## PROJECT REPORT

## PROJECT NAME:TODO APP

## TEAM MEMBERS:

## GOTTIPILLI HARITHA(TEAM LEADER)

## DHANALA SAINANDA

## BORA VIJAY

## DIVIKALA ANUSHA

## 

## 

|  |  |
| --- | --- |
|  |  |

Introduction

A to-do list is a simple prioritized list of the tasks a person must complete. People make a list of everything they need to do, ranked according to priority from the most critical task at the top to the least critical task at the bottom. A few of the features of a good to-do list application include:

Plan and execute simple actions.

Prioritize, manage, and reason about tasks.

Record notes, action items and ideas.

To-dos are the tasks or the atomic entities that make up a to-do list. To-dos are made

quickly, the bulk of them do not specify the work; instead, they are typically just

comprehensive enough to serve as a valuable indicator. To be sure, to-do terminology

like "Groceries" or "Car Wash" is frequently grammatically correct. Because the

signal is so quick, it is only useful for a short period of time while the task is

remembered. In certain cases, a simple item like a stack is enough to recall the job

What is a ToDo List? The definition is a simple one. ***It’s a list of tasks you need to complete or things that you want to do.***

Most typically, they’re organised in order of priority. Traditionally, they’re written on a piece of paper or post it notes and act as a memory aid. As technology has evolved we have been able to create a todo lists with excel spreadsheets, word documents, [email](https://en.wikipedia.org/wiki/Email) lists, todo list apps, Microsoft to do and google to do list to name a few. You can[use a to do list](https://checkify.com/blog/how-to-use-a-to-do-list/) in your home and personal life, or in the workplace.

Having a list of everything you need to do written down in one place means you shouldn’t forget anything important. By prioritising the tasks in the list you plan the order in which you’re going to do them and can quickly see what needs your immediate attention and what tasks you can leave until a little later.

Purpose of this project

One of the most important reasons you should use a to do list is that it will help you stay organised. When you write all your tasks in a list, they seem more manageable. When you’ve got a clear outline of the tasks you’ve got to do and those you’ve completed, it helps you stay focused. While freeing up space in your mind for other more creative tasks.

When you complete a task, you can cross it off your list. This gives you a sense of progress and achievement, something you’ll lack if you’re always rushing from one task to the next. If you feel a sense of achievement, it spurs you on and motivates you to keep moving forward.

But that’s not the only benefit of a to do list. Here are a few more:

**Improves your memory**: A to do list acts as an external memory aid. It’s only possible to hold a few pieces of information at one time. Keep a to do list and you’ll be able to keep track of everything, rather than just a few of the tasks you need to do. Your to do list will also reinforce the information, which makes it less likely you’re going to forget something.

**Increases productivity**: A to do list allows you to prioritize the tasks that are more important. This means you don’t waste time on tasks that don’t require your immediate attention. Your list will help you stay focused on the tasks that are the most important.

**Helps with motivation**: To do lists are a great motivational tool because you can use them to clarify your goals. You can divide your long-term goal into smaller, more achievable short-term goals and as you tick each one off your list, your confidence will increase.

## What is a ToDo List in business and why is it important?

It seems such a simple solution by putting pen to paper and taking time out of your day to create a to do list, a plan for your day helps define your challenges and goals. Preventing time from being wasted trying to identify what is the next most important task to tackle next and even more important makes sure you don’t forget to do something important.

To-do lists offer a way to increase productivity, stopping you from forgetting things, helps prioritise tasks, manage tasks effectively, use time wisely and improve time management as well as workflow.

Literature survey

Existing problem:

Social media and other easily accessible online distractions make it hard for us to stay focused on our tasks and make it difficult for us to do our work efficiently.

Also, constantly switching between tasks may give us the false feeling that we are being productive when we are, in fact, not. It’s more important for us to prioritize tasks and work on those that are most important, rather than focusing on deleting small items from our todo list just for the sake of appearances.

