**Task 3 Report**

* **Task – 3 : To-DoApp.-**
* **Project Description:**

Task 3 - TO-DO App Development! This project will challenge your skills in HTML, CSS, and JavaScript as you embark on the journey of creating a functional TO-DO App. The goal is to design a user-friendly interface that allows users to manage their tasks seamlessly.

* **Project Details:**

HTML: Structure your app.

CSS: Style your app for an appealing look.

JavaScript: Add dynamic and interactive functionalities

1. **Task Management:**

Enable usersto add, edit, and delete tasks.

Implement task prioritization and status (completed or pending).

1. **Local Storage:**

Provide functionality forsaving taskslocally, allowing usersto revisit their tasks later.

* **Project Timeline:**

• Start Date: May 31th, 2024.

• Deadline: June 8th, 2024.

* **Steps Taken:**
* Created a folder given name is – Task3
* Opened Task3 folder in Code Editor like VS Code
* Created a HTML file which name is – index.HTML
* Created a CSS file which name is – styles.css
* Inserted the image for logo
* Started Creating Web app using HTML, CSS, JavaScript
* Checked Code by clicking Run Button.
* On browser seen the Output.
* **Challenges Faced:**
  + Create a JavaScript Function for To-do app
  + Create a Time and Date tag to save time and date
* **Solution Implemented**

Create a HTML Tag

 <div class="container">

        <h1>TO-DO App</h1>

        <div class="input-group">

            <input type="text" id="task-input" placeholder="Add a new task...">

            <input type="datetime-local" id="task-datetime">

            <button id="add-task-btn">Add Task</button>

        </div>

        <ul id="task-list"></ul>

    </div>

Script for Save the task and save local time and date edit delete option

document.addEventListener('DOMContentLoaded', () *=>* {

*const* taskInput = document.getElementById('task-input');

*const* taskDateTime = document.getElementById('task-datetime');

*const* addTaskBtn = document.getElementById('add-task-btn');

*const* taskList = document.getElementById('task-list');

    // Set the default value of the date-time input to the current date and time in IST

    setDefaultDateTime();

    // Load tasks from local storage

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

    tasks.forEach(*task* *=>* renderTask(*task*));

    addTaskBtn.addEventListener('click', addTask);

    taskList.addEventListener('click', handleTaskActions);

*function* setDefaultDateTime() {

*const* now = new *Date*();

*const* nowIST = new *Date*(now.getTime() + (5.5 \* 60 \* 60 \* 1000)); // Convert to IST

        taskDateTime.value = nowIST.toISOString().slice(0, 16);

    }

*function* addTask() {

*const* taskText = taskInput.value.trim();

*const* taskDateTimeValue = taskDateTime.value;

        if (taskText === '' || taskDateTimeValue === '') return;

*const* task = {

            id: *Date*.now(),

            text: taskText,

            dateTime: taskDateTimeValue,

            completed: false

        };

        tasks.push(task);

        renderTask(task);

        saveTasks();

        taskInput.value = '';

        setDefaultDateTime(); // Reset to current date and time in IST

    }

*function* renderTask(*task*) {

*const* li = document.createElement('li');

        li.classList.add('task');

        if (*task*.completed) li.classList.add('completed');

        li.dataset.id = *task*.id;

        li.innerHTML = `

            <span>${*task*.text}</span>

            <span class="task-datetime">${new *Date*(*task*.dateTime).toLocaleString('en-IN', { timeZone: 'Asia/Kolkata' })}</span>

            <div class="task-buttons">

                <button class="edit-btn">Edit</button>

                <button class="delete-btn">Delete</button>

            </div>

        `;

        taskList.appendChild(li);

    }

*function* handleTaskActions(*e*) {

*const* li = *e*.target.closest('li');

*const* taskId = li.dataset.id;

        if (*e*.target.classList.contains('delete-btn')) {

            deleteTask(taskId);

        } else if (*e*.target.classList.contains('edit-btn')) {

            editTask(taskId);

        }

    }

*function* deleteTask(*id*) {

        tasks = tasks.filter(*task* *=>* *task*.id != *id*);

        document.querySelector(`[data-id='${*id*}']`).remove();

        saveTasks();

    }

*function* editTask(*id*) {

*const* task = tasks.find(*task* *=>* *task*.id == *id*);

*const* newTaskText = prompt('Edit your task', task.text);

*const* newTaskDateTime = prompt('Edit your date and time', task.dateTime);

        if (newTaskText !== null && newTaskDateTime !== null) {

            task.text = newTaskText;

            task.dateTime = newTaskDateTime;

            document.querySelector(`[data-id='${*id*}'] span`).textContent = newTaskText;

            document.querySelector(`[data-id='${*id*}'] .task-datetime`).textContent = new *Date*(newTaskDateTime).toLocaleString('en-IN', { timeZone: 'Asia/Kolkata' });

            saveTasks();

