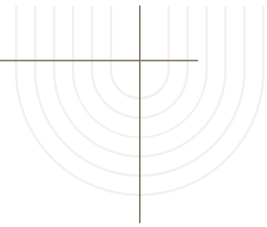




# Report



## Overview of the To-Do List Project

### 1. Project Introduction

This project aims to create a simple web application for managing daily tasks using HTML, CSS, and JavaScript. The application allows users to add and delete tasks upon completion, with design enhancements to ensure a smooth and engaging user experience.

### 2. Project Components

#### A. Frontend (User Interface)

The interface is designed using HTML and CSS, consisting of:

- \* **Input Field:** Allows users to enter new tasks.
- \* **Add Button:** Enables users to add tasks to the list.
- \* **Task List:** Displays all added tasks.
- \* **Remove Button:** Allows users to delete tasks from the list.

CSS is used to style the elements, ensuring a visually appealing design with interactive effects when hovering over buttons.

## **B. JavaScript Functionalities**

The application logic is handled using JavaScript, which includes:

- \* Adding new tasks when the “Add” button is clicked.**
- \* Validating input to prevent empty tasks from being added.**
- \* Creating a new list item containing the task text and a remove button.**
- \* Allowing users to delete tasks by clicking the “Remove” button.**
- \* Marking tasks as completed when clicked, changing their color and applying a strikethrough effect (text-decoration: line-through).**

## **3. Challenges Faced During Development**

While working on the To-Do List project, I encountered a challenge in connecting the code components properly to ensure smooth interaction between them.

To resolve this, I followed these steps:

- 1. Analyzed the code and traced the data flow between different files.**
- 2. Referred to trusted educational sources such as W3Schools to better understand how to link the components.**
- 3. Checked Stack Overflow for similar issues and best practices.**
- 4. Experimented with multiple approaches and debugged errors until I successfully achieved the desired functionality and ensured smooth application performance.**

## **4. Technologies Used**

- \* HTML: For structuring the webpage.**
- \* CSS: For styling and improving user experience.**
- \* JavaScript: For adding interactive features.**

## **5. Conclusion**

**The To-Do List project is a simple yet effective way to practice JavaScript, CSS, and HTML. Despite the challenges encountered during development, continuous research and testing helped resolve issues and improve the application's usability.**

**Future enhancements, such as cloud storage and cross-device synchronization, can further expand its functionality, making it a powerful productivity tool.**

## **Introduction to Git**

**This project uses Git for version control and facilitates collaboration among team members, making it easy to attach projects, manage files, and work together efficiently.**

**A repository for the project has been created on GitHub, allowing all group members to collaborate seamlessly.**

## **Project Branches Created Using Git**

### **1. main (Main Branch)**

- \* Contains the final version of the project after all modifications are completed.**

### **2. report (Report Branch)**

- \* Includes the project report, which contains all important information about the work.**

### **3. presentation (Presentation Branch)**

- \* Dedicated to files related to the presentation, including any Git-related content.**

## **Workflow and Collaboration Using Git**

**Each team member is assigned a specific part of the project.**

**Fatima Salman Yatimi was responsible for:**

- \* Creating the repository on GitHub.**
- \* Setting up branches for each team member to ensure organized workflow.**

## **Git Commands Used**

## Creating and Managing the Repository

1. `dir` → Displays files on the device.
2. `cd` → Navigates to the desired folder.
3. `git init` → Initializes Git in the folder.
4. `touch` → Creates a new file.
5. `git status` → Checks the status of files in the repository.
6. `git add` → Adds files to Git.
7. `git commit -m "message"` → Records changes with a message.
8. `git push origin main` → Pushes files to GitHub.

## Branch Management

9. `git branch` → Lists all branches in the repository.
10. `git checkout -b branch_name` → Creates and switches to a new branch.
11. `git push origin branch_name` → Pushes the new branch to GitHub.

## Problems and Solutions Encountered

1. Problem: Could not find the repository after running `git init`.

✅ Solution: Create a folder on the desktop, then open it using Git Bash for easier access.

**2. Problem: Slow internet affected Git's performance.**

**✓ Solution: Ensure a stable internet connection before running commands that require GitHub access.**

**3. Problem: Error when trying to push files to the repository.**

**✓ Solution: Log in to GitHub and ensure Git is linked to the correct repository.**

## **My Role in the Project**

**✓ Setting up the repository and organizing branches.**

**✓ Ensuring the team follows best practices for Git usage.**

**✓ Assisting in resolving technical issues related to Git.**

## **Resources Used to Learn Git**

**To ensure proper execution of commands and a clear understanding of Git and GitHub, I used several educational resources, including:**

### **1. Dr. Mazen's Lecture**

**\* Explained the fundamentals of Git and its management in software projects.**

### **2. YouTube Course: "Learn Git And GitHub From Zero to Hero in Arabic 2022"**

- \* Provided a comprehensive explanation of Git, from beginner to advanced levels, helping me understand repository and branch management practically.

### 3. YouTube Video: “Easily Upload a Project to GitHub Using Git”

- \* Helped me apply the steps for uploading a project to GitHub and troubleshoot common errors.

#### Task Distribution Among Team Members

- \* Report: [4551744 بيان الرفاعي]
- \* Git Commands: [4553327 فاطمه يتيهي]
- \* Presentation: [4550792 هديل الجهنني]
- \* Project Development: [4559142 منار العنزي]

