Beginner Project 2

To-Do List App (Frontend)

OBJECTIVE : Build a basic To-Do list web application using HTML, CSS, JavaScript

SKILLS : Learn how to organize your code into commits and set up a GitHub repository

GITHUB Concepts : Regularly commit your progress, create branches for features and make pull requests

Git Bash

STEPS For Pushing Files into GitHub

- **Navigate to Your Project Folder**

  - `cd C:\Users\Administrator\Desktop\ssss\To-Do list`

- **Initialize Git Repository (if not already done)**

  - `git init`

- **Add Remote Repository**

  - `git remote add origin https://github.com/Srikar2610/beg-project2.git`

- **Check Git Status**

  - `git status`

- **Stage Files for Commit**

- git add . (to stage all files)

- **Commit Changes**

    - git commit -m "Your commit message"

- **Push to GitHub**

    - git push -u origin master

After PUSHING the files to the GITHUB
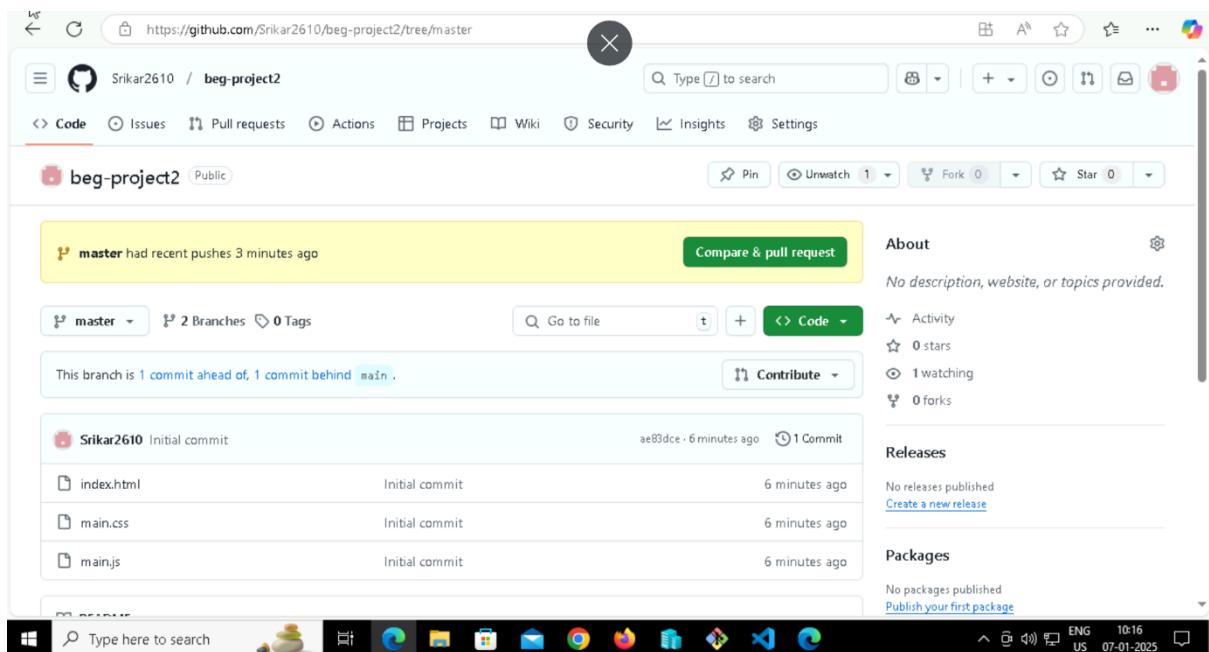
OUTPUT OF TO-DO List :
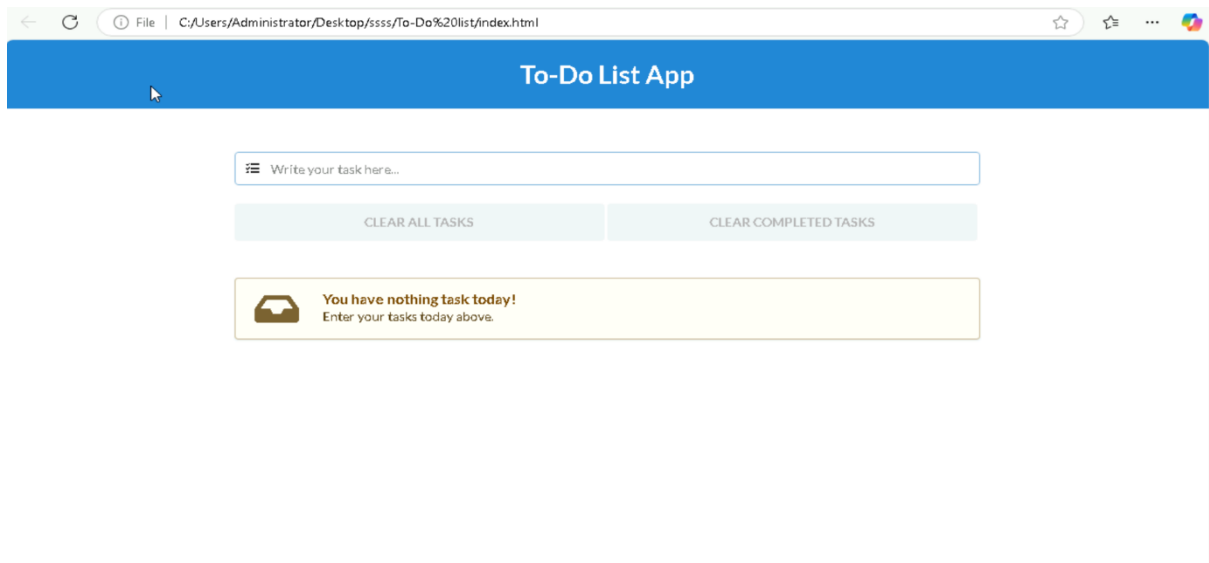
Output of To-Do list After Adding all the tasks
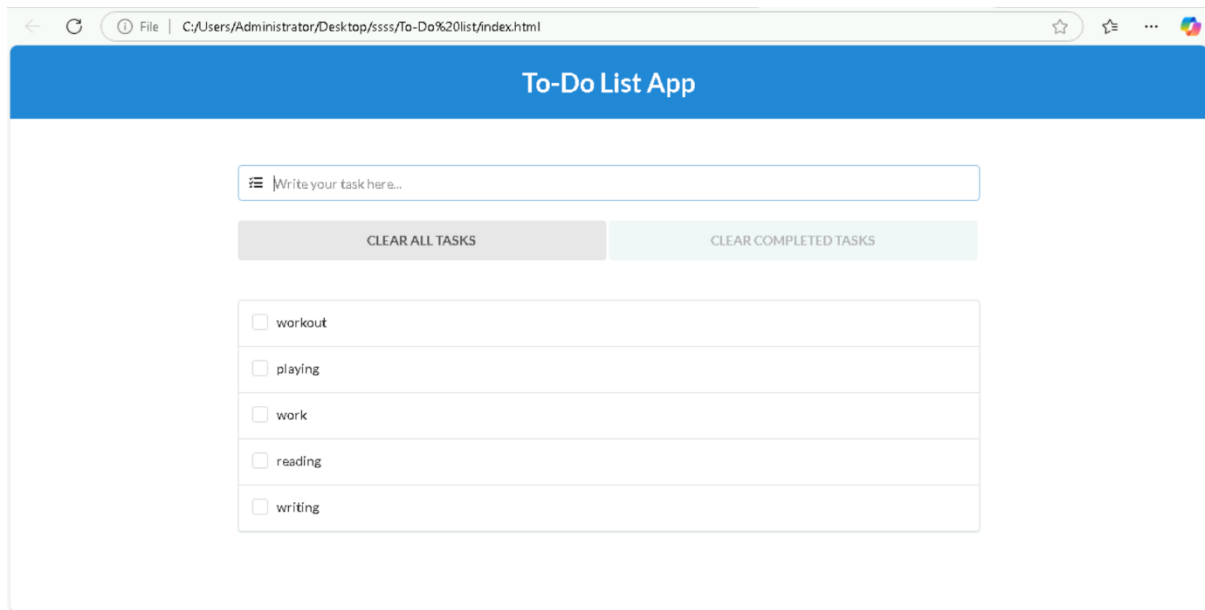
Steps for creating a branch and merging it

1. Create a New Branch

```
git checkout -b your-branch-name
```

2. Make Changes and Commit

Make changes to your files, then stage and commit them:

```
git add .
git commit -m "Your commit message"
```

3. Push the Branch to GitHub

```
git push -u origin your-branch-name
```

4. Switch to the Main Branch

```
git checkout main
```

5. Merge the Branch into Main

```
git merge your-branch-name
```

6. Push the Merged Changes to GitHub

```
git push origin main
```

7. Create a Pull Request on GitHub

- Go to your repository on GitHub.

