Task Manager Web Application

Team Members: Naziya Kouser H-2022115030, Rasiga Priya S-2022115036, Kiruthiga J-2022115049

Introduction:

This project is a web-based **Task Manager** that allows users to manage their daily tasks efficiently. The application provides the ability to **add**, **edit**, **delete**, and **mark tasks as completed**, making it an ideal personal productivity tool.

The application is fully client-side, built using **HTML**, **CSS**, and **JavaScript**, and it leverages the browser's **localStorage** to persist tasks across sessions. It is hosted on a **free-tier Amazon EC2 instance**, providing a practical experience with deploying web apps to the cloud.

Technologies Used:

Layer	Technology
Frontend	HTML, CSS, JavaScript
Backend	None (Fully frontend with localStorage)
Hosting	Amazon EC2 (Amazon Linux 2023)
Web Server	Apache HTTP Server
Development Tools	MobaXterm (SSH and File Transfer)

Key Features:

- **Add Tasks:** Users can add new tasks dynamically.
- **Edit Tasks:** Tasks can be modified in-place and saved again.
- **Delete Tasks:** Irrelevant tasks can be removed.
- **Mark as Completed:** Tasks can be marked as done, striking them out instead of removing.
- **Persistent Storage:** Tasks are saved using browser localStorage.
- **Modern UI:** Clean black and green professional color scheme.

Project Structure:

All code is written inside a **single HTML file** containing:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Task Manager</title>
  <style>
    body {
      font-family: 'Segoe UI', sans-serif;
      background-color: #121212;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: flex-start;
      height: 100vh;
      color: #eee;
    }

    .container {
      margin-top: 50px;
      width: 420px;
      background-color: #1f1f1f;
      padding: 25px 30px;
      border-radius: 14px;
      box-shadow: 0 0 20px rgba(0, 255, 0, 0.15);
    }

    h1 {
      text-align: center;
      margin-bottom: 20px;
      color: #00ff88;
    }

    .input-section {
      display: flex;
      gap: 10px;
    }

    input[type="text"] {
      flex: 1;
      padding: 10px;
      border-radius: 6px;
      border: 1px solid #00ff88;
      font-size: 16px;
      background-color: #222;
      color: #fff;
    }

    button {
      background-color: #00ff88;
      color: #000;
      border: none;
      padding: 10px 14px;
      border-radius: 6px;
      cursor: pointer;
      font-size: 15px;
      font-weight: bold;
      transition: 0.2s;
    }

    button:hover {
      background-color: #00cc66;
    }

    ul {
      list-style-type: none;
      padding: 0;
      margin-top: 20px;
    }

    li {
      background-color: #2a2a2a;
      margin-bottom: 12px;
      padding: 12px;
      border-radius: 6px;
      display: flex;
      flex-direction: column;
    }
```

```css
.top-row {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 8px;
}

li input[type="text"] {
  flex: 1;
  border: none;
  background: transparent;
  font-size: 16px;
  color: #eee;
}

.completed input[type="text"] {
  text-decoration: line-through;
  color: #888;
}

.actions {
  display: flex;
  justify-content: space-between;
  gap: 10px;
}
```

```
    .actions button {
      flex: 1;
      padding: 8px;
      font-size: 14px;
    }

    .edit-btn:hover {
      background-color: #ffc107;
      color: #000;
    }

    .delete-btn:hover {
      background-color: #f44336;
      color: #fff;
    }

    .complete-btn:hover {
      background-color: #2196f3;
      color: white;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>📝 Task Manager</h1>
    <div class="input-section">
```

```
      <input type="text" id="taskInput" placeholder="Enter a task..." />
      <button onclick="addTask()">Add</button>
    </div>
    <ul id="taskList"></ul>
  </div>

  <script>
    let tasks = JSON.parse(localStorage.getItem('tasks')) || [];

    function renderTasks() {
      const taskList = document.getElementById('taskList');
      taskList.innerHTML = '';

      tasks.forEach((taskObj, index) => {
        const li = document.createElement('li');
        if (taskObj.completed) {
          li.classList.add('completed');
        }

        const topRow = document.createElement('div');
        topRow.className = 'top-row';

        const input = document.createElement('input');
        input.type = 'text';
        input.value = taskObj.name;
        input.disabled = true;
        topRow.appendChild(input);


        const actions = document.createElement('div');

        actions.className = 'actions';


        const completeBtn = document.createElement('button');

        completeBtn.textContent = taskObj.completed ? 'Undo' : 'Completed';

        completeBtn.className = 'complete-btn';

        completeBtn.onclick = () => {

          tasks[index].completed = !tasks[index].completed;

          localStorage.setItem('tasks', JSON.stringify(tasks));

          renderTasks();

