Create a To Do List using HTML programming languages

Abstract:

This study aims to develop a simple yet effective To-Do list application using HTML and evaluate its usability and functionality. In today's fast-paced world, individuals often struggle to manage their tasks efficiently, leading to stress and decreased productivity. A digital To-Do list offers a convenient solution for organizing and prioritizing tasks. This research focuses on creating an HTML-based To-Do list application with basic features such as task entry, deletion, and completion tracking. Through user testing and feedback collection, the study aims to assess the application's usability and identify areas for improvement. This study focuses on the development and evaluation of a simple HTMLbased To-Do list application. With the increasing demands of modern life, effective task management has become essential for individuals to stay organized and productive. Digital To-Do list applications offer a convenient solution for managing tasks, but many existing options are complex and feature-rich, which may overwhelm users seeking a basic task management tool. Therefore, this research aims to create a straightforward HTML To-Do list application with essential features such as task entry, deletion, and completion tracking. Through usability testing and feedback collection, the study evaluates the usability and functionality of the application, aiming to provide insights into creating user-friendly and accessible task management solutions. Against this backdrop, this research aims to explore the development, evaluation, and optimization of a To-Do list application that prioritizes simplicity, usability, and effectiveness.

The research seeks to address the following objectives:

Designing a User-Centric To-Do List Application:

The first objective is to design a To-Do list application that places the needs and preferences of users at the forefront. This involves understanding the diverse needs and workflows of potential users through user research, interviews, and surveys. Based on these insights, the application will be designed with a focus on simplicity, intuitiveness, and functionality to ensure a positive user experience.

Implementing Essential Task Management Features:

The second objective is to implement essential features within the To-Do list application that enable users to effectively organize, prioritize, and track their tasks. These features may include task creation, editing, and deletion functionalities, as well as options for setting due dates, assigning priorities, and categorizing tasks. Additionally, the application may incorporate reminders, notifications, and collaboration features to enhance usability and productivity.

Assessing Usability and User Experience:

The third objective is to assess the usability and user experience of the To-Do list application through rigorous testing and evaluation. Usability testing will involve observing users as they interact with the application and collecting feedback on their experiences. This feedback will be used to identify usability issues, usability strengths, and areas for improvement within the application.

Evaluating System Functionality and Performance:

The fourth objective is to evaluate the functionality and performance of the To-Do list application to ensure it meets the desired specifications and requirements. System functionality will be assessed through a series of tests to verify that all essential features are functioning as intended and meeting user needs. Performance testing will involve measuring the application's response times, resource utilization, and scalability to ensure it can accommodate varying levels of user activity and workload.

Iterative Refinement and Optimization: The fifth objective is to iteratively refine and optimize the To-Do list application based on feedback and evaluation results. This iterative process will involve analyzing usability testing data, user feedback, and application performance metrics to identify areas for improvement and implement necessary changes. These changes may include modifying the user interface, enhancing feature functionality, or addressing technical issues to improve the overall usability and effectiveness of the application.

Providing Recommendations for Future Development: The final objective is to provide recommendations for future development and enhancement of the To-Do list application based on research findings and insights. These recommendations may include suggestions for additional features or functionalities, improvements to existing features, or enhancements to the user interface and user experience. Additionally, recommendations may be provided for addressing specific usability issues, optimizing application performance, or accommodating emerging trends and technologies.

Objectives:

Task management is a crucial aspect of daily life, influencing productivity, efficiency, and overall well-being. In today's fast-paced world, individuals are constantly juggling multiple responsibilities, ranging from work assignments and personal projects to household chores and social commitments. Effective task management requires a systematic approach to organizing, prioritizing, and tracking tasks to ensure they are completed in a timely manner. To address the challenges associated with task management, the development and evaluation of a task management system are paramount. Therefore, this research aims to delineate the objectives of creating and assessing a task management system, encompassing usability, functionality, and user satisfaction.

Designing a User-Centric Task Management System: The first objective of this research is to design a user-centric task management system that prioritizes simplicity, intuitiveness, and functionality. The system will be developed with a focus on user experience, ensuring that it meets the diverse needs and preferences of users. Key design considerations will include an intuitive user interface, streamlined task entry and management processes, and clear navigation pathways. By adopting a user-centric approach to design, the task management system aims to enhance usability and facilitate efficient task management for users of all skill levels.

Implementing Essential Task Management Features: The second objective is to implement essential task management features within the system, enabling users to organize, prioritize, and track their tasks effectively. These features will include task creation, editing, and deletion functionalities, as well as options for setting due dates, assigning priorities, and categorizing tasks. Additionally, the system will incorporate reminders and notifications to alert users of upcoming deadlines and overdue tasks. By implementing these essential features, the task management system aims to provide users with comprehensive tools for managing their tasks efficiently.

Assessing Usability and User Experience: The third objective is to assess the usability and user experience of the task management system through rigorous testing and evaluation. Usability testing will involve observing users as they interact with the system and collecting feedback on their experiences. This feedback will be used to identify usability issues, usability strengths, and areas for improvement within the system. Additionally, user satisfaction surveys will be administered to gather quantitative data on users' perceptions of the system's usability, functionality, and overall user experience. By assessing usability and user experience, the research aims to validate the effectiveness of the task management system in meeting users' needs and expectations.

Evaluating System Functionality and Performance: The fourth objective is to evaluate the functionality and performance of the task management system to ensure it meets the desired specifications and requirements. System functionality will be assessed through a series of tests to verify that all essential features are functioning as intended and meeting user needs. Performance testing will involve measuring the system's response times, resource utilization, and scalability to ensure it can accommodate varying levels of user activity and workload. By evaluating system functionality and performance, the research aims to identify any technical issues or limitations that may impact the system's usability and effectiveness.

Iterative Refinement and Optimization : The fifth objective is to iteratively refine and optimize the task management system based on feedback and evaluation results. This iterative process will involve analyzing usability testing data, user feedback, and system performance metrics to identify areas for improvement and implement necessary changes. These changes may include modifying the user interface, enhancing feature functionality, or addressing technical issues to improve the overall usability and effectiveness of the system. By iteratively refining and optimizing the system, the research aims to ensure that it remains relevant, functional, and user-friendly in the face of evolving user needs and technological advancements.

Providing Recommendations for Future Development: The final objective is to provide recommendations for future development and enhancement of the task management system based on research findings and insights. These recommendations may include suggestions for additional features or functionalities, improvements to existing features, or enhancements to the user interface and user experience. Additionally, recommendations may be provided for addressing specific usability issues, optimizing system performance, or accommodating emerging trends and technologies. By providing recommendations for future development, the research aims to guide ongoing efforts to improve and evolve the task management system to better meet the needs of users and adapt to changing environments.

