

To-Do Project (selected App)

Introduction:-

- ⇒ To Do list is an application specially built to keep track of tasks that needed to be done.
- ⇒ It will be like a task keeper where the user would be able to enter the tasks that they need to do.
- ⇒ We use some specific hardware and software components in order to build a to do app.
- ⇒ We had created using the CSS, Javascript and HTML files. We conducted a simple survey and realized to build a app like this.
- ⇒ By using the features in this, we can manage our tasks without any ignorance.
- ⇒ In this the user can mark the complete and uncompleted tasks. They can also delete the task if not needed.

- All the uses of the TO DO APP are given in this project.
- And also there are some advantages and disadvantages of the TO DO APP which are mentioned in the project.

1.2 Purpose OF TO DO App

(3)

- ⇒ Nowadays, people are very much busy with a lot of tasks and are not capable of reminding the tasks to do.
- ⇒ We can also see the lack of organizing the tasks on basis of the priority.
- ⇒ By using this app we can complete the tasks without any pending i.e. "ontime"
- ⇒ This can improve the task and time management by which we can be a good task organizer.
- ⇒ This can help students as well as in business to work effectively and efficiently.
- ⇒ By using this app our mind will not be stressed as there is no need to think a lot about the task we have forgot.
- ⇒ So, to solve these it is necessary to build a "TO DO APP".

2. Literature Survey :-

(4)

- ⇒ We studied the research paper "studies OF task Manager". It has been referred to in 166 papers till now.
- ⇒ The aim of this paper was to find the factors which influence task completion and provide a strong foundation for design of personal to do list.
- ⇒ It was found that people do not correctly explain their task in case of offline or manual to do list.
- ⇒ We studied the case study titled "The Best to-do list App right now" published on theverge.com
- ⇒ Its main aim is to compare the popular to do lists app in the market and give them a score based on the "user-performance".
- ⇒ The approach is simple that the to do list is the one which is perfectly maps individual meets.



- The main finding from this website is to prepare a to do list which will fit most of the user's needs.
- ⇒ We also studied the article from orange scrum. It states that there are few problems if we do the tasks manually.

2.1 Proposed solution:-

(6)

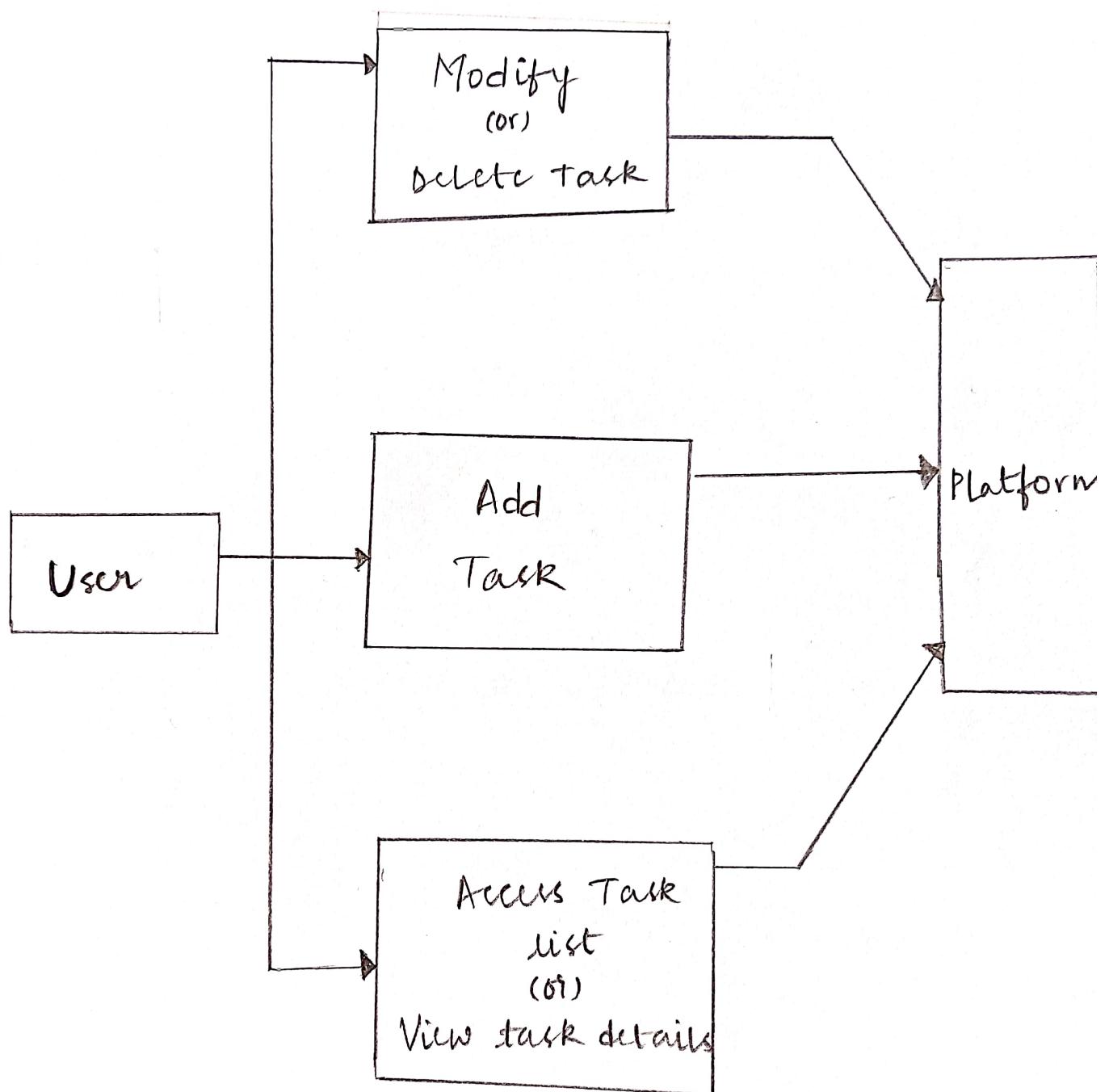
- ⇒ For the problems based on the task management, I would like to propose the idea of creating a todo app which can be accessible on all devices and should be easily usable by the users.
- ⇒ The management of tasks will be easily as they have priority and deadlines.
- ⇒ In this app we can mark the work as completed when its done. So, they are well organized.
- ⇒ In future by linking the todo app with voice assistant it will be easy for all users to complete their tasks in a smart way.
- ⇒ It helps people having memory problem to remind their tasks.
- ⇒ So, I believe that completing our tasks is more important and by using the To Do App we can achieve it full pledgedly.

