Activity Software Developer

File: #3147247

Present by:
Carlos Mario Martinez Trillos

Present to:
Dario

Servicio Nacional de Aprendizaje
SENA

Software Analysis and Development
Bogotá D.C 24/09/2025

Hi Professor, I wasn’t sure about the visual guidelines, so I gave it a retro/gamer style. I hope it’s okay.

Retro Gamer Task Page:

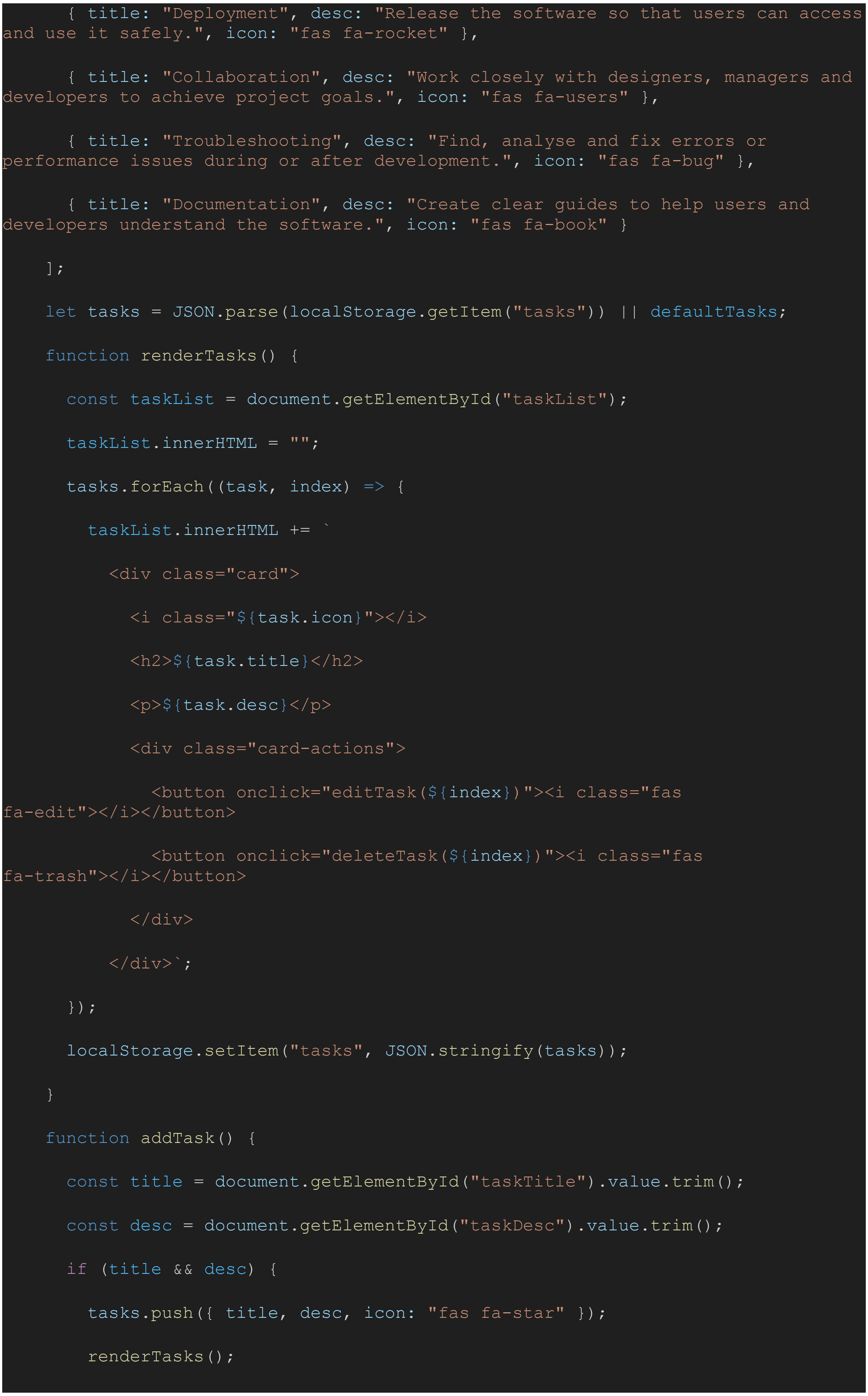
Save the HTML and CSS files in the same folder.