This research aims to delineate the objectives of creating and assessing a task management system, encompassing usability, functionality, and user satisfaction. By designing a user-centric system, implementing essential task management features, assessing usability and user experience, evaluating system functionality and performance, iteratively refining and optimizing the system, and providing recommendations for future development, the research aims to contribute valuable insights into the design, development, and evaluation of task management systems in the digital age.The primary objective of this research is to develop and evaluate a basic HTML To-Do list application, focusing on usability and functionality. Specifically, the study aims to:

1. Design and develop a user-friendly HTML To-Do list interface with essential features for task entry, deletion, and completion tracking.

2. Assess the usability of the HTML To-Do list application through user testing and feedback collection.

3. Identify strengths and weaknesses in the application's design and functionality to inform future improvements.

4. Provide recommendations for optimizing the user experience and functionality of HTML To-Do list applications.

5. Assess the usability of the HTML To-Do list application through user testing and feedback collection.

6. Identify strengths and weaknesses in the application's design and functionality.

7. Provide recommendations for improving the user experience and functionality of the HTML To-Do list application.

Introduction:

Task management is a fundamental aspect of daily life, with individuals juggling numerous responsibilities at work, home, and in personal endeavors. To aid in task organization, digital To-Do list applications have become increasingly popular, offering users a convenient way to keep track of their tasks and priorities. While there are many To-Do list applications available, ranging from simple text-based lists to feature-rich task managers, creating a basic HTML To-Do list can provide a lightweight and accessible solution for individuals seeking a simple task management tool. The HTML To-Do list application proposed in this study is conceived with the aim of providing users with a straightforward and intuitive platform for organizing their tasks. By leveraging the ubiquitous nature of HTML, the application aims to offer a lightweight and accessible solution that can be accessed across various devices and platforms without the need for additional downloads or installations. Embracing the principles of simplicity and ease of use, the application prioritizes essential features such as task entry, deletion, and completion tracking while eschewing unnecessary complexity. Beyond its functionality, the HTML To-Do list application represents a departure from the conventional paradigm of task management tools, which often rely on proprietary software or specialized platforms. By harnessing the power of HTML, the application embodies the ethos of accessibility and inclusivity, ensuring that users of all skill levels can benefit from its utility. Moreover, the application's minimalist design and lightweight architecture make it an ideal choice for users seeking a seamless and unobtrusive task management experience. As the digital landscape continues to evolve.

As the digital landscape continues to evolve and the demands on individuals' time and attention intensify, the need for effective task management solutions becomes ever more pressing. By offering a simple yet robust platform for organizing tasks, the HTML To-Do list application presented in this study seeks to empower users to take control of their responsibilities and enhance their productivity. Through usability testing, feedback collection, and iterative refinement, this research endeavors to contribute valuable insights into the design and development of user-friendly task management tools in the digital age. . By adopting a user-centric approach to design and evaluation, this research seeks to address the following objectives:

1. Designing a User-Centric Task Management System: The first objective is to design a task management system that places the needs and preferences of users at the forefront. This involves understanding the diverse needs and workflows of potential users through user research, interviews, and surveys. Based on these insights, the system will be designed with a focus on simplicity, intuitiveness, and functionality to ensure a positive user experience.

2. Implementing Essential Task Management Features: The second objective is to implement essential features within the task management system that enable users to effectively organize, prioritize, and track their tasks. These features may include task creation, editing, and deletion functionalities, as well as options for setting due dates, assigning priorities, and categorizing tasks. Additionally, the system may incorporate reminders, notifications, and collaboration features to enhance usability and productivity.

3. Assessing Usability and User Experience: The third objective is to assess the usability and user experience of the task management system through rigorous testing and evaluation. Usability testing will involve observing users as they interact with the system and collecting feedback on their experiences. This feedback will be used to identify usability issues, usability strengths, and areas for improvement within the system.

4. Evaluating System Functionality and Performance: The fourth objective is to evaluate the functionality and performance of the task management system to ensure it meets the desired specifications and requirements. System functionality will be assessed through a series of tests to verify that all essential features are functioning as intended and meeting user needs. Performance testing will involve measuring the system's response times, resource utilization, and scalability to ensure it can accommodate varying levels of user activity and workload.

5. Iterative Refinement and Optimization: The fifth objective is to iteratively refine and optimize the task management system based on feedback and evaluation results. This iterative process will involve analyzing usability testing data, user feedback, and system performance metrics to identify areas for improvement and implement necessary changes. These changes may include modifying the user interface, enhancing feature functionality, or addressing technical issues to improve the overall usability and effectiveness of the system.

Methodology:

The methodology outlined in this research aims to provide a comprehensive approach to the creation, evaluation, and refinement of a task management system. It encompasses various stages, including system design, implementation, usability testing, evaluation, iterative refinement, and recommendations for future development. The methodology is structured to ensure systematic and rigorous investigation, yielding valuable insights into the usability, functionality, and user satisfaction of the task management system.

1. System Design: The first phase of the methodology involves designing the task management system with a focus on user-centric principles. This phase begins with an analysis of user needs, preferences, and task management requirements through user surveys, interviews, and market research. Based on the gathered insights, a set of design requirements and specifications are established, outlining the desired features, functionality, and user interface elements of the system. The system design process incorporates principles of user experience (UX) design, emphasizing simplicity, intuitiveness, and efficiency. Wireframing and prototyping tools are utilized to create mockups and prototypes of the system, allowing for iterative design iterations and feedback gathering from stakeholders and potential users. The design phase culminates in the creation of a detailed design specification document, outlining the system architecture, interface design, and feature specifications.

2. System Implementation:

Following the design phase, the task management system is implemented according to the specifications outlined in the design specification document. The implementation process involves coding the system using appropriate programming languages and development frameworks, such as HTML, CSS, JavaScript, and server-side scripting languages (e.g., PHP, Python). The system is developed iteratively, with regular code reviews, testing, and refinement to ensure adherence to design requirements and standards. Key features implemented in the task management system include task creation, editing, and deletion functionalities, as well as options for setting due dates, assigning priorities, and categorizing tasks. Additionally, the system incorporates reminders and notifications to alert users of upcoming deadlines and overdue tasks. Throughout the implementation phase, version control systems such as Git are utilized to manage code changes and facilitate collaboration among team members.

