



INSTITUTE OF AERONAUTICAL ENGINEERING (AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

Complex Problem-Solving Self-Assessment Form

1	Name of the Student	GANAPAVARAPU MEGHANA	
2	Roll Number	25951A6692	
3	Branch and Section	CSE(AI-ML)-B	
4	Program	BTECH	
5	Course Name	FRONTEND WEB DEVELOPMENT LABORATORY	
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	Profiles	(√)
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	Identify the basic structure and components of a web page using HTML5 elements.	✓
CO 2	Apply CSS3 properties and layouts to enhance the presentation of web pages	✓
CO 3	Implement JavaScript for client-side validation, event handling, and interactivity.	✓
CO 4	Analyze different CSS frameworks to choose appropriate tools for responsive web design	✓
CO 5	Develop APIs and asynchronous JavaScript features to build dynamic content-driven applications	✓
CO 6	Implement APIs and asynchronous JavaScript features to build dynamic content-driven applications	✓

9	Course ELRV Video Lectures Viewed	Number of Videos	Viewing time in Hours
10	Justify your understanding of WK1	Applying concepts of engineering science fundamentals for solving respective problems.	
11	Justify your understanding of WK2 – WK9	<p>WK2: Enables precise analysis and modeling using mathematical and computational techniques, essential for solving engineering problems.</p> <p>WK3: Provides a strong theoretical foundation to develop systematic solutions grounded in engineering principles.</p> <p>WK4: Offers advanced knowledge to innovate and address challenges at the forefront of the discipline.</p> <p>WK5: Promotes sustainable design and operations by integrating environmental considerations and resource efficiency.</p> <p>WK6: Facilitates practical application of technology to implement effective engineering solutions.</p> <p>WK7: Highlights the societal and ethical responsibilities of engineers to ensure safety and sustainability.</p> <p>WK8: Encourages critical thinking and engagement with current research to address emerging engineering issues.</p> <p>WK9: Reinforces ethical conduct and inclusivity, ensuring equitable and responsible engineering practices.</p>	
	How many Wks from WK2 to WK9 were implanted?	ALL	
12	Mention them	WK2 – WK9	

Date:

Signature of the Student

CARECOMPANION

A Project Report submitted

In partial fulfilment of the requirements

for the award of the degree of

Bachelor of Technology

In

CSE (Artificial Intelligence & Machine Learning)

By

**GANAPAVARAPU MEGHANA
24951A6692**



Department of

CSE (Artificial Intelligence & Machine Learning)

Institute of Aeronautical Engineering

(Autonomous)

Dundigal, Hyderabad-500043, Telangana

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 in preparing the report.
- d. I have conformed to the norms and guidelines given in the Ethical 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

Place: Hyderabad

Signature of the student

Date:

Roll No:

CERTIFICATE

This is to certify that the project report entitled **CARECOMPANION** submitted by **G.MEGHANA** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfilment of the requirements for the award of the Degree Bachelor of Technology in **CSE (Artificial Intelligence and Machine Learning)** is a Bonafide record of work carried out by her under our 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:

APPROVAL SHEET

This project report entitled **CARECOMPANION** by **G.MEGHANA** is approved for the award of the Degree Bachelor of Technology in Branch **CSE (Artificial Intelligence & Machine Learning)**

Examiners

Supervisor(s)

Principal

Date:

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 & Machine Learning)**, for her kind help and for giving me the necessary guidance and valuable suggestions for this project work.

I am grateful to **Dr P Ashok Babu, 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.

ABSTRACT

CareCompanion is a modern web application developed to support caregivers and families in managing the daily activities, medicine routines, and appointments of elderly individuals. With increasing life expectancy and busy family schedules, the need for an organized, accessible, and reliable care-planning system has grown significantly.

The platform provides task scheduling, automated reminders, a calendar interface, and optional collaboration tools for multiple caregivers. The system promotes safety, consistency, and emotional well-being for elders by reducing missed medications and appointments.

This report presents the overall system design, development methodology, features, technologies used, results, and potential future improvements.

	Table of Contents	Page
Self -Assessment Form		1-3
Title Page		4
Declaration		5
Certificate		6
Approval Sheet		7
Acknowledgement		8
Abstract		9
Contents		10

Chapter 1 – Introduction

- 1.1 Project overview
- 1.2 Problem Statement
- 1.3 Objectives

Chapter 2 – Prerequisites and Requirements

- 2.1 Prerequisites
- 2.2 Requirements

Chapter 3 – Methodology

- 3.1 Algorithm
- 3.2 Flow of the project
- 3.3 Code
- 3.4 Technology used

Chapter 4 - Results and Discussions

Chapter 5 - Conclusions and Future Scope

Chapter 1 - Introduction

1.1 Project Overview

CareCompanion is designed to act as a virtual assistant for elderly caregivers.