1. Open the HTML file in a modern web browser (Chrome, Firefox, Edge).
2. The CSS will automatically style the HTML.
3. Click anywhere on the page to see the retro explosion animation.
4. Use the buttons inside the cards to edit or delete tasks.

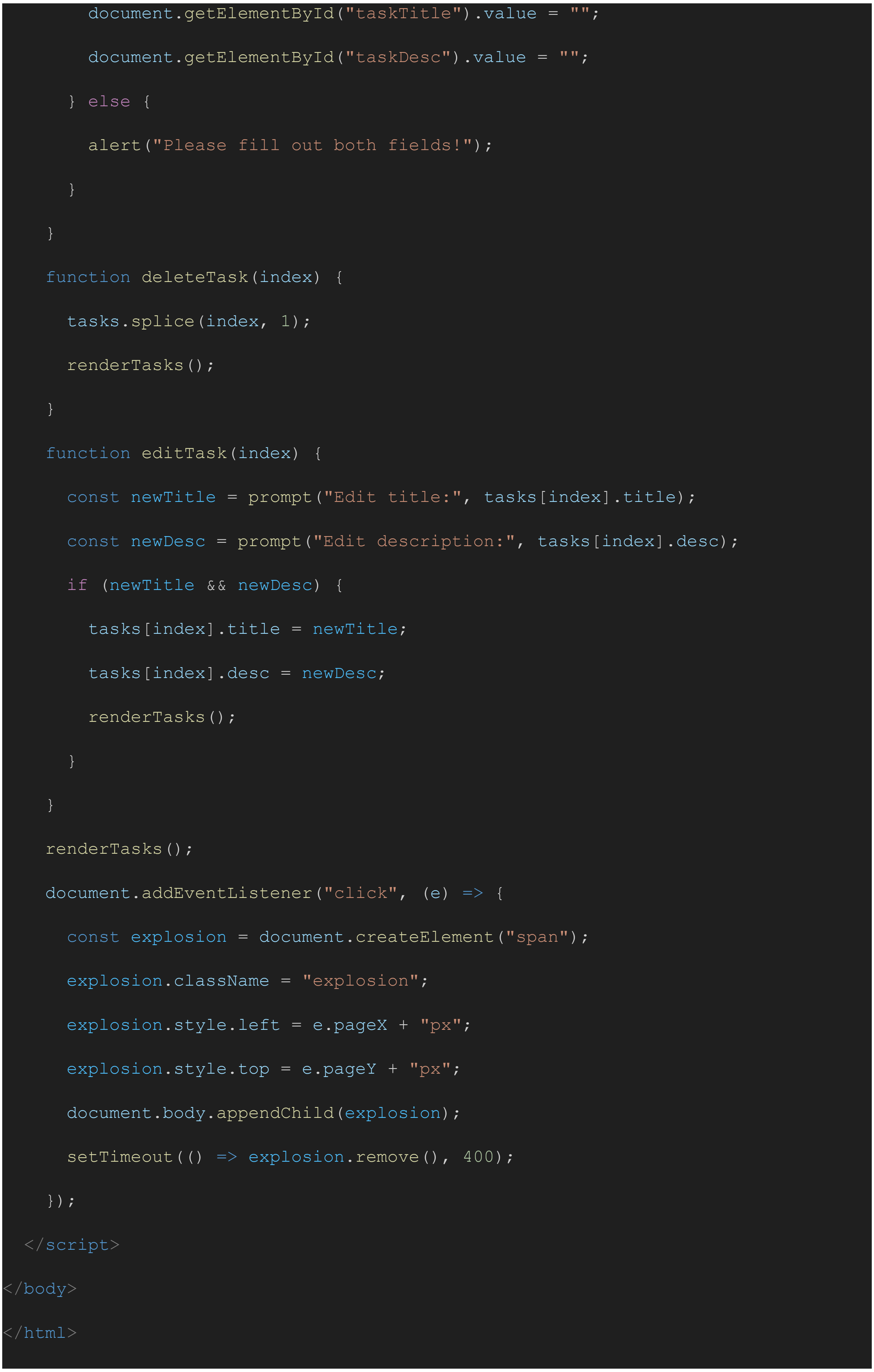
5. For easier access, download the page ZIP file and extract it. Then, double-click the `index.html` file and it will open in your default browser. You can choose a different browser if you want.

