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 obvioussolution 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 tojustify 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 engineeringapplication inthe 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	-
12	How many WKs from WK2 to WK9 were implanted?	-
	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

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 **Head of the Department**

Date: 13-12-2025

Principal

APPROVAL SHEET

This project report entitled **ExamEase** submitted by **Vadala Anjana**is 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. Rajashekar 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

CHAPTER 1
INTRODUCTION

1.1 Problem Statement

In today's fast-paced world, many individuals experience health symptoms but delay seeking medical attention due to lack of awareness, uncertainty about symptom severity, or limited access to healthcare resources. People often rely on unstructured internet searches, which can lead to misinformation, anxiety, and incorrect self-diagnosis. There is a need for a reliable, user-friendly digital platform that helps users understand their symptoms in a structured manner and guides them toward appropriate next steps without replacing professional medical advice.

SymptomSense aims to address this problem by providing an interactive web-based symptom self-assessment system. The application allows users to input their symptoms and receive possible condition suggestions along with basic guidance and recommendations for seeking medical consultation when necessary. By organizing symptom information and presenting preliminary insights clearly, SymptomSense promotes early health awareness, informed decision-making, and timely medical intervention while ensuring user data privacy and accessibility.

1.2 Introduction

In today's fast-paced world, timely access to healthcare information plays a vital role in maintaining personal well-being. Many individuals often experience symptoms but delay seeking medical advice due to lack of awareness, accessibility issues, or uncertainty about the seriousness of their condition. To address this gap, **SymptomSense** has been developed as an interactive web-based health assessment platform.

SymptomSense enables users to input their symptoms and receive preliminary insights into possible health conditions along with general recommendations for further action. The system is designed to support early awareness and encourage informed decision-making, without replacing professional medical consultation. By providing a structured and user-friendly interface, the platform helps users better understand their health concerns and take timely steps toward appropriate care.

The application leverages symptom-based analysis to offer condition suggestions, health tips, and guidance on whether medical attention may be required. With optional features such as symptom history tracking and personalized recommendations, SymptomSense aims to enhance user engagement and promote proactive health monitoring. Overall, SymptomSense serves as a valuable digital tool for improving health awareness and supporting early intervention in healthcare management.

1.3 Requirements

1. Functional Requirements (Front-End)

1.1 Symptom Sense

Functional Requirements

These define what the system should do.

User Registration and Login

Users should be able to create an account using email/username and password.

Registered users should be able to log in and log out securely.

Symptom Input

The system should allow users to enter one or more symptoms.

Symptoms can be selected from a predefined list or entered manually.

Symptom Analysis

The system should analyze the entered symptoms using a rule-based or database-driven approach.

It should match symptoms with possible medical conditions.

Condition Suggestions

The system should display a list of possible conditions based on the symptoms entered.

Each suggestion should include a brief description.

Recommendations

The system should provide general guidance such as:

Home care suggestions

When to consult a doctor

Emergency warning indicators

Symptom History (Optional Enhancement)

The system may store previous symptom entries for registered users.

Users should be able to view their past symptom checks.

User Interface

The application should provide a simple, user-friendly, and responsive interface.

It should be accessible on desktops, tablets, and mobile devices.

Non-Functional Requirements

These define how the system should perform.

Usability

The system should be easy to use for people of all age groups.

Clear instructions and labels should be provided.

Performance

The system should generate results within a few seconds after symptom submission.

It should handle multiple users simultaneously.

Security

User data should be securely stored and protected.

Passwords should be encrypted.

The system should comply with basic data privacy principles.

Reliability

The system should be available with minimal downtime.

It should handle invalid inputs gracefully.

Scalability

The system should support future enhancements like AI-based diagnosis or chatbot integration.

Maintainability

The system should be modular and easy to update.

Code and database should be well documented.

Hardware Requirements

A computer or mobile device with internet access

Minimum 2 GB RAM (for client device)

Standard input devices (keyboard, mouse, touchscreen)

Software Requirements

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

Backend: Python (Flask/Django) or Node.js

Database: MySQL / MongoDB

Browser: Google Chrome, Mozilla Firefox, or any modern web browser

Operating System: Windows, macOS, or Linux

Constraints

The system is not a replacement for professional medical diagnosis.