= .

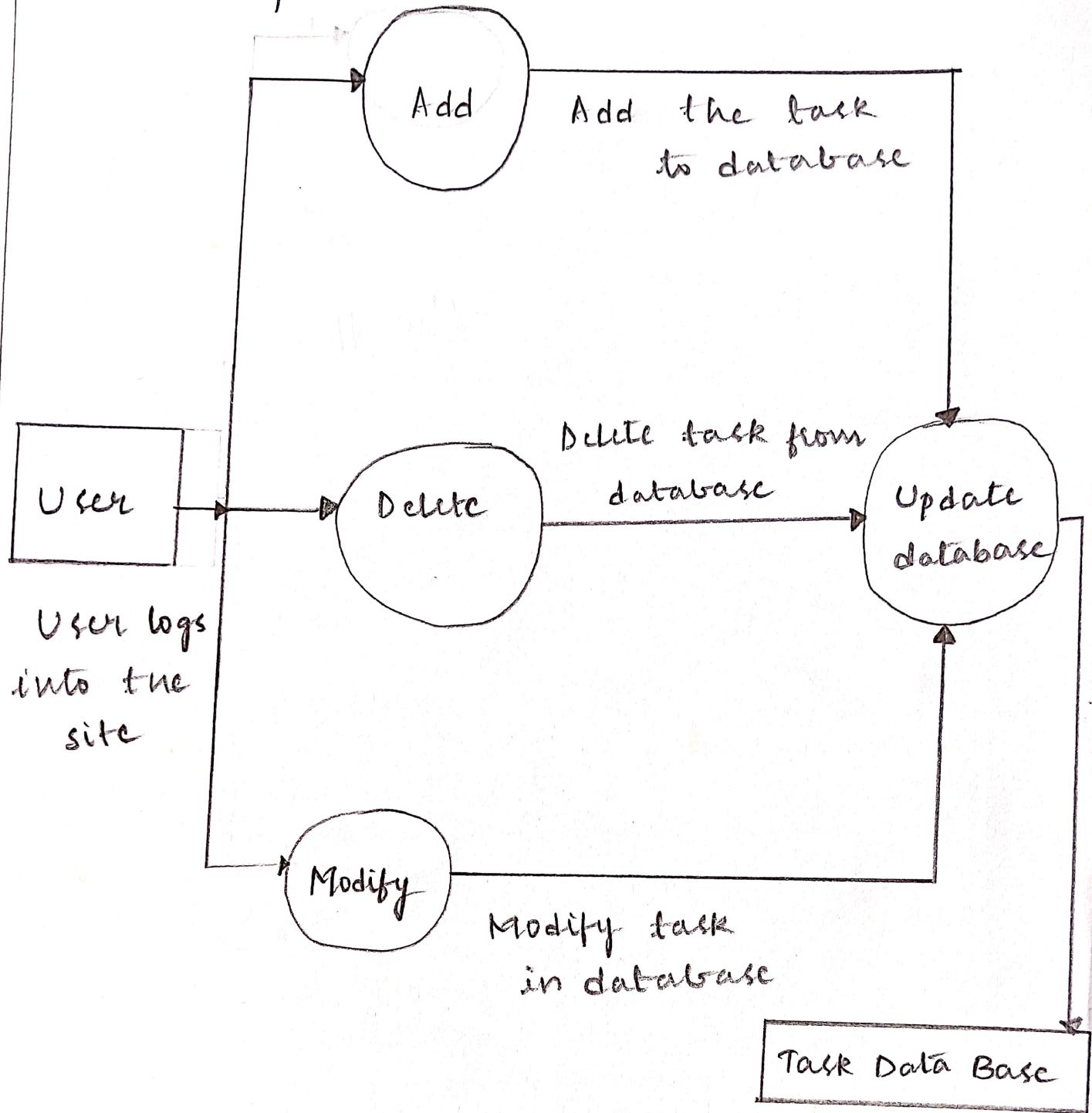
3. Theoretical Analysis:-

3.1

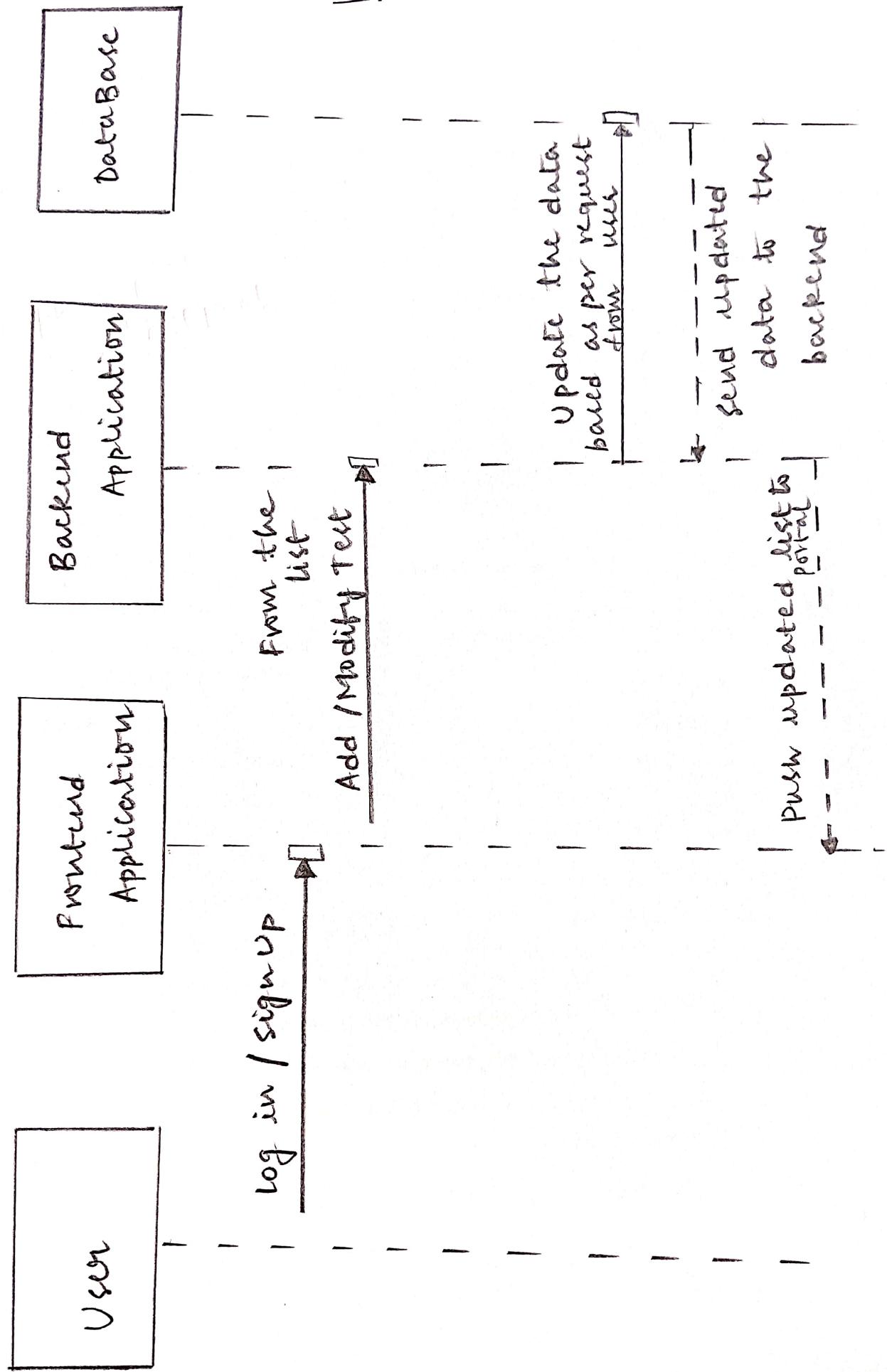
Block diagram :-



Graphical Abstract :-



Data Flow Diagram :-



```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>ToDo List</title>
    <link rel="stylesheet" href="./assets/css/style.css">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
        rel="stylesheet"
        integrity="sha384-9ndCyUalbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
        crossorigin="anonymous">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
</head>

<body>
    <section id="section1">

        <nav class="navbar bg-transparent fixed-top">
            <div class="container-fluid">
                <button class="navbar-toggler me-auto" type="button" data-bs-toggle="offcanvas"
                    data-bs-target="#offcanvasNavbar" aria-controls="offcanvasNavbar">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="offcanvas offcanvas-start" tabindex="-1" id="offcanvasNavbar"
                    aria-labelledby="offcanvasNavbarLabel">
```

```
<div class="offcanvas-header">  
    <h5 class="offcanvas-title" id="offcanvasNavbarLabel">TODO APP</h5>  
    <button type="button" class="btn-close" data-bs-dismiss="offcanvas"  
        aria-label="Close"></button>  
</div>  
  
<div class="offcanvas-body">  
    <ul class="navbar-nav justify-content-start flex-grow-1 pe-3">  
  
        <li class="nav-item">  
            <a class="nav-link active" aria-current="page" href="index.html">  
                <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"  
                    fill="currentColor" class="bi bi-list-task" viewBox="0 0 16 16">  