The goal of this app is to help us become more aware of how we spend time in the process of doing those tasks and how productive that time is. It can help set some constraints on social media to reduce distraction and track the time we spend working on the todo items. When we have a better sense of the estimated time we’ll need to spend on our tasks, along with the validated time spent on the items for reference or personal/team reviews, we are able to manage our daily routines more efficiently.

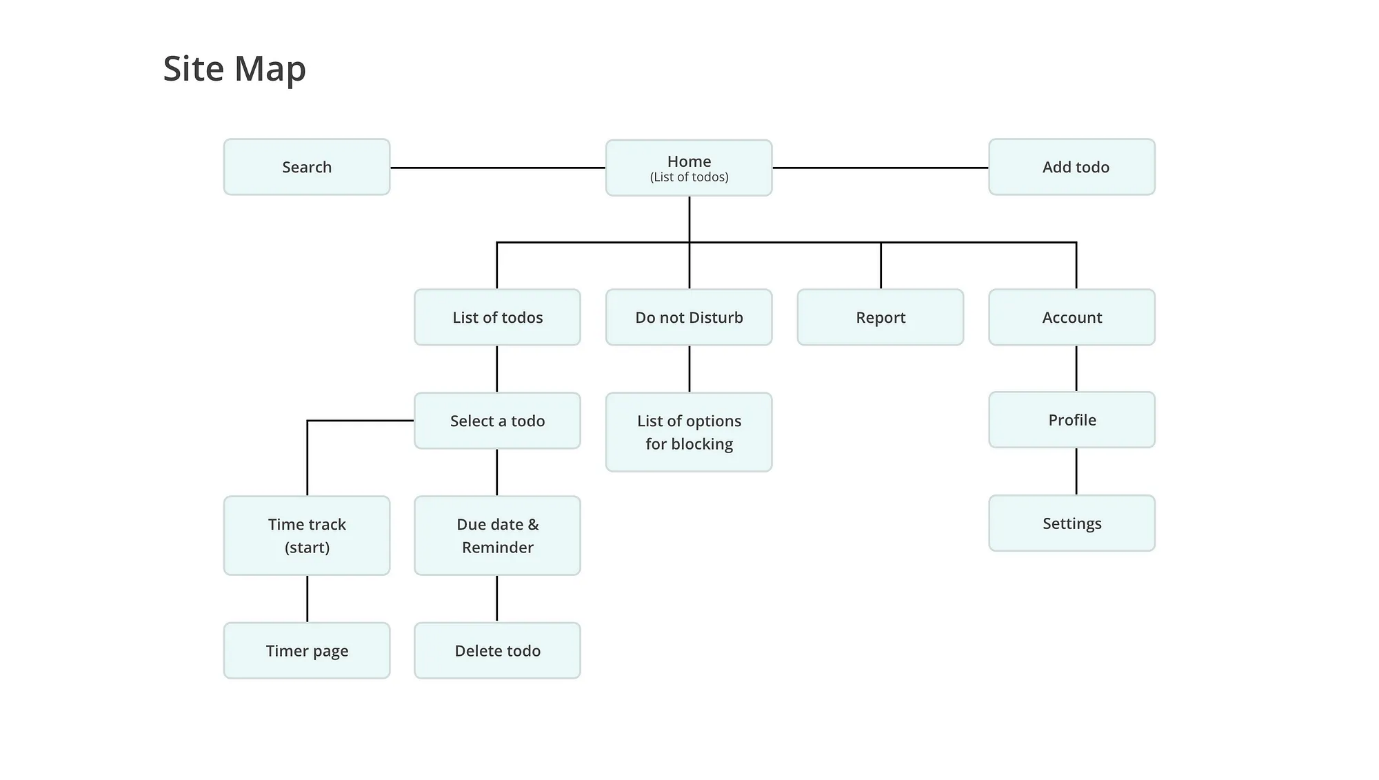
Proposed solution

* **Set some constraints on social media, apps, and online distractions.**  
  (My research indicates that users are heavily distracted in this area.)
* **Manage time on tasks, then validate time spent.**  
  (My research shows that users often lose track of their time and delay work.)
* **Generate a report showing where we spend our time on tasks.**  
  (Users want to know better how their time is spent and how to improve their efficiency.)
* **Make it easy for users to manage todo lists, track time and set constraints in one place.**  
  (Users typically depend on different tools to help organize tasks and track time.)
* **Add the ability to prioritize tasks via deadline.**  
  (Users wish to prioritize todos mainly based on deadlines.)
* **Add the ability to easily add tasks and to search todo lists.**  
  (Users want be able to easily find todo items.)
* **Reduce distractions from mobile by allowing device settings to be synced from desktop.**(Users want to have data sync cross platform.)

It’s very tempting to include many functions in our quest to help users, but all those functions might not lead us toward the target we’re aiming for. The [MoSCow](https://www.toolshero.com/project-management/moscow-method/" \t "_blank) method helps to set the path of product planning and trim down the scope we need for a minimum viable product(MVP). The point is to get the product out for real-world testing so we can have feedback and continue iterating the product.

With social media and other easily accessible online distractions, it’s more difficult than ever for us to stay focused through the course of doing our tasks. In order to handle tasks more efficiently, we must learn to manage digital distractions. This app helps us when we don’t want to engage with these distractions and maintain boundaries at work. The app also provides reporting that will make us more aware of how we spend our time and more informed about how productive we are. As a result, we are able to more efficiently manage our tasks.