Accuracy depends on the quality of symptom data provided by users.

Internet connectivity is required for full functionality

1.4 Pre-requisites

To design, develop, and effectively use the SymptomSense health self-assessment application, the following pre-requisites are required:

1. Technical Knowledge
   - Basic understanding of web technologies such as HTML, CSS, and JavaScript
   - Familiarity with a frontend framework (React, Angular, or Vue) – optional but recommended
   - Knowledge of backend development (Node.js, Python, or PHP)
   - Understanding of database management systems (MySQL, MongoDB, or PostgreSQL)
2. Software Requirements
   - Code editor (e.g., VS Code)
   - Web browser (Google Chrome, Firefox, etc.)
   - Backend runtime environment (Node.js / Python)
   - Database server or cloud database
   - Version control system (Git)
3. Hardware Requirements
   - Computer or laptop with:
   - Minimum 4 GB RAM (8 GB recommended)
   - Stable internet connection
   - Server or cloud hosting platform (optional for deployment)
4. Domain Knowledge
   - Basic awareness of common medical symptoms and conditions
   - Understanding of healthcare terminology
   - Knowledge of ethical considerations in health-related applications
5. Data & Security Requirements
   - Sample symptom and condition datasets
   - Awareness of data privacy and security principles
   - Compliance with basic healthcare data protection standards
6. User Requirements
   - Users should have basic digital literacy
   - Ability to input symptoms accurately
   - Understanding that the application provides preliminary guidance only and is not a medical diagnosis
7. Optional Enhancements Pre-requisites
   - Knowledge of APIs for symptom databases
   - Experience with machine learning (for advanced predictions)
   - Notification services (Email/SMS integration)

CHAPTER 2

REVIEW OF RELEVANT LITERATURE

The rapid growth of digital healthcare technologies has led to the development of various **symptom-checking and health self- assessment systems** aimed at improving early disease detection and health awareness. Several studies and existing systems form the foundation for the development of applications like **SymptomSense**.

Early research in **clinical decision support systems (CDSS)** highlights the importance of computerized tools that assist users in identifying potential health conditions based on symptoms. According to Miller (2019), CDSS applications can enhance healthcare accessibility by providing preliminary guidance and encouraging timely medical consultation. These systems are particularly useful in regions with limited access to healthcare professionals.

Web-based symptom checker applications such as **WebMD Symptom Checker**, **Ada Health**, and **Babylon Health** have been widely studied. Semigran et al. (2015) analyzed the accuracy of online symptom checkers and found that while they are not a replacement for professional diagnosis, they effectively raise awareness and guide users toward appropriate care. This finding supports the core objective of SymptomSense, which focuses on **preliminary self-assessment rather than diagnosis**.

Recent literature emphasizes the role of **user-friendly interfaces** and structured symptom input methods. Studies by Zhang et al. (2020) indicate that applications with simple symptom selection, clear recommendations, and visual dashboards significantly improve user engagement and trust. SymptomSense aligns with these findings by incorporating an intuitive interface and clear health suggestions.

CHAPTER 3

METHODOLGY

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>SymptomSense — Self Assessment</title>
  <style>
    :root{
      --lavender:#E6E6FA; /* light purple */
      --deep-lav:#B39DDB;
      --peach:#FFDAB9; /* interpreted 'leach' as warm peach */
      --muted:#6b6b6b;
      --card-bg:#ffffff;
      --glass: rgba(255,255,255,0.6);
      font-family: Inter, system-ui, -apple-system, 'Segoe UI', Roboto, 'Helvetica Neue', Arial;
    }
    *{box-sizing:border-box}
    body{
      margin:0; min-height:100vh; background: linear-gradient(180deg,var(--lavender) 0%, #F8F5FF
50%); color:#222;
      display:flex; align-items:flex-start; justify-content:center; padding:32px;
    }
    .container{width:100%; max-width:1100px}
    header{display:flex; align-items:center; gap:16px; margin-bottom:18px}
    .brand{display:flex; gap:12px; align-items:center}
    .logo{width:56px; height:56px; border-radius:12px; background:linear-gradient(135deg,var(--
deep-lav), var(--peach)); display:flex; align-items:center; justify-content:center; color:white; font-
weight:700}
    h1{margin:0; font-size:20px}
    p.lead{margin:0; color:var(--muted); font-size:13px}