                    <path  
                        d="M1.5 2.5A.5.5 0 0 1 2 2h12a.5.5 0 0 1 0 1h-12a.5.5 0 0 1 -5-.5zm0  
4A.5.5 0 0 1 2 6h12a.5.5 0 0 1 0 1h-12a.5.5 0 0 1 -5-.5zm0 4A.5.5 0 0 1 2 10h12a.5.5 0 0 1 0 1h-  
12a.5.5 0 0 1 -5-.5zm0 4A.5.5 0 0 1 2 14h12a.5.5 0 0 1 0 1h-12a.5.5 0 0 1 -5-.5z" />  
                </svg>  
                Task  
            </a>  
        </li>  
  
    </ul>  
  
</div>  
</div>  
</div>  
</nav>  
</section>
```

```

<section id="section2">

    <div class="container">
        <!-- Page content -->
        <p class="p">
            <svg xmlns="http://www.w3.org/2000/svg" width="40" height="25" fill="currentColor"
                class="bi bi-house-door" viewBox="0 0 16 16">
                <path
                    d="M8.354 1.146a.5.5 0 0 0-.708 0l-6 6a.5.5 0 0 1 7.5v7a.5.5 0 0 0 .5.5h4.5a.5.5
0 0 0 .5v-4h2v4a.5.5 0 0 0 .5.5h14a.5.5 0 0 1 .5v-7a.5.5 0 0 0 -.146-.354l13 5.793V2.5a.5.5 0 0
0 -.5h-1a.5.5 0 0 0 -.5v1.293l8.354 1.146zM2.5 14v7.707l5.5-5.5 5.5 5.5V14H10v-4a.5.5 0 0 0 -.5
.5h-3a.5.5 0 0 0 -.5v4H2.5Z" />
            </svg>
            <span class="fs-4">Task</span>
        </p>

        <div class="card card1">
            <div class="card-body">
                <input type="text" id="task-input" class="form-control" placeholder="Enter your
task">
                <button id="calendar-btn" class="btn mt-2">
                    <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor"
                        class="bi bi-calendar2-event" viewBox="0 0 16 16">
                        <path
                            d="M11 7.5a.5.5 0 0 1 .5-.5h1a.5.5 0 0 1 .5.5v1a.5.5 0 0 1 -.5h-1a.5.5 0 0 1 -.5
.5v-1z" />
                    <path