Theoretical analysis

Hardware/softwaredesigning

Hardware and software requirements of the project

Index.html

<!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="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" 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>

                                   <!-- <li class="nav-item">

                                        <a class="nav-link" href="#">

                                             <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"

                                                  fill="currentColor" class="bi bi-calendar" viewBox="0 0 16 16">

                                                  <path

                                                       d="M13 2h-1V1a1 1 0 0 0-1-1h-2a1 1 0 0 0-1 1v1H6V1a1 1 0 0 0-1-1H3a1 1 0 0 0-1 1v1H1a1 1 0 0 0-1 1v11a1 1 0 0 0 1 1h12a1 1 0 0 0 1-1V3a1 1 0 0 0-1-1zM2 5h12v8H2V5zm10-3V2h-1v1h-2V2H5v1H4V2H3v1H2V2H1v11h14V2h-1z" />

                                             </svg>

                                             My Day

                                        </a>

                                   </li>

                                   <li class="nav-item">

                                        <a class="nav-link" href="#">

                                             <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"

                                                  fill="currentColor" class="bi bi-exclamation-triangle"

                                                  viewBox="0 0 16 16">

                                                  <path

                                                       d="M8.252 0.275a1.063 1.063 0 0 1 1.496 0l6.98 6.77A1.063 1.063 0 0 1 16 8.464v5.274c0 .587-.476 1.063-1.063 1.063H1.063C.476 15.8 0 15.324 0 14.737V8.463a1.062 1.062 0 0 1 .273-0.71l6.979-6.771ZM8 4.166c-.26 0-.474.177-.474.395v4.938c0 .219.214.396.474.396s.474-0.177.474-0.396V4.561c0-.218-.214-0.396-.474-0.396Zm-0.032 6.14c-.348 0-.63.24-.63.536 0 .295.282.535.63.535 0.349 0 0.63-0.24 0.63-0.535 0-0.296-0.281-0.536-0.63-0.536Z" />

                                             </svg>

                                             Important

                                        </a>

                                   </li>

                                   <li class="nav-item">

                                        <a class="nav-link" href="#">

                                             <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"