- You'll see a prompt to create a pull request for your branch. Click **"Compare & pull request"**.
- Add a title and description, then click **"Create pull request"**.

After Creating a branch and merging the branch
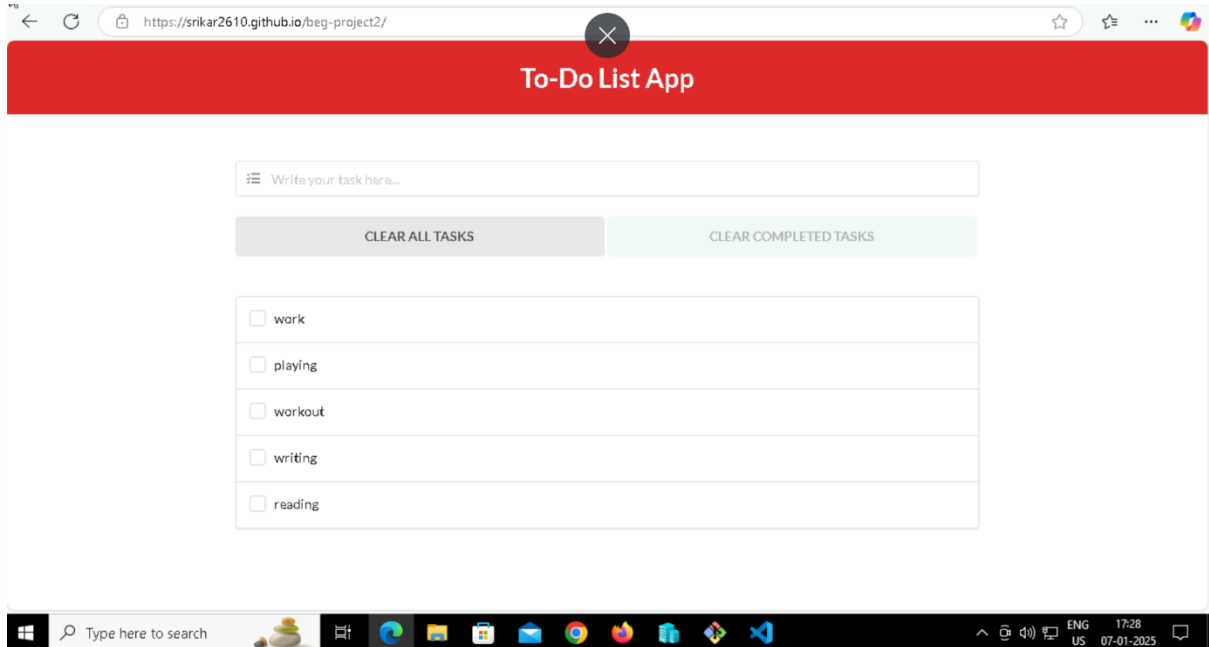
HTML CODE:

```
<!DOCTYPE html>

<html lang="en">

 <head>

  <title>To-Do List App</title>

  <!-- Meta Tags -->

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <!-- Links -->

  <link rel="icon" href="img/favicon.ico" type="image/x-icon" />

  <link

   href="https://fonts.googleapis.com/css2?family=Lato:wght@300;400;600&display=swap"

   rel="stylesheet"/>

  <link rel="stylesheet"

   href="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.css"
```

```html
    integrity="sha512-
8bHTC73gkZ7rZ7vpqUQThUDhqcNFyYi2xgDgPDHc+GXVGHXq+xPjynxIopALmOPqzo9JZj0k6OqqewdGO3EsrQ==
"
    crossorigin="anonymous" />
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/noty/3.1.4/noty.css"
    integrity="sha512-
NXUhxhkDgZYOMjaIgd89zF2w51Mub53Ru3zCNp5LTlEzMbNNAjTjDbpURYGS5Mop2cU4b7re1nOIucsVlrx9fA=="
    crossorigin="anonymous"/>
  <link rel="stylesheet" href="css/main.css" />
 </head>
 <body>
  <!-- Main Header -->
  <header class="main-header">
   <div class="ui container">
    <h1>To-Do List App</h1>
   </div>
  </header>
  <!-- End Main Header -->
  <div class="ui container">
   <!-- Add Task -->
   <form id="add-task" class="ui left icon input fluid">
    <input
      id="add-task-input"
      class="large"
      type="text"
      placeholder="Write your task here..." />
    <i class="tasks icon"></i>
   </form>
   <!-- End Add Task -->
   <!-- Buttons -->
   <div id="buttons" class="ui stackable grid equal width">
    <button
      id="clear-all-tasks"
```

```html
    class="ui button column"
    onclick="showClearAllTasksModal()">
    Clear all tasks
  </button>
  <button
    disabled
    class="ui button column"
    onclick="showClearCompletedTasksModal()">
    Clear completed tasks
  </button>
</div>
<!-- End Buttons -->
<!-- Tasks List -->
<ul id="tasks-list" class="ui segments"></ul>
<!-- End Tasks List -->
<!-- Edit Modal -->
<div id="edit-modal" class="ui tiny modal">
  <div class="header">Update Your Task</div>
  <div class="content">
    <div class="ui fluid input">
      <input id="task-id" type="hidden" />
      <input id="task-text" type="text" />
    </div>
  </div>
  <div class="actions">
