TODO APP

Submitted By

Yajjala Srinu – 20981A4662

Kollati Ravi Varma – 20981A4632

Submitted To

INTRODUCTION

1.1 Overview

The Todo App is a web application that allows users to create and manage their daily tasks or to-do lists. Users can add new tasks, mark tasks as completed, and delete tasks. The app provides a user-friendly interface and uses JavaScript, HTML, CSS, and Bootstrap for its implementation.

1.2 Scope of the project

The scope of a todo app can vary based on its intended purpose and features. Generally, a todo app is designed to help users organize, manage, and track their tasks, to-do lists, and other activities. Here are some common features that can be included in a typical todo app:

1. Task Creation: Users can add new tasks or to-do items to their list, providing a title and optional details such as due dates, priority levels, tags, or descriptions.

2. Task Management: Users can edit, delete, mark as completed, or reschedule tasks.

3. Task Categories/Tags: The app may support categorizing tasks with labels or tags to help users group related tasks together.

4. Task Sorting and Filtering: Users can sort tasks based on priority, due date, or completion status. They can also filter tasks based on specific criteria.

5. Reminders and Notifications: The app can provide reminders or notifications for upcoming or overdue tasks, helping users stay on track.

It's important to note that the scope of a todo app can vary depending on its target audience and the platform it is built for (e.g., mobile, web, desktop). Developers may choose to include more advanced features or keep it simple and focused on the core task management functionalities.

Always consider the target users and their specific needs when designing a todo app to ensure it provides a smooth and efficient task management experience.

1.3 Softwares Used To Compile

1.Visual Studio

2.SublimeText

3.NotePad etc…

Advantages :

Todo apps, also known as task management or to-do list apps, have become popular tools to help individuals and teams stay organized and productive. Below are some advantages and disadvantages of using todo apps:

1. Organization: Todo apps provide a structured way to organize tasks and responsibilities, making it easier to keep track of what needs to be done.

2. Prioritization: Most todo apps allow you to prioritize tasks, helping you focus on the most important ones first and avoid feeling overwhelmed.

3. Reminders: Todo apps often come with reminder features, which can be set to notify you about upcoming or overdue tasks, reducing the risk of forgetting important deadlines.

4. Accessibility: With many todo apps available on multiple devices (smartphones, tablets, and computers), you can access your tasks from anywhere, ensuring you never miss an update or important task.

5. Collaboration: Some todo apps offer collaboration features, allowing teams to share tasks, delegate responsibilities, and work together seamlessly.

6. Time Management: By organizing tasks, setting due dates, and using reminders, todo apps can help you manage your time more effectively, leading to increased productivity.

7. Goal Setting: Todo apps can aid in setting and tracking progress towards goals, as you can break larger projects into smaller tasks and monitor your achievements.

Disadvantages:

1. Overwhelm: With too many tasks or an unclear organization, a todo app can become overwhelming, leading to decreased productivity and added stress.

2. Dependency: Relying solely on a todo app can lead to a lack of mental engagement with tasks, potentially affecting long-term memory retention and overall understanding.

3. Learning Curve: Some todo apps come with a learning curve, especially if they have advanced features. Learning to use the app effectively may take time initially.

4. Distractions: Constant notifications from the app can be distracting and disrupt your workflow if not managed properly.

5. Technical Issues: Like any software, todo apps can have technical glitches or outages, which could temporarily disrupt access to your tasks.

6. Over-Organizing: Spending too much time organizing and categorizing tasks in the app may lead to "analysis paralysis" and waste valuable time that could be spent on completing tasks.

7. Lack of Adaptability: Some todo apps may not be flexible enough to accommodate changing needs or different types of workflows, making them less suitable for certain individuals or teams.

In summary, todo apps can be powerful tools for task management and productivity if used wisely. However, it's essential to strike a balance, avoid over-reliance, and choose an app that aligns with your specific needs and preferences.

