





```
MINGW64/c/Users/Administrator/desktop/TODOLIST
$ git merge --no-commit --no-push origin/main
Merge branch 'main' of https://github.com/KyathamRohith/todolist
# Please enter a commit message to explain why this merge is necessary,
# merge branch 'main' of https://github.com/KyathamRohith/todolist
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
From https://github.com/KyathamRohith/todolist
+ branch      main      -> FETCH_HEAD
Auto-merging styles.css
Merge made by the 'ort' strategy.
 styles.css | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Administrator@e1d77021d9b0504 MINGW64 ~/desktop/TODOLIST (main)
$ git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 665 bytes | 665.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To https://github.com/KyathamRohith/todolist.git
 4e17417..b66908e  main -> main

Administrator@e1d77021d9b0504 MINGW64 ~/desktop/TODOLIST (main)
$
```

HTML 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="stylesheet" href="../css/styles.css">
```

```
</head>
```

```
<body>
```

```
  <div class="container">
```

```
    <header>
```

```
      <h1>To-Do List</h1>
```

```
      <p id="date"></p>
```

```
    </header>
```

```
    <main>
```

```
      <div class="input-section">
```

```
        <input type="text" id="task-input" placeholder="Add a new task">
```

```
        <button id="add-task-btn">Add Task</button>
```

```

        <button id="view-all-tasks-btn">View All Tasks</button>
    </div>
</main>
</div>

<!-- Modal for viewing all tasks -->
<div id="modal" class="modal">
    <div class="modal-content">
        <span class="close-btn">&times;</span>
        <h2>All Tasks</h2>
        <ul id="all-tasks-list"></ul>
    </div>
</div>

<script src="../js/script.js"></script>
</body>
</html>

```

CSS CODE:

```

/* Body and container styles */
body {
    font-family: 'Poppins', sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: url('../assets/background.jpg') no-repeat center center/cover;
    color: #fff;
    overflow: hidden;
}

```

```
.container {  
  background: rgba(0, 0, 0, 0.8);  
  padding: 30px;  
  border-radius: 12px;  
  text-align: center;  
  width: 90%;  
  max-width: 500px;  
  box-shadow: 0 8px 15px rgba(0, 0, 0, 0.5);  
  animation: fadeIn 1s ease-in-out;  
}
```

```
@keyframes fadeIn {  
  from {  
    opacity: 0;  
    transform: scale(0.95);  
  }  
  to {  
    opacity: 1;  
    transform: scale(1);  
  }  
}
```

```
header h1 {  
  font-size: 2.5rem;  
  margin: 0;  
  color: #FFD700;  
}
```

```
header p {  
  font-size: 1.1rem;  
  color: #ccc;
```

```
    margin-top: 10px;
}
```

```
.input-section {
    display: flex;
    gap: 10px;
    margin-bottom: 20px;
    flex-wrap: wrap;
}
```

```
#task-input {
    flex: 1;
    padding: 12px;
    border: none;
    border-radius: 8px;
    font-size: 1rem;
    color: #333;
}
```

```
#add-task-btn, #view-all-tasks-btn {
    padding: 12px 20px;
    background: linear-gradient(45deg, #6a11cb, #2575fc);
    border: none;
    border-radius: 8px;
    color: #fff;
    cursor: pointer;
    transition: transform 0.2s, box-shadow 0.2s;
}
```

```
#view-all-tasks-btn {
    background: linear-gradient(45deg, #ff416c, #ff4b2b);
}
```

```
#add-task-btn:hover, #view-all-tasks-btn:hover {  
    transform: scale(1.05);  
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.5);  
}
```

```
/* Modal styles */
```

```
.modal {  
    display: none;  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    background: rgba(0, 0, 0, 0.8);  
    justify-content: center;  
    align-items: center;  
}
```

```
.modal-content {  
    background: #333;  
    color: #fff;  
    padding: 20px;  
    border-radius: 10px;  
    width: 90%;  
    max-width: 500px;  
    text-align: center;  
    position: relative;  
    animation: slideDown 0.5s ease-in-out;  
}
```

```
@keyframes slideDown {
```

```
from {  
    transform: translateY(-100%);  
    opacity: 0;  
}  
to {  
    transform: translateY(0);  
    opacity: 1;  
}  
}
```

```
.close-btn {  
    font-size: 1.5rem;  
    color: #fff;  
    float: right;  
    cursor: pointer;  
    margin-top: -10px;  
}
```

```
#all-tasks-list {  
    list-style: none;  
    padding: 0;  
    margin: 20px 0;  
}
```

```
#all-tasks-list li {  
    background: #444;  
    margin: 10px 0;  
    padding: 15px;  
    border-radius: 8px;  
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.4);  
    color: #fff;  
}
```


JS CODE:

```
document.addEventListener("DOMContentLoaded", () => {  
    const taskInput = document.getElementById("task-input");  
    const addTaskBtn = document.getElementById("add-task-btn");  
    const viewAllTasksBtn = document.getElementById("view-all-tasks-btn");  
    const dateElement = document.getElementById("date");  
    const modal = document.getElementById("modal");  
    const modalCloseBtn = modal.querySelector(".close-btn");  
    const allTasksList = document.getElementById("all-tasks-list");  
  
    // Display current date  
    const today = new Date();  
    const options = { weekday: 'long', month: 'long', day: 'numeric', year: 'numeric' };  
    dateElement.textContent = today.toLocaleDateString('en-US', options);  
  
    // Add task  
    addTaskBtn.addEventListener("click", () => {  
        const task = taskInput.value.trim();  
        if (task) {  
            const now = new Date();  
            const timeString = now.toLocaleTimeString([], { hour: '2-digit', minute: '2-digit' });  
  
            const modalTask = document.createElement("li");  
            modalTask.textContent = `${task} (Added at: ${timeString})`;  
            allTasksList.appendChild(modalTask);  
  
            taskInput.value = "";  
        }  
    });  
  
    // View all tasks  
    viewAllTasksBtn.addEventListener("click", () => {
```

```
        modal.style.display = "flex";
    });

    // Close modal
    modalCloseBtn.addEventListener("click", () => {
        modal.style.display = "none";
    });
});
```

Steps:

Steps to Prepare a TODOLIST Using HTML, CSS, and JavaScript and Add It to GitHub

Step 1:

1. Create a folder for your project (e.g., portfolio-project).
2. Inside the folder, create three files:
 - o index.html (for your HTML content)
 - o styles.css (for your CSS styling)
 - o script.js (for your JavaScript functionality)

Step 2: Write Your TODO LIST Code

1. Add styling to styles.css and functionality to script.js.

Step 3:

1. Create a GitHub Repository

1. Log in to your [GitHub](#) account.
2. Click on **New Repository**.
3. Provide a repository name and click **Create Repository**.

2. Initialize Git in Your Local Project

1. Open Git Bash and navigate to your project folder:
2. Initialize the folder as a Git repository: git init

3. Link Your Local Repository to GitHub

1. Copy the remote repository URL from GitHub.
-

Step 4: Commit and Push Your Code to GitHub

1. Add files to staging state using `git add .`
2. `git commit -m "Initial commit: Portfolio project"`
3. `git branch -M main`
4. `git push -u origin main`

Step 5: Create and Switch to a New Branch:

1. `git branch branch-name`
2. `git checkout branch-name`

Step 6: Make Changes and Commit:

1. `git add .`
2. `git commit -m "Description of changes"`

Step 7: Merge and Push:

1. `git checkout main`
2. `git merge branch-name`
3. `git branch -d branch-name`
4. `git push origin main`