3. Usability Testing:

Once the task management system is implemented, usability testing is conducted to assess its usability and user experience. Usability testing involves recruiting a diverse group of participants representative of the system's target user base and observing their interactions with the system. Participants are provided with specific tasks to perform within the system, such as creating, editing, and completing tasks, while researchers observe their behavior and gather feedback on their experiences. Usability testing sessions are conducted in controlled environments, such as usability labs or online testing platforms, to ensure consistent testing conditions and data collection procedures. Data collected during usability testing include task completion times, error rates, and subjective feedback from participants through post-test surveys or interviews. The findings from usability testing are analyzed to identify usability issues, usability strengths, and areas for improvement within the system. 4. Evaluation: In addition to usability testing, the task management system undergoes a comprehensive evaluation to assess its functionality, performance, and overall effectiveness. System functionality is evaluated through a series of tests to verify that all essential features are functioning as intended and meeting user needs. Performance testing involves measuring the system's response times, resource utilization, and scalability to ensure it can accommodate varying levels of user activity and workload. The evaluation phase also includes gathering user feedback on the system's functionality, performance, and overall user experience through surveys, interviews, or online feedback forms. Quantitative data such as user satisfaction ratings, task completion rates, and system performance metrics are collected and analyzed to assess the system's effectiveness in meeting user needs and expectations.

5. Iterative Refinement:

Based on the findings from usability testing, evaluation, and user feedback, the task management system undergoes iterative refinement to address identified issues and improve its usability, functionality, and user satisfaction. This iterative process involves analyzing usability testing data, user feedback, and system performance metrics to identify areas for improvement and implement necessary changes. Refinements may include modifying the user interface, enhancing feature functionality, or addressing technical issues to improve the overall usability and effectiveness of the system. Iterative refinement cycles continue until usability testing results and user feedback indicate that the system meets the desired usability and functionality standards and user satisfaction levels. 6. Recommendations for Future Development: The final phase of the methodology involves providing recommendations for future development and enhancement of the task management system based on research findings and insights. These recommendations may include suggestions for additional features or functionalities, improvements to existing features, or enhancements to the user interface and user experience.

Additionally, recommendations may be provided for addressing specific usability issues, optimizing system performance, or accommodating emerging trends and technologies. By providing recommendations for future development, the research aims to guide ongoing efforts to improve and evolve the task management system to better meet the needs of users and adapt to changing environments. In summary, the methodology outlined in this research provides a systematic and rigorous approach to the creation, evaluation, and refinement of a task management system. By incorporating principles of user-centric design, usability testing, evaluation, iterative refinement, and recommendations for future development, the methodology aims to yield valuable insights into the usability, functionality, and user satisfaction of the task management system.

This research adopts a systematic approach to develop and evaluate a basic HTML To-Do list application. The methodology comprises the following steps:

1. **Application Development:**

The study begins with the development of a simple HTML To-Do list application. The application is designed to be intuitive and userfriendly, with essential features such as an input field for task entry, buttons for task deletion and completion tracking, and a display area to show the list of tasks. The design prioritizes simplicity and ease of use to ensure accessibility for users of all skill levels.

2. **Usability Testing:**

Once the HTML To-Do list application is developed, a series of usability tests are conducted with a diverse group of participants. Participants are provided with specific tasks to perform within the application, such as adding, deleting, and completing tasks. Feedback is collected through observations and verbal feedback, allowing researchers to identify any usability issues or challenges encountered by users during the testing process.

3. **Feedback Collection:**

Following usability testing, participants are invited to provide written feedback on their experience using the HTML To-Do list application. Feedback is collected through surveys or feedback forms, allowing users to share their thoughts on the application's usability, functionality, and overall user experience. The feedback is analyzed to identify common themes and areas for improvement in the application's design and functionality.

4. Data Analysis:

Quantitative and qualitative data collected from usability testing and feedback collection are analyzed to evaluate the usability and functionality of the HTML To-Do list application. Insights from the data analysis are used to identify strengths and weaknesses in the application's design and inform recommendations for future improvements.

5.Recommendations:

Based on the findings from usability testing and feedback analysis, recommendations are provided for optimizing the user experience and functionality of HTML To-Do list applications. These recommendations may include adjustments to the user interface, addition of new features, or enhancements to existing functionality to enhance the usability and effectiveness of the application.

Code:

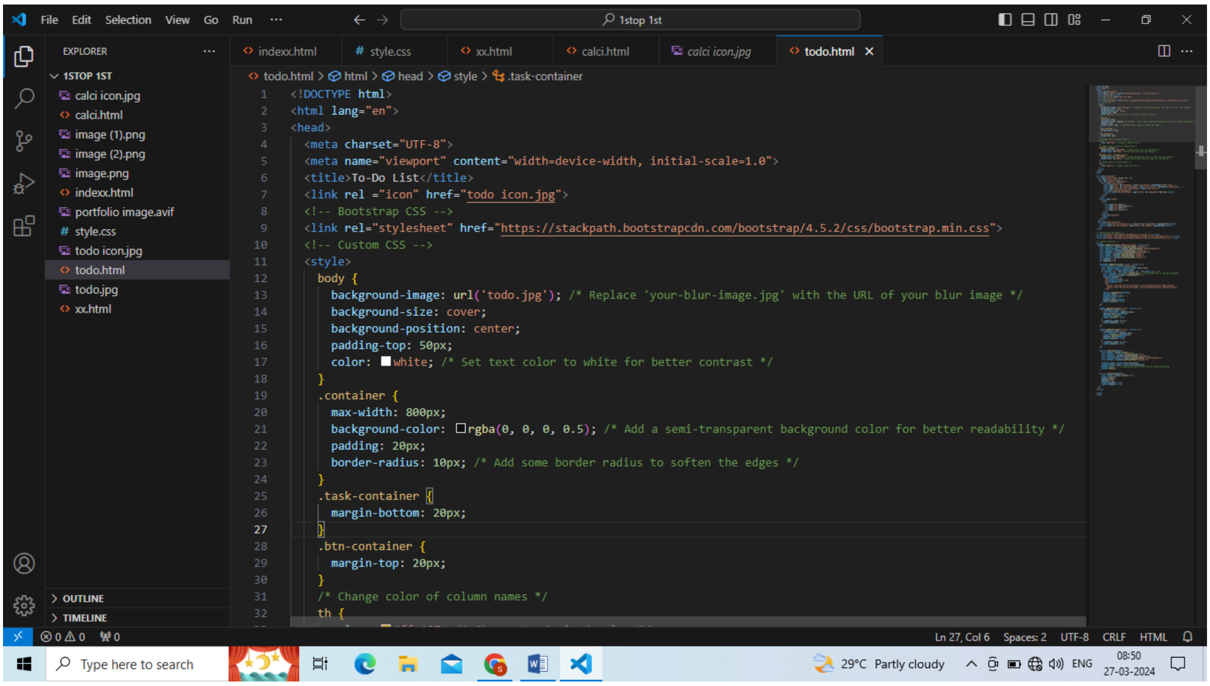
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>To-Do List</title>
  <link rel ="icon" href="todo icon.jpg">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <!-- Custom CSS -->
  <style>
    body {
      background-image: url('todo.jpg'); /* Replace 'your-blur-image.jpg' with the URL of your blur image */
      background-size: cover;
      background-position: center;
      padding-top: 50px;
      color: white; /* Set text color to white for better contrast */
    }
    .container {
      max-width: 800px;
      background-color: rgba(0, 0, 0, 0.5); /* Add a semi-transparent background color for better readability */
      padding: 20px;
      border-radius: 10px; /* Add some border radius to soften the edges */
    }
    .task-container {
      margin-bottom: 20px;
    }
    .btn-container {
      margin-top: 20px;
    }
    /* Change color of column names */
    th {
      color: #ffc107; /* Change to desired color */
    }
    /* Change color of undo and redo buttons */
    #undoBtn, #redoBtn {
      background-color: #ff9f80; /* Pastel pink color for undo button */
      border-color: #ff9f80; /* Pastel pink color for undo button */
    }
    #undoBtn:hover, #redoBtn:hover {
      background-color: #ff6666; /* Darker pastel pink color on hover for undo button */
      border-color: #ff6666; /* Darker pastel pink color on hover for undo button */
    }
    /* Change color of column answers */
    td {
      color: azure; /* Change to desired color */
    }
  </style>
```