        };


        const editBtn = document.createElement('button');

        editBtn.textContent = 'Edit';

        editBtn.className = 'edit-btn';

        editBtn.onclick = () => {

          if (input.disabled) {

            input.disabled = false;

            input.focus();

            editBtn.textContent = 'Save';

          } else {

            tasks[index].name = input.value;

            localStorage.setItem('tasks', JSON.stringify(tasks));
        input.disabled = true;
        editBtn.textContent = 'Edit';
      }
    };

    const deleteBtn = document.createElement('button');
    deleteBtn.textContent = 'Delete';
    deleteBtn.className = 'delete-btn';
    deleteBtn.onclick = () => {
      tasks.splice(index, 1);
      localStorage.setItem('tasks', JSON.stringify(tasks));
      renderTasks();
    };

    actions.appendChild(completeBtn);
    actions.appendChild(editBtn);
    actions.appendChild(deleteBtn);

    li.appendChild(topRow);
    li.appendChild(actions);
    taskList.appendChild(li);
  });
}

function addTask() {
  const taskInput = document.getElementById('taskInput');
  const taskName = taskInput.value.trim();
```

```
      if (taskName) {
        tasks.push({ name: taskName, completed: false });
        localStorage.setItem('tasks', JSON.stringify(tasks));
        taskInput.value = '';
        renderTasks();
      }
    }