                                                  fill="currentColor" class="bi bi-clock" viewBox="0 0 16 16">

                                                  <path

                                                       d="M7.5 1a6.5 6.5 0 0 0-6.496 6.499L1 8.5a.5.5 0 0 0 .5.5h1.645a.5.5 0 0 0 .415-.777L3.8 7.3A4.5 4.5 0 1 1 7.5 2a.5.5 0 0 0 0 1Zm0-1A7.5 7.5 0 1 0 15 7.5a.5.5 0 0 0-1 0A6.5 6.5 0 0 1 7.5 0Z" />

                                                  <path

                                                       d="M7.5 3a.5.5 0 0 0-.5.5v4.085l3.25 1.925a.5.5 0 0 0 .5-0.866L7.5 7.707V3.5A.5.5 0 0 0 7.5 3Z" />

                                             </svg>

                                             Planned

                                        </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 0 1.5 7.5v7a.5.5 0 0 0 .5.5h4.5a.5.5 0 0 0 .5-.5v-4h2v4a.5.5 0 0 0 .5.5H14a.5.5 0 0 0 .5-.5v-7a.5.5 0 0 0-.146-.354L13 5.793V2.5a.5.5 0 0 0-.5-.5h-1a.5.5 0 0 0-.5.5v1.293L8.354 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-.5.5v4H2.5Z" />

                    </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-.5.5h-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-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="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 0 2.704l.944.92-.016 1.32a1.89 1.89 0 0 0 1.912 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 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><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"

                         class="bi bi-list-nested" viewBox="0 0 16 16">

                         <path fill-rule="evenodd"

                              d="M4.5 11.5A.5.5 0 0 1 5 11h10a.5.5 0 0 1 0 1H5a.5.5 0 0 1-.5-.5zm-2-4A.5.5 0 0 1 3 7h10a.5.5 0 0 1 0 1H3a.5.5 0 0 1-.5-.5zm-2-4A.5.5 0 0 1 1 3h10a.5.5 0 0 1 0 1H1a.5.5 0 0 1-.5-.5z" />

                    </svg> Saved Tasks</h5>

               <div id="card-container" class="row">

                    <!-- Cards will be dynamically added here -->

               </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/68SIy3Te4Bkz"

          crossorigin="anonymous"></script>

</body>

</html>

Style.css

#section2{

     margin-top: 10vh;

     padding:  1vh ;

}

.p {

     display: flex;

     align-items: center;

}

svg .bi {

     margin-right: 5px;

     /\* Adjust the spacing between the icon and text \*/

}

.important {

     background-color: #fd826f !important;

     /\* Change the background color to light red \*/

}

#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(0, 0, 0, 0.19) 0px 10px 20px, rgba(0, 0, 0, 0.23) 0px 6px 6px;

}

.card1{

     background-color: rgb(238, 177, 136) !important;