    <div class="ui cancel button">Cancel</div>
    <div id="update-button" class="ui positive right labeled icon button">
      Update
      <i class="checkmark icon"></i>
    </div>
  </div>
</div>
<!-- End Edit Modal -->
```

```html
<!-- Delete Modal -->
<div id="remove-modal" class="ui tiny modal">
  <div class="header">Delete Your Task</div>
  <div class="content">
    <div class="ui fluid input">
      <input id="task-id" type="hidden" />
      <p>Are you sure you want to delete your task?</p>
    </div>
  </div>
  </div>
  <div class="actions">
    <div class="ui cancel red button">No</div>
    <div
      id="remove-button"
      onclick="clearAllTasks()"
      class="ui positive right labeled icon button">
      Yes
      <i class="checkmark icon"></i>
    </div>
  </div>
</div>
<!-- End Clear All Tasks Modal -->
<!-- Clear Completed Tasks Modal -->
<div id="clear-completed-tasks-modal" class="ui tiny modal">
  <div class="header">Clear Completed Tasks</div>
  <div class="content">
    <div class="ui fluid input">
      <input id="task-id" type="hidden" />
      <p>Are you sure you want to clear completed tasks?</p>
    </div>
  </div>
  <div class="actions">
    <div class="ui cancel red button">No</div>
```

```html
    <div
      id="remove-button"

      onclick="clearAllTasks()"

      class="ui positive right labeled icon button"

    >

      Yes

      <i class="checkmark icon"></i>

    </div>

    </div>

  </div>

  <!-- End Clear Completed Tasks Modal -->

</div>

<!-- Scripts -->

<script

  src="https://cdnjs.cloudflare.com/ajax/libs/noty/3.1.4/noty.min.js"

  integrity="sha512-
lOrm9FgT1LKOJRUXF3tp6QaMorJftUjowOWiDcG5GFZ/q7ukof19V0HKx/GWzXCdt9zYju3/KhBNdCLzK8b90Q=="

  crossorigin="anonymous"

></script>

<script

  src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"

  integrity="sha512-
bLT0Qm9VnAYZDflyKcBaQ2gg0hSYNQrJ8RilYldYQ1FxQYoCLtUjuuRuZo+fjqhx/qtq/1itJ0C2ejDxltZVFg=="

  crossorigin="anonymous"

></script>

<script

  src="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.js"

  integrity="sha512-
dqw6X88iGgZlTsONxZK9ePmJEFrmHwpuMrsUChjAw1mRUhUITE5QU9pkcSox+ynfLhL15Sv2al5A0LVyDCmtUw=
="

  crossorigin="anonymous"

></script>

<script

  type="text/javascript"
```

```html
      src="https://cdn.jsdelivr.net/npm/toastify-js"
    ></script>
    <script type="text/javascript" src="js/main.js"></script>
  </body>
</html>
```