index.html

```html
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Software Developer Functions</title>
  <link rel="stylesheet" href="styles.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">
</head>
<body>
  <header>
    <h1>What does a Software Developer do?</h1>
    <p>Main responsibilities and assignments explained in retro gamer style.</p>
  </header>
  <section class="form-container">
    <input type="text" id="taskTitle" placeholder="Task title">
    <input type="text" id="taskDesc" placeholder="Task description">
    <button onclick="addTask()"><i class="fas fa-plus"></i> Add</button>
  </section>

  <main class="cards-container" id="taskList"></main>
  <script>
    const defaultTasks = [
      { title: "Analyse", desc: "Study requirements and identify user and system needs to plan efficient software solutions.", icon: "fas fa-search" },
      { title: "Design", desc: "Create architecture, user interfaces and structures before starting development.", icon: "fas fa-drafting-compass" },
      { title: "Coding", desc: "Write clean and maintainable code to build the planned software functionalities.", icon: "fas fa-code" },
      { title: "Version Control", desc: "Use Git and other tools to track changes, collaborate and manage versions.", icon: "fas fa-code-branch" },
      { title: "Testing", desc: "Run different tests to ensure the software works correctly and without critical bugs.", icon: "fas fa-vial" },
```



```
      { title: "Deployment", desc: "Release the software so that users can access and use it safely.", icon: "fas fa-rocket" },

      { title: "Collaboration", desc: "Work closely with designers, managers and developers to achieve project goals.", icon: "fas fa-users" },

      { title: "Troubleshooting", desc: "Find, analyse and fix errors or performance issues during or after development.", icon: "fas fa-bug" },

      { title: "Documentation", desc: "Create clear guides to help users and developers understand the software.", icon: "fas fa-book" }

    ];

    let tasks = JSON.parse(localStorage.getItem("tasks")) || defaultTasks;

    function renderTasks() {

      const taskList = document.getElementById("taskList");

      taskList.innerHTML = "";

      tasks.forEach((task, index) => {

        taskList.innerHTML += `

          <div class="card">

            <i class="${task.icon}"></i>

            <h2>${task.title}</h2>

            <p>${task.desc}</p>

            <div class="card-actions">

              <button onclick="editTask(${index})"><i class="fas fa-edit"></i></button>

              <button onclick="deleteTask(${index})"><i class="fas fa-trash"></i></button>

            </div>

          </div>`;

      });

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

    }

    function addTask() {

      const title = document.getElementById("taskTitle").value.trim();

      const desc = document.getElementById("taskDesc").value.trim();

      if (title && desc) {

        tasks.push({ title, desc, icon: "fas fa-star" });

        renderTasks();
```