}

Script.js

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) {

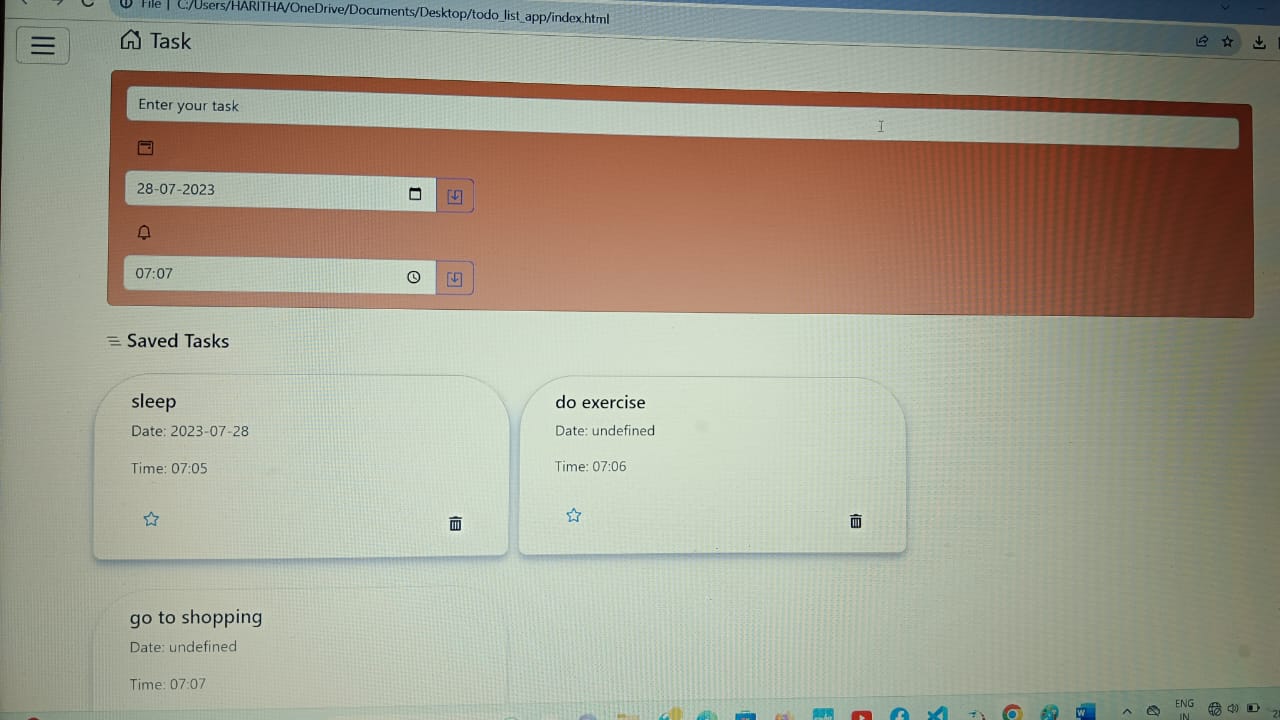
        audio.pause();

        audio.currentTime = 0;

    }

}

Result



Advantages

1. Reorder Tasks – This one is really simple. One of the advantages of an app-based todo list is that you are not limited by ink on paper. Yet, many apps won’t let you easily re-organize the order of the todos on your list.
2. Prioritization – This is another basic time management feature. It amazes me how many apps don’t allow prioritization of tasks. (Or have it poorly implemented.) It is important to be able to separate important tasks from the lesser ones. This should be by default.
3. Quick Entry/Capture – Todo apps should always be ready to *quickly* capture a random task. It shouldn’t require 16 steps to enter a todo. They should be quick to allow you to add a todo and get it “out of your head.”
4. Simple UI – Following on from quick capture, overall the app should be *simple*. Simple to use. Simple user interface. Too many features just clutter things up. And 99% of those extra bells and whistles never get used.
5. Pictures on Tasks – This is one of my favorites, and I feel it is underestimated by most todo apps. Sometimes a photo is a great way to add a task to your list. There  are a few task apps that have this feature, but it is often buried if it exists. (Looking at you Wunderlist.)
6. Sync – Again, one of the benefits of an electronic list should be the ability to sync is across all platforms and locations that the user works. (See #10 below)
7. Today List – You can’t do it all. And one of the best ways to “get work done” is to only concentrate on your top tasks for today. The ability to see a subset or “Hit List” for Today is important.
8. Future Tasks – Todo lists should remember so you don’t have to. Tasks should be able to “send tasks to a future date” and then be out-of-mind until that date. Some of the bigger apps have this, but I would like to see it be easier to use and more intuitive. (Todoist does a good job at making this feature easy to use.)
9. Voice Capture – One of the quickest ways to interact with our devices on the go is by voice commands. Yet, it is still not a ubiquitous feature, and many apps use workarounds or third-party systems. I would like to see more apps that allow voice capture from the lock screen.
10. **On Every Platform** – Your todo list needs to be *wherever*you are. Regardless of device or platform. The ideal todo app must be available for the majority of platforms. For me this list must include: Mobile (iPhone and Android), and desktop apps (Mac and Windows). A web interface is good too, or could be a stepping-stone to desktop apps. It is hard for smaller developers to cover all these bases, but as they grow, apps need to be available everywhere their users work.
11. Pretty Design – I bet you didn’t expect this one. But, yes, I am saying that the app has to be visually appealing. Sarcasm aside, this is basic UI design. The app has to look good and be *readable*, too.
12. Fun to Use – Last but not least, users have to *want* to use your productivity app. (Or *any app!)* People use tools they enjoy and work for them. One of the best recent examples of this is Slack. Users *love* using it. Unlike their email app, Slack is *fun to use and gets the work done.* The same must be true of the ideal todo app.