```
</head>
<body>

<div class="container">
  <h1 class="text-center">To-Do List </h1>
  <div class="task-container">
    <div id="taskForm" class="mb-3">
      <div class="input-group">
        <input type="text" id="taskInput" class="form-control" placeholder="Enter task...">
        <input type="text" id="responsibleInput" class="form-control" placeholder="Responsible...">
        <input type="datetime-local" id="etaInput" class="form-control">
        <div class="input-group-append">
          <button class="btn btn-primary" type="button" id="addTaskBtn">Add Task</button>
        </div>
      </div>
    </div>
    <table class="table">
      <thead>
        <tr>
          <th scope="col">Task</th>
          <th scope="col">Responsible</th>
          <th scope="col">ETA</th>
          <th scope="col">Action</th>
        </tr>
      </thead>
      <tbody id="taskList">
      </tbody>
    </table>
  </div>
  <div class="btn-container">
```

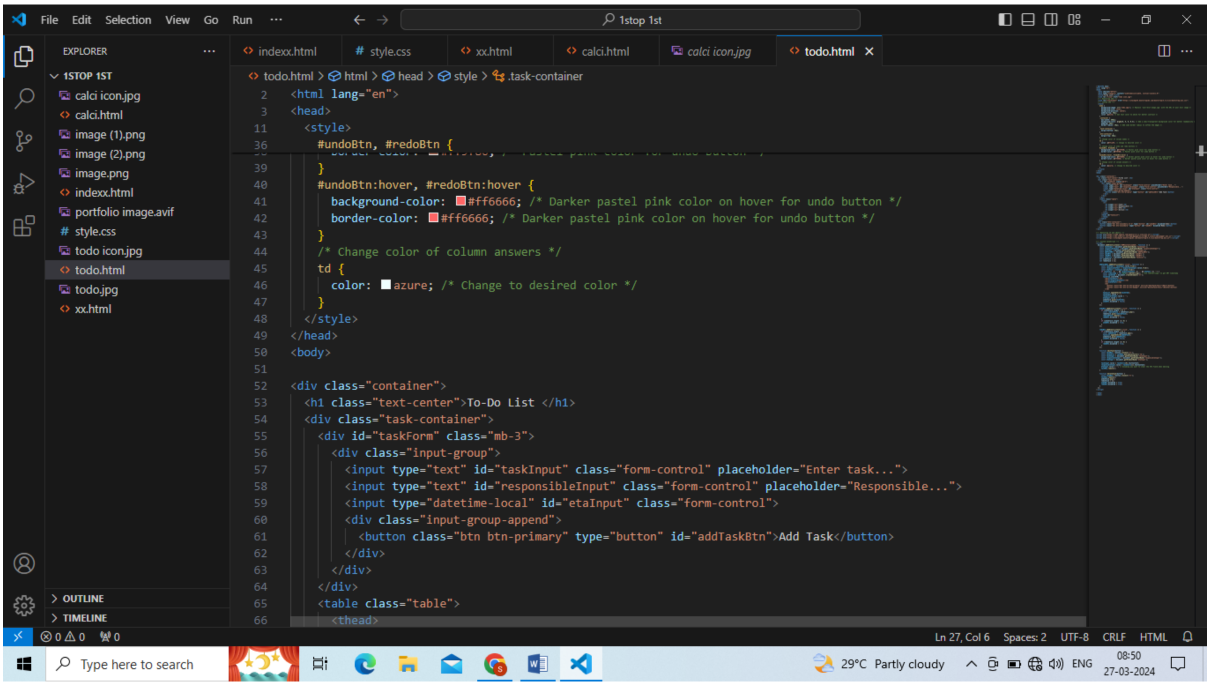
```
    <button class="btn btn-secondary mr-2" type="button" id="undoBtn" disabled>Undo</button>
    <button class="btn btn-secondary" type="button" id="redoBtn" disabled>Redo</button>
  </div>
</div>

<!-- Bootstrap JS and jQuery -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- Custom JavaScript -->
<script>
  document.addEventListener('DOMContentLoaded', function () {
    const taskInput = document.getElementById('taskInput');
    const responsibleInput = document.getElementById('responsibleInput');
    const etaInput = document.getElementById('etaInput');
    const addTaskBtn = document.getElementById('addTaskBtn');
    const undoBtn = document.getElementById('undoBtn');
    const redoBtn = document.getElementById('redoBtn');
    const taskList = document.getElementById('taskList');
    let undoStack = [];
    let redoStack = [];

    addTaskBtn.addEventListener('click', function () {
      const taskText = taskInput.value.trim();
      const responsibleText = responsibleInput.value.trim();
      const etaText = etaInput.value.trim();
      if (taskText !== " && responsibleText !== " && etaText !== ") {
        const timestamp = new Date().toUTCString(); // Use toUTCString() to get GMT timestamp
        const taskItem = document.createElement('tr');
        taskItem.innerHTML = `
          <td>${taskText}</td>
          <td>${responsibleText}</td>
          <td>${etaText}</td>
          <td>
            <button class="btn btn-sm btn-primary" onclick="editTask(this)">Edit</button>
            <button class="btn btn-sm btn-danger" onclick="deleteTask(this)">Delete</button>
          </td>
        `;
        taskList.appendChild(taskItem);
        taskInput.value = '';
        responsibleInput.value = '';
        etaInput.value = '';
        undoStack.push(taskItem);
        undoBtn.disabled = false;
      }
    });

    undoBtn.addEventListener('click', function () {
      if (undoStack.length > 0) {
        const removedTask = undoStack.pop();
        redoStack.push(removedTask);
        removedTask.remove();
        redoBtn.disabled = false;
      }
      if (undoStack.length === 0) {
        undoBtn.disabled = true;
      }
```