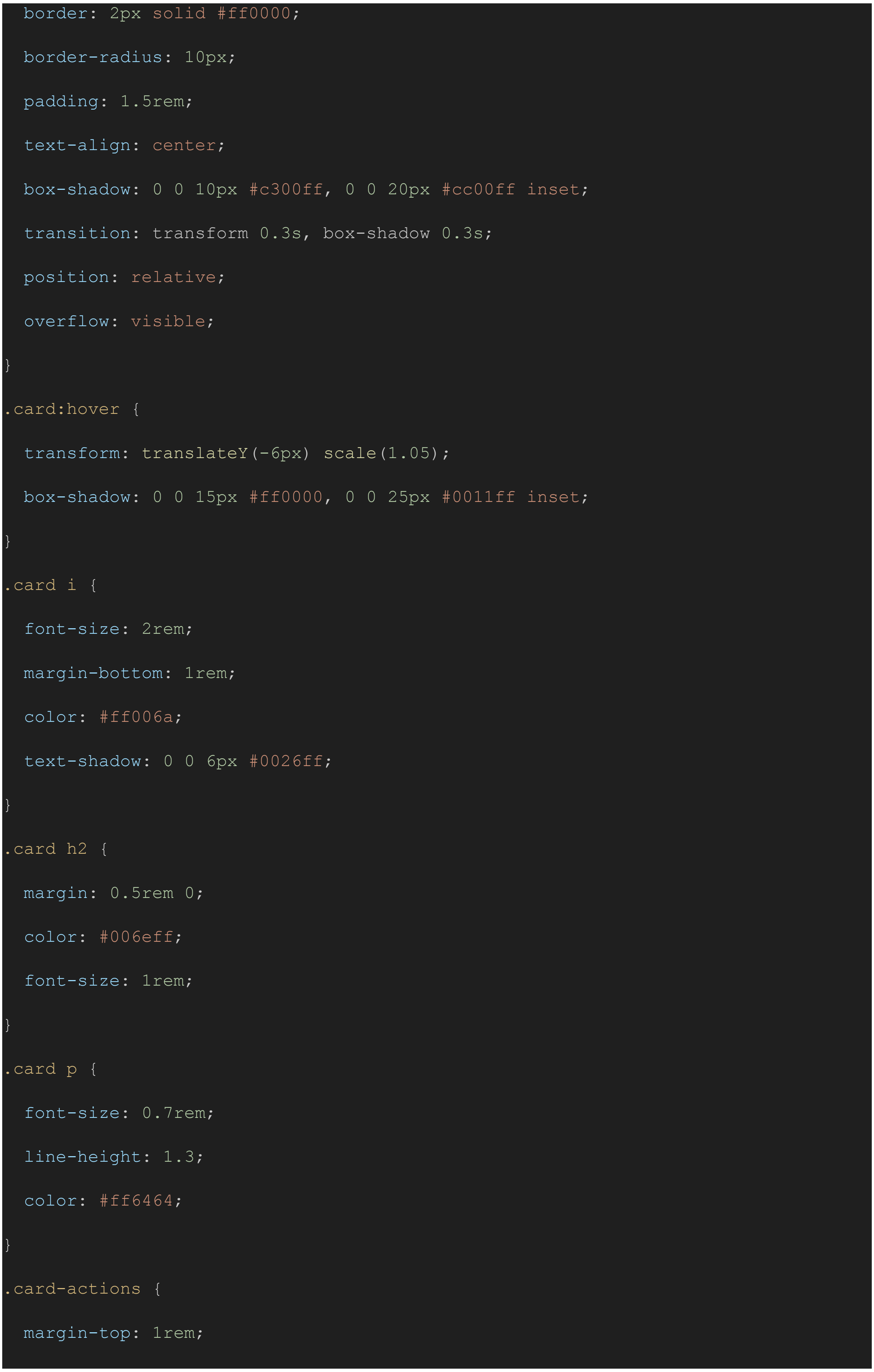


```
        document.getElementById("taskTitle").value = "";

        document.getElementById("taskDesc").value = "";

      } else {

        alert("Please fill out both fields!");

      }

    }

    function deleteTask(index) {

      tasks.splice(index, 1);

      renderTasks();

    }

    function editTask(index) {

      const newTitle = prompt("Edit title:", tasks[index].title);

      const newDesc = prompt("Edit description:", tasks[index].desc);

      if (newTitle && newDesc) {

        tasks[index].title = newTitle;

        tasks[index].desc = newDesc;

        renderTasks();

      }

    }

    renderTasks();

    document.addEventListener("click", (e) => {

      const explosion = document.createElement("span");

      explosion.className = "explosion";

      explosion.style.left = e.pageX + "px";

      explosion.style.top = e.pageY + "px";

      document.body.appendChild(explosion);

      setTimeout(() => explosion.remove(), 400);

    });

  </script>

</body>

</html>
```