Source Code:

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css
" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous" />
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>
    <script src="https://kit.fontawesome.com/5f59ca6ad3.js"
crossorigin="anonymous"></script>

<link rel="stylesheet" type="text/css" href="todo1 app.css">
  </head>
  <body>
    <div class="todos-bg-container">
      <div class="container">
        <div class="row">
          <div class="col-12">
            <h1 class="todos-heading">Todos</h1>
            <h1 class="create-task-heading">
              Create <span class="create-task-heading-subpart">Task</span>
            </h1>
            <input type="text" id="todoUserInput" class="todo-user-input"
placeholder="What needs to be done?"/>
            <button class="button" id="addTodoButton">Add</button>
            <h1 class="todo-items-heading">
              My <span class="todo-items-heading-subpart">Tasks</span>
            </h1>
            <ul class="todo-items-container" id="todoItemsContainer"></ul>
            <button class="button" id="saveTodoButton">Save</button>
          </div>
```

```
        </div>
      </div>
    </div>
    <script src="todo1 app.js"></script>
  </body>
</html>
```

CSS

```
.todos-bg-container {
  background-color: #f9fbfe;
  height: 100vh;
}

.todos-heading {
  text-align: center;
  font-family: "Roboto";
  font-size: 46px;
  font-weight: 500;
  margin-top: 20px;
  margin-bottom: 20px;
}

.create-task-heading {
  font-family: "Roboto";
  font-size: 32px;
  font-weight: 700;
}

.create-task-heading-subpart {
  font-family: "Roboto";
  font-size: 32px;
  font-weight: 500;
}

.todo-items-heading {
  font-family: "Roboto";
  font-size: 32px;
  font-weight: 700;
}

.todo-items-heading-subpart {
  font-family: "Roboto";
  font-size: 32px;
  font-weight: 500;
}
```

```css
.todo-items-container {
  margin: 0px;
  padding: 0px;
}

.todo-item-container {
  margin-top: 15px;
}

.todo-user-input {
  background-color: white;
  width: 100%;
  border-style: solid;
  border-width: 1px;
  border-color: #e4e7eb;
  border-radius: 10px;
  margin-top: 10px;
  padding: 15px;
}

.button {
  color: white;
  background-color: #4c63b6;
  font-family: "Roboto";
  font-size: 18px;
  border-width: 0px;
  border-radius: 4px;
  margin-top: 20px;
  margin-bottom: 50px;
  padding-top: 5px;
  padding-bottom: 5px;
  padding-right: 20px;
  padding-left: 20px;
}

.label-container {
  background-color: #e6f6ff;
  width: 100%;
  border-style: solid;
  border-width: 5px;
  border-color: #096f92;
  border-right: none;
  border-top: none;
  border-bottom: none;
  border-radius: 4px;
}

.checkbox-input {
```

```
  width: 20px;
  height: 20px;
  margin-top: 12px;
  margin-right: 12px;
}

.checkbox-label {
  font-family: "Roboto";
  font-size: 16px;
  font-weight: 400;
  width: 82%;
  margin: 0px;
  padding-top: 10px;
  padding-bottom: 10px;
  padding-left: 20px;
  padding-right: 20px;
  border-radius: 5px;
}

.delete-icon-container {
  text-align: right;
  width: 18%;
}

.delete-icon {
  padding: 15px;
}

.checked {
  text-decoration: line-through;
}
```

JavaScript

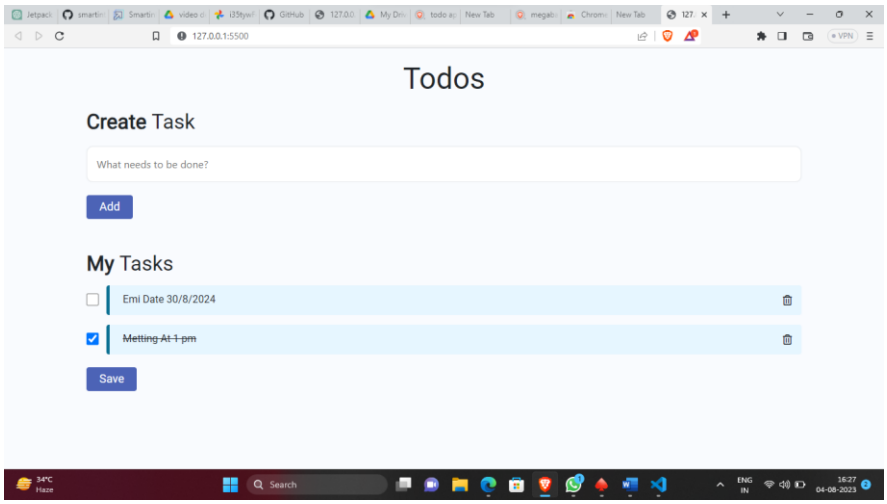
```
let todoItemsContainer = document.getElementById("todoItemsContainer");
let addTodoButton = document.getElementById("addTodoButton");
let saveTodoButton = document.getElementById("saveTodoButton");

function getTodoListFromLocalStorage() {
  let stringifiedTodoList = localStorage.getItem("todoList");
  let parsedTodoList = JSON.parse(stringifiedTodoList);
  if (parsedTodoList === null) {
    return [];
  } else {
    return parsedTodoList;
  }
}
```