    .grid{display:grid; grid-template-columns: 1fr 420px; gap:18px}

    /* Card */
    .card{background:var(--card-bg); border-radius:14px; padding:18px; box-shadow:0 6px 20px
rgba(64,44,96,0.08)}
    label{font-size:13px; color:var(--muted)}
    input[type=text], select, textarea{width:100%; padding:10px 12px; margin-top:8px; border-
radius:8px; border:1px solid #eee}
    .symptom-row{display:flex; gap:8px; margin-top:10px}
    .symptom-row input{flex:1}
    .severity{width:140px}
    .btn{background:linear-gradient(90deg,var(--deep-lav), #9575CD); color:white; border:none;
padding:10px 14px; border-radius:10px; cursor:pointer}
    .btn.ghost{background:transparent; color:var(--deep-lav); border:1px dashed
rgba(100,50,150,0.08)}
```

```
    .history-list{max-height:300px; overflow:auto; margin-top:12px}
    .history-item{padding:10px; border-radius:8px; background:linear-gradient(180deg,
rgba(242,234,255,0.8), rgba(255,250,245,0.6)); margin-bottom:8px; font-size:13px}

    .result{margin-top:12px; padding:12px; border-radius:10px; background:linear-
gradient(180deg,var(--peach), #fff)}
    .tags{display:flex; gap:8px; flex-wrap:wrap; margin-top:8px}
    .tag{padding:6px 8px; border-radius:999px; background:var(--deep-lav); color:white; font-
size:12px}

    .footer{margin-top:12px; font-size:12px; color:var(--muted)}

    /* Responsive */
    @media (max-width:980px){
      .grid{grid-template-columns:1fr;}
      .card{padding:14px}
    }

    /* small helpers */
    .muted{color:var(--muted); font-size:13px}
    .flex{display:flex; gap:10px; align-items:center}
    canvas{width:100%; height:160px}
  </style>
</head>
<body>
  <div class="container">
    <header>
      <div class="brand">
        <div class="logo">SS</div>
        <div>
          <h1>SymptomSense</h1>
          <p class="lead">Preliminary self-assessment · Awareness & timely consultation</p>
        </div>
      </div>
      <div style="margin-left:auto" class="muted">Offline demo • Data stored in browser</div>
    </header>

    <div class="grid">
      <!-- Left: main interaction -->
      <div>
        <div class="card">
          <h3>Describe your symptoms</h3>
          <p class="muted">Type symptoms (e.g., cough, fever, headache). Add severity and a short
note.</p>

          <div style="margin-top:12px">
            <label>Symptom</label>
            <div class="symptom-row">
              <input id="symptomInput" type="text" placeholder="e.g., cough" />
              <select id="severity" class="severity">
                <option value="mild">Mild</option>
                <option value="moderate">Moderate</option>
                <option value="severe">Severe</option>
              </select>
              <button id="addBtn" class="btn">Add</button>
            </div>

            <label style="margin-top:12px">Notes (optional)</label>
            <textarea id="note" rows="2" placeholder="Any extra details... (duration, onset,
meds)"></textarea>

            <div style="display:flex; gap:8px; margin-top:12px">
              <button id="assessBtn" class="btn">Assess</button>
              <button id="clearBtn" class="btn ghost">Clear Current</button>
            </div>