Disadvantage

* Gives you an excuse to remove tasks from your to-do list that you SHOULD do, but CAN put off
* Allows you to avoid these tasks as long as possible
* Promotes procrastination and task avoidance
* A lengthy running to-do list can be intimidating and cluttered

Applications

1. It's accessible anywhere and on every device. Browser, Android or IOS it works everywhere
2. You can add images, Audio or you can scribble.
3. It allows to add check list, which helps while I go for grocery shopping.
4. You can share the notes with anyone else, it makes easy to collaborate.
5. It's easy to use, fast and very reliable.
6. You can put labels on notes, it makes easy to separate notes from one another

Conclusion

The to-do list developed for this project is meant to be a minimal free and open source application that can help improve the productivity of a user without taking anything.It aims to solve at least some of the problems discussed in the earlier report.

Futurescope

Deviations from expected results and way ahead;

The feature to and list tasks by their priority was implemented towards the end of the project as we had initially planned to classify tasks into

Two lists.

* Several implementation problems did not have any solutions that could be solved directly using third party libraries so,they had to be solved by manually implementing the functionalities.
* Improvement in security and integrity
* Integrations with other applications such as calender,mail,etc.
* Implementation of authentication
* Hosting the web application online to make it accessible to more users.

To-dos are the tasks or the atomic entities that make up a to-do list. To-dos are made

quickly, the bulk of them do not specify the work; instead, they are typically just

comprehensive enough to serve as a valuable indicator. To be sure, to-do terminology

like "Groceries" or "Car Wash" is frequently grammatically correct. Because the

signal is so quick, it is only useful for a short period of time while the task is

remembered. In certain cases, a simple item like a stack is enough to recall the job

without the need for a note

To-dos are the tasks or the atomic entities that make up a to-do list. To-dos are made

quickly, the bulk of them do not specify the work; instead, they are typically just

comprehensive enough to serve as a valuable indicator. To be sure, to-do terminology

like "Groceries" or "Car Wash" is frequently grammatically correct. Because the

signal is so quick, it is only useful for a short period of time while the task is

remembered. In certain cases, a simple item like a stack is enough to recall the job

without the need for a note

To-dos are the tasks or the atomic entities that make up a to-do list. To-dos are made

quickly, the bulk of them do not specify the work; instead, they are typically just

comprehensive enough to serve as a valuable indicator. To be sure, to-do terminology

like "Groceries" or "Car Wash" is frequently grammatically correct. Because the

signal is so quick, it is only useful for a short period of time while the task is

remembered. In certain cases, a simple item like a stack is enough to recall the job

without the need for a note

To-dos are the tasks or the atomic entities that make up a to-do list. To-dos are made

quickly, the bulk of them do not specify the work; instead, they are typically just

comprehensive enough to serve as a valuable indicator. To be sure, to-do terminology

like "Groceries" or "Car Wash" is frequently grammatically correct. Because the

signal is so quick, it is only useful for a short period of time while the task is

remembered. In certain cases, a simple item like a se

quickly, the bulk of them do not specify the work; instead, they are typically just

comprehensive enough to serve as a valuable indicator. To be sure, to-do terminology

like "Groceries" or "Car Wash" is frequently grammatically correct. Because the

signal is so quick, it is only useful for a short period of time while the task is

remembered. In certain cases, a simple item like a stack is enough to recall the job

without the need for a note