```

```
d="M3.5 0a.5.5 0 0 1 .5.5V1h8V.5a.5.5 0 0 1 1 0V1h1a2 2 0 0 1 2 2v11a2 2 0 0  
1-2 2H2a2 2 0 0 1-2-2V3a2 2 0 0 1 2-2h1V.5a.5.5 0 0 1 .5-.5zM2 2a1 1 0 0 0-1 1v11a1 1 0 0 0 1  
1h12a1 1 0 0 0 1-1V3a1 1 0 0 0-1-1H2z" />  
  
<path  
  
d="M2.5 4a.5.5 0 0 1 .5-.5h10a.5.5 0 0 1 .5.5v1a.5.5 0 0 1-5.5H3a.5.5 0 0 1-5-  
.5V4z" />  
  
</svg>  
  
</button>  
  
  
<div id="calendar-picker-container" class="mt-2 d-none">  
  
<div class="input-group">  
  
<input type="date" id="calendar-picker" class="form-control"  
  
placeholder="Select date">  
  
<button id="calendar-save-btn" class="btn btn-outline-primary"><svg  
  
xmlns="http://www.w3.org/2000/svg" width="16" height="16"  
  
fill="currentColor" class="bi bi-save" viewBox="0 0 16 16">  
  
<path  
  
d="M2 1a1 1 0 0 0-1 1v12a1 1 0 0 0 1 1h12a1 1 0 0 0 1-1V2a1 1 0 0 0-1-  
1H9.5a1 1 0 0 0-1 1v7.293l2.646-2.647a.5.5 0 0 1 .708.708l-3.5 3.5a.5.5 0 0 1 -.708 0l-3.5-3.5a.5.5 0 1  
1 .708-.708l7.5 9.293V2a2 2 0 0 1 2-2H14a2 2 0 0 1 2 2v12a2 2 0 0 1-2 2H2a2 2 0 0 1-2-2V2a2 2 0 0 1  
2-2h2.5a.5.5 0 0 1 0 1H2z" />  
  
</svg></button>  
  
</div>  
  
</div>  
  
  
<button id="notification-btn" class="btn mt-2">  
  
<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"  
fill="currentColor"  
  
class="bi bi-bell" viewBox="0 0 16 16">  
  
<path
```

```

d="M8 16a2 2 0 0 0 2-2H6a2 2 0 0 0 2 2zM8 1.918l-.797.161A4.002 4.002 0 0 0
4 6c0 .628-.134 2.197-.459 3.742-.16.767-.376 1.566-.663 2.258h10.244c-.287-.692-.502-1.49-.663-
2.258C12.134 8.197 12 6.628 12 6a4.002 4.002 0 0 0-3.203-3.92l8 1.917zM14.22
12c.223.447.481.801.78 1H1c.299-.199.557-.553.78-1C2.68 10.2 3 6.88 3 6c0-2.42 1.72-4.44 4.005-
4.901a1 1 0 1 1 1.99 0A5.002 5.002 0 0 1 13 6c0 .88.32 4.2 1.22 6z" />

</svg>
</button>

<div id="time-picker-container" class="mt-2 d-none">
  <div class="input-group">
    <input type="time" id="time-picker" class="form-control" placeholder="Select
time">
    <button id="save-btr" class="btn btn-outline-primary"><svg
      xmlns="http://www.w3.org/2000/svg" width="16" height="16"
      fill="currentColor" class="bi bi-save" viewBox="0 0 16 16">
      <path
        d="M2 1a1 1 0 0 0-1 1v12a1 1 0 0 1 1h12a1 1 0 0 0 1-1V2a1 1 0 0 0-1
1H9.5a1 1 0 0 0-1 1v7.293l2.646-2.647a.5.5 0 0 1 .708.708l-3.5 3.5a.5.5 0 0 1 -.708 0l-3.5-3.5a.5.5 0 1
1 .708-.708l7.5 9.293V2a2 2 0 0 1 2-2H14a2 2 0 0 1 2 2v12a2 2 0 0 1-2 2H2a2 2 0 0 1-2-2V2a2 2 0 0 1
2-2h2.5a.5.5 0 0 1 0 1H2z" />
    </svg></button>
  </div>
</div>
<button id="add-btn" class="btn mt-2" style="display: none;">
  <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
    fill="currentColor"
    class="bi bi-patch-plus" viewBox="0 0 16 16">
    <path fill-rule="evenodd"
      d="M8 5.5a.5.5 0 0 1 .5.5v1.5H10a.5.5 0 0 1 0 1H8.5V10a.5.5 0 0 1-1
0V8.5H6a.5.5 0 0 1 0-1h1.5V6a.5.5 0 0 1 .5-.5z" />

