



# INSTITUTE OF AERONAUTICAL ENGINEERING (AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

## Complex Problem-Solving Self-Assessment Form

1	Name of the Student	VADALA ANJANA
2	Roll Number	25951A6626
3	Branch and Section	CSE-(AI&ML) - A
4	Program	B. Tech
5	Course Name	Front-End Web Development
6	Course Code	ACSE04
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles	
EC	Profiles	(✓)
EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	✓
EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	
EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	✓
EC 5	Conceptualises alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	✓
EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	

	EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	
	EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	✓
	EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	

	EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	✓
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	✓
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	✓

8	Please tick (✓) relevant Course Outcomes (COs) Covered		
	CO	Course Outcomes	(✓)
	CO 1	Describe language basics like alphabet, strings, grammars, productions, derivations, and Chomsky hierarchy, construct DFA, NFA, and conversion of NFA to DFA, Moore and Mealy machines and interpret differences between them.	✓
	CO 2	Recognize regular expressions, formulate, and build equivalent finite automata for various languages.	✓
	CO 3	Identify closure, and decision properties of the languages and prove the membership.	✓
	CO4	Demonstrate context-free grammars, check the ambiguity of the grammar, and design equivalent PDA to accept the context-free languages.	
	CO 5	Uses mathematical tools and abstract machine models to solve complex problems.	✓
	CO 6	Analyze and distinguish between decidable and undecidable problems.	✓
9	Course ELRVVideo Lectures Viewed		Number of Videos
			Viewing time in Hours
10	Justify your understanding of WK1		-

11	Justify your understanding of WK2 – WK9	-
	How many WKS from WK2 to WK9 were implanted?	-
12	Mention them	-

Date: 13-12-2025

Vadala Anjana

Signature of the Student

**COMPLEX ENGINEERING PROBLEM**

**A COURSE SIDE PROJECT**

**ON**

**Front-End Web Development**

*Vadala Anjana*

**25951A6626**

# **ExamEase**

*A Project Report submitted  
in partial fulfillment of the*

*requirements for the award of the degree of*

**Bachelor of Technology  
in**

**CSE (Artificial Intelligence & Machine Learning)**

*By*

**Vadala Anjana**

**25951A6626**



**Department of CSE (Artificial Intelligence & Machine Learning)**

**INSTITUTE OF AERONAUTICAL ENGINEERING**

**(Autonomous)**

**Dundigal, Hyderabad – 500 043, Telangana**

**November, 2025**

2025, Vadala Anjana, All rights reserved.

## **DECLARATION**

I certify that

- a. The work contained in this report is original and has been done by me under the guidance of my supervisor (s).
- b. The work has not been submitted to any other Institute for any degree or diploma.
- c. I have followed the guidelines provided by the Institute for preparing the report.
- d. I have conformed to the norms and guidelines given in the Code of Conduct of the Institute.
- e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Vadala Anjana

**Place: Hyderabad**

**Signature of the Student**

**Date: 05-12-2025**

## **CERTIFICATE**

This is to certify that the project report entitled **ExamEase** submitted by **Vadala Anjana** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **CSE - (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** is a Bonafide record of work carried out by his guidance and supervision.

The Contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

**Supervisor**

**Date: 13-12-2025**

**Head of the Department**

**Principal**

## **APPROVAL SHEET**

This project report entitled **ExamEase** submitted by **Vadala Anjanais** approved for the award of the Degree Bachelor of Technology in Branch **CSE (Artificial Intelligence & Machine Learning)**.

**Examiner**

**Supervisor(s)**

**Principal**

**Date: 13-12-2025**

**Place: Hyderabad**

## **ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide **Mr. Vidyasagar Vidapu, Assistant Professor, Department of CSE – (Artificial Intelligence and Machine Learning)**, for his kind help and for giving me the necessary guidance and valuable suggestions for this project work.

I am grateful to **Dr. M. Purushotham Reddy, Professor and Head of the Department, Department of CSE (Artificial Intelligence & Machine Learning)**, for extending his support to carry on this project work. I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

I express my sincere gratitude to **Dr. L. V. Narasimha Prasad, Professor and Principal** who has been a great source of information for my work.

I thank our college management and respected **Sri M. Rajashekhar Reddy, Chairman, IARE, Dundigal** for providing me with the necessary infrastructure to conduct the project work.

I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

## ABSTRACT

ExamEase is an interactive, web-based platform designed to simplify exam preparation through an efficient and user-friendly front-end interface. This project focuses on developing a responsive and engaging learning environment using core web technologies such as **HTML5, CSS3, and JavaScript**.

The aim of ExamEase is to provide students with an accessible digital space where they can practice subject-specific quizzes, track their progress, and receive instant feedback to enhance their academic readiness.