            <div class="tags" id="currentTags" aria-live="polite"></div>
          </div>

          <div class="result" id="assessmentResult" aria-live="polite" style="display:none">
            <strong>Possible conditions</strong>
            <div id="conditionsList"></div>
            <div style="margin-top:8px">
              <strong>Recommendations</strong>
              <div id="recommendations"></div>
```

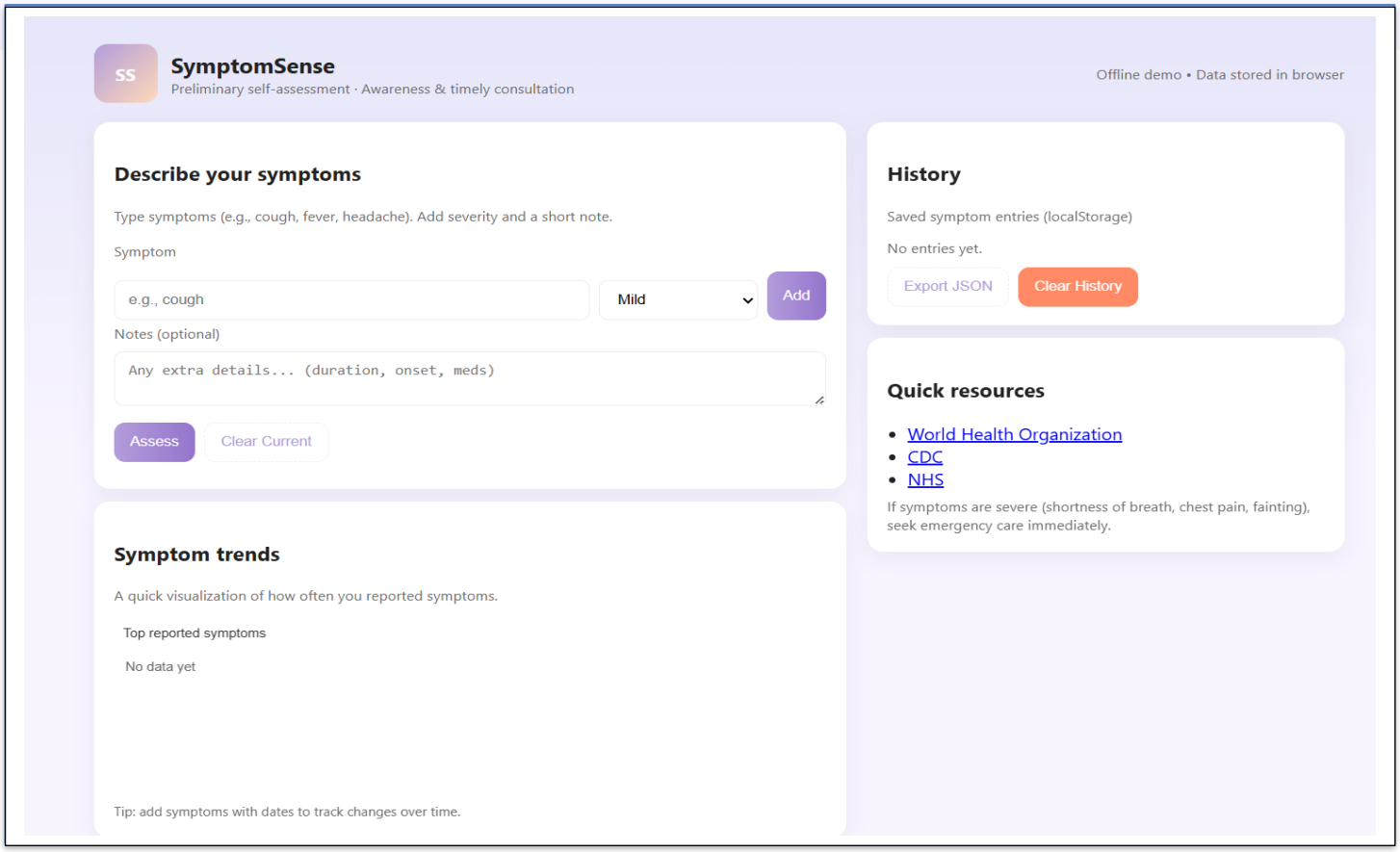
```
          </div>
          <div style="margin-top:8px; font-size:12px" class="muted">Remember: this tool is for informational purposes only and is NOT a medical diagnosis.</div>
        </div>

      </div>

      <div class="card" style="margin-top:12px">
        <h3>Symptom trends</h3>
        <p class="muted">A quick visualization of how often you reported symptoms.</p>
        <canvas id="trendCanvas"></canvas>
        <div class="footer">Tip: add symptoms with dates to track changes over time.</div>
      </div>
    </div>