It helps manage essential routines such as:

- Medication timings
- Doctor visits
- Daily health activities
- Hydration and diet reminders
- Exercise schedules
- Personal tasks

The application emphasizes accessibility with simple navigation, readable fonts, large buttons, color-coded reminders, and mobile-friendly design suitable for elderly and non-technical users.

CareCompanion supports families, attendants, and elder-care centers by organizing tasks and improving consistency in elderly care.\

1.2 Problem Statement

Elderly people often depend on structured routines to maintain health.

Caregivers handling multiple responsibilities face challenges such as:

- Forgetting medicine times
- Missing doctor appointments
- Lack of daily health tracking
- Difficulty in managing multiple patients
- No proper system for scheduling

To solve these issues, a digital tool is required that can organize elderly-care activities and automate reminders.

1.3 Objectives

Primary Objectives:

- Develop an intuitive web interface to add and track daily elderly care tasks.
- Implement an automated reminder system using browser notifications.
- Display tasks visually in a calendar format for better readability.
- Ensure easy editing, updating, and deleting of scheduled tasks.

Secondary Objectives:

- Provide collaboration features for multiple caregivers.
- Allow priority tagging (High/Medium/Low).
- Integrate optional voice-based reminders.
- Design UI/UX following W3C accessibility standards for elderly users.

Long-Term Objectives:

- Enable cloud-based data synchronization.
- Provide analytics for tracking health activity patterns.

Chapter 2 – Prerequisites and Requirements

2.1 Prerequisites

- Basic knowledge of **HTML, CSS, JavaScript**
- Understanding of **React.js** (recommended)
- Familiarity with scheduling and reminder logic
- Awareness of accessibility and UI design principles

2.2 Requirements

Hardware Requirements

- Laptop / PC with stable internet
- 4GB RAM minimum

Software Requirements

- VS Code
- Node.js & npm (for React)
- Git/GitHub (optional for version control)

Libraries / Tools Used

- React.js
- React Router
- React Calendar
- Browser Notification API
- LocalStorage or a lightweight backend

Chapter 3 - Methodology

3.1 Algorithm

- User logs into the system
- Creates personalized care schedules
- Adds tasks like medication, appointments, or activities
- System stores data using LocalStorage or backend API
- Reminder triggers based on time
- Calendar displays upcoming events
- User edits or deletes tasks as needed
- Optional features enhance usability (collaboration, voice reminders, etc.)

3.2 Project work flow

- Requirement Analysis
- UI/UX Prototyping
- Frontend Component Development
- Reminder & Notification Setup
- Calendar Integration
- User Testing
- Final Deployment

3.3 Code

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>CareCompanion</title>

<style>

body {
    font-family: Arial, sans-serif;
    background: #f5f5f5;
    margin: 0;
    padding: 0;
}

header {
```

```
background: #3a6ea5;  
color: white;  
padding: 20px;  
text-align: center;  
font-size: 28px;  
}  
  
.container {  
width: 90%;  
max-width: 800px;  
margin: 30px auto;  
background: white;  
padding: 20px;  
border-radius: 10px;  
box-shadow: 0 0 10px #ccc;  
}  
  
label {  
font-size: 16px;  
font-weight: bold;  
}  
  
input, select {  
width: 100%;  
padding: 10px;  
margin: 8px 0 20px 0;  
border-radius: 5px;
```

```
border: 1px solid #777;  
font-size: 15px;  
}  
  
button {  
background: #3a6ea5;  
color: white;  
padding: 12px 20px;  
border: none;  
cursor: pointer;  
border-radius: 5px;  
font-size: 17px;  
}  
  
button:hover {  
background: #2e5786;  
}  
  
.task-box {  
padding: 15px;  
background: #e8f0fe;  
margin-top: 10px;  
border-radius: 8px;  
}  
  
.delete-btn {  
background: red;  
padding: 8px 13px;
```

```
float: right;  
font-size: 14px;  
}  
</style>  
</head>  
<body>  
  
<header>CareCompanion – Elderly Care Reminder System</header>  
  
<div class="container">  
  
<h2>Add New Task</h2>  
  
<label>Task Name</label>  
<input type="text" id="taskName" placeholder="Enter activity (e.g., Take Medicine)">  
  