```

```
<path  
d="m10.273 2.513-.921-.944.715-.698.622.637.89-.011a2.89 2.89 0 0 1 2.924  
2.924l-.01.89.636.622a2.89 2.89 0 0 1 0 4.134l-.637.622.011.89a2.89 2.89 0 0 1-2.924 2.924l-.89-.01-  
.622.636a2.89 2.89 0 0 1-4.134 0l-.622-.637-.89.011a2.89 2.89 0 0 1-2.924-2.924l.01-.89-.636-  
.622a2.89 2.89 0 0 1 0-4.134l.637-.622-.011-.89a2.89 2.89 0 0 1 2.924-2.924l.89.01.622-.636a2.89  
2.89 0 0 1 4.134 0l-.715.698a1.89 1.89 0 0 0-2.704 0l-.92.944-1.32-.016a1.89 1.89 0 0 0-1.911  
1.912l.016 1.318-.944.921a1.89 1.89 0 0 0 2.704l.944.92-.016 1.32a1.89 1.89 0 0 0 1.912l-  
1.911l1.318-.016.921.944a1.89 1.89 0 0 0 2.704 0l.92-.944 1.32.016a1.89 1.89 0 0 0 1.911-1.912l-  
.016-1.318.944-.921a1.89 1.89 0 0 0-2.704l-.944-.92.016-1.32a1.89 1.89 0 0 0-1.912-1.911l-  
1.318.016z" />  
  
</svg>  
</button>  
  
</div>  
</div>  
  
</div>  
  
<div class="container mt-4">  
  
<h5><img alt="List icon" style="vertical-align: middle;"/> Saved Tasks</h5>  
  
<div id="card-container" class="row">  
    <!-- Cards will be dynamically added here -->  
    <div>  
        <!-- Card content -->  
    </div>  
</div>
```

```
</section>

<script src=".//assets/js/script.js"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
       integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68Sly3Te4Bkz"
       crossorigin="anonymous"></script>

</body>

</html>
```

```
const cardContainer = document.getElementById('card-container');

const tasks = [];

const dates = [];

const times = [];

const calendarBtn = document.getElementById('calendar-btn');

const calendarPickerContainer = document.getElementById('calendar-picker-container');

const calendarSaveBtn = document.getElementById('calendar-save-btn');

calendarBtn.addEventListener('click', function () {

    calendarPickerContainer.classList.toggle('d-none');

});

calendarSaveBtn.addEventListener('click', function () {

    const selectedDate = document.getElementById('calendar-picker').value;

    if (selectedDate !== "") {

        dates.push(selectedDate);

        calendarPickerContainer.classList.add('d-none');

        saveDataToLocalStorage();

    }

});

const notificationBtn = document.getElementById('notification-btn');

const timePickerContainer = document.getElementById('time-picker-container');

const saveBtn = document.getElementById('save-btn');

notificationBtn.addEventListener('click', function () {
```

```
timePickerContainer.classList.toggle('d-none');

});

saveBtn.addEventListener('click', function () {
    const selectedTime = document.getElementById('time-picker').value;
    if (selectedTime !== "") {
        times.push(selectedTime);
        timePickerContainer.classList.add('d-none');
        saveDataToLocalStorage();
    }
});

const taskInput = document.getElementById('task-input');
const addBtn = document.getElementById('add-btn');

taskInput.addEventListener('input', function () {
    if (taskInput.value.trim() !== "") {
        addBtn.style.display = 'inline-block';
    } else {
        addBtn.style.display = 'none';
    }
});

addBtn.addEventListener('click', function () {
    const task = taskInput.value.trim();
    if (task !== "") {
        tasks.push({ task: task, date: dates[dates.length - 1], time: times[times.length - 1], notified: false
    });
}
```

```
const card = createCard(task, dates[dates.length - 1], times[times.length - 1]);

cardContainer.appendChild(card);

taskInput.value = "";

addBtn.style.display = 'none';

saveDataToLocalStorage();

playNotificationSound(); // Play the notification sound on adding a task

}

});

});
```

```
document.addEventListener('DOMContentLoaded', function () {

loadDataFromLocalStorage();

checkDateTime();

setInterval(checkDateTime, 1000);

});
```

```
function createCard(task, date, time) {

const card = document.createElement('div');

card.classList.add('card', 'col-md-4', 'my-3');

const cardBody = document.createElement('div');

cardBody.classList.add('card-body');

card.style.marginRight = '10px';

card.style.marginBottom = '10px';

const taskTitle = document.createElement('h5');

taskTitle.classList.add('card-title');
```

```
taskTitle.textContent = task;

const dateText = document.createElement('p');
dateText.classList.add('card-text');
dateText.textContent = 'Date: ' + date;

const timeText = document.createElement('p');
timeText.classList.add('card-text');
timeText.textContent = 'Time: ' + time;

const actionContainer = document.createElement('div');
actionContainer.classList.add('action-container', 'd-flex', 'justify-content-between', 'align-items-center');

const importantIcon = document.createElement('button');
importantIcon.classList.add('btn', 'btn-link', 'important-icon');
importantIcon.innerHTML = '<i class="far fa-star"></i>';
let isImportant = false;

importantIcon.addEventListener('click', function () {
    isImportant = !isImportant;
    card.classList.toggle('important');
    saveDataToLocalStorage();
    if (isImportant) {
        playNotificationSound(); // Play the notification sound when the task is marked as important
    }
});
```

```
const deleteBtn = document.createElement('button');

deleteBtn.classList.add('btn', 'btn-outline', 'mt-3');

deleteBtn.innerHTML = '<i class="fas fa-trash-alt"></i>';

deleteBtn.addEventListener('click', function () {

    cardContainer.removeChild(card);

    const index = tasks.findIndex((item) => item.task === task);

    if (index !== -1) {

        tasks.splice(index, 1);

        dates.splice(index, 1);

        times.splice(index, 1);

    }

    saveDataToLocalStorage();

});

actionContainer.appendChild(importantIcon);

actionContainer.appendChild(deleteBtn);

cardBody.appendChild(taskTitle);

cardBody.appendChild(dateText);

cardBody.appendChild(timeText);

cardBody.appendChild(actionContainer);

card.appendChild(cardBody);

return card;

})
```

```
function saveDataToLocalStorage() {  
    localStorage.setItem('tasks', JSON.stringify(tasks));  
    localStorage.setItem('selectedDates', JSON.stringify(dates));  
    localStorage.setItem('selectedTimes', JSON.stringify(times));  
}  
  
function loadDataFromLocalStorage() {  
    const savedTasks = localStorage.getItem('tasks');  
    const savedDates = localStorage.getItem('selectedDates');  
    const savedTimes = localStorage.getItem('selectedTimes');  
  
    if (savedTasks && savedDates && savedTimes) {  
        tasks.push(...JSON.parse(savedTasks));  
        dates.push(...JSON.parse(savedDates));  
        times.push(...JSON.parse(savedTimes));  
  
        for (let i = 0; i < tasks.length; i++) {  
            const card = createCard(tasks[i].task, dates[i], times[i]);  
            cardContainer.appendChild(card);  
        }  
    }  
}  
  
function checkDateTime() {  
    const currentDate = new Date();  
    const notificationTimeFrame = 0; // Specify the time frame in minutes
```

```
let hasUnexpiredTasks = false;

for (let i = tasks.length - 1; i >= 0; i--) {
    const taskDate = new Date(tasks[i].date);
    const taskTime = tasks[i].time.split(':');
    const taskDateTime = new Date(
        taskDate.getFullYear(),
        taskDate.getMonth(),
        taskDate.getDate(),
        taskTime[0],
        taskTime[1]
    );

    if (
        currentDate.getTime() >= taskDateTime.getTime() &&
        currentDate.getTime() <= taskDateTime.getTime() + notificationTimeFrame * 60 * 1000
    ) {
        // Task is due within the specified time frame
        if (!tasks[i].notified) {
            tasks[i].notified = true;
            playNotificationSound();
        }
        hasUnexpiredTasks = true;
    }
}

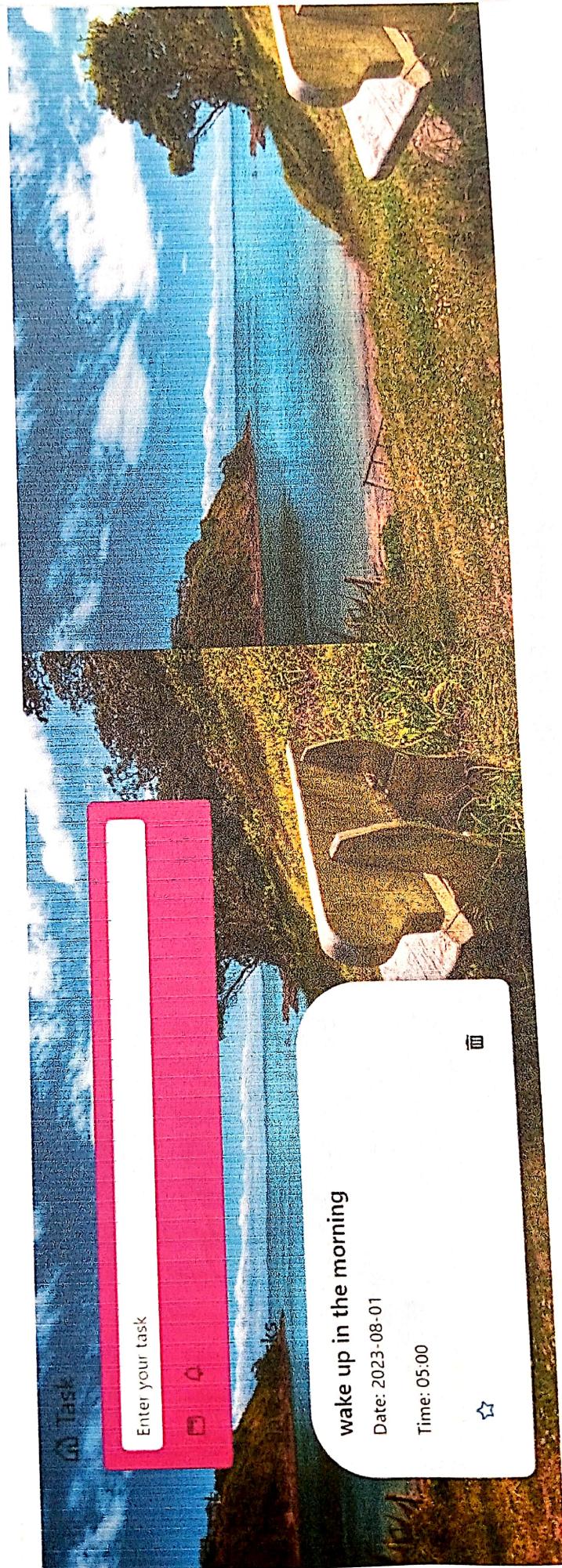
// Check if all tasks have been completed
```