    <!-- Right: history & dashboard -->
    <aside>
      <div class="card">
        <h3>History</h3>
        <p class="muted">Saved symptom entries (localStorage)</p>
        <div class="history-list" id="historyList"></div>
        <div style="display:flex; gap:8px; margin-top:10px">
          <button id="exportBtn" class="btn ghost">Export JSON</button>
          <button id="clearHistory" class="btn" style="background:#FF8A65">Clear History</button>
        </div>
      </div>

      <div class="card" style="margin-top:12px">
        <h3>Quick resources</h3>
        <ul style="padding-left:18px; margin:6px 0;">
          <li><a href="https://www.who.int/" target="_blank">World Health Organization</a></li>
          <li><a href="https://www.cdc.gov/" target="_blank">CDC</a></li>
          <li><a href="https://www.nhs.uk/" target="_blank">NHS</a></li>
        </ul>
        <div class="muted">If symptoms are severe (shortness of breath, chest pain, fainting), seek emergency care immediately.</div>
      </div>
    </aside>
  </div>
</div>

<script>
  // Simple client-side logic for SymptomSense demo
  const symptomInput = document.getElementById('symptomInput');
  const addBtn = document.getElementById('addBtn');
  const currentTags = document.getElementById('currentTags');
  const assessBtn = document.getElementById('assessBtn');
  const clearBtn = document.getElementById('clearBtn');
  const severity = document.getElementById('severity');
  const note = document.getElementById('note');
  const assessmentResult = document.getElementById('assessmentResult');
  const conditionsList = document.getElementById('conditionsList');
  const recommendations = document.getElementById('recommendations');
  const historyList = document.getElementById('historyList');
  const exportBtn = document.getElementById('exportBtn');
  const clearHistory = document.getElementById('clearHistory');

  let currentSymptoms = [];

  // Example symptom -> condition mapping (simplified). In real apps use validated clinical data.
  const conditionRules = [
    {conds:['fever','cough','fatigue'], name:'Flu-like illness', urgency:'moderate', rec:'Rest, hydrate, consult primary care if persists >48 hrs or worsens.'},
    {conds:['fever','cough','shortness of breath'], name:'Possible respiratory infection (seek medical advice)', urgency:'high', rec:'If shortness of breath or high fever, seek urgent care.'},
    {conds:['headache','stiff neck','fever'], name:'Possible meningitis (rare)', urgency:'high', rec:'Seek emergency care if severe headache with neck stiffness.'},
    {conds:['sore throat','fever','swollen glands'], name:'Tonsillitis / Strep throat', urgency:'moderate', rec:'See GP for testing if severe or persistent.'},
    {conds:['runny nose','sneezing','itchy eyes'], name:'Allergic rhinitis', urgency:'low', rec:'Consider antihistamines; avoid triggers.'},
    {conds:['chest pain','shortness of breath'], name:'Possible cardiac or pulmonary emergency', urgency:'critical', rec:'Call emergency services immediately.'}
  ];
```

```
  // Utility: normalize symptom text
  function norm(s){ return s.trim().toLowerCase(); }

  function renderTags(){
    currentTags.innerHTML = '';
    currentSymptoms.forEach((s, idx) => {
      const t = document.createElement('div');
      t.className = 'tag';
      t.textContent = s.symptom + (s.severity? ' ('+s.severity+')':'');
      t.style.cursor='pointer';
      t.title = s.note||'';
      t.onclick = ()=>{ currentSymptoms.splice(idx,1); renderTags(); };
      currentTags.appendChild(t);
    });
  }

  addBtn.addEventListener('click', ()=>{
    const val = norm(symptomInput.value);
    if(!val) return; // ignore empty
    currentSymptoms.push({symptom:val, severity:severity.value, note:note.value, date:new
Date().toISOString()});
    symptomInput.value=''; note.value=''; renderTags();
  });

  clearBtn.addEventListener('click', ()=>{ currentSymptoms = []; renderTags();
assessmentResult.style.display='none'; });

  assessBtn.addEventListener('click', ()=>{
    if(currentSymptoms.length===0){ alert('Add at least one symptom to assess.'); return; }
    // derive condition matches
    const keys = currentSymptoms.map(s=>s.symptom);
    const found = [];
    conditionRules.forEach(rule=>{
      const matchCount = rule.conds.filter(c=> keys.includes(c)).length;
      // heuristic: at least 2 matches or contains a high-urgency symptom
      if(matchCount>=2 || rule.conds.some(c=> keys.includes(c) && rule.urgency==='critical')){
        found.push(Object.assign({},rule,{matchCount}));
      }
    });

