MAJOR PROJECT

Project Name **:** Creation of a HTML To Do List

Task **:** Create a To Do list using HTML Programming Language

CONTENTS

ABSTRACT

This project introduces a user-friendly and interactive HTML-based to-do list application designed to enhance task organization and productivity. Leveraging the power of HTML, CSS, and JavaScript, the application provides a seamless and responsive user interface for creating, updating, and deleting tasks. The Key features in this Application are Dynamic Task Creation, Intuitive User Interface,Real-time Updates, Task Modification and Deletion, Sorting and Filtering, Local Storage Integration,Cross-Browser Compatibility and Accessibility. This HTML to-do list application provides a user-friendly and efficient solution for individuals seeking a digital platform to organize and manage their tasks. Whether for personal use or collaborative projects, the application's features contribute to a streamlined and effective task management experience.

OBJECTIVE

The primary objective of an HTML to-do list is to provide a user-friendly and efficient digital platform for organizing and managing tasks. At its core, the application is designed to offer a clean and intuitive user interface through the use of HTML, CSS, and JavaScript. The aim is to create an environment where users can easily input, modify, and delete tasks, fostering a seamless experience in organizing their daily activities. The objectives that such an application aims to achieve are Task Organisation, User-Friendly Interface, Dynamic Task Creation , Real-time Updates, Task Modification and deletion. the application adheres to web accessibility standards to make it usable by individuals with disabilities. Features like keyboard navigation and alternative text for non-text content contribute to an inclusive and accessible user experience. The HTML to-do list application aims to enhance task management, productivity, and organization by combining a user-friendly interface with dynamic features, real-time updates, and accessibility considerations.

INTRODUCTION

Web development refers to the process of creating and maintaining websites and web applications for the internet or an intranet. It involves a combination of various skills, technologies, and methodologies to design, develop, and deploy functional and interactive web solutions. The web development is categorized as front-end development , back – end development and full-stack development . The HTML to-do list is a simple yet effective web-based application designed to help users organize and manage their tasks efficiently. Using basic HTML, CSS for styling, and JavaScript for interactivity, the to-do list provides a clean and intuitive user interface. Users can easily add, modify, or delete tasks, and the application dynamically updates in real-time without requiring page reloads. With features like sorting, filtering, and local storage integration, this lightweight tool offers a convenient way for users to stay on top of their daily activities, promoting a streamlined and organized approach to task management. This is done with the help of web development technologies i.e; **HTML , CSS, JAVASCRIPT , BOOTSTRAP** and **JQUERY** .

HTML (Hyper Text Markup Language) : HTML (Hypertext Markup Language) is the standard markup language used to create web pages and structure their content. It consists of a set of tags and elements that define the various components of a web page, such as headings, paragraphs, links, images, forms, and more. HTML plays a fundamental role in web development and is used in conjunction with CSS (Cascading Style Sheets) and JavaScript to create visually appealing, interactive, and responsive websites. It provides the structure and semantics needed to organize and present information on the internet, making it accessible to both humans and web browsers. HTML is one of the web development technology that is being used by all the people all over the world.

CSS (Cascading Style Sheet) : CSS (Cascading Style Sheets) is a language used for describing the presentation and styling of web documents written in HTML. It allows web developers to control the layout, appearance, and design of web pages, ensuring a consistent

and visually appealing user experience across different devices and screen sizes. CSS achieves this by defining rules that specify how HTML elements should be displayed, including attributes like colors, fonts, spacing, positioning, and more. CSS is an essential part of modern web development, enabling the separation of content (HTML) from presentation (CSS), making websites easier to maintain, style, and adapt for various platforms and devices. It also plays a crucial role in enhancing user accessibility and optimizing page loading times.

JS (Javascript) : JavaScript is a versatile and widely-used programming language primarily used for adding interactivity and dynamic behavior to web pages. It allows developers to create responsive and feature-rich web applications by manipulating the content and behavior of websites in real-time within a user's web browser. JavaScript is an essential part of front-end web development, enabling actions like form validation, animations, real-time updates, and more. It can also be used in the back-end (Node.js) to build server-side applications. JavaScript is supported by all major web browsers and has a large ecosystem of libraries and frameworks, making it a powerful tool for creating modern web applications.

BOOTSTRAP : Bootstrap is a popular open-source front-end framework for web development that simplifies the process of designing and building responsive and visually appealing websites and web applications. Developed by Twitter, Bootstrap provides a collection of pre-designed HTML, CSS, and JavaScript components, such as navigation bars, buttons, forms, modals, and more. These components can be easily customized and integrated into web projects, saving developers time and effort. The features of bootstrap include Responsive design , Customization , Accessibility , Cross browser compatibility and Community and Ecosystem. Bootstrap simplifies front-end development by providing a well-documented and consistent set of tools and components, making it a valuable resource for both beginners and experienced web developers.