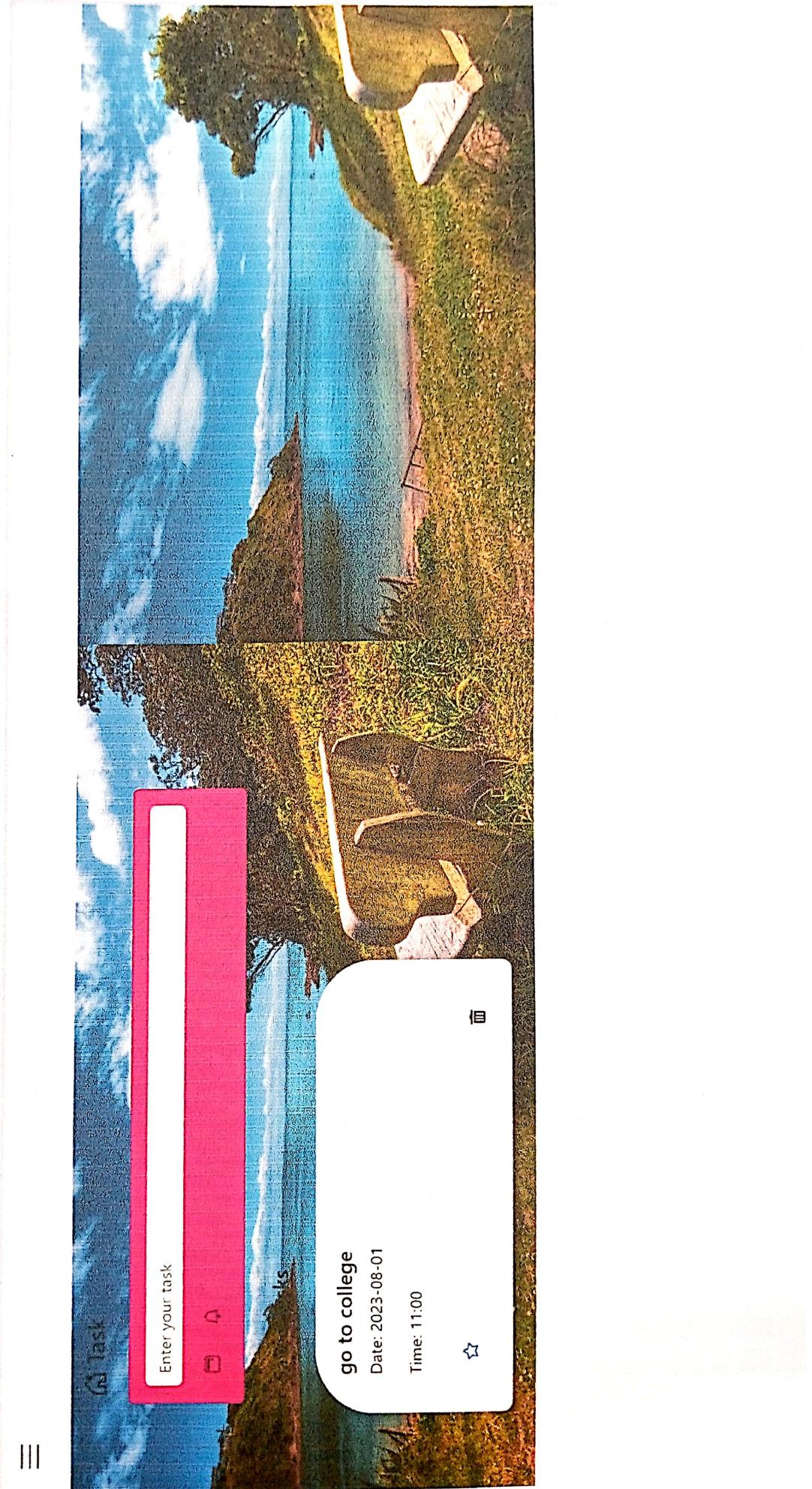
```
if (!hasUnexpiredTasks) {  
    stopNotificationSound();  
}  
}  
  
let audio; // Variable to store the notification audio  
  
async function playNotificationSound() {  
    try {  
        if (audio) {  
            audio.pause();  
            audio.currentTime = 0;  
        }  
        audio = new Audio('./assets/img/mp3.wav');  
        await audio.play();  
        alert('Task notification111');  
    } catch (error) {  
        console.error('Error playing notification sound:', error);  
    }  
}  
  
function stopNotificationSound() {  
    if (audio) {  
        audio.pause();  
        audio.currentTime = 0;  
    }  
}
```