The platform incorporates an intuitive layout that enables smooth navigation and quick access to learning modules. Interactive elements such as dynamic quiz cards, progress bars, timers, and real-time score updates contribute to a more immersive learning experience. By applying modern design principles, including responsive layouts and clean visual hierarchies, ExamEase ensures compatibility across various devices such as smartphones, tablets, and desktop computers.

A key focus of the project is improving user engagement through thoughtfully designed UI/UX components. Visual consistency, color psychology, and minimalistic design elements were implemented to reduce cognitive load and keep learners focused on content. JavaScript-driven interactivity enhances the functionality of the platform by managing quiz logic, storing user progress, and delivering instant results without requiring back-end processes.

Overall, ExamEase demonstrates the practical application of front-end web development skills in creating a meaningful educational tool. It highlights the importance of usability, visual appeal, and interactivity in modern web applications, ultimately providing learners with a convenient and effective method for preparing for exams.

## CONTENTS

<b>Name of Contents</b>	<b>Page No.</b>
Title Page	I
Declaration	II
Certificate	III
Approval Sheet	IV
Acknowledgement	V
Abstract	VI
Contents	VII
Chapter 1- Introduction	8-9
1.1 Problem Statement	8
1.2 Introduction	8
1.3 Requirements	9 -10
1.4 Prerequisites	11
Chapter 2 - Review of Relevant Literature	12
Chapter 3- Methodology	13-18
Chapter 4- Results and Discussions	19
Chapter 5- Conclusions and Future Scope	20

# CHAPTER 1

## INTRODUCTION

### 1.1 Problem Statement

Medication non-adherence is a major challenge in healthcare, particularly among elderly patients, individuals with chronic illnesses, and patients managing multiple prescriptions. Many patients forget to take their medications on time, take incorrect dosages, or fail to follow prescribed schedules due to busy lifestyles, lack of awareness, or complex medication routines. This can lead to poor treatment outcomes, increased hospitalizations, and higher healthcare costs.

Currently, traditional methods such as handwritten notes, pill boxes, or verbal reminders are often unreliable and insufficient to ensure consistent medication adherence. There is a need for a simple, reliable, and user-friendly digital solution that helps patients manage their medications effectively.

The **Medication Reminder System** aims to address this problem by providing an automated platform that allows users to record their medications, set dosage schedules, and receive timely reminders. The system also tracks medication intake history and provides adherence reports, helping users and caregivers monitor compliance. By improving medication adherence, the system supports better health management, reduces missed doses, and enhances overall patient safety and well-being.

### 1.2 Introduction

Medication adherence is a critical factor in effective healthcare management, especially for patients with chronic illnesses, elderly individuals, and those prescribed multiple medications. Forgetting to take medicines on time or taking incorrect dosages can lead to serious health complications, reduced treatment effectiveness, and increased healthcare costs.

A **Medication Reminder** system is a digital solution designed to help users manage their medication schedules efficiently. The application allows users to add prescribed medicines, set dosage details, and schedule reminders at specific times. By providing timely alerts and easy-to-use tracking features, the system ensures that medications are taken correctly and consistently.

The Medication Reminder application aims to improve patient compliance, reduce missed doses, and promote better health outcomes. With features such as daily schedules, adherence tracking, and optional notifications via alarms, email, or SMS, the system supports users in maintaining a disciplined and organized medication routine. This solution serves as a practical tool for enhancing personal healthcare management and supporting long-term treatment plans.

## **1.3 Requirements**

### **1. Functional Requirements (Front-End)**

#### **1.1 Medication Reminder**

#### **Functional Requirements**

##### **User Registration and Login**

Users should be able to register using email/phone number.

Secure login and logout functionality.

Password recovery option.

##### **User Profile Management**

Users can add and update personal details (name, age, gender).

Option to add health-related information (optional).

##### **Medication Management**

Add new medications with details:

Medication name

Dosage

Frequency (daily, weekly, custom)

Start and end date

Edit or delete existing medications.

##### **Reminder Scheduling**

Set reminders for each medication.

Support multiple reminders per day.

Flexible time scheduling.

##### **Notifications and Alerts**

Timely alerts via:

In-app notifications

Email/SMS (optional)

Snooze and dismiss options for reminders.

##### **Medication Tracking**

Mark medication as taken or missed.

Maintain daily, weekly, and monthly adherence records.

##### **Dashboard and Reports**

Display today's medication schedule.

Visual representation of adherence statistics.

Generate medication history reports.

## Caregiver Support (Optional)

Allow caregivers to monitor medication adherence.

Send alerts to caregivers if doses are missed.

## Non-Functional Requirements

### Usability

Simple and intuitive user interface.

Easy navigation for elderly users.

