NAME: **VIJAY RAJMOD**

Design and implement a web application using the Angular or React framework

That allows users to view a list of profiles and interactively explore the addresses

Of each profile on a map. The application aims to provide an intuitive and

User-friendly way to navigate through profiles and visualise the geographic

Locations associated with each individual.

The key functionalities of the application include:

1. Profile Display: Create a webpage that presents a collection of profiles,

Each comprising essential information such as the person’s name,

Photograph, and a brief description.

1. Interactive Mapping: Incorporate an interactive map component that can

Dynamically display addresses based on user interactions. This map will

Allow users to see the geographical location associated with each profile.

1. Summary Integration: Implement a “Summary” button adjacent to each

Profile. Clicking this button should trigger the display of the map

Component with a marker indicating the precise address of the selected

Profile.

1. (Good to have ) Map Services Integration: Utilise external map services like

Google Maps or Mapbox to integrate the mapping functionality into the

Application. This entails setting up markers and correctly rendering

Addresses on the map.

1. User-Friendly Experience: Ensure that the application offers a smooth and

Intuitive user experience, enabling users to easily navigate profiles and

Access mapped addresses without confusion.

1. Profile Data Management : Allow administrators to add, edit, or delete

Profiles.

1. This will require an admin panel or dashboard to manage the profile data

Efficiently.

1. Search and Filter Functionality : Provide users with the ability to search and

Filter profiles based on different criteria, such as name, location, or other

Attributes. This enhances the usability of the application.

1. Responsive Design : Ensure that the application is responsive and

Mobile-friendly so that users can access it from various devices, including

Smartphones and tablets.10. Error Handling Implement robust error handling and validation

Mechanisms to handle issues gracefully, such as invalid addresses or

Failed map service requests.

1. Loading Indicators : Include loading indicators of progress bars to give

Users feedback when the application is fetching data or rendering the map.

1. Profile Details: Create a separate profile details view that provides more

In-depth information about each profile when a user clicks on a profile

Card. This can include additional details like contact information, interests,

Etc.

By considering these additional points, you can create a more robust and

User-friendly web application that meets the needs of both users and

Administrators while ensuring its long-term viability and success.

The successful completion of this project will result in a fully functional web

Application that demonstrates proficiency in utilising Angular’s capabilities to

Create an engaging interface for profile viewing and location mapping. The

Application will offer an innovative solution for users interested in exploring the

Geographic distribution of profiles, enhancing user engagement and interaction.

**HTML**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset=”UTF-8” />**

**<title>Profile Viewer</title>**

**<link rel=”stylesheet” href=”styles.css” />**

**</head>**

**<body>**

**<div id=”app”>**

**<h1>Profiles</h1>**

**<div id=”profiles”>**

**<div class=”profile” \*ngFor=”let profile of profiles”>**

**<img [src]=”profile.photo” alt=”{{ profile.name }}” />**

**<h2>{{ profile.name }}</h2>**

**<p>{{ profile.description }}</p>**

**<button (click)=”showSummary(profile.address)”>Summary</button>**

**</div>**

**</div>**

**<div id=”map” \*ngIf=”selectedAddress”></div>**

**</div>**

**<script src=”main.js”></script>**

**</body>**

**</html>**

**CSS**

**Body {**

**Font-family: Arial, sans-serif;**

**Margin: 0;**

**Padding: 0;**

**}**

**#app {**

**Display: flex;**

**Flex-direction: column;**

**Align-items: center;**

**Padding: 20px;**

**}**

**#profiles {**

**Display: flex;**

**Flex-wrap: wrap;**

**Justify-content: center;**

**Margin-top: 20px;**

**}**

**.profile {**

**Width: 300px;**

**Margin: 10px;**

**Box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);**

**Text-align: center;**

**}**

**.profile img {**

**Width: 100%;**

**Height: 200px;**

**Object-fit: cover;**

**}**

**#map {**

**Width: 100%;**

**Height: 500px;**

**Border: 1px solid #ccc;**

**Margin-top: 20px;**

**}**

**Js**

**Const profiles = [**

**{**

**Name: “vijay rajmod”,**

**Photo:** [**https://via.placeholder.com/150**](https://via.placeholder.com/150)**,**

**Description: “vijay is a student.”,**

**Address: “dehu phata alandi ”**

**},**

**// More profiles…**

**];**

**Let selectedAddress = null;**

**Function showSummary(address) {**

**// Implement address parsing and geocoding logic here**

**selectedAddress = address;**

**}**

**// Render profiles**

**Const profilesDiv = document.getElementById(“profiles”);**

**Profiles.forEach(profile => {**

**Const profileDiv = document.createElement(“div”);**

**profileDiv.classList.add(“profile”);**

**profileDiv.innerHTML = `**

**<img src=”${profile.photo}” alt=”${profile.name}” />**

**<h2>${profile.name}</h2>**

**<p>${profile.description}</p>**

**<button onclick=”showSummary(‘${profile.address}’)”>Summary</button>**

**`;**

**profilesDiv.appendChild(profileDiv);**

**});**

**// Initialize map**

**// You can use the Google Maps or Mapbox API to render the map and marker**