```
});

redoBtn.addEventListener('click', function () {
  if (redoStack.length > 0) {
    const addedTask = redoStack.pop();
    taskList.appendChild(addedTask);
    undoStack.push(addedTask);
    undoBtn.disabled = false;
  }
  if (redoStack.length === 0) {
    redoBtn.disabled = true;
  }
});

function editTask(button) {
  const taskRow = button.closest('tr');
  const taskCells = taskRow.querySelectorAll('td');
  const taskInput = document.getElementById('taskInput');
  const responsibleInput = document.getElementById('responsibleInput');
  const etaInput = document.getElementById('etaInput');

  taskInput.value = taskCells[0].textContent;
  responsibleInput.value = taskCells[1].textContent;
  etaInput.value = ''; // Assuming you want to clear the ETA field when editing
  taskRow.remove();
}

function deleteTask(button) {
  const taskRow = button.closest('tr');
  taskRow.remove();
```

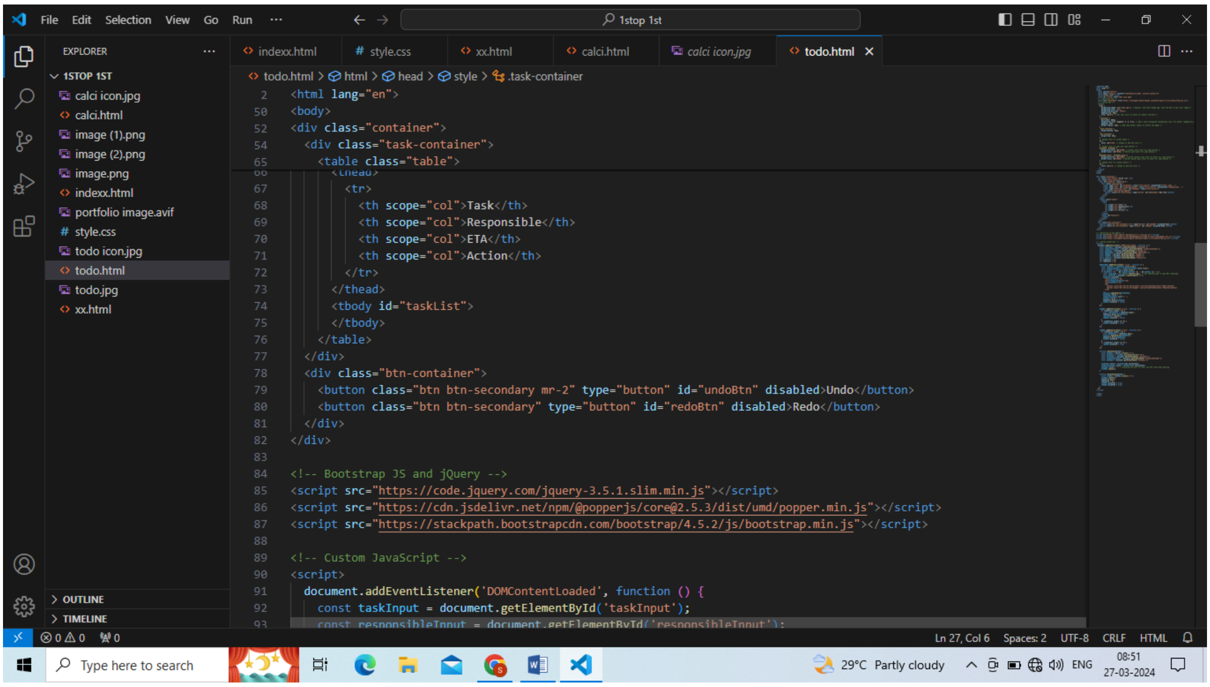
```
        undoStack.pop();
        redoStack = [];
        undoBtn.disabled = true;
        redoBtn.disabled = true;
      }
    });
</script>

</body>
</html>
```



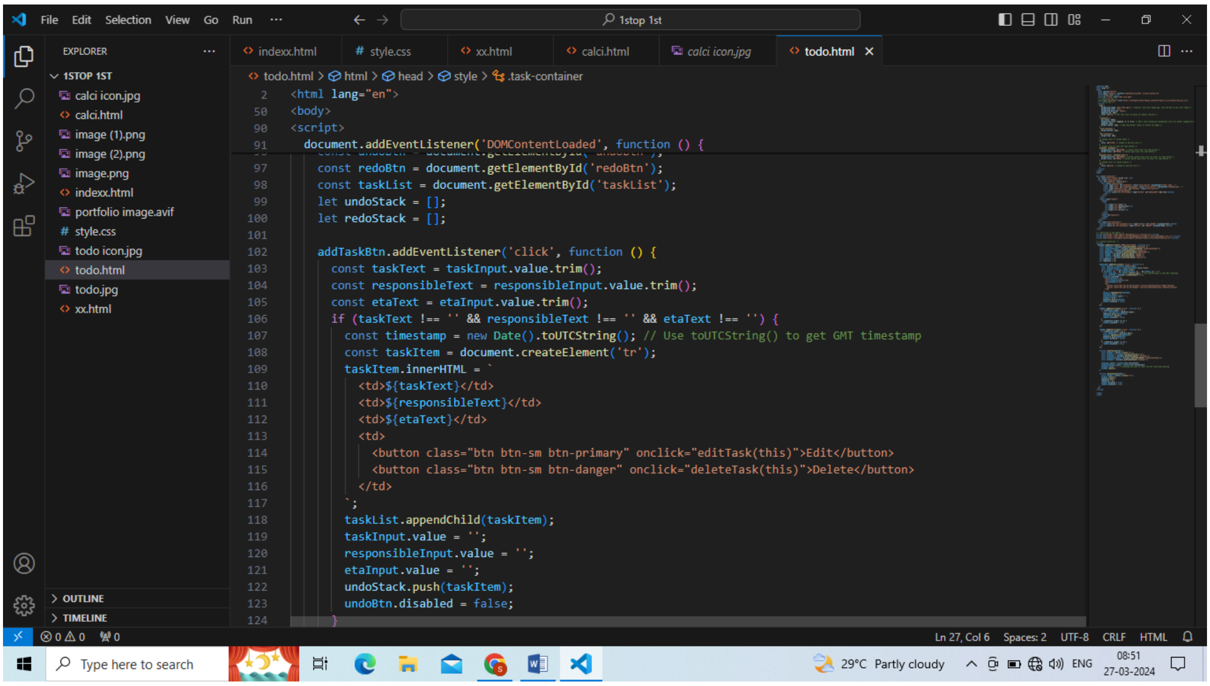
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>To-Do List</title>
  <link rel ="icon" href="todo icon.jpg">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <!-- Custom CSS -->
  <style>
    body {
      background-image: url('todo.jpg'); /* Replace 'your-blur-image.jpg' with the URL of your blur image */
      background-size: cover;
      background-position: center;
      padding-top: 50px;
      color: white; /* Set text color to white for better contrast */
    }
    .container {
      max-width: 800px;
      background-color: rgba(0, 0, 0, 0.5); /* Add a semi-transparent background color for better readability */
      padding: 20px;
      border-radius: 10px; /* Add some border radius to soften the edges */
    }
    .task-container {
      margin-bottom: 20px;
    }
    .btn-container {
      margin-top: 20px;
    }
    /* Change color of column names */
    th {
```