### Performance

Reminders should trigger without delay.

Support multiple users simultaneously.

### Security

Secure authentication and authorization.

Encrypt sensitive user data.

Role-based access control.

### Reliability

System should function consistently without crashes.

Backup and recovery mechanisms.

### Scalability

Support an increasing number of users and reminders.

### Availability

System should be available 24/7.

Minimal downtime during maintenance.

### Compatibility

Accessible via web and mobile devices.

Compatible with major browsers and operating systems.

## Hardware Requirements

Smartphone or computer

Internet connection

Notification-enabled device (speaker/vibration)

## Software Requirements

Frontend: HTML, CSS, JavaScript (React/Angular optional)

Backend: Node.js / Python / Java

Database: MySQL / MongoDB

Notification Services: Email/SMS APIs

Web browser (Chrome, Firefox, Edge)

## Constraints

Dependence on internet connectivity for notifications.

SMS/Email notifications may incur additional costs.

User must grant notification permissions.

#### Assumptions

Users will input correct medication details.

Devices support notifications.

Users regularly check the application.

### 1.4 Pre-requisites

#### Hardware Requirements

Desktop/Laptop or Smartphone

Minimum 2 GB RAM (4 GB recommended)

Stable Internet connection

Notification-enabled device (sound/vibration support)

#### Software Requirements

Operating System: Windows / Linux / macOS / Android / iOS

Web Browser: Google Chrome, Mozilla Firefox, Edge, or Safari

Development Environment (if applicable):

IDE: VS Code / Android Studio

Backend: Node.js / Python / Java

Database: MySQL / MongoDB / Firebase

#### Notification Services:

Email (SMTP)

SMS or Push Notification API (optional)

#### Technical Knowledge Requirements

Basic understanding of:

Web or Mobile Application Development

HTML, CSS, JavaScript (for web-based systems)

Backend logic and API integration

Database management concepts

Familiarity with:

Scheduling and time-based triggers

Authentication and user management

Functional Pre-requisites

User registration and login mechanism

System clock synchronization

Ability to store medication data (name, dosage, time)

Reminder scheduling functionality

Alert/notification mechanism

## User Requirements

Basic digital literacy

Ability to set medication schedules

Permission to enable notifications on the device

## Security and Privacy Pre-requisites

Secure user authentication

Data encryption for sensitive information

Compliance with basic data privacy standards

Secure database access

## Optional Pre-requisites

Cloud hosting services

Multi-language support

Accessibility features for elderly users

Integration with wearable devices

## CHAPTER 2

### REVIEW OF RELEVANT LITERATURE

Several researchers have highlighted that medication non-adherence leads to poor disease control, increased hospitalizations, and higher mortality rates. Osterberg and Blaschke reported that nearly 50% of patients with chronic illnesses do not take medications as prescribed.

Factors contributing to non-adherence include forgetfulness, complex medication schedules, lack of awareness, side effects, and inadequate patient-provider communication. Among these factors, forgetfulness has been identified as one of the most common and preventable causes, making reminder-based interventions highly relevant.

Recent literature emphasizes the effectiveness of web-based and smartphone applications that combine reminders with additional features such as medication tracking, adherence analytics, and user-friendly dashboards. These systems allow patients to log medication intake, receive notifications, and review adherence history. Research indicates that applications offering feedback and progress visualization are more effective than simple reminder alerts.

Some studies also highlight the role of personalization in reminder systems. Customizable alert tones, flexible scheduling, and language support have been shown to increase user engagement.

Integration with cloud databases enables secure storage and retrieval of medication data, supporting continuity of care

## CHAPTER 3

### METHODOLOGY

The application underwent **testing** for functionality, responsiveness, and data persistence. User feedback guided refinements to ensure intuitive navigation and effective task management. Finally, the web application was deployed using **GitHub Pages**, with version control managed via **Git & GitHub**.