METHODOLOGY

In the process of creating this single page portfolio website , many steps have been involved. We will clearly see the steps now :

1) **Setting up the Tool :** The first step involved in the creation this single page portfolio website is deciding the tool and preparing the tool for usage. I used virtual studio code shortly known as VScode.
2) **Setting up XAMPP :**
   - XAMPP is a free and open-source cross-platform web server solution stack package that consists mainly of the Apache HTTP Server, MariaDB database (MySQL replacement), and interpreters for scripts written in the PHP and Perl programming languages.
   - Installation of XAMPP from the google in the apachefriends.org website and install it.
3) **Creating required files for the website :** A file is created in the XAMPP folder that file consists of Bootstrap, CSS , JS and HTML.
4) **Creating the HTML page :** creating a html page in the VScode.
5) **Associating the CSS and JS :**
   - Downloading the Bootstrap CSS and JS files from the Bootstrap official website and add file with the html page .
   - Linking the CSS file and Bootstrap icons link which is copied from the Bootstrap website in the head section of the html page.
   - Downloading fontawesome files from the fontawesome website and Linking the css file from the fontawesome to the HTML page.
   - Then, linking the JS file and JQuery link which is copied from the JQuery website in the body section with the script tag of the html page.
   - Copy the popper.js links from the bootstrap website and paste them in the html page.
6) **Designing the webpage :** writing the code using HTML , CSS and JS.
7) **Execution of the code :** Executing the written code in the html page and checking whether we got the required result or not.

CODE

Index.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML To Do List</title>
  <link href="bootstrap/css/bootstrap.css" rel="stylesheet"> <!--CSS-->
  <link href="style.css" rel="stylesheet">
  <link href="fontawesome/css/all.css" rel="stylesheet"> <!--fontawesome-->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.1/font/bootstrap-icons.css">
  <!--bootstrap icons-->
</head>

<body>
  <nav>
    <div class="container-fluid">
      <a href="#" class="navbar-brand">
        <img src="images/logo.jpg" class="img-fluid" alt="logo" width="100"
style="padding-top: 10px;">
      </a>
  </nav>
  <div class="wrapper">
    <div class="task-input">
      <img src="images/bars-icon.png" alt="icon">
      <input type="text" placeholder="Add a new task">
    </div>
```

```
    <div class="controls">
      <div class="filters">
        <span class="active" id="all">All</span>
        <span id="pending">Pending</span>
        <span id="completed">Completed</span>
      </div>
      <button class="clear-btn">Clear All</button>
    </div>
    <ul class="task-box"></ul>
  </div>


  <script src="JS/main.js" type="text/javascript"> </script>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script><!--
jquery-->
  <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
    integrity="sha384-
I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r"
    crossorigin="anonymous"></script> <!--popper.js-->
  <script src="bootstrap/js/bootstrap.js"></script><!--javascript-->
  <script>
    const taskInput = document.querySelector(".task-input input"),
      filters = document.querySelectorAll(".filters span"),
      clearAll = document.querySelector(".clear-btn"),
      taskBox = document.querySelector(".task-box");

    let editId,
      isEditTask = false,
      todos = JSON.parse(localStorage.getItem("todo-list"));

    filters.forEach((btn) => {
```

```
    btn.addEventListener("click", () => {
      document.querySelector("span.active").classList.remove("active");
      btn.classList.add("active");
      showTodo(btn.id);
    });
  });

  function showTodo(filter) {
    let liTag = "";
    if (todos) {
      todos.forEach((todo, id) => {
        let completed = todo.status == "completed" ? "checked" : "";
        if (filter == todo.status || filter == "all") {
          liTag += `<li class="task">
                      <label for="${id}">
                        <input onclick="updateStatus(this)" type="checkbox" id="${id}"
${completed}>
                        <p class="${completed}">${todo.name}</p>
                      </label>
                      <div class="settings">
                        <i onclick="showMenu(this)" class="uil uil-ellipsis-h"></i>
                        <ul class="task-menu">
                          <li onclick='editTask(${id}, "${todo.name}")'><i class="uil uil-
pen"></i>Edit</li>
                          <li onclick='deleteTask(${id}, "${filter}")'><i class="uil uil-
trash"></i>Delete</li>
                        </ul>
                      </div>
                    </li>`;
        }
      });
```