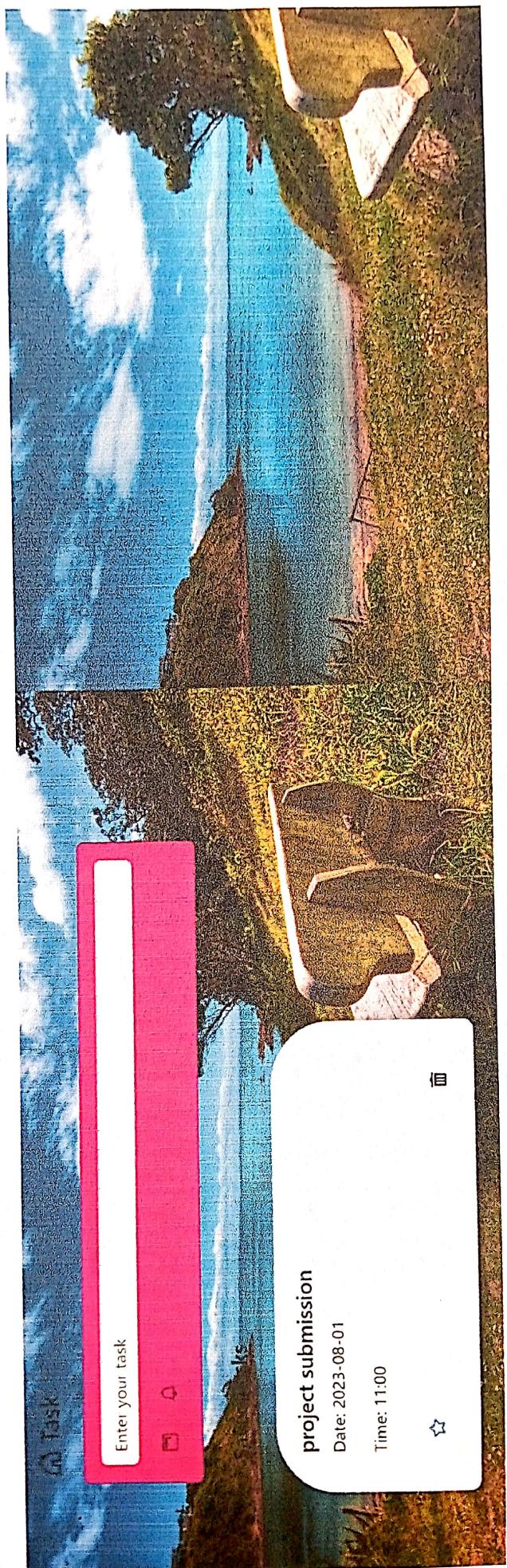
```
.card.card1{  
    width: 600px;  
  
}  
  
#section2{  
    margin-top: 10vh;  
    padding: 1vh;  
    background-image: url("https://getwallpapers.com/wallpaper/full/d/8/3/303843.jpg");  
    background-size: contain;  
  
}  
  
.p {  
    display: flex;  
    align-items: center;  
  
}  
  
svg .bi {  
    margin-right: 5px;  
    /* Adjust the spacing between the icon and text */  
}  
  
.important {  
    background-color: #dc1586 !important;  
    /* Change the background color to light red */  
}  
  
.offcanvas-body {
```

```
background-color: rgb(251, 249, 249);  
}  
  
#calendar-picker-container , #time-picker-container{  
    width: 50vh;  
}  
  
.action-Container{  
    margin: 20px;  
}  
  
div .col-md-4{  
    padding: 0vh 3vh;  
    border-radius: 60px 60px 10px 10px;  
    box-shadow: rgba(7, 6, 5, 0.19) 0px 10px 20px, rgba(0, 0, 0, 0.23) 0px 6px 6px;  
}  
  
.card1{  
    background-color: rgb(252, 7, 174) !important;  
}
```

