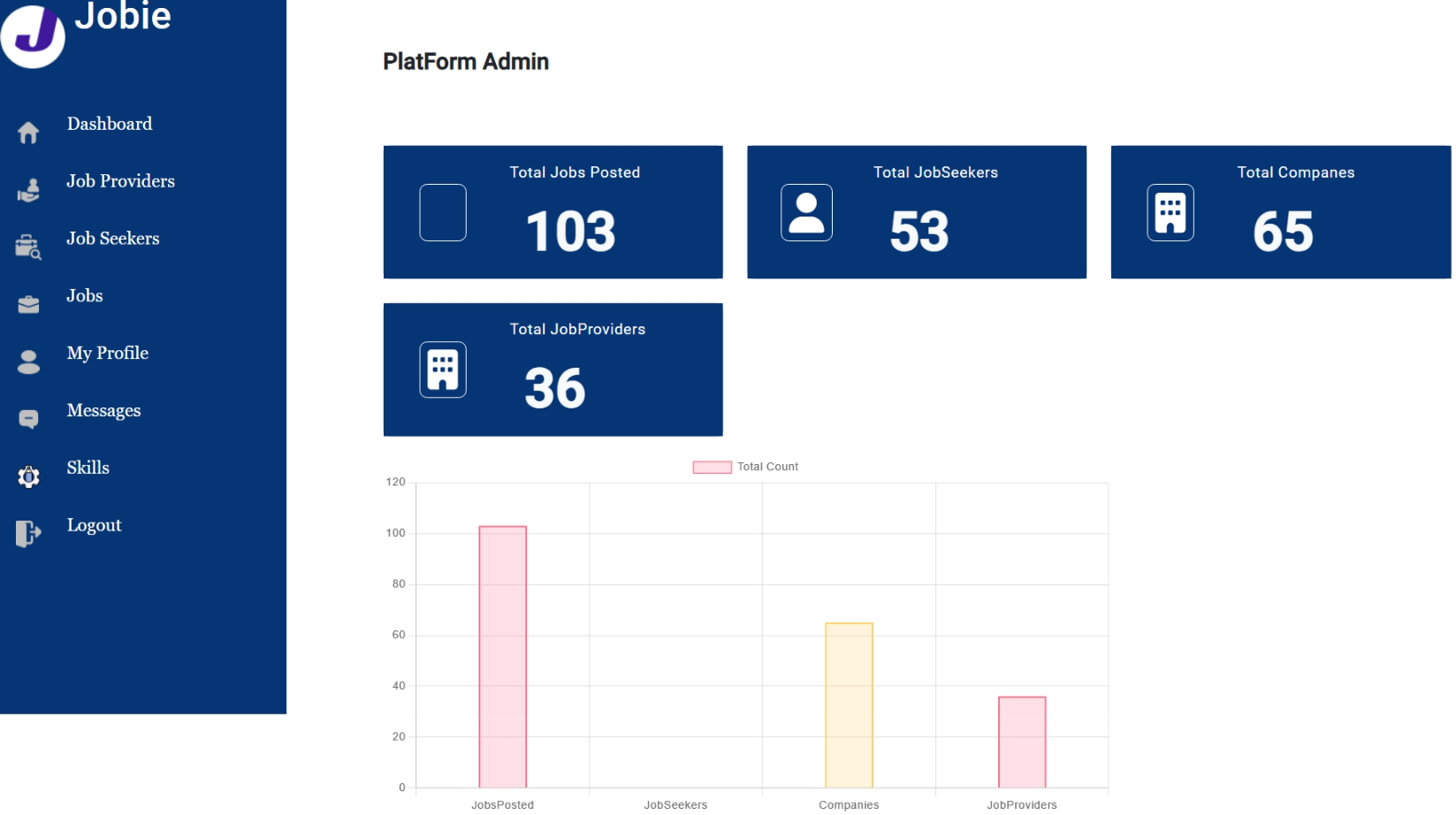
**HTTP – CRUD Operations**

**Exercise: 1**

**Platform Admin Module: Dashboard**

****

**Objective**:

You are tasked with creating a dashboard for an administration panel in an Angular application. The dashboard should display various counts related to job postings, job seekers, companies, and job providers. To achieve this, you have already created a service class named `dashboardService` that fetches these counts from the backend using HTTP requests. The service class has methods such as `getTotaljobsPosted()`, `getTotaljobSeekers()`, `getTotalcompanies()`, and `getTotalJobProvider()`.

**Task:**

* Now, your task is to use JSON Server to simulate a backend API and implement the functionality to display these counts on cards in your Angular application. Additionally, you need to create an Angular chart to visually represent these counts.