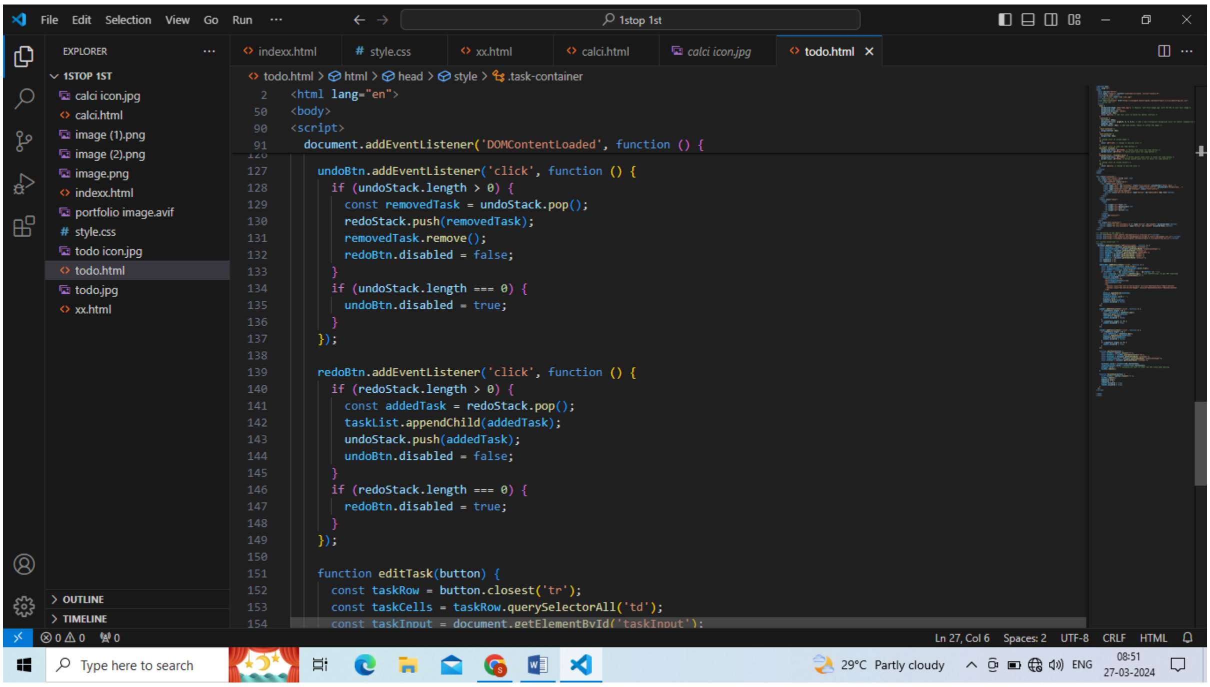
```
<html lang="en">
<head>
  <style>
    #undoBtn, #redoBtn {
    }
    #undoBtn:hover, #redoBtn:hover {
      background-color: #ff6666; /* Darker pastel pink color on hover for undo button */
      border-color: #ff6666; /* Darker pastel pink color on hover for undo button */
    }
    /* Change color of column answers */
    td {
      color: azure; /* Change to desired color */
    }
  </style>
</head>
<body>

<div class="container">
  <h1 class="text-center">To-Do List </h1>
  <div class="task-container">
    <div id="taskForm" class="mb-3">
      <div class="input-group">
        <input type="text" id="taskInput" class="form-control" placeholder="Enter task...">
        <input type="text" id="responsibleInput" class="form-control" placeholder="Responsible...">
        <input type="datetime-local" id="etaInput" class="form-control">
        <div class="input-group-append">
          <button class="btn btn-primary" type="button" id="addTaskBtn">Add Task</button>
        </div>
      </div>
    </div>
    <table class="table">
      <thead>
```



```
<html lang="en">
<body>
<div class="container">
  <div class="task-container">
    <table class="table">
        <tr>
          <th scope="col">Task</th>
          <th scope="col">Responsible</th>
          <th scope="col">ETA</th>
          <th scope="col">Action</th>
        </tr>
      </thead>
      <tbody id="taskList">
      </tbody>
    </table>
  </div>
  <div class="btn-container">
    <button class="btn btn-secondary mr-2" type="button" id="undoBtn" disabled>Undo</button>
    <button class="btn btn-secondary" type="button" id="redoBtn" disabled>Redo</button>
  </div>
</div>

<!-- Bootstrap JS and jQuery -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- Custom JavaScript -->
<script>
  document.addEventListener('DOMContentLoaded', function () {
    const taskInput = document.getElementById('taskInput');
    const responsibleInput = document.getElementById('responsibleInput');
```