styles.css:

```
@import
url('https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap');

body {

  margin: 0;

  font-family: 'Press Start 2P', cursive;

  background: #0A0A0A;

  color: #00ff99;

}

header {

  text-align: center;

  background: #111;

  padding: 2rem 1rem;

  border-bottom: 4px solid #09ff00;

  text-shadow: 2px 2px #000;

}

header h1 {

  margin: 0;

  font-size: 1.5rem;

  color: #ff00bf;

}

header p {

  margin-top: 0.5rem;

  font-size: 0.8rem;

  color: #0c44ff;

}

.form-container {

  display: flex;

  justify-content: center;

  gap: 0.5rem;

  padding: 1rem;

  background: #111;
```

```css
  border-bottom: 2px solid #00ff99;
}
.form-container input {
  font-family: 'Press Start 2P', cursive;
  padding: 0.5rem;
  border: 2px solid #00ff0d;
  background: #000;
  color: #ff3c00;
  width: 150px;
}
.form-container button {
  font-family: 'Press Start 2P', cursive;
  background: #6f00ff;
  border: none;
  padding: 0.5rem 1rem;
  cursor: pointer;
  color: #000;
  transition: 0.2s;
}
.form-container button:hover {
  background: #ffc400;
  color: #ff0095;
}
.cards-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(230px, 1fr));
  gap: 1.5rem;
  padding: 2rem;
}
.card {
  background: #111;
```



```css
  border: 2px solid #ff0000;

  border-radius: 10px;

  padding: 1.5rem;

  text-align: center;

  box-shadow: 0 0 10px #c300ff, 0 0 20px #cc00ff inset;

  transition: transform 0.3s, box-shadow 0.3s;

  position: relative;

  overflow: visible;

}

.card:hover {

  transform: translateY(-6px) scale(1.05);

  box-shadow: 0 0 15px #ff0000, 0 0 25px #0011ff inset;

}

.card i {

  font-size: 2rem;

  margin-bottom: 1rem;

  color: #ff006a;

  text-shadow: 0 0 6px #0026ff;

}

.card h2 {

  margin: 0.5rem 0;

  color: #006eff;

  font-size: 1rem;

}

.card p {

  font-size: 0.7rem;

  line-height: 1.3;

  color: #ff6464;

}

.card-actions {

  margin-top: 1rem;
  display: flex;
  justify-content: center;
  gap: 0.5rem;
}
.card-actions button {
  background: transparent;
  border: none;
  padding: 0.3rem 0.6rem;
  cursor: pointer;
  font-size: 0.7rem;
}
.card-actions button:hover {
  color: #8f1111;
}
::-webkit-scrollbar {
  width: 10px;
  height: 10px;
}
::-webkit-scrollbar-track {
  background: #000;
  border: 2px solid #00ff99;
}
::-webkit-scrollbar-thumb {
  background: #00ff99;
  box-shadow: 0 0 10px #00ff99;
}
::-webkit-scrollbar-thumb:hover {
  background: #ff00cc;
  box-shadow: 0 0 15px #ff00cc;
}
body {
  scrollbar-width: thin;

  scrollbar-color: #00ff99 #000;

}

.card:active::after,

button:active::after {

  content: "";

  position: absolute;

  top: 50%;

  left: 50%;

  width: 6px;

  height: 6px;

  background: #ff00cc;

  box-shadow:

    12px 0 #00ff99,

    -12px 0 #00ff99,

    0 12px #ff00cc,

    0 -12px #ff00cc;

  transform: translate(-50%, -50%) scale(1);

  animation: retro-explosion 0.25s ease-out forwards;

  border-radius: 2px;

  pointer-events: none;

}

.explosion {

  position: absolute;

  width: 6px;

  height: 6px;

  background: #ff00cc;

  border-radius: 2px;

  animation: retro-explosion 0.25s ease-out forwards;

  pointer-events: none;

  box-shadow:
    8px 0 #00ff99,
    -8px 0 #00ff99,
    0 8px #ffea00,
    0 -8px #ff00cc,
    6px 6px #ff00cc,
    -6px -6px #00ff99,
    6px -6px #ffea00,
    -6px 6px #ff00cc;
}
@keyframes retro-explosion {
  0% {
    opacity: 1;
    transform: translate(-50%, -50%) scale(0.8);
  }
  50% {
    opacity: 1;
    transform: translate(-50%, -50%) scale(1.3);
  }
  100% {
    opacity: 0;
    transform: translate(-50%, -50%) scale(2);
  }
}
```