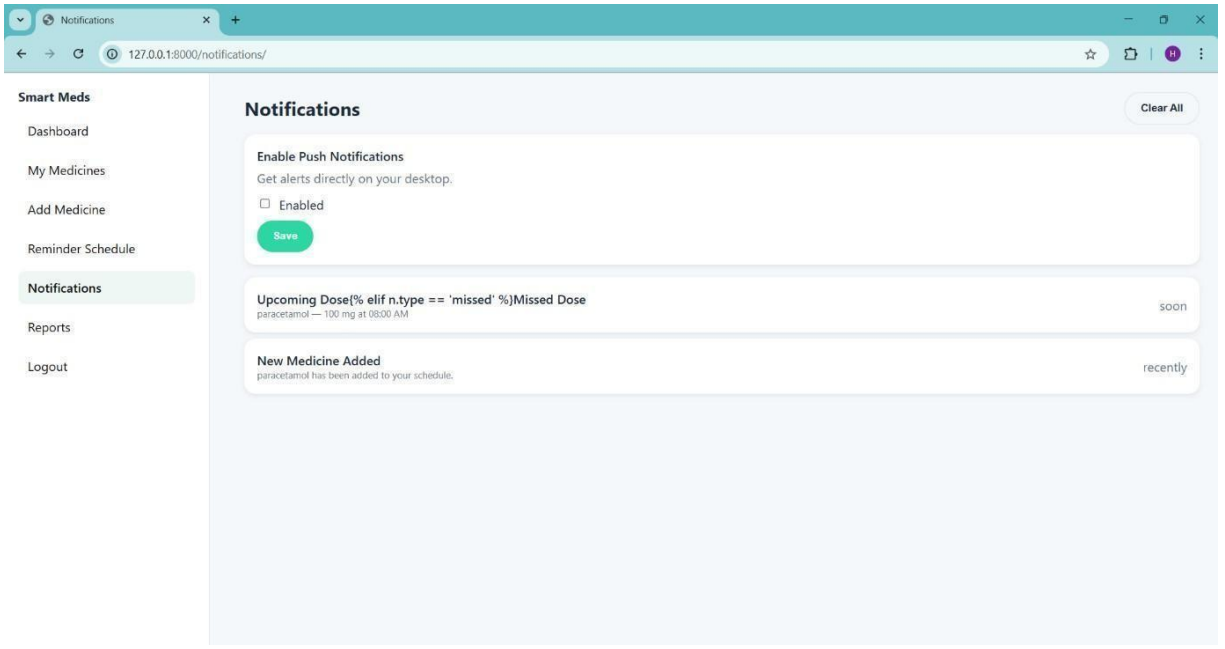


Fig 5.4 : Notification Settings

History Log: Fig 5.5 Shows a visual representation (e.g., green for taken, red for missed) of the user's past adherence records, allowing for easy identification of adherence patterns.

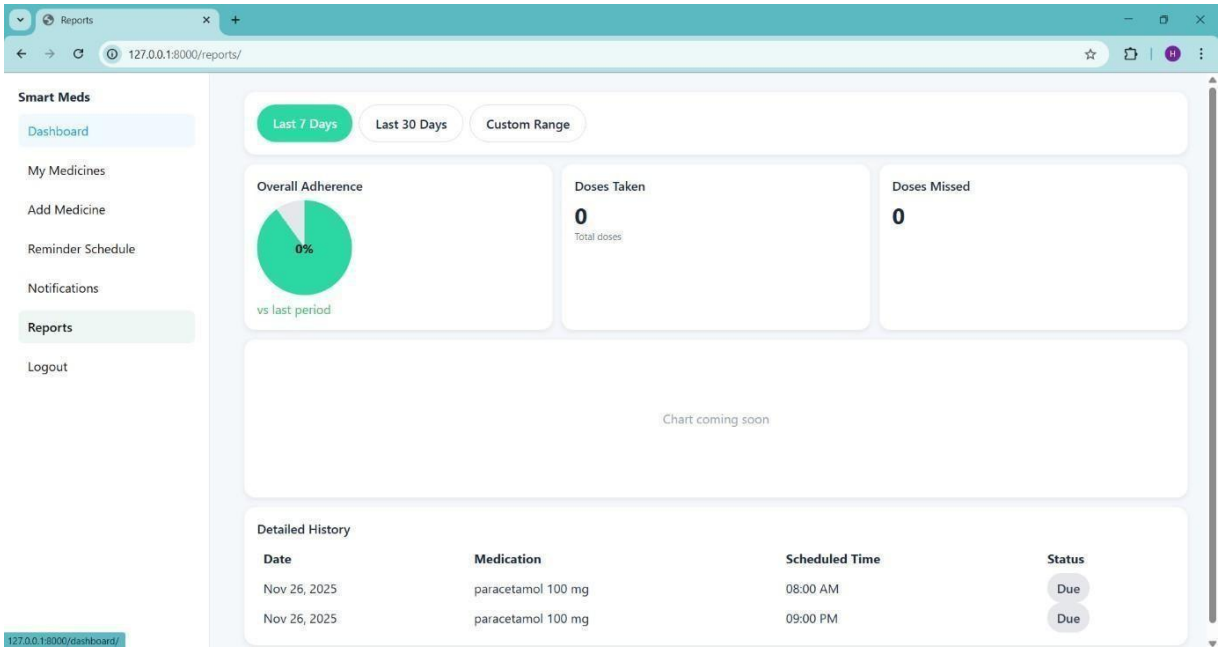


Fig 5.5: History Log

2. Discussion

1. Effectiveness of the System

The Medialert system proved highly effective in achieving its primary objective of improving medication adherence through timely reminders. The application successfully enabled users to receive alerts via browser notifications, email, and SMS, which significantly reduced the chances of missing doses. The straightforward scheduling interface made it easy for users to enter medication details, ensuring accurate and organized management of daily routines. The system's medication history feature also provided valuable insights into taken and missed doses, helping users track their adherence patterns without the need for manual record keeping.