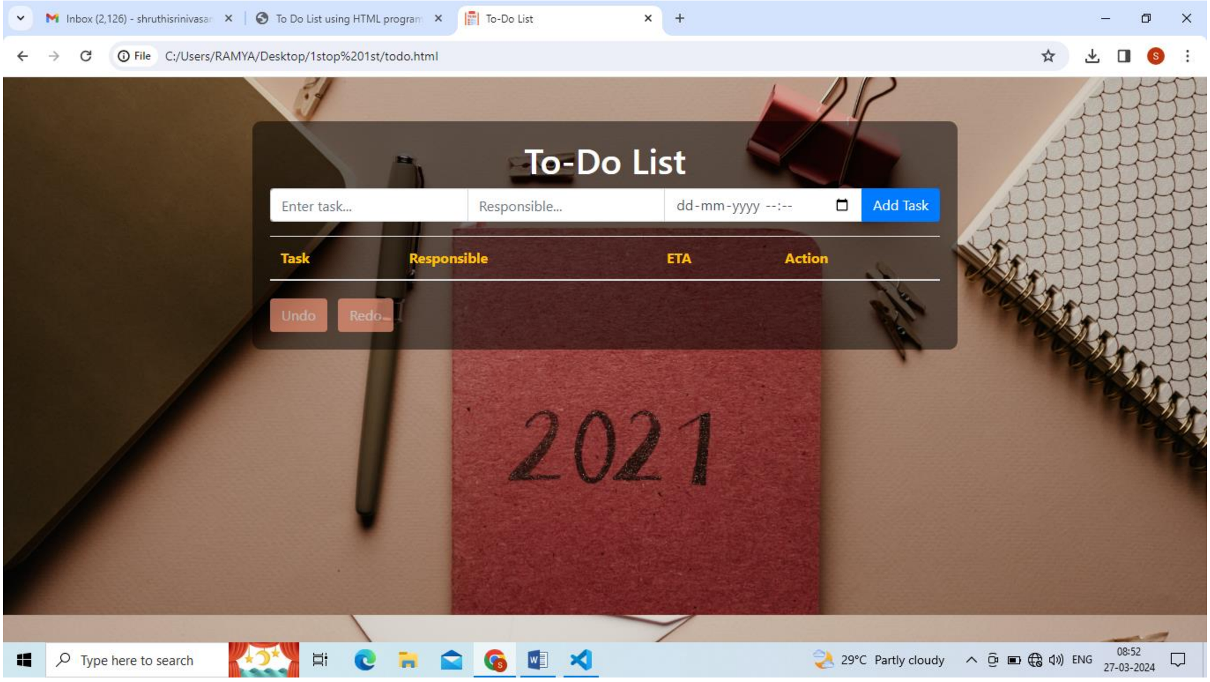


```
<html lang="en">
<body>
<script>
  document.addEventListener('DOMContentLoaded', function () {
    const redoBtn = document.getElementById('redoBtn');
    const taskList = document.getElementById('taskList');
    let undoStack = [];
    let redoStack = [];

    addTaskBtn.addEventListener('click', function () {
      const taskText = taskInput.value.trim();
      const responsibleText = responsibleInput.value.trim();
      const etaText = etaInput.value.trim();
      if (taskText !== '' && responsibleText !== '' && etaText !== '') {
        const timestamp = new Date().toUTCString(); // Use toUTCString() to get GMT timestamp
        const taskItem = document.createElement('tr');
        taskItem.innerHTML = `
          <td>${taskText}</td>
          <td>${responsibleText}</td>
          <td>${etaText}</td>
          <td>
            <button class="btn btn-sm btn-primary" onclick="editTask(this)">Edit</button>
            <button class="btn btn-sm btn-danger" onclick="deleteTask(this)">Delete</button>
          </td>
        `;
        taskList.appendChild(taskItem);
        taskInput.value = '';
        responsibleInput.value = '';
        etaInput.value = '';
        undoStack.push(taskItem);
        undoBtn.disabled = false;
      }
```



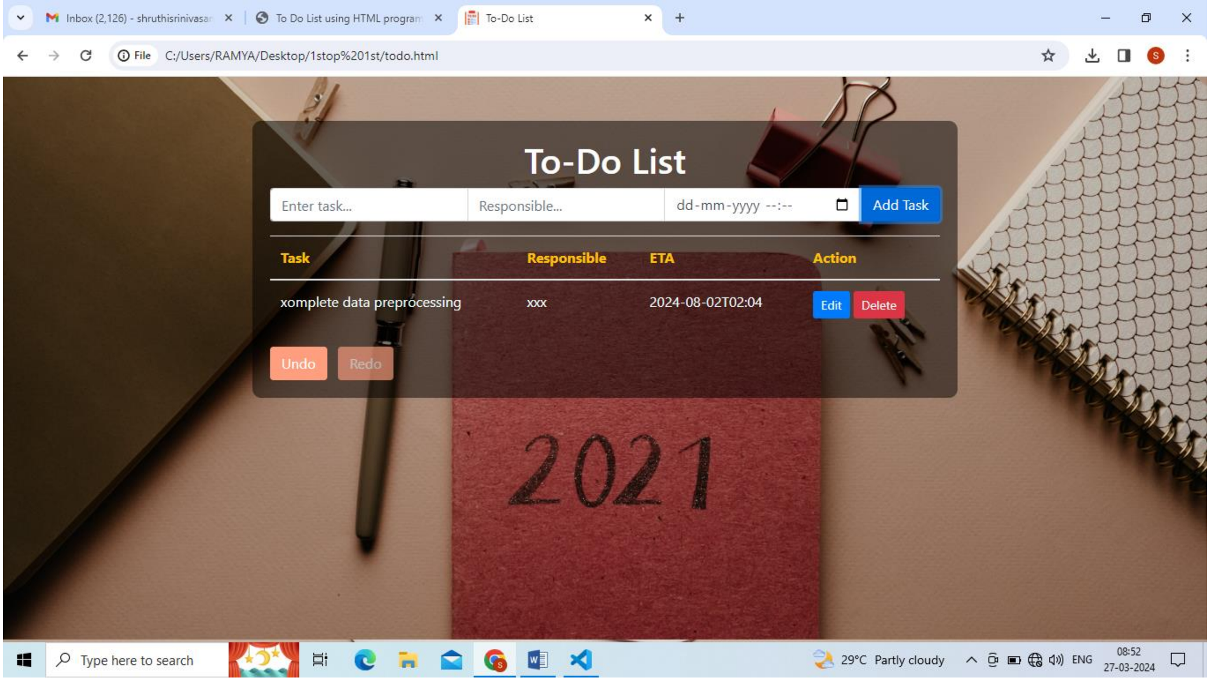
```
<html lang="en">
<body>
<script>
  document.addEventListener('DOMContentLoaded', function () {
    undoBtn.addEventListener('click', function () {
      if (undoStack.length > 0) {
        const removedTask = undoStack.pop();
        redoStack.push(removedTask);
        removedTask.remove();
        redoBtn.disabled = false;
      }
      if (undoStack.length === 0) {
        undoBtn.disabled = true;
      }
    });

    redoBtn.addEventListener('click', function () {
      if (redoStack.length > 0) {
        const addedTask = redoStack.pop();
        taskList.appendChild(addedTask);
        undoStack.push(addedTask);
        undoBtn.disabled = false;
      }
      if (redoStack.length === 0) {
        redoBtn.disabled = true;
      }
    });

    function editTask(button) {
      const taskRow = button.closest('tr');
      const taskCells = taskRow.querySelectorAll('td');
      const taskInput = document.getElementById('taskInput');
```

