Introduction

The Task Tracker application is a simple, user-friendly tool designed to help manage tasks effectively. Built with vanilla JavaScript, HTML, and CSS, it provides basic CRUD (Create, Read, Update, Delete) operations for tasks.

Setup Instructions

1. **Create Project Directory**

bash

Copy code

```
mkdir task-tracker

cd task-tracker
```

2. **Create Basic Files**
    - **index.html**
    - **styles.css**
    - **script.js**
3. **Folder Structure**

Copy code

```
task-tracker/

 ├── index.html

 ├── styles.css

└── script.js
```

4. **Code for index.html**

html

Copy code

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Task Tracker</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Task Tracker</h1>
    <div class="task-input">
      <input type="text" id="taskInput" placeholder="Add a new task">
      <button id="addTaskButton">Add Task</button>
    </div>
    <ul id="taskList"></ul>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

5. **Code for styles.css**

css

Copy code

```
body {
  font-family: Arial, sans-serif;
  background: linear-gradient(to right, #ff7e5f, #feb47b);
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  animation: fadeInBackground 1s ease-in-out;
}

@keyframes fadeInBackground {
  from { opacity: 0; }
  to { opacity: 1; }
}

.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);
  width: 350px;
  text-align: center;
  animation: fadeInContainer 1s ease-in-out;
}

@keyframes fadeInContainer {
  from { transform: translateY(-50px); opacity: 0; }
  to { transform: translateY(0); opacity: 1; }
}

h1 {
  margin-bottom: 20px;
  color: #ff7e5f;
}
```

```
.task-input {

  display: flex;

  justify-content: space-between;

  margin-bottom: 20px;

}


#taskInput {

  width: 70%;

  padding: 10px;

  border: 1px solid #ccc;

  border-radius: 5px;

  outline: none;

  transition: all 0.3s ease;

}


#taskInput:focus {

  border-color: #ff7e5f;

  box-shadow: 0 0 5px rgba(255, 126, 95, 0.5);

}


#addTaskButton {

  padding: 10px;

  background-color: #ff7e5f;

  color: #fff;

  border: none;

  border-radius: 5px;
  cursor: pointer;
  transition: all 0.3s ease;
}

#addTaskButton:hover {
  background-color: #feb47b;
  transform: scale(1.05);
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  background-color: #f9f9f9;
  padding: 10px;
  margin-bottom: 10px;
  border-radius: 5px;
  display: flex;
  justify-content: space-between;
  align-items: center;
  transition: all 0.3s ease;
  animation: fadeInTask 0.5s ease-in-out;
}

@keyframes fadeInTask {
  from { transform: translateX(-50px); opacity: 0; }
  to { transform: translateX(0); opacity: 1; }
}

li .actions {
  display: flex;
}

li .actions button {
  background-color: #dc3545;
  color: #fff;
  border: none;
  border-radius: 5px;
  margin-left: 5px;
  cursor: pointer;
  padding: 5px;
  transition: all 0.3s ease;
}

li .actions button.edit {
  background-color: #007bff;
}

li .actions button:hover {
  opacity: 0.8;
}
```

6. **Code for script.js**

javascript

Copy code

```
document.addEventListener("DOMContentLoaded", function() {
  const taskInput = document.getElementById("taskInput");
  const addTaskButton = document.getElementById("addTaskButton");
  const taskList = document.getElementById("taskList");

  addTaskButton.addEventListener("click", addTask);
  taskList.addEventListener("click", handleTaskActions);

  function addTask() {
    const taskText = taskInput.value.trim();
    if (taskText === "") {
      alert("Please enter a task.");
      return;
    }
    const taskItem = createTaskItem(taskText);
    taskList.appendChild(taskItem);
    taskInput.value = "";
  }

  function createTaskItem(taskText) {
    const li = document.createElement("li");
    li.textContent = taskText;

    const actions = document.createElement("div");
    actions.classList.add("actions");
    const editButton = document.createElement("button");
    editButton.textContent = "Edit";
    editButton.classList.add("edit");
    editButton.onclick = () => editTask(li);

    const deleteButton = document.createElement("button");
    deleteButton.textContent = "Delete";
    deleteButton.onclick = () => deleteTask(li);

    actions.appendChild(editButton);
    actions.appendChild(deleteButton);
    li.appendChild(actions);

    return li;
  }

  function handleTaskActions(event) {
    if (event.target.tagName === "BUTTON") {
      const action = event.target.textContent.toLowerCase();
      const taskItem = event.target.closest("li");

      if (action === "edit") {
        editTask(taskItem);
      } else if (action === "delete") {
        deleteTask(taskItem);
      }
    }
   }

   function editTask(taskItem) {
    const newTaskText = prompt("Edit task:", taskItem.firstChild.textContent);
    if (newTaskText !== null && newTaskText.trim() !== "") {
     taskItem.firstChild.textContent = newTaskText.trim();
    }
   }

   function deleteTask(taskItem) {
    taskItem.remove();
   }
  });
```

Functionalities

1. **Add Task**
   - Users can add a new task by entering text in the input field and clicking the "Add Task" button.
   - The task will be displayed in the task list.
2. **Display Tasks**
   - All tasks are displayed in a list format.
3. **Update Task**
   - Users can edit an existing task by clicking the "Edit" button next to the task.
   - A prompt will appear, allowing the user to enter a new task description.
4. **Delete Task**
   - Users can delete a task by clicking the "Delete" button next to the task.

Testing

1. **Add Task**
   - Enter text in the input field and click the "Add Task" button.
   - Verify that the task appears in the task list.
2. **Edit Task**
   - Click the "Edit" button next to a task.
   - Enter a new task description in the prompt and verify that the task updates in the list.
3. **Delete Task**
   - Click the "Delete" button next to a task.
   - Verify that the task is removed from the list.