This methodology ensures a **lightweight, front-end focused solution** that supports student productivity through interactive planning, real-time reminders, and progress tracking, with potential for future enhancements such as analytics and AI-based study recommendations.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>MedicationReminder</title>
  <style>
    :root {
      --bg:#f6f8fb; --card:#fff; --muted:#6b7280; --accent:#2563eb; --success:#10b981; --
      danger:#ef4444;
      --glass: rgba(255,255,255,0.6);
    }
    [data-theme="dark"]{
      --bg:#0b1020; --card:#0f1724; --muted:#9aa4b2; --accent:#60a5fa; --success:#34d399; --
      danger:#f87171; --glass: rgba(255,255,255,0.03);
    }
    *{box-sizing:border-box}
    body{margin:0;font-family:Inter,ui-sans-serif,system-ui,-apple-system,"Segoe
    UI",Roboto,"Helvetica Neue",Arial; background:linear-gradient(180deg,var(--bg),#e9eef8);min-
    height:100vh;color:var(--muted)}
    container{max-width:1100px;margin:24px auto;padding:18px}
    header{display:flex;gap:12px;align-items:center;justify-content:space-between}
    h1{margin:0;font-size:20px;color:var(--accent)}
    .controls{display:flex;gap:8px;align-items:center}
    .card{background:var(--card);border-radius:12px;padding:14px;box-shadow:0 6px 18px
    rgba(16,24,40,0.06);}
    .grid{display:grid;grid-template-columns:1fr 360px;gap:16px;margin-top:16px}
    @media (max-width:900px){.grid{grid-template-columns:1fr}}
    /* Form */
    form .row{display:flex;gap:8px;margin-bottom:8px}
    label{font-size:12px;color:var(--muted);display:block;margin-bottom:6px}
    input[type=text], input[type=time], select{width:100%;padding:8px;border-radius:8px;border:1px
    solid #e6e9ef;background:transparent}
    input[type=number]{width:120px}
    .btn{padding:8px 12px;border-radius:10px;border:0;background:var(--accent);color:white;cursor:pointer}
    .btn.ghost{background:transparent;color:var(--accent);border:1px solid rgba(0,0,0,0.06)}
    .med-list{display:flex;flex-direction:column;gap:10px;margin-top:10px}
    .med{display:flex;align-items:center;justify-content:space-between;padding:10px;border-
    radius:10px;border:1px dashed rgba(0,0,0,0.03)}
```

```

.med .meta {display:flex;align-items:center;gap:12px}
.pill {padding:6px 8px;border-radius:8px;background:var(--glass);color:var(--muted);font-weight:600}
.small {font-size:12px;color:var(--muted) }

/* Dashboard */
.stats {display:flex;gap:8px}
.stat {flex:1;padding:12px;border-radius:10px;background:linear-gradient(180deg,rgba(255,255,255,0.02),transparent);}
.progress {height:10px;background:#e6e9ef;border-radius:8px;overflow:hidden}
.progress > i {display:block;height:100%;background:var(--accent)}

/* Agenda */
.agenda {margin-top:12px}
.agenda ul {list-style:none;padding:0;margin:0;display:flex;flex-direction:column;gap:8px}
.agenda li {display:flex;justify-content:space-between;align-items:center;padding:8px;border-radius:8px}

footer {margin-top:16px;text-align:center;color:var(--muted);font-size:13px}

.darktoggle {display:inline-flex;align-items:center;gap:8px;padding:6px 8px;border-radius:999px;background:var(--glass);cursor:pointer}

.actions {display:flex;gap:8px}
.tiny {font-size:12px;padding:6px 8px;border-radius:8px}

/* modal */
.modal {position:fixed;inset:0;display:none;align-items:center;justify-content:center;background:rgba(2,6,23,0.5)}
.modal.open {display:flex}
.modal .card {width:560px;max-width:95%}

</style>
</head>
<body>
<div class="container">
  <header>
    <div>
      <h1>MedicationReminder</h1>
      <div class="small">Manage meds • Schedule reminders • Track adherence</div>
    </div>
    <div class="controls">
      <div class="darktoggle" id="themeToggle">🌙 <span id="themeLabel">Dark</span></div>
      <button class="btn" id="exportCsv">Export CSV</button>
      <button class="btn ghost" id="importBtn">Import</button>
    </div>
  </header>

  <div class="grid">
    <main>
      <div class="card">
        <h3 style="margin:0 0 8px 0">Add / Edit Medication</h3>
        <form id="medForm">
          <input type="hidden" id="medId" />
          <div class="row">
            <div style="flex:1">
              <label for="name">Medication name</label>
              <input id="name" type="text" placeholder="e.g. Metformin" required />
            </div>
            <div>
              <label for="dose">Dose</label>
              <input id="dose" type="text" placeholder="e.g. 500 mg" />
            </div>
          </div>
          <div class="row">
            <div style="flex:1">
              <label for="times">Times (select or add times)</label>
              <div style="display:flex;gap:8px;align-items:center">
                <input id="newTime" type="time" />
                <button class="btn" type="button" id="addTimeBtn">Add time</button>
              </div>
              <div id="timesList" style="margin-top:8px;display:flex;flex-wrap:wrap;gap:6px"></div>
            </div>
            <div style="width:160px">

```

