# **Angular CRUD Operations Project Documentation**

## **Project Overview**

The project is a **CRUD (Create, Read, Update, Delete)** application for managing employees with their respective roles. The application is divided into three main folders:

1. **Core Folder**
2. **Shared Module**
3. **Modules Folder**

Each folder has a specific purpose and contains components, services, and utilities that are essential for the application's functionality.  
  
**Admin users:** to have an access to display/delete/add/edit users data.  
**Basic users:** to have an access to display/add only.  
  
  
***P.S.: I’m using the localstorage to add and display the users data***

## **1. Core Folder**

The **Core Folder** contains the foundational elements of the application, such as API methods, generic functions, base URLs, entities, guards, and interfaces.

### **Key Components:**

#### **a) API Methods**

* Contains generic HTTP methods (GET, POST, PUT, DELETE) to interact with the backend.

#### **b) Generic Functions**

* Contains reusable utility functions, such as error handling, logging, and data transformation.

#### **c) Base URL**

* Stores the base URL for API endpoints.

#### **d) Entities**

* Defines the structure of the data models (e.g., User).

#### **e) Guards**

* Contains route guards to protect routes (e.g., authentication guard).

#### **f) Interfaces**

* Contains interfaces for shared data structures, such as language settings for Arabic/English content.

## **2. Shared Module**

The **Shared Module** contains components, services, and utilities that are used across multiple modules in the application.

### **Key Components:**

#### **a) Navbar**

* A reusable navigation bar component.

#### **b) Footer**

* A reusable footer component.

#### **c) Shared Buttons**

* Reusable buttons with consistent styling and functionality.

#### **d) Shared Services**

* Services that provide shared functionality, such as user authentication or language switching.

## **3. Modules Folder**

The **Modules Folder** contains feature modules for different parts of the application. Each module has its own routing and components.

### **Key Modules:**

#### **a) Login Module**

* Handles user authentication.
* Contains:
  + Login component (form for username/password).
  + Routing for the login page.

#### **b) Home Module**

* Displays the main dashboard or homepage.
* Contains:
  + Home component (displays welcome message or summary).
  + Routing for the home page.

#### **c) Task Module**

* Manages CRUD operations for employees.
* Contains:
  + **Add New User Component**: A form to add a new employee.
  + **All Users Component**: Displays a list of all employees with options to edit or delete.
  + **Edit User Component**: A popup opens with a form to edit an existing employee's details.
  + **Dialog Component**: A pop-up dialog used for editing users. This component does not have its own route and is opened as a pop-up.

## **Workflow**

### **1. User Authentication**

* Users log in through the **Login Module**.
* Upon successful login, they are redirected to the **Home Module**.

### **2. Employee Management**

* From the **Home Module**, users can navigate to the **Task Module** to manage employees.
* In the **Task Module**:
  + Users can view a list of all employees in the **All Users Component**.
  + Users can add a new employee using the **Add New User Component** (opened as a dialog).
  + Users can edit an existing employee's details using the **Edit User Component** (opened as a dialog).
  + Users can delete an employee directly from the **All Users Component**.

### **3. Shared Functionality**

* The **Shared Module** provides reusable components like the **Navbar** and **Footer**, which are used across all modules.
* Shared services, such as authentication and language switching, are also provided by the **Shared Module**.