```
  }
  taskBox.innerHTML = liTag || `<span>You don't have any task here</span>`;
  let checkTask = taskBox.querySelectorAll(".task");
  !checkTask.length
    ? clearAll.classList.remove("active")
    : clearAll.classList.add("active");
  taskBox.offsetHeight >= 300
    ? taskBox.classList.add("overflow")
    : taskBox.classList.remove("overflow");
}
showTodo("all");

function showMenu(selectedTask) {
  let menuDiv = selectedTask.parentElement.lastElementChild;
  menuDiv.classList.add("show");
  document.addEventListener("click", (e) => {
    if (e.target.tagName != "I" || e.target != selectedTask) {
      menuDiv.classList.remove("show");
    }
  });
}

function updateStatus(selectedTask) {
  let taskName = selectedTask.parentElement.lastElementChild;
  if (selectedTask.checked) {
    taskName.classList.add("checked");
    todos[selectedTask.id].status = "completed";
  } else {
    taskName.classList.remove("checked");
    todos[selectedTask.id].status = "pending";
```

```
  }
  localStorage.setItem("todo-list", JSON.stringify(todos));
}

function editTask(taskId, textName) {
  editId = taskId;
  isEditTask = true;
  taskInput.value = textName;
  taskInput.focus();
  taskInput.classList.add("active");
}

function deleteTask(deleteId, filter) {
  isEditTask = false;
  todos.splice(deleteId, 1);
  localStorage.setItem("todo-list", JSON.stringify(todos));
  showTodo(filter);
}

clearAll.addEventListener("click", () => {
  isEditTask = false;
  todos.splice(0, todos.length);
  localStorage.setItem("todo-list", JSON.stringify(todos));
  showTodo();
});

taskInput.addEventListener("keyup", (e) => {
  let userTask = taskInput.value.trim();
  if (e.key == "Enter" && userTask) {
    if (!isEditTask) {
```

```
        todos = !todos ? [] : todos;
        let taskInfo = { name: userTask, status: "pending" };
        todos.push(taskInfo);
       } else {
        isEditTask = false;
        todos[editId].name = userTask;
       }
       taskInput.value = "";
       localStorage.setItem("todo-list", JSON.stringify(todos));
       showTodo(document.querySelector("span.active").id);
      }
     });


  </script>
</body>

</html>
```

Style.css :

```
/* Import Google Font - Poppins */
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap');

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}

body {
  width: 100%;
  height: 100vh;
  overflow: hidden;
  background-color: teal;
}

::selection {
  color: #fff;
  background: #3C87FF;
```

```
}

.wrapper {
  max-width: 600px;
  padding: 28px 0 30px;
  margin: 137px auto;
  background: #fff;
  border-radius: 12px;
  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);
}

.task-input {
  height: 100px;
  padding: 0 25px;
  position: relative;
}

.task-input img {
  top: 50%;
  position: absolute;
  height: 25px;
  transform: translate(17px, -50%);
}
```

```
.task-input input {
  height: 100%;
  width: 100%;
  outline: none;
  font-size: 18px;
  border-radius: 5px;
  padding: 0 20px 0 53px;
  border: 1px solid #999;
}

.task-input input:focus,
.task-input input.active {
  padding-left: 52px;
  border: 2px solid #3C87FF;
}

.task-input input::placeholder {
  color: #bfbfbf;
}

.controls,
li {
  display: flex;
  align-items: center;
```

```
  justify-content: space-between;
}

.controls {
  padding: 18px 25px;
  border-bottom: 1px solid #ccc;
}

.filters span {
  margin: 0 8px;
  font-size: 17px;
  color: #444444;
  cursor: pointer;
}

.filters span:first-child {
  margin-left: 0;
}

.filters span.active {
  color: #3C87FF;
}

.controls .clear-btn {
```

```
  border: none;
  opacity: 0.6;
  outline: none;
  color: #fff;
  cursor: pointer;
  font-size: 13px;
  padding: 7px 13px;
  border-radius: 4px;
  letter-spacing: 0.3px;
  pointer-events: none;
  transition: transform 0.25s ease;
  background: linear-gradient(135deg, #1798fb 0%, #2D5CFE 100%);
}

.clear-btn.active {
  opacity: 0.9;
  pointer-events: auto;
}

.clear-btn:active {
  transform: scale(0.93);
}

.task-box {
```

```
  margin-top: 20px;
  margin-right: 5px;
  padding: 0 20px 10px 25px;
}

.task-box.overflow {
  overflow-y: auto;
  max-height: 300px;
}

.task-box::-webkit-scrollbar {
  width: 5px;
}

.task-box::-webkit-scrollbar-track {
  background: #f1f1f1;
  border-radius: 25px;
}

.task-box::-webkit-scrollbar-thumb {
  background: #e6e6e6;
  border-radius: 25px;
}
```