```

<label for="days">Days</label>
<select id="days">
  <option value="daily">Daily</option>
  <option value="weekdays">Weekdays (Mon-Fri)</option>
  <option value="custom">Custom</option>
</select>
</div>
</div>
<div class="row" style="align-items:center">
  <div style="flex:1">
    <label for="notes">Notes (optional)</label>
    <input id="notes" type="text" placeholder="e.g. take with food" />
  </div>
  <div style="display:flex;flex-direction:column;gap:6px">
    <label class="small">Sound</label>
    <select id="tone" style="width:160px">
      <option value="">Default</option>
      <option value="beep">Beep</option>
      <option value="chime">Chime</option>
    </select>
  </div>
</div>
<div style="display:flex;gap:8px;margin-top:10px">
  <button class="btn" type="submit">Save Medication</button>
  <button class="btn ghost" type="button" id="clearForm">Clear</button>
</div>
</form>

<div class="med-list" id="medList"></div>
</div>

<div class="card agenda" id="agendaCard">
  <h3 style="margin:0 0 8px 0">Today's Agenda</h3>
  <ul id="agendaList"></ul>
</div>
</main>

<aside>
  <div class="card">
    <h3 style="margin:0 0 8px 0">Dashboard</h3>
    <div class="stats">
      <div class="stat">
        <div class="small">Medications</div>
        <div style="font-size:22px" id="statMeds">0</div>
      </div>
      <div class="stat">
        <div class="small">Today's doses</div>
        <div style="font-size:22px" id="statDoses">0</div>
      </div>
      <div class="stat">
        <div class="small">Adherence (30d)</div>
        <div style="font-size:22px" id="statAdherence">0%</div>
        <div class="progress" style="margin-top:8px"><i id="adherenceBar" style="width:0%"></i></div>
      </div>
    </div>
  </div>

  <div style="margin-top:12px">
    <h4 style="margin:8px 0">Quick Actions</h4>
    <div class="actions">
      <button class="tiny" id="markAllTaken">Mark all due as taken</button>
      <button class="tiny" id="snoozeAll">Snooze 10 min</button>
    </div>
  </div>
</div>

<div class="card" style="margin-top:12px">
  <h3 style="margin:0 0 8px 0">Settings & Export</h3>
  <div style="display:flex;flex-direction:column;gap:8px">
    <label class="small">Export logs</label>
    <button class="btn" id="exportLog">Export Logs (CSV)</button>
    <button class="btn ghost" id="clearData">Clear All Data</button>
  </div>
</div>

```

```
</aside>
</div>
```

```
<footer>MedicationReminder • Works offline in browser • Uses notifications (allow when prompted)</footer>
```

```
<div class="modal" id="modal">
  <div class="card">
    <h3 id="modalTitle">Confirm</h3>
    <p id="modalText">Are you sure?</p>
    <div style="display:flex;gap:8px;justify-content:flex-end;margin-top:10px">
      <button class="btn ghost" id="modalCancel">Cancel</button>
      <button class="btn" id="modalOk">OK</button>
    </div>
  </div>
</div>
</div>
```

```
<audio id="sound-beep"><source
src="data:audio/wav;base64,UkIGRIQAAABXQVZFZm10IBAAAAAABAAEAIYAAESsAACAB
AAZGF0YQAAAAA=" type="audio/wav"></audio>
<audio id="sound-chime"><source
src="data:audio/wav;base64,UkIGRhQAAABXQVZFZm10IBAAAAAABAAEAIYAAESsAACAB
AAZGF0YQAAAAA=" type="audio/wav"></audio>
```

```
<script>
  // Basic app - single-file progressive web app behaviour
  (function(){
    const $ = sel => document.querySelector(sel);
    const $$ = sel => Array.from(document.querySelectorAll(sel));

    // Data model in localStorage
    const DB_KEY = 'med_reminder_db_v1';
    let db = { meds: [], logs: [] };

    function saveDB() { localStorage.setItem(DB_KEY, JSON.stringify(db)); render(); }
    function loadDB() { const raw = localStorage.getItem(DB_KEY); if(raw) db = JSON.parse(raw); }

    // Utils
    function uid() { return 'm'+Math.random().toString(36).slice(2,9); }
    function nowISO() { return new Date().toISOString(); }

    // Permission for notifications
    async function ensureNotifications() { if('Notification' in window) { if(Notification.permission === 'default') await Notification.requestPermission(); } }

    // Time handling
    function timeStrToDate(timeStr, date = new Date()) {
      const [hh,mm] = timeStr.split(':').map(Number);
      const d = new Date(date);
      d.setHours(hh,mm,0,0); return d;
    }

    // CRUD
    function addMed(med) { med.id = uid(); db.meds.push(med); saveDB(); }
    function updateMed(id, patch) { const m = db.meds.find(x=>x.id===id); Object.assign(m, patch); saveDB(); }
    function deleteMed(id) { db.meds = db.meds.filter(x=>x.id!==id); saveDB(); }