CSS CODE :

```css
/* Variables */
:root {
        --dark: #1b1c1d;

        --blue: #2185d0;

        --green: #1ebc30;

        --red: #db2828;

        --white: #ffffff;

        --gray: #babac0;

        --secondary: #fbfdff;
}
/* Default Settings */
* {
        padding: 0;

        margin: 0;

        box-sizing: border-box;
}
body {
        color: var(--dark);

        background: var(--secondary);

        font-family: 'Lato', sans-serif;
}
.ui.grid + .grid {
        margin-top: 0;
}
/* Classes */
body::-webkit-scrollbar {
        background-color: var(--secondary);
```

```css
        width: 16px;
}

body::-webkit-scrollbar-track {
        background-color: var(--secondary);
}

body::-webkit-scrollbar-thumb {
        background-color: var(--gray);
        border-radius: 16px;
        border: 4px solid var(--white);
}

/* Header */

.main-header {
        color: var(--white);
        background: var(--blue);
        padding: 1.5rem 0;
        text-align: center;
}

/* Add Task */

#add-task {
        width: 70%;
        margin: 0 auto;
        margin-top: 2rem;
}

#add-task input {
        margin: 1.5rem auto;
}

/* Buttons */

#buttons {
        width: 70%;
        margin: 0 auto;
        margin-bottom: 3rem;
}

/* Tasks List */
```

```css
#tasks-list {

        margin: 0 auto;

        width: 70%;

        list-style: none;

}

#tasks-list li {

        width: 100%;

}

#tasks-list li .remove,

#tasks-list li .edit {

        opacity: 0;

        transition: all 0.3s ease;

}

#tasks-list li:hover .remove,

#tasks-list li:hover .edit {

        opacity: 1;

}

#tasks-list li .column:last-child {

        text-align: right;

}

#tasks-list li i {

        cursor: pointer !important;

}

/* Alert */

.alert {

        width: 70%;

        margin: 0 auto;

        margin-bottom: 1rem;

        text-align: center;

        padding: 0.75rrem 1.25rrem;

        border-radius: 10px;

        line-height: 1.6rem;

}
```

```css
.alert-warning {

        color: #856404;

        background-color: #fff3cd;

        border-color: #ffeeba;

}

/* Noty */

.noty_theme__metroui.noty_bar .noty_body {

        background: var(--green);

        color: var(--white);

        border-radius: 0.28571429rem;

        box-shadow: 0 0 0 1px var(--green) inset, 0 0 0 0 transparent;

}

.noty_theme__metroui.noty_type__error {

        background: none;

}

.noty_theme__metroui.noty_bar.noty_type__error .noty_body {

        background: var(--red);

        color: var(--white);

        border-radius: 0.28571429rem !important;

        box-shadow: 0 0 0 1px var(--red) inset, 0 0 0 0 transparent;

}


/* Responsive */

@media screen and (max-width: 770px) {

        #add-task,

        #buttons,

        #tasks-list {

                width: 100%;

        }

}
```