    window.onload = renderTasks;
  </script>
</body>
</html>
```

Deployment Steps on AWS EC2:

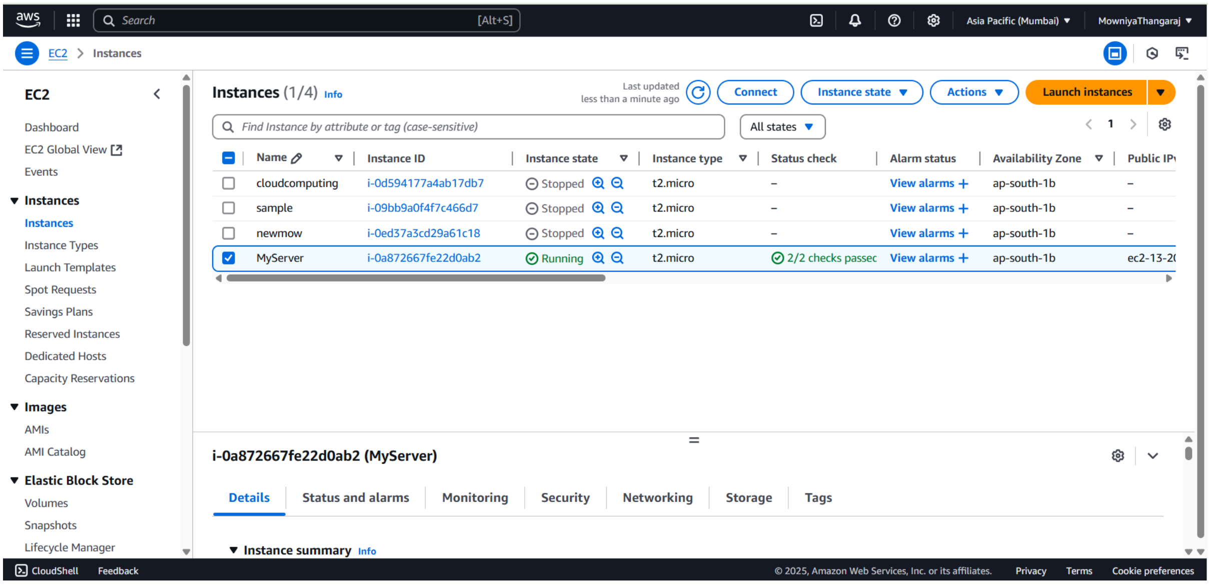
1. **Created an EC2 Instance** on AWS using Amazon Linux 2023.
2. **Connected via MobaXterm** using the `.pem` file (SSH).
3. **Installed Apache** using:
   sudo yum install httpd -y
4. **Started Apache server** using:

   sudo service httpd start

5. **Uploaded `index.html`** to `/var/www/html/` using MobaXterm or `mv` command.
6. **Accessed the app** via EC2 Public IP in a browser.

Screenshots:

- EC2 Instance Running



Instances (1/4) Info

Last updated less than a minute ago

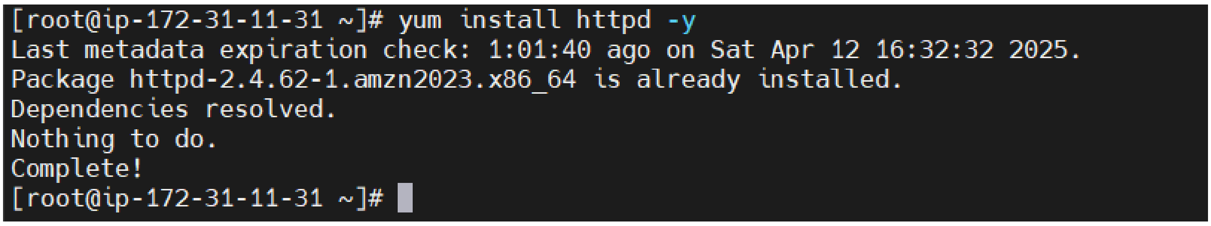
Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
□	cloudcomputing	i-0d594177a4ab17db7	Stopped	t2.micro	–	View alarms +	ap-south-1b	–
□	sample	i-09bb9a0f4f7c466d7	Stopped	t2.micro	–	View alarms +	ap-south-1b	–
□	newmow	i-0ed37a3cd29a61c18	Stopped	t2.micro	–	View alarms +	ap-south-1b	–
☑	MyServer	i-0a872667fe22d0ab2	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-2(

i-0a872667fe22d0ab2 (MyServer)

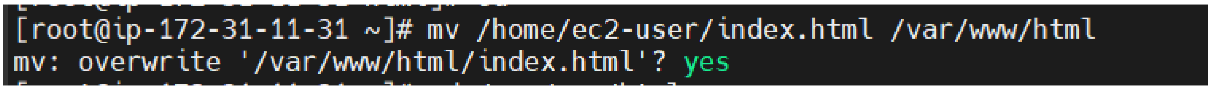
▼ Instance summary Info

- Apache Installation



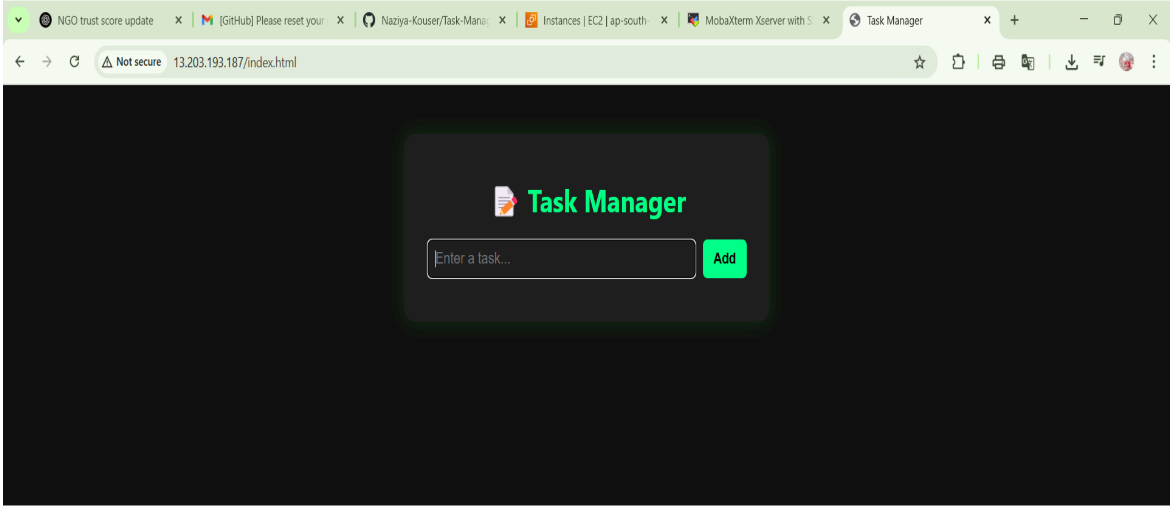
```
[root@ip-172-31-11-31 ~]# yum install httpd -y
Last metadata expiration check: 1:01:40 ago on Sat Apr 12 16:32:32 2025.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-11-31 ~]#
```

- File Upload in MobaXterm



```
[root@ip-172-31-11-31 ~]# mv /home/ec2-user/index.html /var/www/html
mv: overwrite '/var/www/html/index.html'? yes
```

- Webpage Running in Browser



Outcome & Learnings:

- Understood the practical use of **HTML/CSS/JavaScript** to build a real application.
- Learned how to **deploy a static website on AWS EC2**.
- Worked with **localStorage** for browser-side persistence.
- Gained experience with **MobaXterm**, **Linux commands**, and **Apache server setup**.

Conclusion:

This project helped me understand both **frontend web development** and **cloud deployment**. It’s a minimal yet effective application showcasing core CRUD functionalities and deployment skills.