<label>Date</label>  
<input type="date" id="taskDate">  
  
<label>Time</label>  
<input type="time" id="taskTime">  
  
<button onclick="addTask()">Add Task</button>
```

```
<h2 style="margin-top:40px;">Scheduled Tasks</h2>
<div id="taskList"></div>
</div>

<script>
// Load tasks from LocalStorage when page opens
document.addEventListener("DOMContentLoaded", loadTasks);

function addTask() {
    let name = document.getElementById("taskName").value;
    let date = document.getElementById("taskDate").value;
    let time = document.getElementById("taskTime").value;

    if (name === "" || date === "" || time === "") {
        alert("Please fill all fields!");
        return;
    }

    let task = {
        name: name,
        date: date,
```

```

time: time

};

saveTask(task);

displayTask(task);

scheduleReminder(task);

document.getElementById("taskName").value = "";

document.getElementById("taskDate").value = "";

document.getElementById("taskTime").value = "";

}

function saveTask(task) {

let tasks = JSON.parse(localStorage.getItem("tasks")) || [];

tasks.push(task);

localStorage.setItem("tasks", JSON.stringify(tasks));

}

function loadTasks() {

let tasks = JSON.parse(localStorage.getItem("tasks")) || [];

tasks.forEach(task => displayTask(task));

}

function displayTask(task) {

```

```

let taskDiv = document.createElement("div");

taskDiv.className = "task-box";

taskDiv.innerHTML = `

<strong>${task.name}</strong><br>

Date: ${task.date} — Time: ${task.time}

<button class="delete-btn" onclick="deleteTask('${task.name}', '${task.date}', '${task.time}', this)">Delete</button>

`;

document.getElementById("taskList").appendChild(taskDiv);

}

function deleteTask(name, date, time, element) {

let tasks = JSON.parse(localStorage.getItem("tasks")) || [];

let updated = tasks.filter(t => !(t.name === name && t.date === date && t.time === time));

localStorage.setItem("tasks", JSON.stringify(updated));

element.parentElement.remove();

}

```

// Reminder Function

```

function scheduleReminder(task) {

let now = new Date();

let reminderTime = new Date(task.date + " " + task.time);

```

```

let timeout = reminderTime - now;

if (timeout > 0) {
    setTimeout(() => {
        showNotification(task.name);
    }, timeout);
}

function showNotification(taskName) {
    if (Notification.permission === "granted") {
        new Notification("Reminder: " + taskName);
    } else {
        Notification.requestPermission().then(permission => {
            if (permission === "granted") {
                new Notification("Reminder: " + taskName);
            }
        });
    }
}

</script>

</body>

```

</html>

3.4 Technology used

- HTML5: Structure of application
- CSS3: Styling and responsive layout
- JavaScript: Logic for tasks and reminders
- React.js: Component-based UI
- Notification API: Browser-based reminders
- LocalStorage / API: Temporary or persistent data storage

Chapter 4 - Results and Discussions

The application successfully performs:

- Creating, editing, and deleting care schedules
- Showing tasks in a clear calendar view
- Triggering reminders at correct times
- Displaying readable and accessible UI
- Smooth performance across laptop and mobile devices

User testing showed that:

- Elderly users found the interface easy to understand
- Caregivers appreciated priority tagging and collaboration options
- Reminders significantly reduced missed tasks

Accessibility tests confirmed:

- Proper color contrast
- Large readable typography
- Clean layout
- Touch-friendly icons and buttons

Chapter 5 - Conclusions and Future Scope

5.1 Conclusion

CareCompanion is an effective digital solution for elderly-care scheduling. Its intuitive interface, automated reminders, and calendar visualization help caregivers manage responsibilities more efficiently.

The system enhances caregiving quality, reduces stress, and supports healthier routines for elderly individuals.

5.2 Future Scope

- The project can be extended with:
- Integration with Google/Apple Calendar
- AI-based personalized health suggestions
- Voice-based reminders and commands
- Real-time caregiver collaboration
- Mobile application for Android and iOS
- Cloud synchronization for multi-device access
- Dashboard analytics of health activities

References

- React.js Official Documentation
- MDN Web Docs – Web APIs
- W3C Web Accessibility Standards
- Online tutorials and frontend development guides