    // Rendering
    function render() {
      // Med list
      const medList = $('#medList'); medList.innerHTML="";
      db.meds.forEach(m=>{
        const el = document.createElement('div'); el.className='med';
        const meta = document.createElement('div'); meta.className='meta';
        meta.innerHTML = `<div style="min-width:90px"><strong style="color:var(--accent)">${m.name}</strong><div class="small">${m.dose} ${m.unit}</div></div>`;
        const times = document.createElement('div'); times.innerHTML = `<div class="pill">${m.times.join(', ')}</div><div class="small">${m.days} daily</div>`;
        meta.appendChild(times);
        const actions = document.createElement('div');
        actions.innerHTML = `<button class="tiny" data-id="${m.id}" data-act="edit">Edit</button>`;
        el.appendChild(meta);
        el.appendChild(actions);
        medList.appendChild(el);
      });
    }
  })();

```

```

<button class="tiny" data-id="$ {m.id}" data-act="delete">Delete</button>';
el.appendChild(meta); el.appendChild(actions);
medList.appendChild(el);
});

// Stats
$('#statMeds').textContent = db.meds.length;
const todays = computeTodaysDoses();
$('#statDoses').textContent = todays.length;

// Adherence last 30 days
const adh = computeAdherence(30);
$('#statAdherence').textContent = Math.round(adh)+'%';
$('#adherenceBar').style.width = Math.round(adh)+'%';

// Agenda
const agenda = $('#agendaList'); agenda.innerHTML="";
todays.forEach(item=>{
  const li = document.createElement('li');
  li.innerHTML = `<div><strong>$ {item.time}</strong> · <span
class="small">$ {item.med.name} $ {item.med.dose}''</span></div>
<div><button class="tiny" data-id="$ {item.id}" data-act="taken">Taken</button>
<button class="tiny" data-id="$ {item.id}" data-act="snooze">Snooze</button></div>`;
<button class="tiny" data-id="$ {item.id}" data-act="snooze">Snooze</button></div>';
  agenda.appendChild(li);
});

// Compute doses scheduled for today
function computeTodaysDoses(){
  const out=[]; const today = new Date();
  db.meds.forEach(m=>{
    // days: daily, weekdays, custom(not implemented)
    const include = (m.days==='weekdays')? (today.getDay()!==0 && today.getDay()!==6) : true;
    if(!include) return;
    (m.times||[]).forEach(t=>{
      out.push({ id: m.id+'|'+t, med:m, time:t });
    });
  });
  // sort by time
  out.sort((a,b)=> a.time.localeCompare(b.time));
  return out;
}

// Logs: mark taken
function markTaken(itemId){ const [mid, time] = itemId.split('|'); db.logs.push({ medId: mid,
time, at: nowISO() }); saveDB(); }

function computeAdherence(days){
  // approximate: count taken doses in last N days divided by expected doses
  const end = new Date(); const start = new Date(); start.setDate(end.getDate() - days + 1);
  // expected: for each day count doses
  let expected=0, taken=0;
  for(let d = new Date(start); d<=end; d.setDate(d.getDate()+1)){
    db.meds.forEach(m=>{
      const include = (m.days==='weekdays')? (d.getDay()!==0 && d.getDay()!==6) : true;
      if(include) expected += (m.times||[]).length;
    });
  }
  db.logs.forEach(l=>{
    const at = new Date(l.at); if(at >= start && at <= end) taken++;
  });
  if(expected===0) return 0; return (taken/expected)*100;
}

// Scheduler loop (runs every 15s) - checks for due meds in the next minute
let snoozedUntil = null;
function schedulerTick(){
  if(snoozedUntil && new Date() < snoozedUntil) return;
  const now = new Date();
  const nowMin = now.getHours().toString().padStart(2,'0')+':'+
now.getMinutes().toString().padStart(2,'0');
  db.meds.forEach(m=>{
    (m.times||[]).forEach(t=>{
      if(t==nowMin){ // due now
        ...
      }
    });
  });
}

```

```

const key = 'notified'+m.id+'|'+t+'|'+(new Date()).toDateString();
if(!localStorage.getItem(key)){
    notifyDue(m,t); localStorage.setItem(key,'1');
}
};

function notifyDue(med,time){
    // Browser Notification
    if('Notification' in window && Notification.permission === 'granted'){
        const n = new Notification('Medication due: '+med.name, { body: `${med.dose} ${time}` });
    };
    n.onclick = ()=> window.focus();
}
// Play sound
const tone = med.tone || '';
if(tone){ const audio = document.getElementById('sound-'+tone); if(audio)
audio.play().catch(()=>{}); }
// Add to UI (log as ringing)
showModal(`Time to take ${med.name} (${med.dose}) at ${time}`, ()=>{
markTaken(med.id+'|'+time); });
}