JavaScript CODE :

```javascript
// Elements

const tasksList = document.querySelector("#tasks-list")
```

```javascript
const addTaskForm = document.querySelector("form#add-task")

const addTaskInput = document.querySelector("#add-task-input")

const clearAllTasksBtn = document.querySelector("button#clear-all-tasks")

// Total List Of Tasks

let list = JSON.parse(localStorage.getItem("tasks")) || []

/*** Show All Tasks From Local Storage In Page*/

function showTasksList() {

  tasksList.innerHTML = ""

  const list = JSON.parse(localStorage.getItem("tasks")) || []

  if (list.length === 0) {

    clearAllTasksBtn.disabled = true

    const element = String.raw`
                            <div class="ui icon warning message">
                                <i class="inbox icon"></i>
                                <div class="content">
                                    <div class="header">You have nothing task today!</div>
                                    <div>Enter your tasks today above.</div>
                                </div>
                            </div>

    tasksList.style.border = "none"

    return tasksList.insertAdjacentHTML("beforeend", element)

  }

  clearAllTasksBtn.disabled = false

  tasksList.style.border = "1px solid rgba(34,36,38,.15)"

  list.reverse().forEach(task => {

    const element = String.raw`
                            <li class="ui segment grid equal width">
                                <div class="ui checkbox column">
                                    <input type="checkbox" ${task.completed ? "checked" : ""}>

                                    <label>${task.text}</label>
                                </div>
                                <div class="column">
```

```
                                        <i data-id="${task.id}" class="edit outline icon"></i>
                                        <i data-id="${task.id}" class="trash alternate outline
remove icon"></i>
                                    </div>
                        </li>
  tasksList.insertAdjacentHTML("beforeend", element)
 })
 document.querySelectorAll(`li i.edit`).forEach(item => {
  item.addEventListener("click", e => {
   e.stopPropagation()
   showEditModal(+e.target.dataset.id)
  })
 })
 document.querySelectorAll(`li i.trash`).forEach(item => {
  item.addEventListener("click", e => {
   e.stopPropagation()
   showRemoveModal(+e.target.dataset.id)
  })
 })
}
/*** Add new task to local storage*/
function addTask(event) {
 event.preventDefault()
 const taskText = addTaskInput.value
 if (taskText.trim().length === 0) {
  return (addTaskInput.value = "")
 }
 list.push({
  id: list.length + 1,
  text: taskText,
  completed: false,
 })
 localStorage.setItem("tasks", JSON.stringify(list))
```

```javascript
    addTaskInput.value = ""

    showNotification("success", "Task was successfully added")

    showTasksList()

}
// Change Complete State

function completeTask(id) {

  // Get Task

  const taskIndex = list.findIndex(t => t.id == id)

  const task = list[taskIndex]

  // Change State

  task.completed = !task.completed

  list[taskIndex] = task

  // Save Changes

  localStorage.setItem("tasks", JSON.stringify(list))

  showTasksList()

}
/**

 * Remove task

 */

function removeTask(id) {

  list = list.filter(t => t.id !== id)

  localStorage.setItem("tasks", JSON.stringify(list))

  showNotification("error", "Task was successfully deleted")

  showTasksList()

}
/**

 * Edit task

 */

function editTask(id) {

  const taskText = document.querySelector("#task-text").value


  if (taskText.trim().length === 0) return

  const taskIndex = list.findIndex(t => t.id == id)
```

```javascript
    list[taskIndex].text = taskText
    localStorage.setItem("tasks", JSON.stringify(list))
    showNotification("success", "Task was successfully updated")
    showTasksList()
}
// Clear All Tasks
function clearAllTasks() {
  if (list.length > 0) {
    list = []
    localStorage.setItem("tasks", JSON.stringify(list))
    return showTasksList()
  }
  new Noty({
    type: "error",
    text: '<i class="close icon"></i> There is no task to remove.',
    layout: "bottomRight",
    timeout: 2000,
    progressBar: true,
    closeWith: ["click"],
    theme: "metroui",
  }).show()
}
// Clear Complete Tasks
function clearCompleteTasks() {
  if (list.length > 0) {
    if (confirm("Are you sure?")) {
      const filteredTasks = list.filter(t => t.completed !== true)
      localStorage.setItem("tasks", JSON.stringify(filteredTasks))
      return showTasksList()
    }
  }
  Toastify({
    text: "There is no task to remove",
```

```javascript
      duration: 3000,

      close: true,

      gravity: "bottom",

      position: "left",

      backgroundColor: "linear-gradient(to right, #e45757, #d44747)",

      stopOnFocus: true,

    }).showToast()

  }
// Show Edit Modal And Pass Data

function showEditModal(id) {

  const taskIndex = list.findIndex(t => t.id == id)

  const { text } = list[taskIndex]

  document.querySelector("#edit-modal .content #task-id").value = id

  document.querySelector("#edit-modal .content #task-text").value = text.trim()

  document

    .querySelector("#update-button")

    .addEventListener("click", () => editTask(+id))

  $("#edit-modal.modal").modal("show")

  }
// Show Remove Modal

function showRemoveModal(id) {

  document

    .querySelector("#remove-button")

    .addEventListener("click", () => removeTask(+id))

  $("#remove-modal.modal").modal("show")

  }
// Show Clear All Tasks Modal

function showClearAllTasksModal() {

  if (list.length > 0) {

    return $("#clear-all-tasks-modal.modal").modal("show")

  }

  new Noty({

    type: "error",
```

```javascript
    text: '<i class="close icon"></i> There is no task to remove.',

    layout: "bottomRight",

    timeout: 2000,

    progressBar: true,

    closeWith: ["click"],

    theme: "metroui",

  }).show()

}

function showNotification(type, text) {

  new Noty({

    type,

    text: `<i class="check icon"></i> ${text}`,

    layout: "bottomRight",

    timeout: 2000,

    progressBar: true,

    closeWith: ["click"],

    theme: "metroui",

  }).show()

}

// Event Listeners

addTaskForm.addEventListener("submit", addTask)

window.addEventListener("load", () => addTaskInput.focus())

showTasksList()
```