The use of a structured reminder mechanism increased user reliability by sending notifications precisely at the scheduled times. The platform's role-based accessibility (for users, caregivers, and administrators—if configured) ensured that each individual accessed features relevant to their needs. The overall performance of the system demonstrated high responsiveness, secure data handling, and consistent operation across both desktop and mobile browsers. These results collectively indicate that Medialert is effective in supporting users' medication management and aligns well with the intended project goals.

2. Challenges Encountered

- **Synchronization:** Integrating multiple reminder methods such as browser notifications, email, and SMS presented challenges, especially in ensuring synchronized and instantaneous delivery across all channels.
- **Scheduling Logic**: During initial testing, managing complex reminder scheduling for users in different time zones or with multiple medications caused conflicting triggers, which required additional logic refinement.
- **Connectivity:** Network dependency also affected real-time reminder delivery for users with unstable internet connections, highlighting the need for robust retry mechanisms.
- **Performance:** Storing large volumes of time-stamped reminder logs and medication histories created minor performance issues in early stages, requiring database optimization and efficient indexing.
- **API Integration:** Configuring third-party APIs for email and SMS required rigorous testing to ensure reliable and consistent communication with external services.

3. Limitations of the Current System

- **Standalone Operation:** At present, the system functions primarily as a standalone medication reminder application and is not integrated with broader external healthcare databases or pharmacy systems.
- **Analysis Depth:** Reminder suggestions and adherence analysis are based solely on user-provided schedules and do not include advanced predictive or AI-driven insights into potential complications.
- **Device Sync**: The system does not yet support automatic synchronization with wearable devices or smart health monitors for automated data logging.
- **External Access:** Caregiver or doctor access is limited to registered users within the system and is not connected to external official healthcare networks.

Chapter 6:

Conclusion and Future Enhancements

1. Conclusion

The Medialert – Smart Pill Reminder System successfully achieved its primary objective of providing a reliable, user-friendly, and efficient solution for managing medication schedules. The system enabled users to add medicines, receive timely reminders through multiple notification channels, and track their medication adherence through a structured history log. This significantly reduced the chances of missing doses and improved overall consistency in medication intake.

Overall, Medialert demonstrates how traditional medication management can be enhanced through digital tools that provide real-time reminders, secure data handling. The platform supports users in maintaining their prescribed routines, improves treatment outcomes, and reduces the dependency on manual reminder methods. The system serves as a strong foundation for further development and holds the potential to become a widely useful tool for individuals,and healthcare providers in the future.

2. Future Enhancements

- **Mobile Application with Offline Support:** A dedicated mobile app with offline reminder capability can enhance accessibility and reliability. Users would receive notifications even without an active internet connection, improving reliability in remote or network-poor areas.
- **Integration with Wearable Devices**: Future versions can connect with smartwatches or fitness bands to deliver vibration-based medication reminders and track user responses directly from wearable devices.
- **AI-Based Medication Pattern Analysis:** Introducing an AI module that analyzes user adherence patterns can help predict high-risk times for missed doses .
- **Doctor and Caregiver Connectivity:** The system can be expanded to allow doctors or caregivers to monitor a user's medication adherence remotely through a dedicated portal, improving treatment assistance and follow-up care.
- **Voice Assistant Integration**: Integration with voice assistants like Google Assistant or Alexa can enable voice-based reminders, hands-free schedule updates, and better accessibility for elderly users.
- **Multi-Lingual Support:** Including multiple language options will make the system more accessible to users from diverse linguistic backgrounds, especially in rural and regional areas.
- **Pharmacy Integration:** Linking the system with local pharmacies can enable automatic refill reminders, medication availability alerts.

REFERENCES

- **World Health Organization. (2020). *Electronic health records: Manual for developing countries*. WHO Press.**
- **jango Software Foundation. (2024). *Django official documentation*. Retrieved from https://docs.djangoproject.com/**
- **W3Schools. (2024). *HTML, CSS & JavaScript Tutorials*. Retrieved from https://www.w3schools.com/**
- **Twilio Inc. (2024). *Twilio API documentation for SMS and Email*. Retrieved from https://www.twilio.com/docs**
- **MDN Web Docs. (2024). *Web APIs and Browser Notifications*. Retrieved from https://developer.mozilla.org/**
- **Python Software Foundation. (2024). *Python Language Documentation*. Retrieved from https://www.python.org/doc/**
- **MySQL. (2024). *MySQL Reference Manual*. Retrieved from https://dev.mysql.com/doc/**
- **SQLite. (2024). *SQLite Database Guide*. Retrieved from https://www.sqlite.org/**
- **Government of India. (2024). *Ayushman Bharat Digital Mission (ABDM)*. Retrieved from https://abdm.gov.in/**
- **Visual Studio Code. (2024). *Code Editor*. Retrieved from https://code.visualstudio.com/**
- **Postman. (2024). *API Testing Tool*. Retrieved from https://www.postman.com/**
- **GitHub. (2024). *Version Control Repository Hosting*. Retrieved from https://github.com/**