4. Result :-







Task

Enter your task

Q

project submission

Date: 2023-08-01

Time: 11:00



Scanned with OKEN Scanner

Requirements

Hardware Requirement

- * The hardware required to build a todo app are following:
- * Processor: We should need a good processor, so that we can work on it without any problem. The processor required is pentium IV or above to work smoothly.
- * RAM: The RAM should be more than 16GB so that while working there will be no more load on the processor.
- * Secondary storage:
A disk drive of 500mb or more is sufficient to store the work files on local storage.
- * Input devices:
The devices used are keyboard and mouse.
- * Output devices:
The output devices is the monitor where we can see the output or display the code or execute the code.

Software Requirements :-

- * The software / applications required to build the todo list app are :-
- * Operating system :
We know that it is main to work on a PC or computer , we can use versions of windows , mac , ubuntu , Linux . But by using the latest as the system will work efficiently .
- * Front-End :-
we have to create a HTML file for html code a javascript file for JS code and a CSS file for designing the webpage . we can also use the bootstrap to work fastly
- * Local Host :
Here - we can use the microsoft's visual studio code for writings executing all the code needed .
- * Backend :-
we have to connect to a database so that information will be stored . we can also use python , java , PHP , ruby on rails as backend .

5.1 Advantages of

(7)

TODO APP :-

-
1. Organization: A Todo App helps users stay organized by providing a structured way to manage tasks making it easier to keep track of responsibilities and deadlines.
 2. Increased productivity:
By having a clear list of tasks and priorities users can focus on what needs to be done minimizing distractions.
 3. Time management:
Todo apps often include features like marking tasks.
 4. Goal Tracking:
It allows users to set and track their goals, breaking them down into manageable tasks.
 5. Accessibility:
Most todo apps sync across multiple devices, ensuring users can access their tasks from anywhere.
 6. Collaboration:
Some todo apps offer collaboration features making it easier for teams to work together, assign tasks and track shared projects.

To Do App :-

1. Task Neglect :-

In some cases, users may forget to update their todo app regularly, leading to the neglected tasks and missed deadlines.

2. Dependency :-

Relying solely on a "to do app" without developing good organized or organizational habits can make users feel lost or unproductive when they don't have access.

3. Technical issue :-

To Do Apps may experience technical glitches or compatibility issues, which can disrupt task management.

4. Distractions :-

Constant notifications and reminders from a todo app can be distracting and disrupt focus on other important tasks.

5. Learning Curve :-

Some todo apps can be complex to set up or use initially, requiring time and effort to learn their full functionality.

6. Applications :-

⑪

1. Work and Business:-

In a professional setting, individuals can use a to do app to manage work-related tasks projects and helping them stay organized and productive.

2. Study and Academic Planning:-

Students can use to do apps to manage their study schedules, assignment deadlines and exam preparations.

3. Travel Planning:-

For trips and vacations, a to do app can help users create packing lists, plan activities and keep track of travel arrangements.

4. Event Planning:-

Whether its organizing a party, a wedding or a social gathering, a to do app can aid in managing tasks, guest lists and other event details.

5. Personal Development:-

To do apps can be utilized for personal development like tracking fitness goals, reading lists or learning new skills.

F. Conclusion :-

(12)

Conclusion :-

=

- ⇒ To do app is a useful app which helps us to organize and create tasks to do later.
- ⇒ By working to create the to do app frontend, I came to know how to create it, mark it done when the work is completed and some failures of the app.
- ⇒ This app helps the students for work and business as well.
- ⇒ But we should not become overly reliant on the app.
- ⇒ It helps in every applications and can be easily accessible.
- ⇒ So, by this I conclude that "To Do App" is very useful to not distract from the work and complete all the tasks efficiently.

8. Future scope :-

(13)

1. Voice integration:

By combining todo app with voice assistant we will get reminders and other suggestions.

2. Personalize recommendations:

We can even personalize more based on user behaviour and preferences.

3. Predictive Assistance:

By this app we will observe over user moments on past and suggest tasks based on the time.

4. Smart - Home Integration:

By syncing todo app with home devices the daily tasks can be easily completed.

5. Focus on well-being:

By incorporating features to promote mindfulness stress reduction and work life balance.

6. Cross platform functionality:

It will be more crossplatform friendly, ensuring easy synchronization.