        }

    }

*function* saveTasks() {

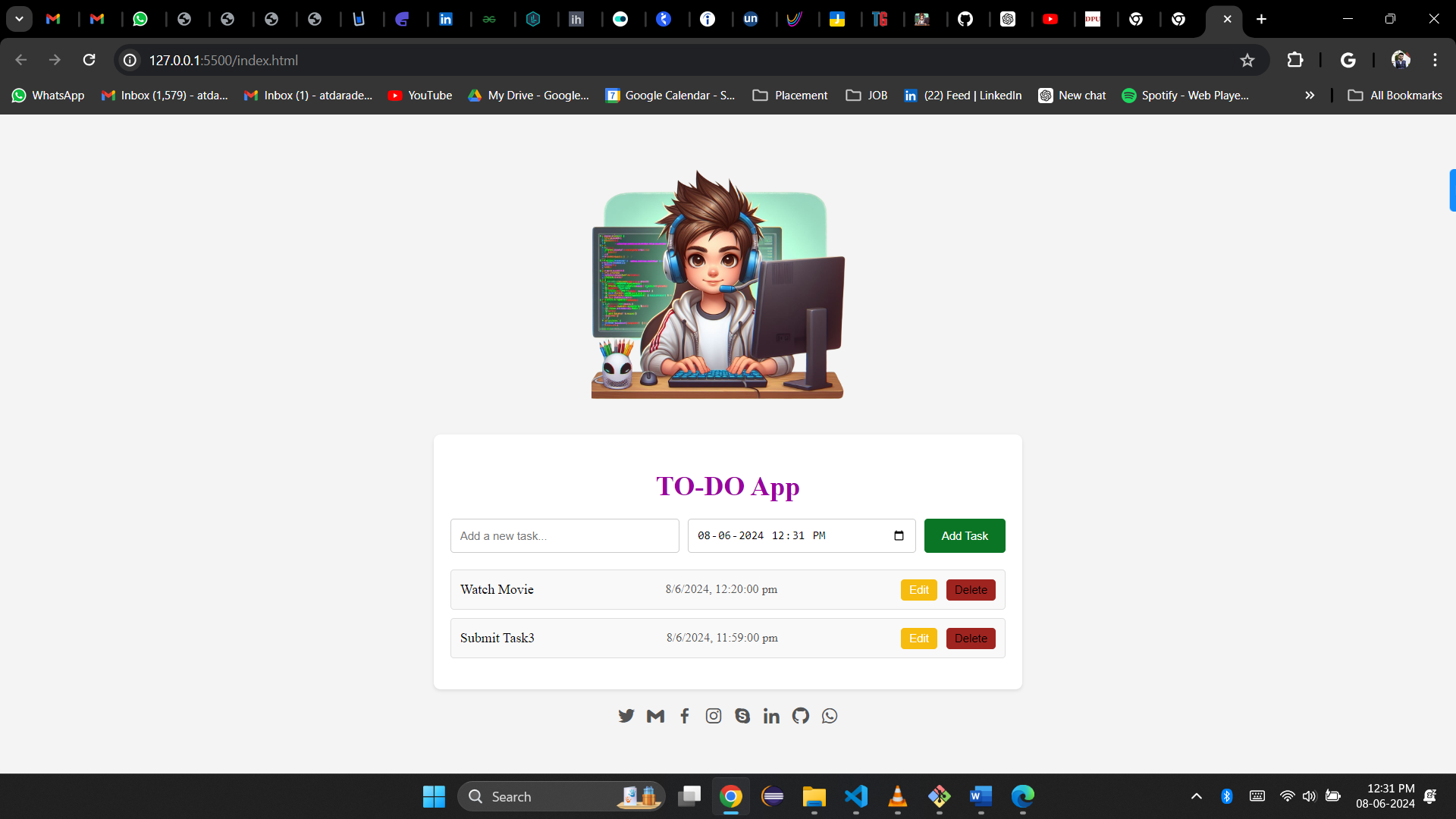
        localStorage.setItem('tasks', JSON.stringify(tasks));

    }

});

* **Output :**

Visit this Website to see the Output- [https://todo-save.netlify.app](https://todo-save.netlify.app/)



* **Learnings:**

How to make objects in HTML using CSS and JavaScript.

How to Save Data in Local Storage.

How to use Time and Date

* **Task1 Planning and Progress Structure:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task- 2 Planning and Progress Report** | | | |
| **Sr. No.** | **Activity** | **Status** | **Due** |
| 1 | Task assigned | **On Track** | 31/05/2024 |
| 2 | Create a folder and files and thinking topic | **On Track** | 01/06/2024 |
| 3 | Create a HTML Page | **On Track** | 02/06/2024 |
| 4 | Apply CSS for HTML page | **On Track** | 03/06/2024 |
| 5 | Insert Date input tag to save date | **On Track** | 04/06/2024 |
| 6 | Function Building and Implementation | **On Track** | 05/06/2024 |
| 7 | Checking and testing | **On Track** | 06/06/2024 |
| 8 | Adding social Links | **On Track** | 07/06/2024 |
| 9 | Git Upload and Submission | **On Track** | 08/06/2024 |