

# Med Track: Comprehensive Project Report

---

**Tagline:** *Your Health, Reimagined.*

**Project Developer:** Harika - RVRJC **Date:** January 2026

---

## 1. Executive Summary

---

MediTrack is a comprehensive web-based healthcare management application designed to assist individuals in managing their medical routines effectively. In an era where personal health management is becoming increasingly complex, MediTrack offers a centralized platform for tracking prescriptions, scheduling doctor appointments, and receiving intelligent health support.

Built with a robust Python Flask backend and a modern, responsive frontend, the application aims to improve medication adherence and streamline patient-doctor interactions.

---

## 2. Introduction

---

### 2.1 Problem Statement

Patients often struggle with:

- Remembering to take multiple medications at specific times.
- Keeping track of appointment schedules and avoiding conflicts.
- getting immediate, reliable answers to basic health queries without waiting for a doctor.
- Maintaining a historical record of their adherence for doctor visits.

### 2.2 Proposed Solution

MediTrack addresses these challenges by providing:

- **Smart Medicine Logging:** A digital inventory of prescriptions with dose tracking.
- **Appointment Management:** A booking system with automatic conflict detection.
- **AI Health Assistant:** A chatbot for 24/7 symptom checking and mental health support.
- **Adherence Analytics:** Visual charts showing how well the patient is following their regimen.

---

## 3. System Architecture

---

The project follows a generic Model-View-Controller (MVC) compatible architecture, adapted for Flask (Blueprints).

## 3.1 Technology Stack

- **Backend:** Python 3.11, Flask 3.0 (Web Framework)
- **Database:** SQLite (Relational DB), SQLAlchemy (ORM)
- **Frontend:** HTML5, CSS3, Bootstrap 5, Vanilla JavaScript
- **Visualization:** Chart.js for data analytics
- **Authentication:** Flask-Login, Bcrypt (Security)
- **Task Scheduling:** APScheduler (Background jobs for reminders)
- **Deployment:** Docker, Docker Compose

## 3.2 Directory Structure

```
med_track_rvrjc_project/ ■■■ app/ ■ ■■■ templates/ # HTML pages (UI) ■  
■■■ static/ # CSS, JS, Images ■ ■■■ models.py # Database Schema ■ ■■■  
routes/ # Blueprints (auth, medicine, etc.) ■ ■■■ __init__.py # App  
Factory ■■■ tests/ # Automated Unit Tests ■■■ migrations/ # Database  
Migrations ■■■ Dockerfile # Container Config ■■■ run.py # Entry Point
```

---

# 4. Key Features & Modules

---

## 4.1 User Authentication System

- **Registration & Login:** Secure user creation with email and password.
- **Security:** Passwords are hashed using `bcrypt` before storage. No plain-text passwords exist in the database.
- **Session Management:** `Flask-Login` handles user sessions, ensuring only authenticated users access the dashboard.

## 4.2 Dashboard & Analytics

- **Overview:** The landing page after login provides a snapshot of the user's health.
- **Adherence Chart:** A dynamic bar chart visualizes the percentage of doses taken vs. scheduled.
- **Quick Actions:** One-click access to log doses or book appointments.

## 4.3 Medicine Management

- **Inventory:** Users can add medicines with details like Name, Dose Strength (e.g., 500mg), and Scheduled Times (e.g., 09:00, 21:00).
- **Dose Logging:** A simple "Check" interface allows users to mark a dose as taken.
- **Adherence Logic:** The system calculates adherence scores:  $(\text{Doses Taken} / \text{Total Scheduled Doses}) * 100$ .

## 4.4 Appointment Scheduler

- **Booking System:** Interface to schedule visits with Title, Date, Time, and Description.
- **Conflict Detection:** The backend algorithm checks existing appointments. If a new booking overlaps ( $\pm 30$  mins) with an existing one, the system rejects it and alerts the user.

## 4.5 AI Health Assistant (Chatbot)

- **Rule-Based AI:** A Python-based chatbot that processes user input for keywords.

### Capabilities:

- **Symptom Check:** Responds to keywords like "fever", "headache".
- **Crisis Detection:** Detects words like "suicide" or "depressed" and provides helpline resources.
- **General Chat:** Friendly greetings and health tips.
- **Disclaimer:** Clearly marked as "Not Medical Advice" for safety.

## 4.6 Notifications & Background Jobs

- **APScheduler:** A background process runs independently of the web requests.
- **Email Reminders:** (Configurable) Scans the database every hour to send email alerts for missed doses or upcoming appointments.

---

# 5. Database Design

---

The system uses a Relational Database with the following entities:

1. **User:** `id`, `email`, `password_hash`, `name`
  2. **Medicine:** `id`, `user_id`, `name`, `dose`, `times` (stored as comma-separated string)
  3. **DoseLog:** `id`, `medicine_id`, `timestamp`, `taken` (Boolean)
  4. **Appointment:** `id`, `user_id`, `title`, `datetime`, `description`
-

## 6. User Interface (UI) Design

---

### 6.1 Design Philosophy

- **Modern & Clean:** Uses a "Glassmorphism" aesthetic with translucent cards and soft gradients.
- **Theme:** Blue and Purple gradients (#667eea to #764ba2) invoke a sense of calm and trust (Color Psychology).
- **Responsive:** Built with Bootstrap 5, ensuring the app works on Desktops, Tablets, and Mobile phones.

### 6.2 Key UI Components

- **Landing Page:** Features a 3D isometric illustration and clear value propositions.
  - **Glass Cards:** Login/Register forms float on a vibrant mesh gradient background.
  - **Interactive Elements:** Hover effects, smooth transitions, and distinct badges for status updates.
- 

## 7. Testing & Quality Assurance

---

- **Unit Testing:** `pytest` is used to test individual functions (e.g., adherence calculation).
  - **Integration Testing:** Tests the full flow from API to Database (e.g., creating a user and logging them in).
  - **CI/CD:** A GitHub Actions workflow (`ci.yml`) automatically runs tests on every push to ensure code stability.
- 

## 8. Conclusion

---

MediTrack successfully digitizes the personal health management process. By combining a secure backend with an intuitive, aesthetically pleasing frontend, it empowers users to take control of their medical routines. The inclusion of AI support and intelligent conflict detection makes it a robust solution for modern healthcare needs.

---

*Generated by MediTrack Automated Reporting System*

---