```
.task-box .task {
  list-style: none;
  font-size: 17px;
  margin-bottom: 18px;
  padding-bottom: 16px;
  align-items: flex-start;
  border-bottom: 1px solid #ccc;
}

.task-box .task:last-child {
  margin-bottom: 0;
  border-bottom: 0;
  padding-bottom: 0;
}

.task-box .task label {
  display: flex;
  align-items: flex-start;
}

.task label input {
  margin-top: 7px;
  accent-color: #3C87FF;
}
```

```
.task label p {
  user-select: none;
  margin-left: 12px;
  word-wrap: break-word;
}

.task label p.checked {
  text-decoration: line-through;
}

.task-box .settings {
  position: relative;
}

.settings :where(i, li) {
  cursor: pointer;
}

.settings .task-menu {
  z-index: 10;
  right: -5px;
  bottom: -65px;
  padding: 5px 0;
```

```
  background: #fff;
  position: absolute;
  border-radius: 4px;
  transform: scale(0);
  transform-origin: top right;
  box-shadow: 0 0 6px rgba(0, 0, 0, 0.15);
  transition: transform 0.2s ease;
}

.task-box .task:last-child .task-menu {
  bottom: 0;
  transform-origin: bottom right;
}

.task-box .task:first-child .task-menu {
  bottom: -65px;
  transform-origin: top right;
}

.task-menu.show {
  transform: scale(1);
}

.task-menu li {
```

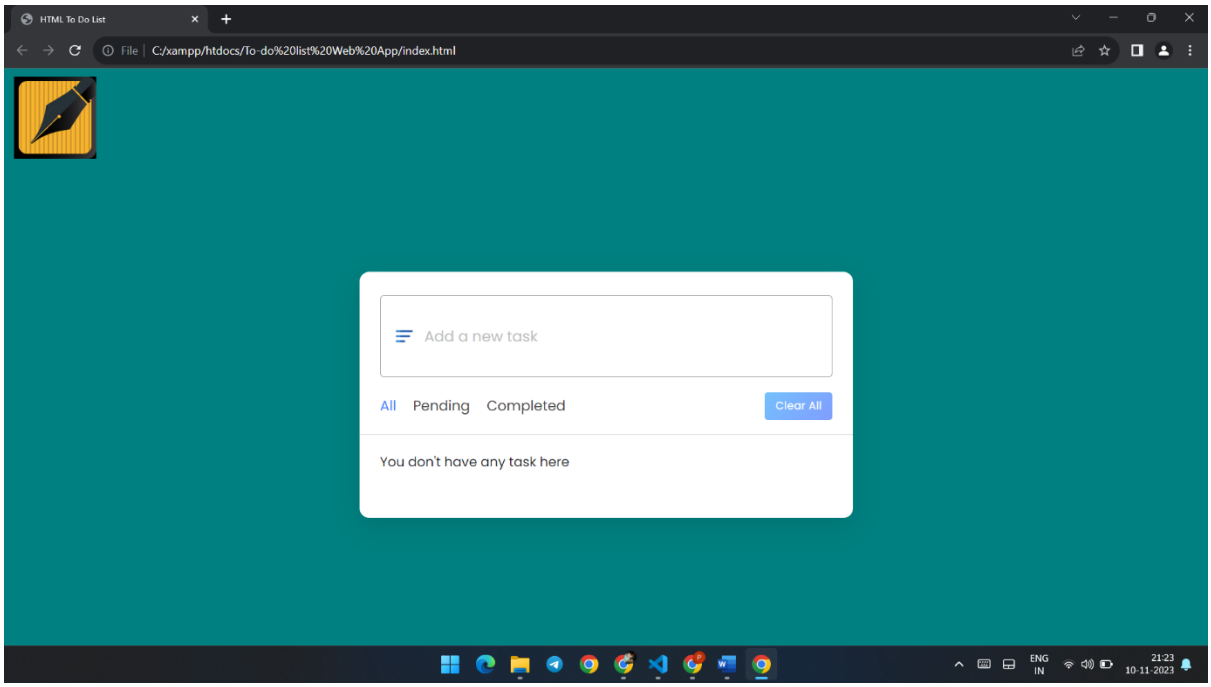
```
  height: 25px;
  font-size: 16px;
  margin-bottom: 2px;
  padding: 17px 15px;
  cursor: pointer;
  justify-content: flex-start;
}