1. Setting up JSON Server to serve mock data for the backend API.

2. Modifying the Angular service class (`dashboardService`) to fetch data from the JSON Server.

3. Creating components to display the counts on cards.

4. Implementing an Angular chart to visualize the data fetched from the backend.

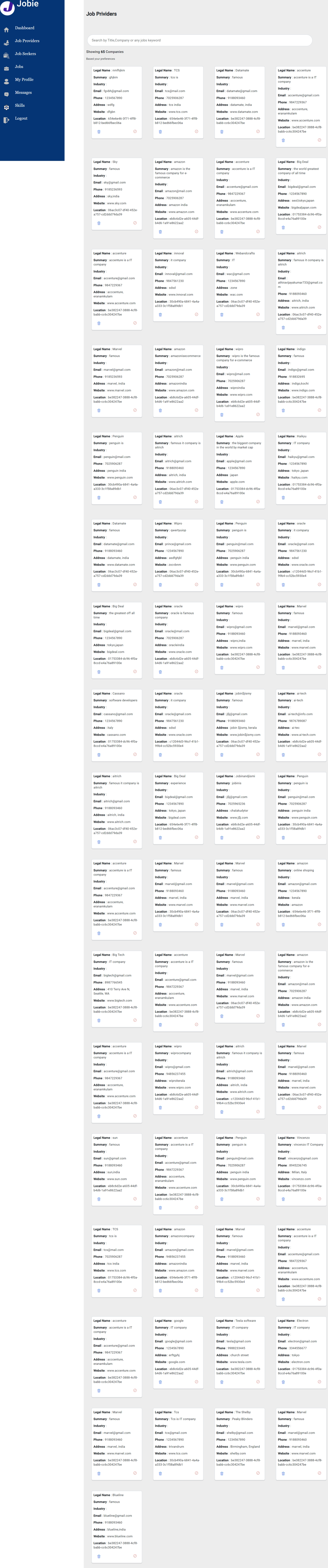
**Instructions**

Please provide step-by-step instructions along with code examples for each part of the solution. Additionally, explain any dependencies or configurations required to ensure the successful implementation of the dashboard.

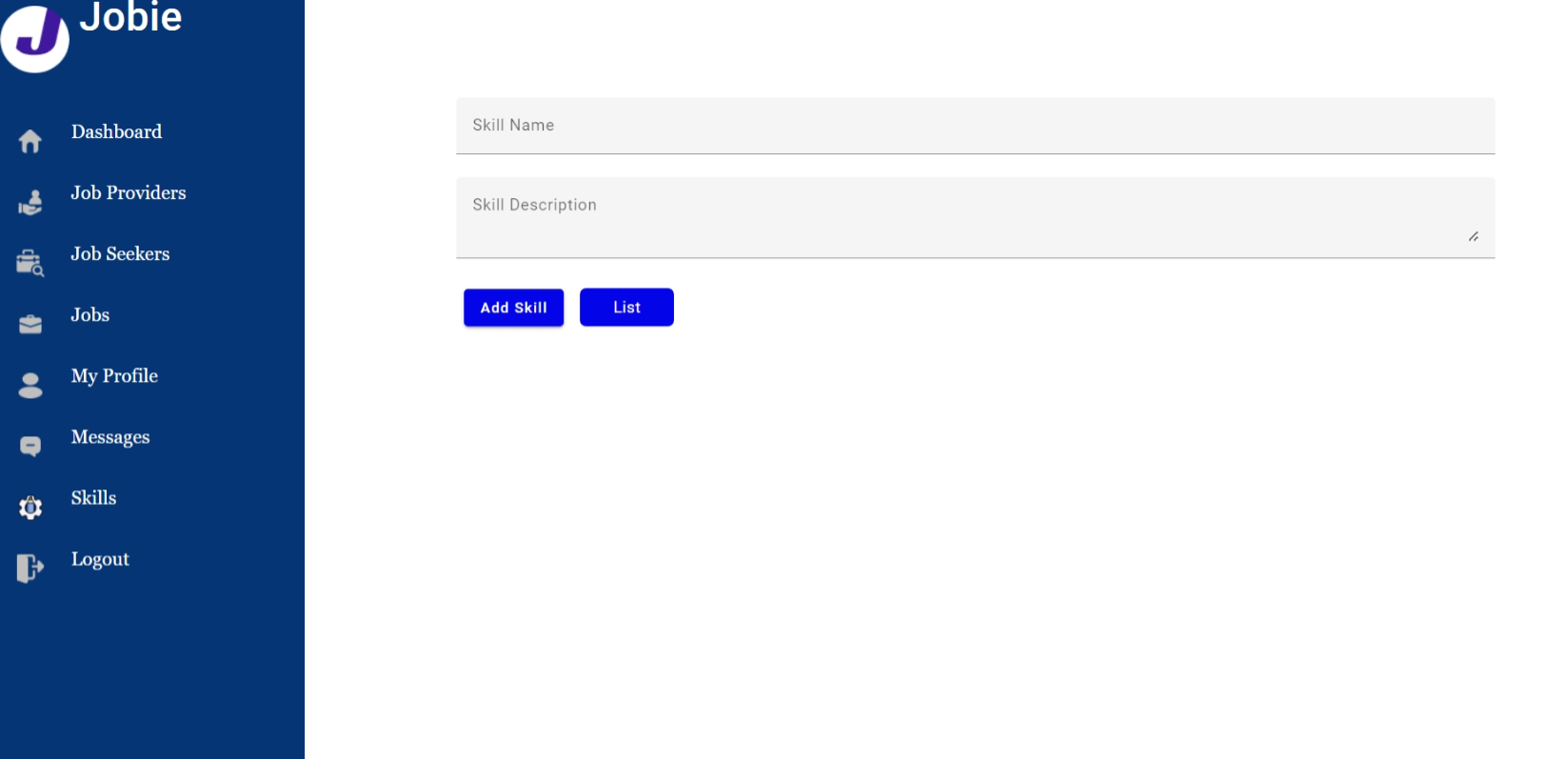
* Installation and setup of JSON Server.
* Configuration of JSON Server to serve mock data.
* Modification of the Angular service class to fetch data from JSON Server.
* Creation of Angular components to display card elements.
* Implementation of an Angular chart library (e.g., Chart.js) to visualize the data.
* Integrating the chart component into your Angular application and populating it with the fetched data.
* Feel free to use any additional Angular libraries or tools that you find suitable for achieving this task.

