SYMPTOMSENSE

A Web-Based Symptom Analysis System

DOMAIN

Web Application Development / Healthcare Technology

TECHNOLOGIES USED

HTML

CSS

JavaScript

TITLE PAGE

Project Title:

SymptomSense – Intelligent Symptom Analysis Web Application

Domain:

Web Development & Healthcare Systems

ABSTRACT

SymptomSense is a web-based healthcare support application designed to assist users in identifying possible medical conditions based on the symptoms they experience. The application uses structured symptom input and logical analysis to provide preliminary health insights. This system does not replace professional medical diagnosis but helps users understand potential conditions and decide when to seek medical assistance.

Developed using HTML, CSS, and JavaScript, SymptomSense focuses on usability, responsiveness, and accuracy. The application demonstrates effective use of front-end web technologies, modular programming concepts, and problem-solving techniques in healthcare software development.

INTRODUCTION

Healthcare accessibility remains a challenge, especially for individuals seeking immediate information about health conditions. Many people ignore early symptoms due to lack of awareness. SymptomSense aims to bridge this gap by providing a simple, interactive platform that allows users to input symptoms and receive potential health condition suggestions.

The project emphasizes ease of use, responsiveness, and clarity of information. It serves as an educational and assistive tool, especially useful in early symptom awareness.

PROBLEM STATEMENT

Lack of immediate access to basic medical guidance

Difficulty in understanding symptoms and their seriousness

Overdependence on unreliable online sources

Need for a simple, user-friendly symptom checker

SymptomSense addresses these issues by offering structured symptom analysis in a clean and accessible web interface.

OBJECTIVES

To develop a web-based symptom analysis system

To allow users to input symptoms easily

To provide possible health condition suggestions

To create a responsive and user-friendly interface

To apply HTML, CSS, and JavaScript concepts effectively

DOMAIN DESCRIPTION

Healthcare Web Applications

Healthcare web applications aim to assist patients and healthcare providers by offering digital tools for information management and diagnosis support. SymptomSense falls under health informatics, focusing on preventive healthcare awareness.

The domain emphasizes:

User safety

Information clarity

Ethical health guidance

SYSTEM OVERVIEW

SymptomSense is a client-side web application where:

Users enter symptoms through a form

JavaScript processes inputs

Logical conditions match symptoms with known conditions

Results are displayed instantly

The system does not store personal data, ensuring privacy.

MODULES DESCRIPTION

1. User Interface Module

Symptom input form

Buttons and navigation

Result display section

2. Symptom Processing Module

JavaScript logic

Conditional statements

Matching symptoms to conditions

3. Result Analysis Module

Displays possible conditions

Provides general advice

4. Styling Module

CSS for layout and responsiveness

Fonts, colors, and spacing

CONCEPTS APPLIED

HTML Concepts

Forms

Input fields

Semantic tags

Div and section elements

CSS Concepts

Flexbox

Responsive design

Media queries

Styling components

JavaScript Concepts

Functions

Conditional logic

DOM manipulation

Event handling

FEATURES IMPLEMENTED

User-friendly symptom input form

Instant analysis without page reload

Responsive design for mobile and desktop

Clean and minimal UI

Clear result presentation

SYSTEM ARCHITECTURE

Architecture Type:

Client-Side Web Application

Flow:

User inputs symptoms

JavaScript validates input

Logic matches symptoms

Results displayed on UI

No backend or database is used in the current version.

CHALLENGES FACED

1. Accurate Symptom Mapping

Difficulty in mapping symptoms correctly

2. User Interface Design

Ensuring simplicity and clarity

3. JavaScript Logic Errors

Handling multiple symptoms

4. Responsiveness

Supporting various screen sizes

SOLUTIONS IMPLEMENTED

Used structured conditional logic

Designed modular JavaScript functions

Implemented CSS Flexbox for layout

Tested application on multiple devices

FUTURE ENHANCEMENTS

Integration with a medical database

Adding AI-based symptom analysis

Backend support with user accounts

Multilingual support

Doctor consultation integration

HTML CODE

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SymptomSense</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

```
<h1>SymptomSense</h1>
<p class="tagline">A Web-Based Symptom Analysis System</p>
<form id="symptomForm">
<label for="symptoms">Enter your symptoms (comma separated):</label>
<textarea id="symptoms" placeholder="Fever, cough, headache" required></textarea>
<button type="submit">Analyze Symptoms</button>
</form>
<div id="result"></div>
</div>
<script src="script.js"></script>
</body>
</html>
```

CSS CODE

```
* {
margin: 0;
padding: 0;
box-sizing: border-box;
font-family: "Times New Roman", serif;
}
body {
background-color: #f4f6f8;
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
}
```

```
.container {
background: #ffffff;
width: 60%;
padding: 30px;
border-radius: 8px;
box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
h1 {
text-align: center;
margin-bottom: 10px;
}
.tagline {
text-align: center;
color: #555;
margin-bottom: 20px;
}
label {
font-weight: bold;
}.
textarea {
width: 100%;
height: 100px;
margin-top: 8px;
padding: 10px;
font-size: 14px;
}
```

```
button {
display: block;
margin-top: 15px;
padding: 10px;
width: 100%;
background-color: #2c7be5;
color: white;
border: none;
cursor: pointer;
font-size: 16px;
}
button:hover {
background-color: #1a5fc4;
}
#result {
margin-top: 20px;
padding: 15px;
background: #eef3ff;
border-radius: 5px;
}
```

Java Script code

```
document.getElementById("symptomForm").addEventListener("submit", function(e) {
e.preventDefault();
const input = document.getElementById("symptoms").value.toLowerCase();
const resultDiv = document.getElementById("result");
let symptoms = input.split(",").map(s => s.trim());
```

```
let condition = "No matching condition found. Please consult a doctor.";

if (symptoms.includes("fever") && symptoms.includes("cough")) {

condition = "Possible Condition: Flu or Viral Infection.";

}

else if (symptoms.includes("headache") && symptoms.includes("nausea")) {

condition = "Possible Condition: Migraine.";

}

else if (symptoms.includes("chest pain") && symptoms.includes("shortness of breath")) {

condition = "Possible Condition: Heart-related issue. Seek medical help immediately.";

}

else if (symptoms.includes("sore throat") && symptoms.includes("cold")) {

condition = "Possible Condition: Common Cold.";

}

resultDiv.innerHTML = `

<h3>Analysis Result</h3>

<p>${condition}</p>

<p class="note">Note: This is not a medical diagnosis.</p>

`;

});
```

CONCLUSION

SymptomSense successfully demonstrates the application of web development technologies in the healthcare domain. The project highlights how simple front-end tools can be used to create meaningful applications that improve health awareness.

The system is scalable and can be enhanced with advanced technologies in the future. It serves as a strong foundation for healthcare-based web application development.

REFERENCES

W3Schools – HTML, CSS, JavaScript

MDN Web Docs

Healthcare Information Systems – Textbooks

Online Medical Resources