**Build a Modular Dashboard Application in Angular 20**

You are tasked with building a **modular dashboard application** for a fictional company called **TaskFlow Inc.** using **Angular 20**. The application should display a list of projects, allow users to view project details, and include a reusable sidebar component. The goal is to demonstrate your understanding of Angular bootstrapping, component architecture, data binding, and configuration.

**🔧 Requirements**

**1. Bootstrapping & Application Entry**

* Identify and explain the **entry point** of the Angular application.
* Configure the **main.ts** file to bootstrap the application using the **AppModule**.
* Describe the role of **Webpack** in the build process and how it interacts with the entry point.

**2. Project Configuration**

* Modify the **angular.json** file to:
  + Add a custom style.
  + Include a new asset folder for icons.
  + Set up a new build configuration for production.

**3. Root Module & AppComponent**

* Create the **AppModule** as the **root module**.
* Define the **AppComponent** as the root component.
* Display a welcome message using **interpolation** and **property binding**.

**4. Component Architecture**

* Create the following components using Angular CLI:
  + ProjectListComponent – displays a list of projects.
  + ProjectDetailComponent – shows details of a selected project.
  + SidebarComponent – reusable sidebar with navigation links.
* Register all components in the appropriate module.

**5. Component Communication**

* Use **Input** and **Output** decorators to pass data between ProjectListComponent and ProjectDetailComponent.
* Display SidebarComponent as a **child component** in AppComponent.

**6. Multiple Root Components (Advanced)**

* Add a second root component called AdminComponent and configure it to bootstrap separately for a different entry point (e.g., admin-main.ts).
* Explain how Angular supports **multiple root components** and how Webpack handles this.

**7. Data Binding**

* Demonstrate the use of:
  + **Interpolation** to display project names.
  + **Property binding** to dynamically set image URLs.
  + **Event binding** to handle click events on project items.
  + **Two-way binding** for a search input field.

**📦 Deliverables**

* Source code with proper folder structure.
* angular.json with custom configurations.
* Screenshots or screen recordings showing:
  + Bootstrapping process.
  + Component interaction.
  + Data binding in action.

**🧠 Evaluation Criteria**

* Correct use of Angular bootstrapping and configuration.
* Clean and modular component design.
* Proper use of Angular data binding techniques.
* Clear understanding of Angular’s build and entry system.