    // generic suggestions if nothing matched
    if(found.length===0){
      found.push({name:'Non-specific symptoms', urgency:'low', rec:'Monitor symptoms. See a
clinician if symptoms persist or worsen.'});
    }

    // Display
    conditionsList.innerHTML=''; recommendations.innerHTML='';
    found.sort((a,b)=> b.matchCount - a.matchCount ).forEach(f=>{
      const d = document.createElement('div');
      d.style.marginTop='8px';
      d.innerHTML = `<div style="font-weight:700">${f.name}</div><div class="muted">Urgency:
${f.urgency}</div>`;
      conditionsList.appendChild(d);

      const r = document.createElement('div'); r.style.marginTop='6px'; r.textContent = f.rec || '';
      recommendations.appendChild(r);
    });

    // Save to history
    const entry = {id:Date.now(), symptoms:currentSymptoms.slice(), conditions:found, created:new
Date().toISOString() };
    saveEntry(entry);

    assessmentResult.style.display='block';
    currentSymptoms = []; renderTags();
    renderHistory();
    drawTrends();
  });

  // History persistence
  function getHistory(){
    try{ return JSON.parse(localStorage.getItem('symptoms_history')||'[]'); } catch(e){ return []; }
  }
  function saveEntry(e){
```

```
    const arr = getHistory(); arr.unshift(e); // newest first
    localStorage.setItem('symptoms_history', JSON.stringify(arr.slice(0,200)));
  }
  function clearAllHistory(){ localStorage.removeItem('symptoms_history'); renderHistory();
drawTrends(); }

  function renderHistory(){
    const arr = getHistory(); historyList.innerHTML='';
    if(arr.length===0){ historyList.innerHTML='<div class="muted">No entries yet.</div>'; return;
}
    arr.forEach(item=>{
      const el = document.createElement('div'); el.className='history-item';
      const date = new Date(item.created).toLocaleString();
      el.innerHTML = `<div style="display:flex; justify-content:space-between;
gap:8px"><div><strong>${item.conditions[0]?.name || 'Assessment'}</strong><div class="muted"
style="font-size:12px">${date}</div></div><div style="text-align:right; font-size:12px"><button
class='btn ghost' data-id='${item.id}'>View</button></div></div>`;
      const details = document.createElement('div'); details.style.marginTop='8px';
details.style.fontSize='13px';
      details.innerHTML = `<div class='muted'>Symptoms:</div> ${item.symptoms.map(s=> `<div
style='display:inline-block; margin:6px 6px 0 0; padding:6px 8px; background:#F3E8FF; border-
radius:8px'>${s.symptom} (${s.severity})</div>`).join('')}`;
      el.appendChild(details);
      historyList.appendChild(el);
    });
    // attach view handlers
    historyList.querySelectorAll('button[data-id]').forEach(b=> b.addEventListener('click',(ev)=>{
      const id = Number(ev.target.getAttribute('data-id'));
      const found = getHistory().find(x=>x.id===id);
      if(found){
        // open a quick modal-like display using alert (simple)
        const lines = [];
        lines.push('Assessment: '+ (found.conditions.map(c=>c.name).join('; ')) );
        lines.push('Symptoms: '+ found.symptoms.map(s=> s.symptom+'('+s.severity+')').join(', '));
        lines.push('Date: '+ new Date(found.created).toLocaleString());
        alert(lines.join('\n'));
      }
    }));
  }

  exportBtn.addEventListener('click', ()=>{
    const data = JSON.stringify(getHistory(), null, 2);
    // create downloadable file
    const blob = new Blob([data], {type:'application/json'});
    const url = URL.createObjectURL(blob);
    const a = document.createElement('a'); a.href = url; a.download = 'symptoms_history.json';
a.click(); URL.revokeObjectURL(url);
  });

  clearHistory.addEventListener('click', ()=>{
    if(confirm('Clear all saved history?')){ clearAllHistory(); }
  });

