# Nexventures Ltd. - Software Engineering Internship Task

**Subject: Take-Home Technical Assessment**

This task is designed to evaluate your technical skills, problem-solving approach, and attention to detail. We do not need you as those who can only write code but also understand user needs and translate them into a functional, clean, and well-structured application.

### The Project: "Project Pulse" - A Simple Project Management Dashboard

Imagine that Nexventures Ltd. needs a simple internal tool for small teams to track their projects. Your task is to build a lightweight web application called "Project Pulse" with the following structure:

1. A Public-Facing Landing Page: This is the page a user would see first.
2. A "Features" Page: A page detailing the application's main features.
3. A User Dashboard: The main application interface after a user logs in (authentication can be simulated).

## Core Requirements

### 1. Landing Page (/ or index.html)

This page should be designed to attract and inform potential users.

- Hero Section: A compelling headline and sub-headline introducing "Project Pulse."
- Call-to-Action (CTA): A prominent button that says "Go to Dashboard" (which will lead to the dashboard page).
- Brief Value Proposition: 2-3 short points or icons explaining why someone should use the app (e.g., "Track Progress," "Manage Tasks," "Collaborate Easily").

### 2. Features Page (/features)

This page provides more detail on what the application does.

- It should list at least three main features of the "Project Pulse" dashboard.
- Each feature should have a title, a short description, and a relevant icon or placeholder image.
- This page should be accessible from the navigation menu on the Landing Page.

### 3. User Dashboard (/dashboard)

This is the core of the application. It should include the following key features:

- Project Overview: A section at the top welcoming the user (e.g., "Welcome, [User]!") and showing a summary stat, like "Total Projects: 3".
- Project List: A main area that displays a list of projects.
  - Each project card/row should show: Project Name, Description, Status (e.g., Not Started, In Progress, Completed).
- "Add New Project" Functionality: A button that opens a form (a modal or a new section) to add a new project. The form should have fields for:
  - Project Name (required)
  - Project Description
  - Status (dropdown selection)
  - Clicking "Save" should add the new project to the Project List dynamically on the frontend (no backend persistence required).
- Status Update: A way to change the status of an existing project (e.g., a dropdown in each project card).
- Navigation: A consistent navigation bar across all pages with links to "Home," "Features," and "Dashboard."

## Technical Guidelines & What We're Looking For

- Preferable Technology Stack: We primarily use the following in our projects:
    - Frontend: React.js (with functional components & hooks) or Vue.js
    - Styling: CSS3, Sass, or a modern CSS-in-JS library. Using a component library like Material-UI, Chakra UI, or Ant Design is a plus.
    - "Backend" Logic: Since this is a frontend-focused task, you can use a simple state management solution (like React's useState, Context API, or Vue's Composition API) to manage the project data. No actual server is needed; data can be lost on refresh.
- However, you are free to use any language or framework you are most comfortable with (e.g., Angular, Svelte, vanilla JavaScript, etc.). The quality of your code and the final output are more important than the specific tool.
- Code Quality:
    - Clean, readable, and well-structured code.
    - Consistent naming conventions.
    - Proper separation of concerns (e.g., component structure, styling files).
    - Thoughtful user experience (UX) and a clean, modern user interface (UI).
- Bonus Points (Optional but Impressive):
    - Make the project data persist in the browser's localStorage.
    - Add the ability to edit or delete a project.
    - Make the dashboard responsive (mobile-friendly).
    - Add simple client-side form validation for the "Add Project" form.
    - Write a few simple unit tests for a core function or component.
    - Deploy the project to a free platform (e.g., Vercel, Netlify, GitHub Pages) and share the live URL.

## Deliverables

Please provide the following within 7 days of receiving this task:

1. Source Code: A link to a Git repository (e.g., GitHub, GitLab) with your complete code. Please include a README.md file with:
    - Project description.
    - Instructions on how to build and run the project locally.
    - A brief explanation of your technical choices and any assumptions you made.
2. Live Demo (Optional but Highly Recommended): A publicly accessible URL where we can interact with your application.

We are excited to see what you build! This is your chance to showcase your skills and creativity. Good luck!

**The Nexventures Ltd. Team**