// Modal helper
function showModal(text, okCb){ const modal = $('#modal'); $('#modalText').textContent = text;
modal.classList.add('open');
$('#modalOk').onclick = ()=>{ modal.classList.remove('open'); okCb && okCb(); };
$('#modalCancel').onclick = ()=> modal.classList.remove('open');
}

// Form handling
$('#addTimeBtn').addEventListener('click', ()=>{
    const t = $('#newTime').value; if(!t) return;
    const list = $('#timesList');
    if(Array.from(list.children).some(c=>c.dataset.time==t)) return;
    const chip = document.createElement('div'); chip.className='pill'; chip.style.cursor='pointer';
    chip.dataset.time=t; chip.textContent = t + ' X';
    chip.onclick = ()=> chip.remove();
    list.appendChild(chip);
    $('#newTime').value="";
});

$('#medForm').addEventListener('submit', e=>{
    e.preventDefault();
    const id = $('#medId').value; const name = $('#name').value.trim(); const dose =
$('#dose').value.trim();
    const times = Array.from($('#timesList').children).map(c=>c.dataset.time);
    const days = $('#days').value; const notes = $('#notes').value; const tone = $('#tone').value;
    if(!name || times.length==0){ alert('Please add a name and at least one time'); return; }
    const med = { name, dose, times, days, notes, tone };
    if(id) { updateMed(id, med); } else addMed(med);
    $('#medForm').reset(); $('#timesList').innerHTML=""; $('#medId').value="";
});

// Click handlers for med list and agenda
document.body.addEventListener('click', e=>{
    const btn = e.target.closest('button'); if(!btn) return; const act = btn.dataset.act; const id =
btn.dataset.id;
    if(act==='edit'){ const m = db.meds.find(x=>x.id==id); if(!m) return; $('#medId').value = m.id;
    $('#name').value = m.name; $('#dose').value = m.dose; $('#notes').value = m.notes || '';
    $('#tone').value = m.tone||''; $('#days').value = m.days||'daily'; $('#timesList').innerHTML =
(m.times||[]).map(t=>{ const chip=document.createElement('div'); chip.className='pill';
chip.dataset.time=t; chip.textContent=t+' X'; chip.onclick=()=>chip.remove();
$('#timesList').appendChild(chip); }).join('');
    window.scrollTo({top:0,behavior:'smooth'}); }
    if(act==='delete'){ confirmModal('Delete medication?', ()=>{ deleteMed(id); }); }
    if(act==='taken'){ markTaken(id); }
    if(act==='snooze'){ snoozedUntil = new Date(Date.now()+10*60*1000); alert('Snoozed 10
minutes'); }
});

function confirmModal(text, okCb){ $('#modalTitle').textContent='Confirm';

```

```

$( '#modalText' ).textContent = text; $( '#modal' ).classList.add('open'); $( '#modalOk' ).onClick = ()=>{
$( '#modal' ).classList.remove('open'); okCb && okCb(); }; $( '#modalCancel' ).onClick = ()=>
$( '#modal' ).classList.remove('open'); }

// Quick actions
$( '#markAllTaken' ).addEventListener('click', ()=>{
computeTodaysDoses().forEach(it=>markTaken(it.id)); alert('Marked all due as taken'); });
$( '#snoozeAll' ).addEventListener('click', ()=>{ snoozedUntil = new Date(Date.now() + 10*60*1000); alert('Snoozed 10 minutes'); });

// Export CSV
function exportCSV(){
let rows = [['medId','name','dose','time','takenAt']];
db.logs.forEach(l=>{
const med = db.meds.find(m=>m.id === l.medId) || {};
rows.push([l.medId, med.name||'', med.dose||'', l.time, l.at]);
});
const csv = rows.map(r=>r.map(c=>""+String(c||").replace(/\//g,"")+"")).join(',')).join('\n');
const blob = new Blob([csv], {type:'text/csv'}); const url = URL.createObjectURL(blob);
const a = document.createElement('a'); a.href=url; a.download='med_logs.csv'; a.click();
URL.revokeObjectURL(url);
}

$( '#exportLog' ).addEventListener('click', exportCSV);
$( '#exportCsv' ).addEventListener('click', ()=>{
// export medication list
let rows = [['id','name','dose','times','days','notes','tone']];
db.meds.forEach(m=> rows.push([m.id,m.name,m.dose, (m.times||[]).join('|'), m.days||'', m.notes||'', m.tone||']));
const csv = rows.map(r=>r.map(c=>""+String(c||").replace(/\//g,"")+"")).join(',')).join('\n');
const blob = new Blob([csv], {type:'text/csv'}); const url = URL.createObjectURL(blob);
const a = document.createElement('a'); a.href=url; a.download='medications.csv'; a.click();
URL.revokeObjectURL(url);
});