```
let todoList = getTodoListFromLocalStorage();
let todosCount = todoList.length;

saveTodoButton.onclick = function() {
  localStorage.setItem("todoList", JSON.stringify(todoList));
};

function onAddTodo() {
  let userInputElement = document.getElementById("todoUserInput");
  let userInputValue = userInputElement.value;

  if (userInputValue === "") {
    alert("Enter Valid Text");
    return;
  }

  todosCount = todosCount + 1;

  let newTodo = {
    text: userInputValue,
    uniqueNo: todosCount,
    isChecked: false
  };
  todoList.push(newTodo);
  createAndAppendTodo(newTodo);
  userInputElement.value = "";
}

addTodoButton.onclick = function() {
  onAddTodo();
};

function onTodoStatusChange(checkboxId, labelId, todoId) {
  let checkboxElement = document.getElementById(checkboxId);
  let labelElement = document.getElementById(labelId);
  labelElement.classList.toggle("checked");

  let todoObjectIndex = todoList.findIndex(function(eachTodo) {
    let eachTodoId = "todo" + eachTodo.uniqueNo;

    if (eachTodoId === todoId) {
      return true;
    } else {
      return false;
    }
  });
```

```
  let todoObject = todoList[todoObjectIndex];

  if(todoObject.isChecked === true){
    todoObject.isChecked = false;
  } else {
    todoObject.isChecked = true;
  }

}

function onDeleteTodo(todoId) {
  let todoElement = document.getElementById(todoId);
  todoItemsContainer.removeChild(todoElement);

  let deleteElementIndex = todoList.findIndex(function(eachTodo) {
    let eachTodoId = "todo" + eachTodo.uniqueNo;
    if (eachTodoId === todoId) {
      return true;
    } else {
      return false;
    }
  });

  todoList.splice(deleteElementIndex, 1);
}

function createAndAppendTodo(todo) {
  let todoId = "todo" + todo.uniqueNo;
  let checkboxId = "checkbox" + todo.uniqueNo;
  let labelId = "label" + todo.uniqueNo;

  let todoElement = document.createElement("li");
  todoElement.classList.add("todo-item-container", "d-flex", "flex-row");
  todoElement.id = todoId;
  todoItemsContainer.appendChild(todoElement);

  let inputElement = document.createElement("input");
  inputElement.type = "checkbox";
  inputElement.id = checkboxId;
  inputElement.checked = todo.isChecked;

  inputElement.onclick = function () {
    onTodoStatusChange(checkboxId, labelId, todoId);
  };

  inputElement.classList.add("checkbox-input");
  todoElement.appendChild(inputElement);
```

```
  let labelContainer = document.createElement("div");
  labelContainer.classList.add("label-container", "d-flex", "flex-row");
  todoElement.appendChild(labelContainer);

  let labelElement = document.createElement("label");
  labelElement.setAttribute("for", checkboxId);
  labelElement.id = labelId;
  labelElement.classList.add("checkbox-label");
  labelElement.textContent = todo.text;
  if (todo.isChecked === true) {
    labelElement.classList.add("checked");
  }
  labelContainer.appendChild(labelElement);

  let deleteIconContainer = document.createElement("div");
  deleteIconContainer.classList.add("delete-icon-container");
  labelContainer.appendChild(deleteIconContainer);

  let deleteIcon = document.createElement("i");
  deleteIcon.classList.add("far", "fa-trash-alt", "delete-icon");

  deleteIcon.onclick = function () {
    onDeleteTodo(todoId);
  };

  deleteIconContainer.appendChild(deleteIcon);
}

for (let todo of todoList) {
  createAndAppendTodo(todo);
}
```

Result:



Conclusion:

By the end of this project, you will be able to:

1. Create a responsive and interactive user interface using HTML, CSS, and Bootstrap.
2. Implement functionality to add, complete, and delete tasks using JavaScript.
3. Provide filtering options to view tasks based on their status.