.task-menu li:last-child {
  margin-bottom: 0;
}

.settings li:hover {
  background: #f5f5f5;
}

.settings li i {
  padding-right: 8px;
}

@media (max-width: 400px) {
  body {
    padding: 0 10px;
  }
```

```
.wrapper {
  padding: 20px 0;
}

.filters span {
  margin: 0 5px;
}

.task-input {
  padding: 0 20px;
}

.controls {
  padding: 18px 20px;
}
.task-box {
  margin-top: 20px;
  margin-right: 5px;
  padding: 0 15px 10px 20px;
}
.task label input {
  margin-top: 4px;
}
```

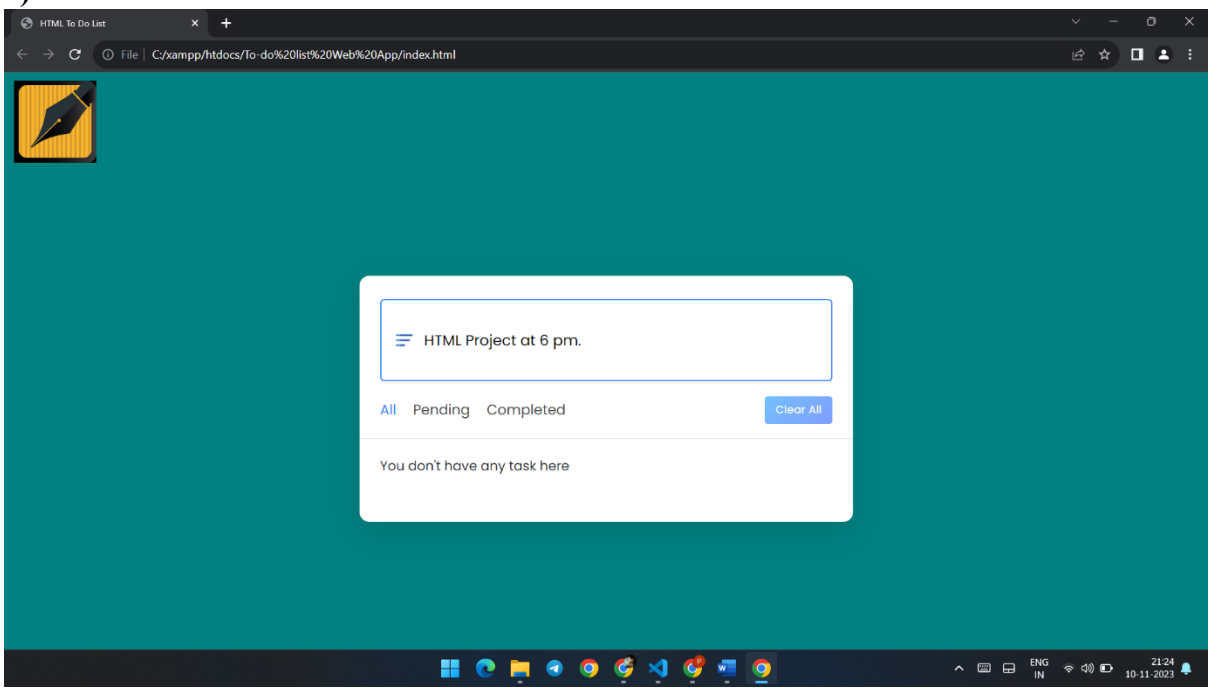
}

RESULT & OUTPUT

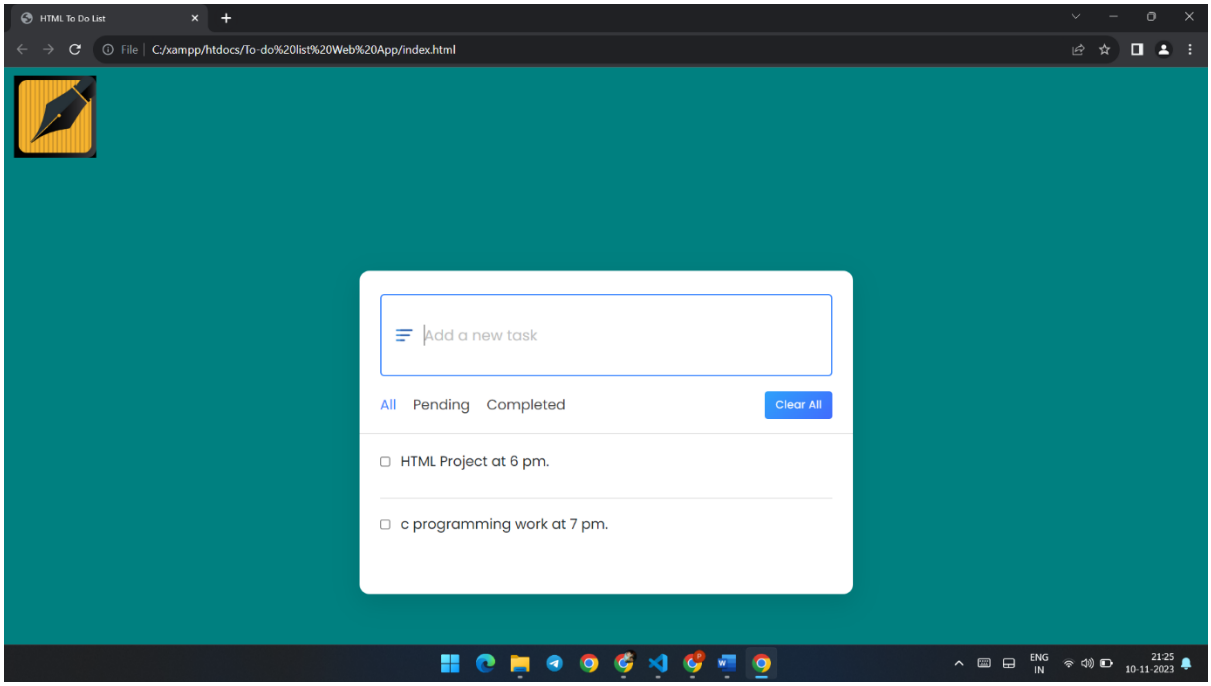
1)



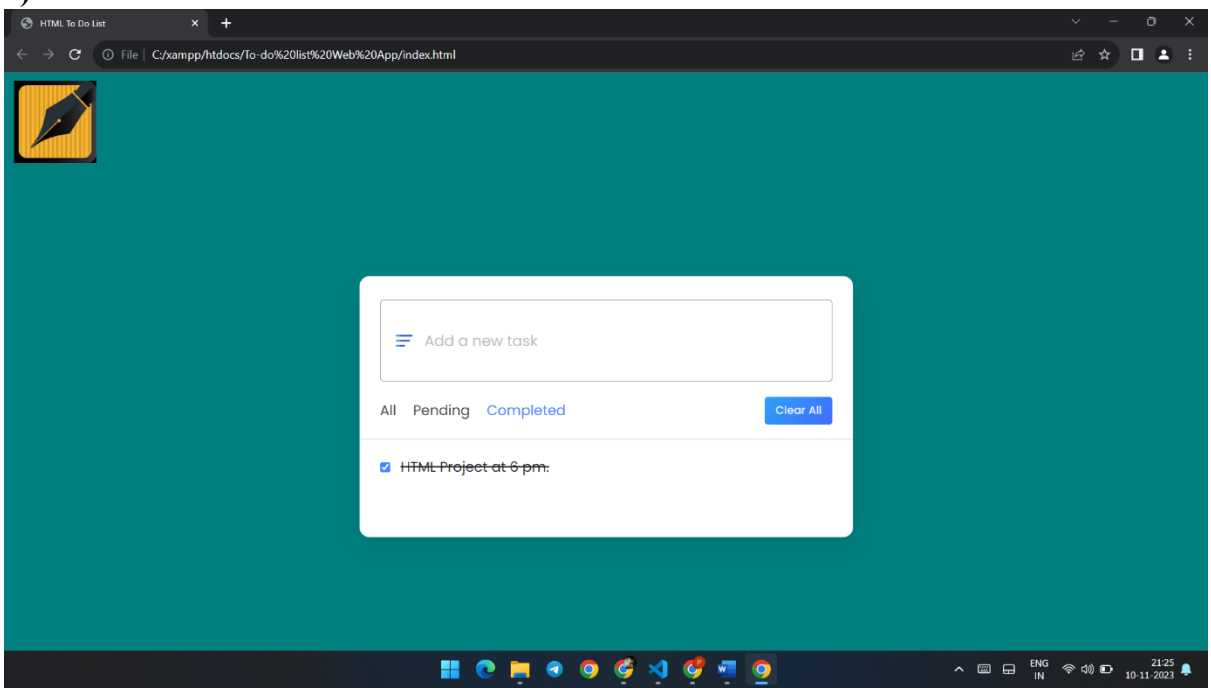