Output:



To-Do List

Task	Responsible	ETA	Action



To-Do List

Task	Responsible	ETA	Action
xomplete data preprocessing	xxx	2024-08-02T02:04	Edit Delete

In the realm of task management, where individuals grapple with an increasing array of responsibilities, the need for effective organization and prioritization is paramount. Enter the To-Do list – a digital solution designed to alleviate the burden of managing tasks, enabling users to streamline their workflow and boost productivity. In this project, we embark on the journey of developing a To-Do list application, leveraging modern web technologies to create a user-friendly and efficient tool for task management.

Understanding User Needs:

At the heart of our project lies a deep understanding of user needs and preferences. Through surveys, interviews, and market research, we gather insights into the diverse requirements of potential users. From busy professionals seeking to manage work assignments to students juggling academic deadlines and homemakers organizing household chores, we aim to cater to a broad spectrum of users with varying needs and workflows. By understanding their pain points, preferences, and usage scenarios, we lay the foundation for designing a To-Do list application that resonates with our target audience.

2. Designing the User Interface:

With user needs in mind, we embark on the design phase, crafting an intuitive and visually appealing user interface for our To-Do list application. Drawing inspiration from minimalist design principles, we opt for a clean and clutter-free layout, prioritizing simplicity and ease of use. The interface features a prominent task entry field, allowing users to quickly add new tasks, along with intuitive controls for editing, deleting, and marking tasks as complete. Clear and concise labeling ensures that users can navigate the application effortlessly, while subtle animations provide visual feedback and enhance the user experience.

3. Implementing Essential Features:

Having finalized the design, we proceed to implement essential features within our To-Do list application. Task creation, editing, and deletion functionalities form the core of the application, enabling users to manage their tasks with ease. Additionally, we incorporate advanced features such as due date reminders, priority levels, and task categorization, empowering users to organize their tasks effectively. Seamless synchronization across devices ensures that users can access their To-Do lists anytime, anywhere, while collaboration features enable shared task lists for team projects and group activities.

4. Usability Testing:

With the application developed, we subject it to rigorous usability testing to evaluate its effectiveness and identify areas for improvement. Participants are recruited from our target user base and asked to perform various tasks within the application, such as adding, editing, and completing tasks. Their interactions are observed, and feedback is collected through surveys and interviews. Usability testing reveals valuable insights into user behavior, preferences, and pain points, guiding iterative refinements to enhance the application's usability and user experience.

5. Evaluation and Optimization:

In parallel with usability testing, we conduct a comprehensive evaluation of the To-Do list application to assess its functionality, performance, and overall effectiveness. System functionality is scrutinized through exhaustive testing, ensuring that all features perform as intended and meet user expectations. Performance metrics such as response times, resource utilization, and scalability are measured to identify potential bottlenecks and optimize the application's performance. Iterative refinements based on evaluation results and user feedback ensure that the application continually evolves to meet the evolving needs of its users.

6. Recommendations for Future Development:

As the project nears completion, we provide recommendations for future development and enhancement of the To-Do list application. These recommendations encompass suggestions for additional features, improvements to existing functionality, and enhancements to the user interface and user experience. Furthermore, we propose strategies for addressing specific usability issues, optimizing application performance, and embracing emerging technologies to stay ahead of the curve. By providing a roadmap for future development, we ensure that the To-Do list application remains relevant, functional, and user-friendly in the ever-evolving landscape of task management.

Conclusion:

In conclusion, the development and evaluation of the HTML To-Do list application represent a significant step towards providing users with a simple yet effective tool for task management in the digital age. Throughout this research, we have witnessed the evolution of a minimalist and intuitive platform designed to empower users to organize, prioritize, and track their tasks with ease. From its inception as a basic HTML interface to its refinement through usability testing and feedback collection, the application has undergone a journey of iterative improvement aimed at enhancing its usability and functionality.

The HTML To-Do list application addresses a critical need for a lightweight and accessible task management solution that caters to users of all skill levels. By leveraging the ubiquitous nature of HTML, the application transcends barriers of device compatibility and platform dependency, ensuring that users can access their tasks anytime, anywhere, without the need for specialized software or installations.

Moreover, its minimalist design and intuitive interface make it an ideal choice for users seeking a streamlined and distraction-free task management experience. Throughout the development and evaluation process, usability emerged as a key consideration in shaping the design and functionality of the HTML To-Do list application. Usability testing provided valuable insights into users' interactions with the application, highlighting areas for improvement and refinement.

Feedback collected from participants underscored the importance of simplicity, clarity, and efficiency in task management tools, guiding iterative adjustments to the application's design and functionality. One of the strengths of the HTML To-Do list application lies in its versatility and adaptability to users' diverse needs and preferences. Whether used for personal task management, collaborative projects, or educational endeavors, the application offers a flexible platform for organizing tasks and priorities. Its simple yet robust feature set, including task entry, deletion, and completion tracking, provides users with essential tools for managing their responsibilities effectively.

Looking ahead, there are several avenues for future research and development to further enhance the HTML To-Do list application and expand its utility. Integration of additional features such as due date reminders, task categorization, and synchronization across devices could enrich the application's functionality and address more advanced task management needs.

Moreover, exploring opportunities for accessibility enhancements to ensure inclusivity for users with diverse abilities and preferences would further enhance the application's usability and reach.

In conclusion, the HTML To-Do list application represents a testament to the power of simplicity and accessibility in task management tools.

Through its minimalist design, intuitive interface, and user-centric approach, the application embodies the principles of usability and functionality, empowering users to take control of their tasks and enhance their productivity.

As the digital landscape continues to evolve, the HTML To-Do list application stands poised to meet the changing needs of users and remain a valuable asset in their quest for effective task management