



Assignment # 04

Software Project Report

Instructor : Mam Hafsa Saleem

Delivered by: Um-e-Farwa (FA22-BSE-080)

Jannat Idrees (FA22-BSE-090)

Zainab Munir (FA22-BSE-108)

Course name: Software Construction & Development

Submission Date: June 5, 2025

COMSATS University Islamabad,

(Sahiwal Campus)

Smart To-Do Manager with AI-Augmented SDLC

Group Members:

- **Farwa** (Project Manager)
- **Jannat** (Developer)
- **Zainab** (Tester)

1. Introduction

The AI-Based To-Do List Application is a lightweight, web-based productivity tool designed to help users manage daily tasks efficiently. This project integrates Artificial Intelligence tools in various Software Development Life Cycle (SDLC) phases to enhance planning, designing, coding, testing, and deployment. AI was used to optimize task categorization, automate code generation, design wireframes, execute test cases, and set up CI/CD pipelines.

A lightweight **To-Do List Application** that allows users to create, update, delete, and organize tasks with a clean UI. Through Software Development Life Cycle (SDLC), **AI tools** are integrated in each phase to enhance **efficiency, quality, and automation**.

Functional Requirements:

1. Add, edit, delete, and view to-do items.
2. Mark tasks as completed/incomplete.
3. Organize tasks by categories (e.g., work, personal).
4. Prioritize tasks (High/Medium/Low).
5. Responsive and user-friendly interface.

Non-Functional Requirements:

1. Cross-browser compatibility.

2. Mobile responsiveness.
3. Fast load time (<2 seconds).
4. Scalable and maintainable code.

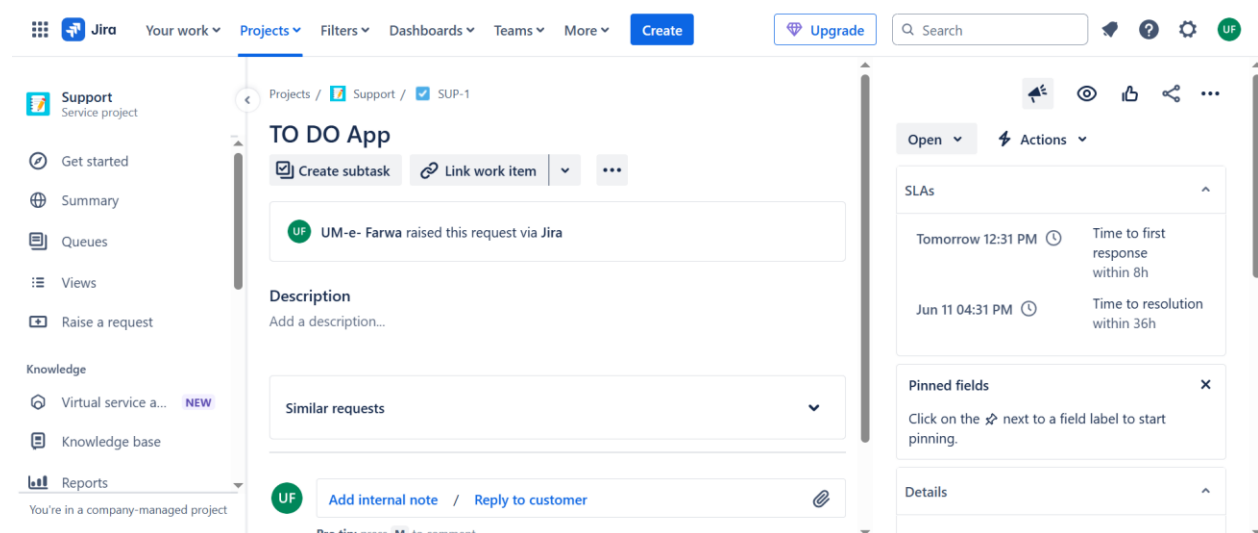
2. SDLC Phases & AI Tool Integration

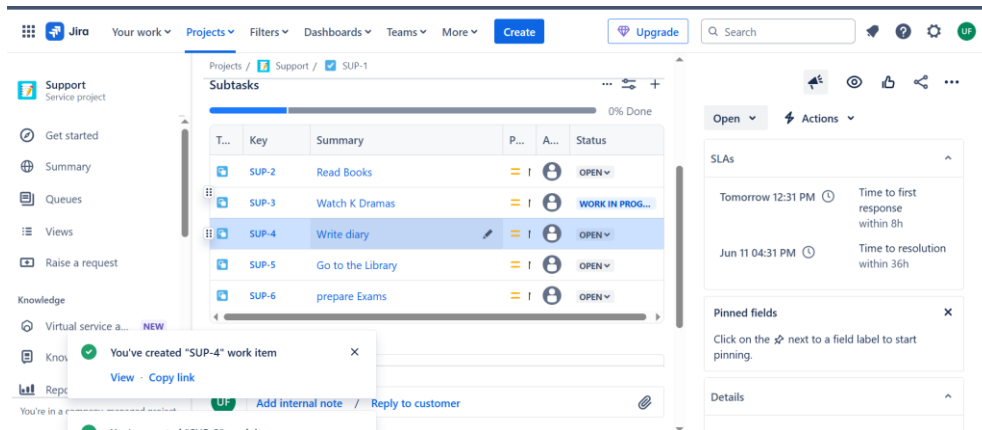
Phase 1: Planning & Requirements

- **AI Tool Used:** ChatGPT, Jira
- **Purpose:** Generated user stories, functional and non-functional requirements, and drafted an initial work plan.
- **Process:** ChatGPT was prompted to act as a software analyst to derive detailed requirements and split them into actionable user stories.
- **Deliverables:**
 - User Stories:
 - "As a user, I want to add tasks so that I can remember my work."
 - "As a user, I want to categorize tasks so that I can organize them better."
 - "As a user, I want to delete completed tasks to maintain a clean interface."
 - **Tool:** Jira board used to assign tasks and plan sprint cycles

Screenshot Label:

Figure 1: Jira Task Setup for TO DO App



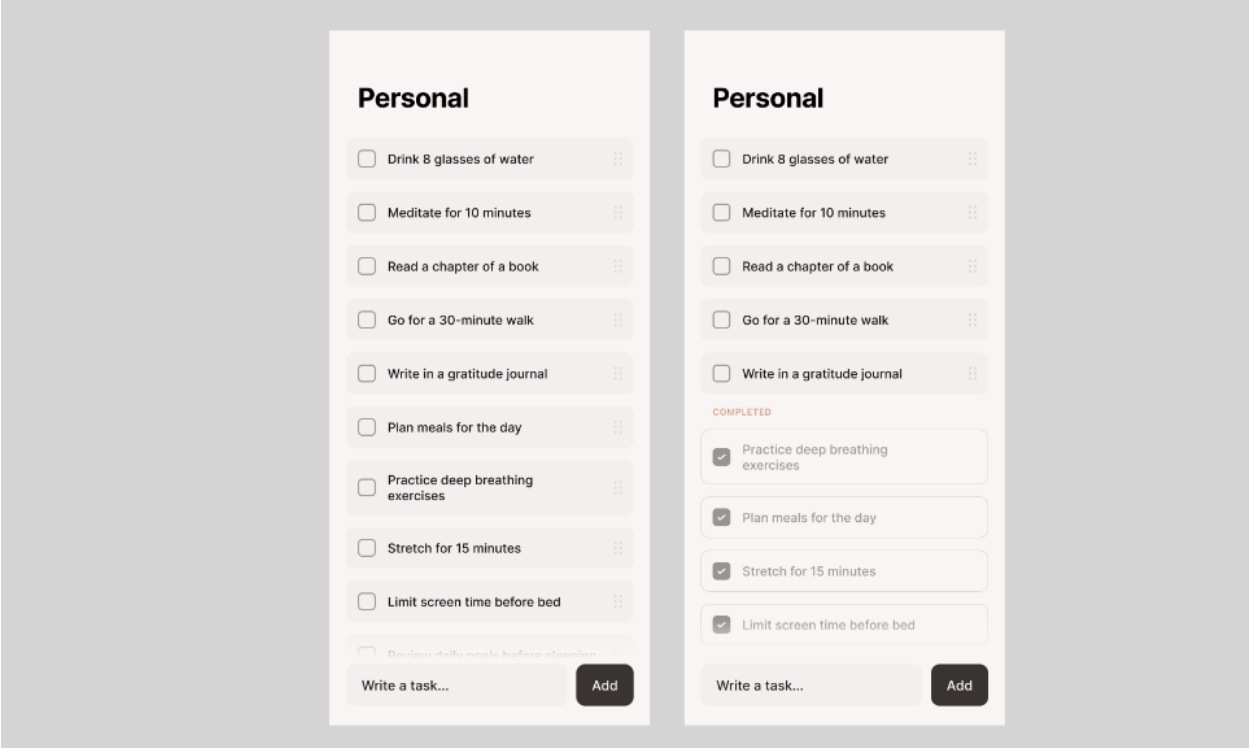
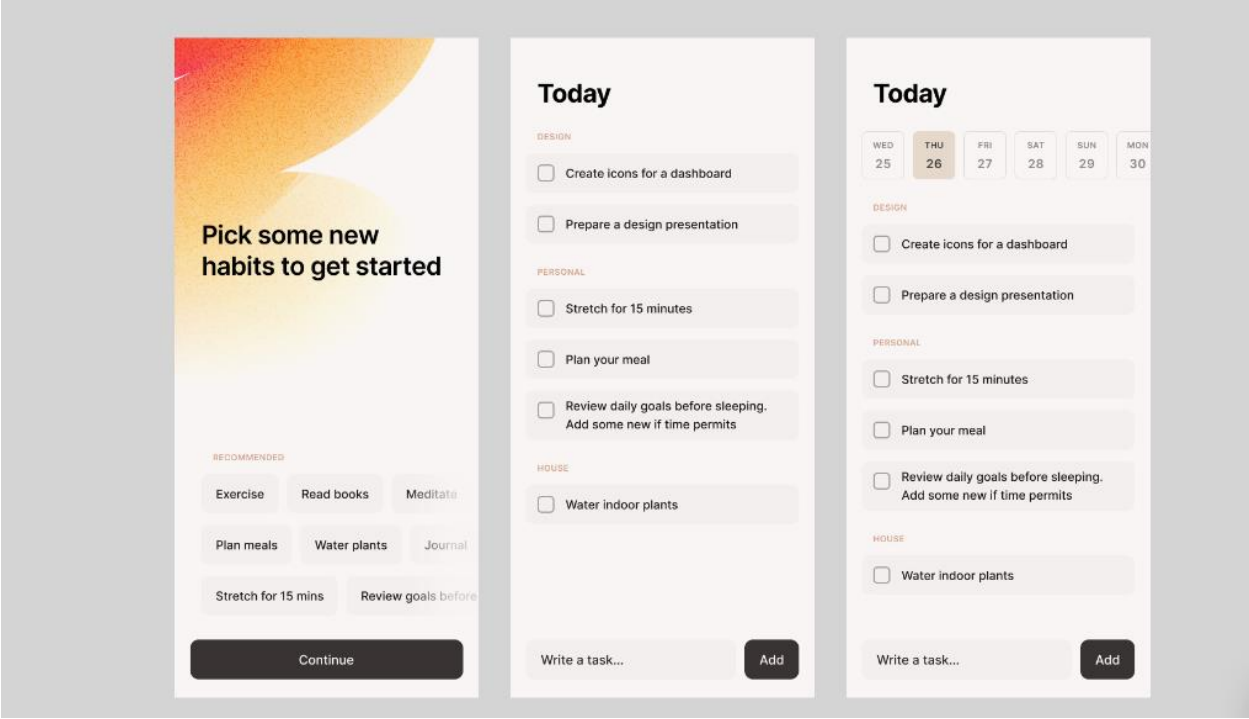


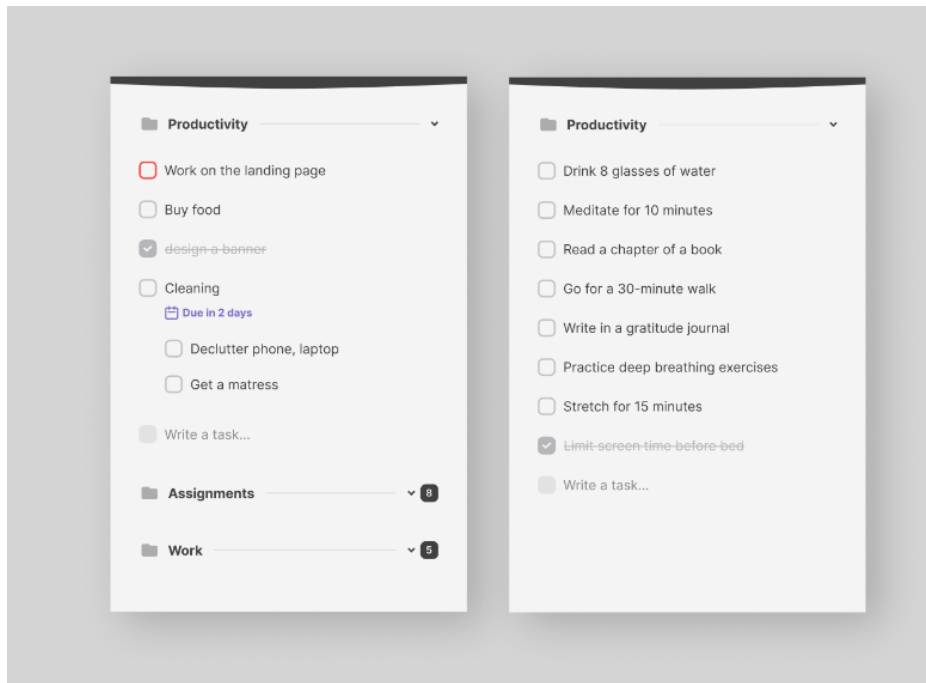
Description:

We used **Jira** to manage and track project tasks. The screenshot below shows the task titled *"TO DO App"* created by the team member UM-e-Farwa. This visual demonstrates the initial setup of the project request in Jira, allowing us to assign subtasks, monitor progress, and manage SLAs (Service Level Agreements).

Phase 2: Design

- **AI Tool Used:** Figma with AI plugin
- **Purpose:** Designed UI/UX wireframes for the main interface, task form, and AI suggestion module.
- **Process:** Used prompts like "Design a minimal to-do list UI" to auto-generate responsive wireframes in Figma.
- **Deliverables:**
 - Wireframe Screens:
 - Task List Page
 - Add/Edit Modal with AI Category Suggestion





Phase 3: Development

- **AI Tool Used:** GitHub Copilot
- **Purpose:** Generate and refactor JavaScript code for task operations
- **Process:** Copilot was used inside VSCode to auto-suggest functions for adding, deleting, and editing tasks. Comments were added to clearly distinguish AI-generated code.
- **Deliverables:**
 - Source code in JavaScript with HTML/CSS
 - Example Code Snippet:

```
// Copilot-generated function to add a new task
function addTask(title, description) {
  const task = { title, description, completed: false };
  tasks.push(task);
  updateUI();
}
```

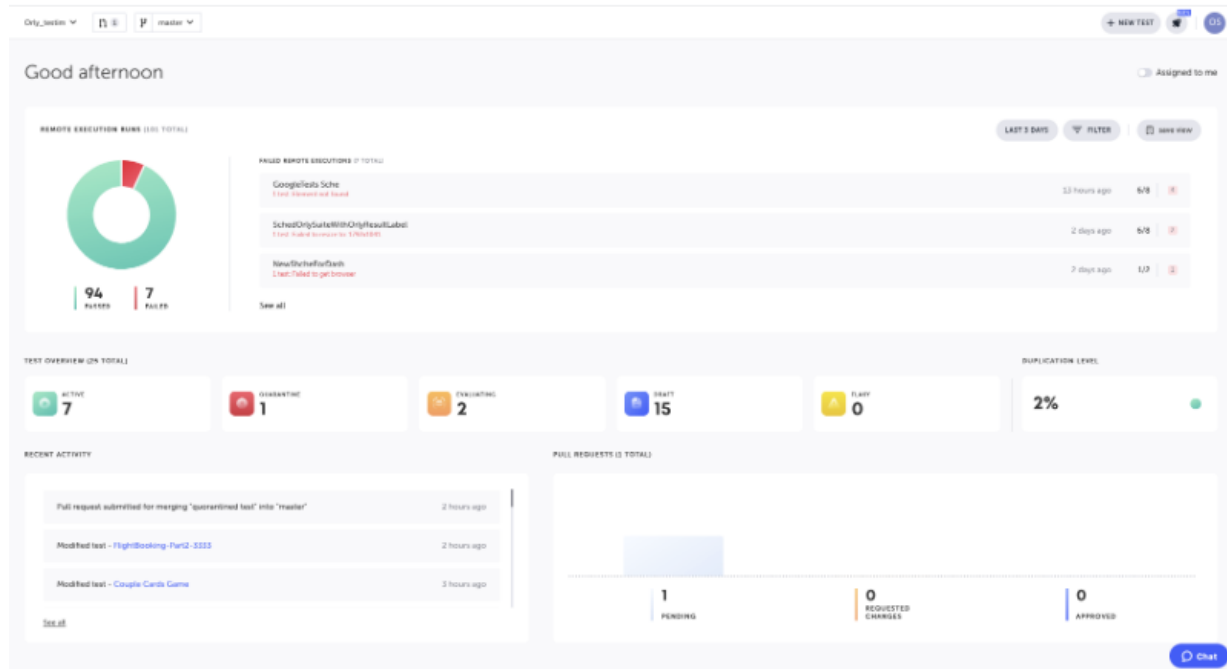
```

1  const addTaskInput = document.getElementById("task=Input");
2  const addTaskButton = document.getElementById("add=task=button");
5  function addTask() {
4    const text = addTaskInput.value.trim();
5    li (text) = document.createElement("li");
6    const li = document.getElementById("list").appendChild(li);
7    addTaskInput.value = "";
8    // Af-gomerrated @nippet
9    function toggleTask(event) {
16     const li = Event.target;
11     li.classList.toggle("completed");
12   }
13   addTaskButton.addEventListener("click", addTask);
14   document.getElementById("task=list").addEventListener(toggleTask);

