Creating And Generating Components

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1 Project Abstract

In modern web development, building modular, scalable, and reusable components is essential for creating maintainable applications. This assignment focuses on designing a simple **To-Do List Application** using Angular. The goal is to understand and implement component-based architecture by dividing the application into meaningful, reusable parts.

The objective is to create a **To-Do List Single Page Application (SPA)** using Angular, structured with multiple components. The application must display a header and a list of default tasks, allowing users to mark tasks as complete or incomplete using checkboxes. The interaction between components should follow Angular standards, utilizing property and event binding where necessary.

2 Problem Statement

You are tasked with building a **To-Do List Single Page Application (SPA)** using Angular. The application should consist of multiple components to:

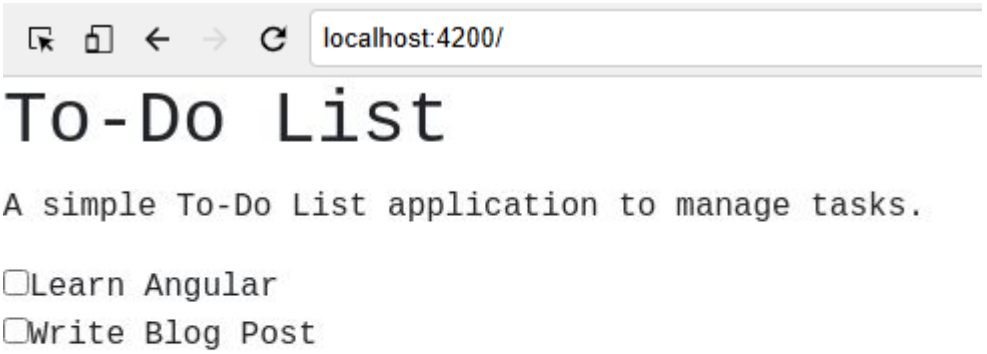
- Display a header and subheader.
- Render a list of to-do tasks.
- Allow users to mark tasks as completed/incomplete.

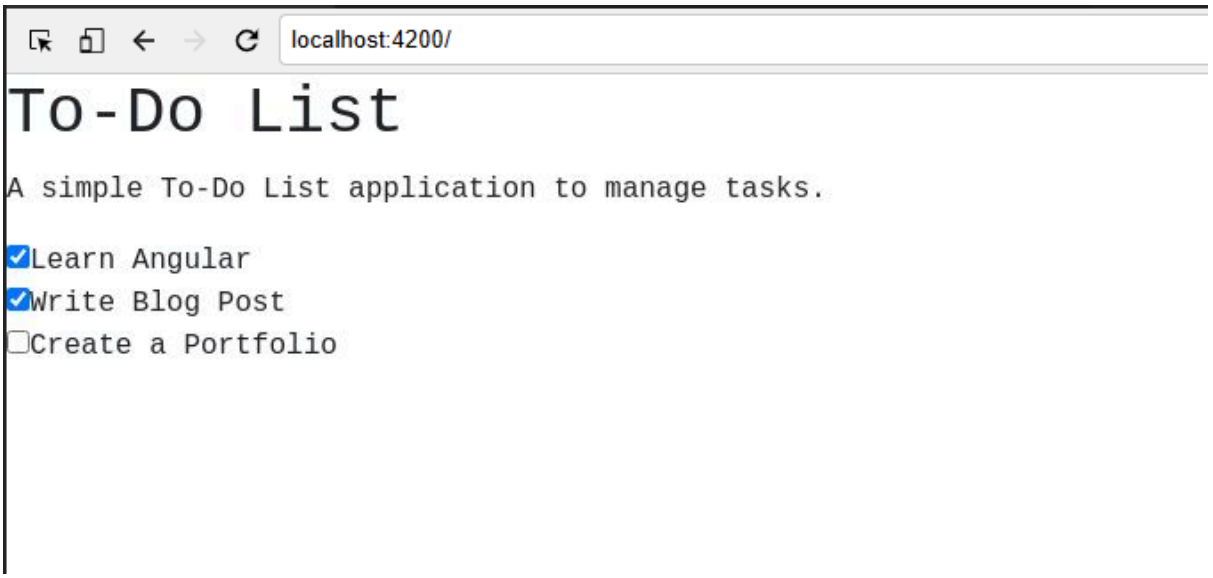
The focus is to split the application logically using components and ensure proper data binding and event handling.

3 Proposed Creating And Generating Components Application Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed.

3.1 ScreenShots





4 BUSINESS-REQUIREMENT:

As an application developer, develop the Creating And Generating Components (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as the default page. Acceptance criteria: **HeaderComponent:** 1. Should have "To-Do List" as heading in **h1**.

		2. Should display the subheading "A simple To-Do List application to manage tasks." in **p** tag. **ToDoListComponent:** 3. Should list the below default tasks (with fields name and completed) each with an empty checkboxes before it: ● Learn Angular ● Write Blog Post ● Create a Portfolio 4. Maintain the default tasks array and pass the task data to TodoItemComponent to display tasks with checkboxes. **ToDoItemComponent:** 5. Use @Input() to receive task details from ToDoListComponent. 6. Users should be able to mark tasks as completed/incomplete by interacting with checkboxes. 7. Application should dynamically update the status of tasks based on checkbox selection. **** Kindly refer to the screenshots for any clarifications. ****

5 CONSTRAINTS

1. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.

6 MANDATORY ASSESSMENT GUIDELINES

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**
   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. **This editor Auto Saves the code.**

4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**

   Note: The application will not run in the local browser
7. **You can follow series of command to setup Angular environment once you are in your project-name folder:**
   a. **npm install --no-bin-links --unsafe-perm -> Will install all dependencies -> takes 10 to 15 min.**
   b. **npm run start -> To compile and deploy the project in browser. You can press the <Ctrl> key while clicking on localhost:4200 to open the project in the browser -> takes 2 to 3 min.**
   c. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min.**
8. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.**
9. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**