  // Simple trends chart: counts of most reported symptoms over time
  const canvas = document.getElementById('trendCanvas');
  function drawTrends(){
    const ctx = canvas.getContext('2d');
    // resize canvas to css size
    canvas.width = canvas.clientWidth * devicePixelRatio;
    canvas.height = canvas.clientHeight * devicePixelRatio;
    ctx.scale(devicePixelRatio, devicePixelRatio);
    // gather data
    const arr = getHistory();
    const counts = {};
    arr.forEach(entry=> entry.symptoms.forEach(s=> counts[s.symptom] = (counts[s.symptom]||0) +
1));
    const items = Object.entries(counts).sort((a,b)=> b[1]-a[1]).slice(0,6);
    // clear
    ctx.clearRect(0,0,canvas.width,canvas.height);
    ctx.font = '12px sans-serif';
    ctx.fillStyle = '#333';
    ctx.fillText('Top reported symptoms', 8, 18);
    if(items.length===0){ ctx.fillStyle='#666'; ctx.fillText('No data yet', 10, 50); return; }
    const chartTop = 30; const chartLeft = 10; const chartW = canvas.clientWidth - 20; const barH =
; const gap = 12;
```

```
    const max = Math.max(...items.map(i=>i[1]));
    items.forEach((it,i)=>{
      const y = chartTop + i*(barH+gap);
      const w = (it[1]/max) * (chartW*0.7);
      // bar
      ctx.fillStyle = 'rgba(179,157,219,0.9)';
      ctx.fillRect(chartLeft, y, w, barH);
      // label
      ctx.fillStyle = '#222';
      ctx.fillText(it[0], chartLeft + w + 8, y + 12);
      ctx.fillStyle = '#222';
      ctx.fillText(String(it[1]), chartLeft + w - 28, y + 12);
    });
  }

  // initial render
  renderHistory(); drawTrends();

  // ensure canvas redraw on resize
  window.addEventListener('resize', ()=>{ drawTrends(); });

  // Accessibility: Enter to add
  symptomInput.addEventListener('keydown', (e)=>{ if(e.key==='Enter'){ addBtn.click(); } });

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

Output:



CHAPTER 4

RESULTS AND DISCUSSIONS

The results highlight the effectiveness of SymptomSense as a supportive health information system rather than a diagnostic tool. By focusing on symptom-based assessment, the application bridges the gap between symptom onset and professional medical consultation.

One of the major strengths of the system is its **simplicity and accessibility**, making it suitable for users without medical knowledge. The application promotes informed decision-making by encouraging timely medical attention when necessary. This is particularly beneficial in scenarios where users may ignore early symptoms due to lack of awareness.

However, the system's accuracy depends on the completeness and correctness of user-provided symptom data. Since symptoms can be subjective and vary in severity, there is a possibility of overlapping condition suggestions. This limitation emphasizes that SymptomSense should be used only as a **preliminary assessment tool**, not a replacement for professional diagnosis.

Additionally, the current implementation relies on a static symptom–condition database. While effective for common conditions, it may not cover rare or complex medical cases. Incorporating machine learning models or real-time medical datasets in future versions could enhance adaptability and precision.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusions

SymptomSense is a practical and effective web-based application designed to assist users in performing preliminary self-assessments of their health symptoms. The system successfully provides users with possible condition suggestions and appropriate recommendations based on the symptoms entered, promoting greater health awareness and timely medical consultation.

The project demonstrates how technology can be utilized to bridge the gap between symptom recognition and professional healthcare services. By offering a structured and user-friendly interface, SymptomSense enables users to better understand their symptoms without requiring medical expertise. The application emphasizes guidance rather than diagnosis, ensuring responsible and ethical use.

Although the current implementation relies on a predefined symptom–condition database, it performs reliably for common health scenarios. The system lays a strong foundation for future enhancements, such as integrating machine learning algorithms, expanding medical databases, and adding personalized health tracking features.

In conclusion, SymptomSense fulfills its objectives of improving early symptom awareness and supporting informed healthcare decisions. With further development and refinement, it has the potential to become a valuable digital health support tool that complements traditi