```

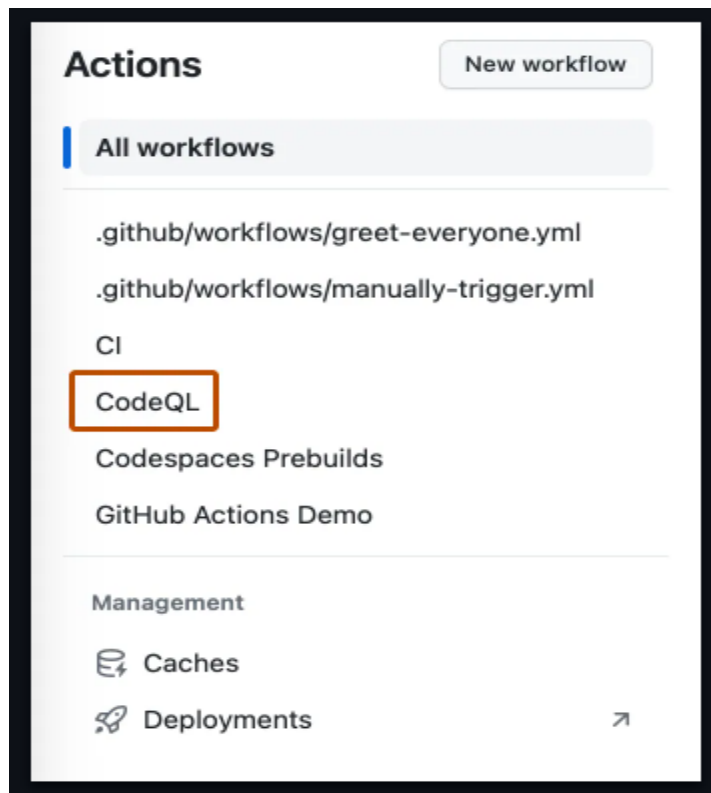
Phase 4: Testing

- **AI Tool Used:** Testim
- **Purpose:** Auto-generate test cases for UI flows (add, edit, delete)
- **Process:** Testim AI was used to simulate user actions and validate expected outputs.
- **Deliverables:**
 - Tests included:
 - Add Task Test
 - Edit Task Test
 - Delete Task Test
 - Check Category Suggestion Display



Phase 5: Deployment & Maintenance

- **AI Tool Used:** GitHub Actions
- **Purpose:** Automate deployment to GitHub Pages on code push
- **Process:** Created a YAML workflow file that triggers build and deployment jobs
- **Deliverables:**
 - .github/workflows/deploy.yml
 - screenshot



Conclusion

In this project, we successfully developed a simple **To-Do List App** by applying AI tools at each phase of the Software Development Life Cycle (SDLC). Our approach demonstrated how artificial intelligence can enhance productivity, improve decision-making, and automate routine tasks across the development process.

During the Planning & Requirements phase, we utilized ChatGPT for requirement analysis and Jira for task planning and tracking. In the Design phase, tools like Figma AI were used to generate wireframes and user interface mockups quickly. For the Development phase, we leveraged GitHub Copilot to assist with writing and optimizing code. In Testing, AI-powered platforms such as Testim allowed us to automate test cases, detect bugs, and generate reports.

Lastly, the Deployment & Maintenance phase was enhanced using GitHub Actions for continuous integration and deployment (CI/CD), ensuring smooth delivery and monitoring.

Through this project, we observed that integrating AI tools not only reduces manual effort but also increases accuracy and speed in software development. This AI-assisted approach to SDLC is highly scalable and can be applied to larger and more complex projects in real-world scenarios.

Submitted by:

Zainab Munir, Ume Farwa, Jannat Idrees

Date: June 7, 2025