$( '#clearData' ).addEventListener('click', ()=>{ confirmModal('Clear all medications and logs? This cannot be undone.', ()=>{ db = {meds:[], logs:[]} ; saveDB(); }); });

$( '#importBtn' ).addEventListener('click', ()=>{ const txt = prompt('Paste medications CSV (id,name,dose,times,days,notes,tone)'); if(!txt) return; const lines = txt.split('\n').slice(1);
lines.forEach(l=>{ const parts = l.split(','); if(parts.length>=3){ const name =
parts[1].replace(/\//g,"").trim(); const dose = parts[2].replace(/\//g,"").trim(); const times =
(parts[3]||"").replace(/\//g,"").split('|').filter(Boolean); addMed({name,dose,times,days:parts[4]||'daily'}); }
}); alert('Imported (best effort)'); });

$( '#clearForm' ).addEventListener('click', ()=>{ $( '#medForm' ).reset();
$( '#timesList' ).innerHTML=""; $( '#medId' ).value=""; });

// theme
const themeToggle = $( '#themeToggle' ); themeToggle.addEventListener('click', ()=>{
const root = document.documentElement; if(root.getAttribute('data-theme')==='dark'){
root.removeAttribute('data-theme'); $( '#themeLabel' ).textContent='Dark'; } else {
root.setAttribute('data-theme','dark'); $( '#themeLabel' ).textContent='Light'; }
});

// load & init
loadDB(); render(); ensureNotifications(); setInterval(schedulerTick, 15000); // check every 15s

// initial scheduler tick to catch immediate times
schedulerTick();

// ask for permission on first load
if('Notification' in window && Notification.permission === 'default'){
const ask = confirm('Enable browser notifications for reminders?'); if(ask)
Notification.requestPermission();
}

// prevent accidental data loss
window.addEventListener('beforeunload', e=>{ if(db.meds.length || db.logs.length){
e.returnValue = 'You have saved data in MedicationReminder.'; } });

})

</script>
</body>
</html>

```

Output:

The screenshot shows the MedicationReminder application interface. On the left, the 'Add / Edit Medication' screen is displayed, allowing users to input medication details like name, dose, times, notes, and sound settings. It includes a 'Save Medication' button and a 'Clear' button. On the right, the 'Dashboard' screen provides a summary of medication counts, today's doses, and adherence percentages. It also features 'Quick Actions' for marking all due as taken or snoozing for 10 minutes, and a 'Settings & Export' section for exporting logs.

## CHAPTER 4

### RESULTS AND DISCUSSIONS

The results highlight that the Medication Reminder system effectively addresses the problem of missed or delayed medication intake. By providing timely alerts and a structured way to manage medication schedules, the system reduces reliance on memory and manual tracking.

One of the key strengths of the system is its adherence monitoring capability. The ability to view adherence statistics encourages users to follow prescribed schedules more consistently. This feature can be particularly beneficial for patients with chronic conditions who require long-term medication management.

The reminder mechanism proved to be reliable; however, its effectiveness depends on external factors such as device availability, notification permissions, and user responsiveness. While the system ensures reminders are sent, actual medication intake still relies on user compliance.

From a usability perspective, the clean interface and organized dashboards enhance user engagement.

Simple navigation and clear alerts reduce cognitive load, making the application suitable for a wide range of users, including elderly patients with basic technical skills.

Despite the positive outcomes, certain limitations were observed. The current system does not automatically verify medication intake and relies on user input for adherence tracking. Additionally, features such as SMS-based reminders, caregiver notifications, and integration with wearable or healthcare systems could further improve effectiveness.

Overall, the results demonstrate that the Medication Reminder system is a practical and efficient solution for medication management. With further enhancements and real-world deployment, it has the potential to significantly improve patient adherence and support better health outcomes.

## **CHAPTER 5**

### **CONCLUSION AND FUTURE SCOPE**

#### **5.1 Conclusions**

The Medication Reminder system effectively addresses the common problem of missed or incorrect medication intake by providing a simple, reliable, and user-friendly digital solution. By allowing users to schedule medications, set timely reminders, and track adherence, the system supports better health management and reduces the risk of medication-related complications.

The application enhances patient awareness and responsibility, particularly for elderly users and individuals with chronic conditions who require strict medication routines. Features such as customizable alerts, adherence tracking, and clear dashboards improve usability and encourage consistent medication habits.

Overall, the Medication Reminder system demonstrates how technology can play a vital role in improving healthcare outcomes. With future enhancements such as SMS/email notifications, integration with wearable devices, and doctor or caregiver access, the system can be further strengthened to provide comprehensive support for medication management and patient well-being.