2)



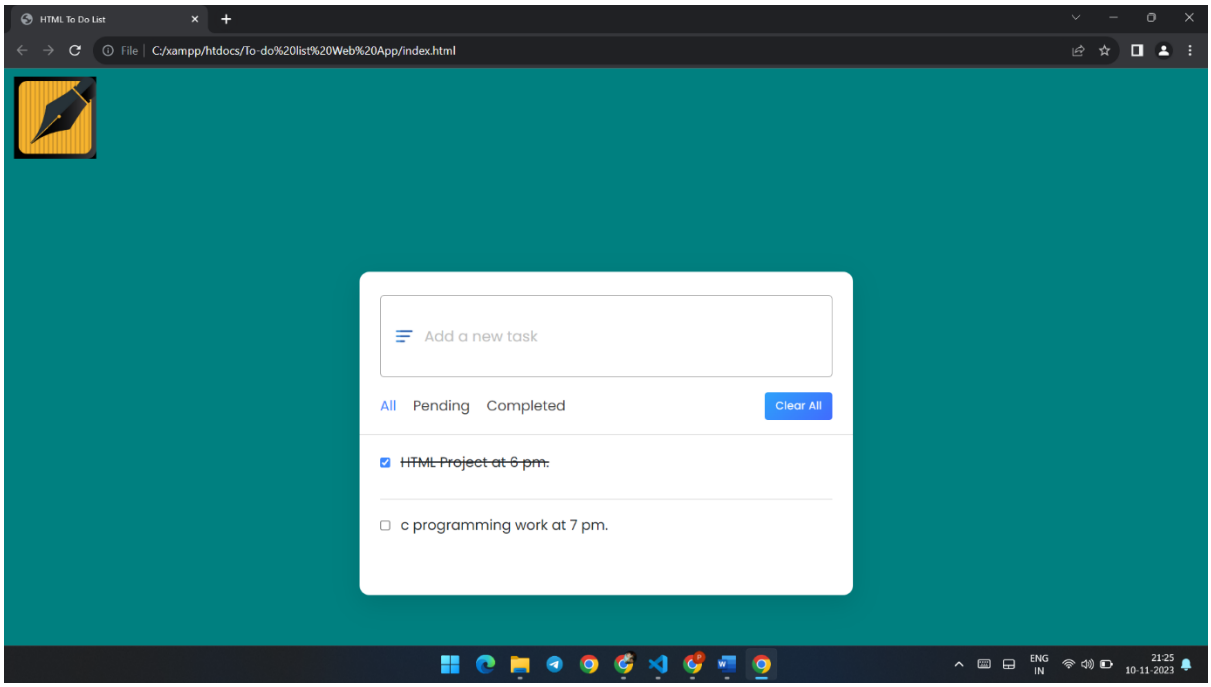
3)


HTML To Do List
File | C:/xampp/htdocs/To-do%20list%20Web%20App/index.html
Add a new task
All
Pending
Completed
Clear All
HTML Project at 6 pm.
c programming work at 7 pm.
ENG
IN
21:25
10-11-2023

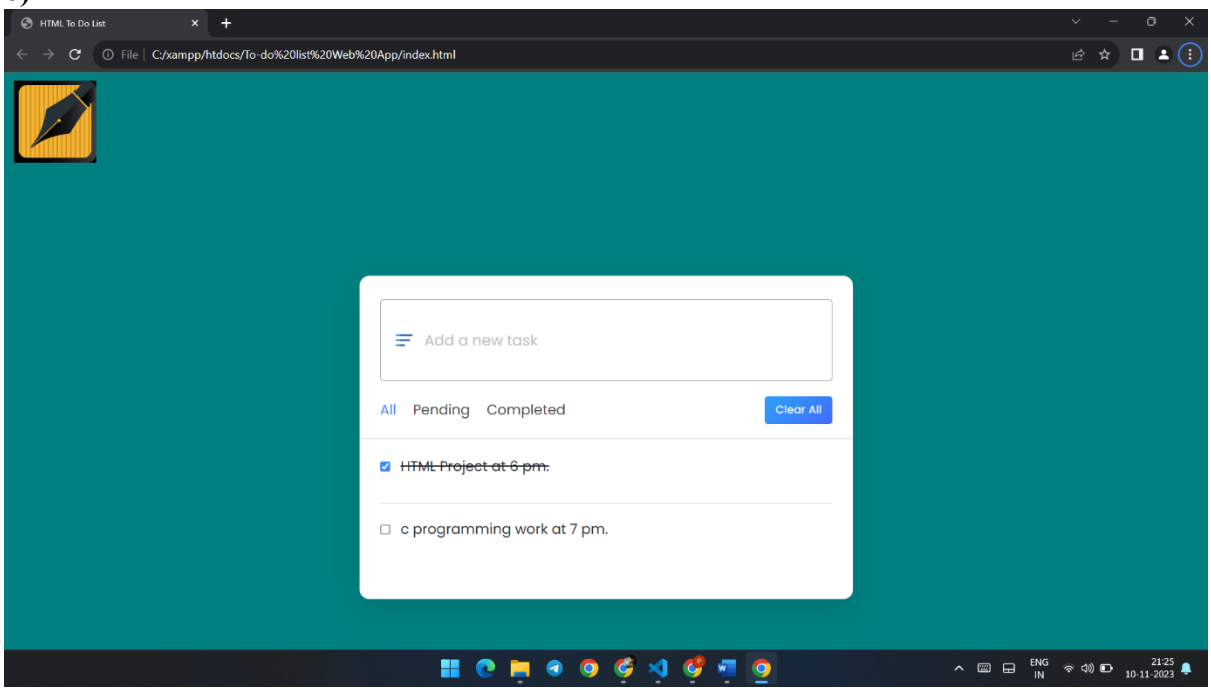
4)


HTML To Do List
File | C:/xampp/htdocs/To-do%20list%20Web%20App/index.html
Add a new task
All
Pending
Completed
Clear All
HTML Project at 6 pm.
ENG
IN
21:25
10-11-2023

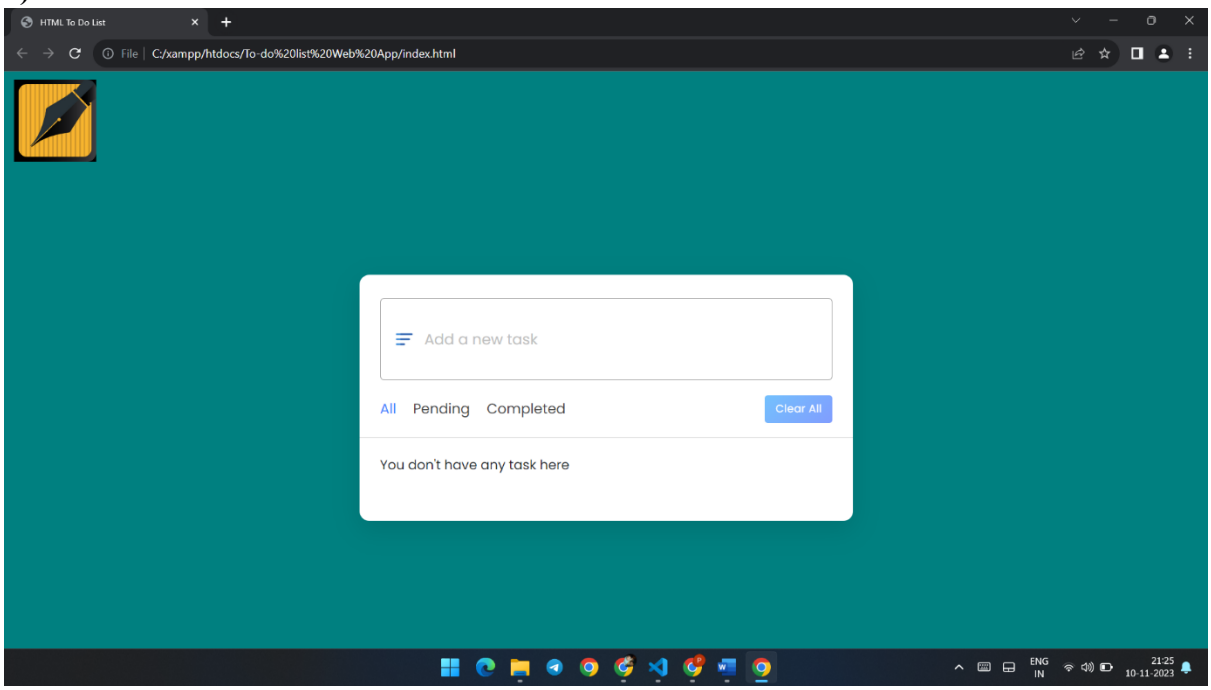
5)


HTML To Do List
File | C:/xampp/htdocs/To-do%20list%20Web%20App/index.html
Add a new task
All
Pending
Completed
Clear All
HTML Project at 6 pm.
c programming work at 7 pm.
ENG
IN
21:25
10-11-2023

6)


HTML To Do List
File | C:/xampp/htdocs/To-do%20list%20Web%20App/index.html
Add a new task
All
Pending
Completed
Clear All
HTML Project at 6 pm.
c programming work at 7 pm.
ENG
IN
21:25
10-11-2023

7)


HTML To Do List
File | C:/xampp/htdocs/To-do%20list%20Web%20App/index.html
Add a new task
All
Pending
Completed
Clear All
You don't have any task here
ENG
IN
21:25
10-11-2023

CONCLUSION

The HTML to-do list serves as a straightforward yet powerful tool for users seeking a digital solution to organize and manage their tasks. By leveraging the capabilities of HTML, CSS, and JavaScript, the application provides a user-friendly interface that allows for easy task creation, modification, and deletion. The real-time updates enhance the responsiveness of the application, offering a seamless experience for users as they interact with their to-do list. The incorporation of features such as sorting, filtering, and local storage ensures a versatile and efficient task management experience. Users can prioritize their activities, adapt to changing circumstances, and access their to-do list across sessions.