**Exercise: 2**

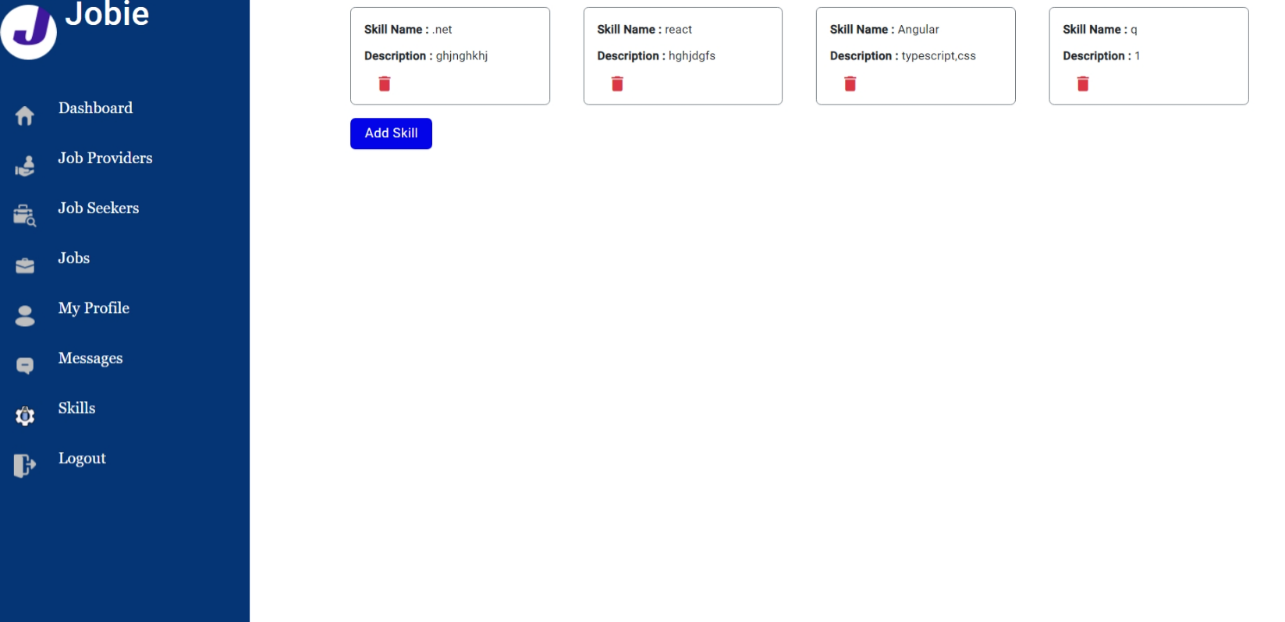
**Platform Admin Module: Job Providers list**

****

**Add Skill**

****

**List Skill**

****

**Job Providers list Management:**

Implement functionality to list job providers in the admin panel for the job provider module. You have a service class named jobService which fetches job providers' data from the backend API using HTTP requests. The service class has methods such as getCompanies() to fetch job providers and removeCompany(id: string) to delete a job provider by ID. You need to integrate this service with JSON Server to simulate a backend API.

**Skills Administration:**

Implement functionality to manage skills for job seekers. This includes listing skills, adding new skills, and deleting existing skills. The admin should be able to set job seekers' skill sets populated with available skills. You need to create a service class for managing skills and integrate it with JSON Server.

You are tasked with implementing a skills management feature in an Angular application. You have a service class named SkillService which interacts with the backend API to perform CRUD operations on skills. The service class provides methods such as getSkills() to fetch existing skills, addSkill(skill: Skill) to add a new skill, and deleteSkill(skillId: string) to remove a skill by its ID.

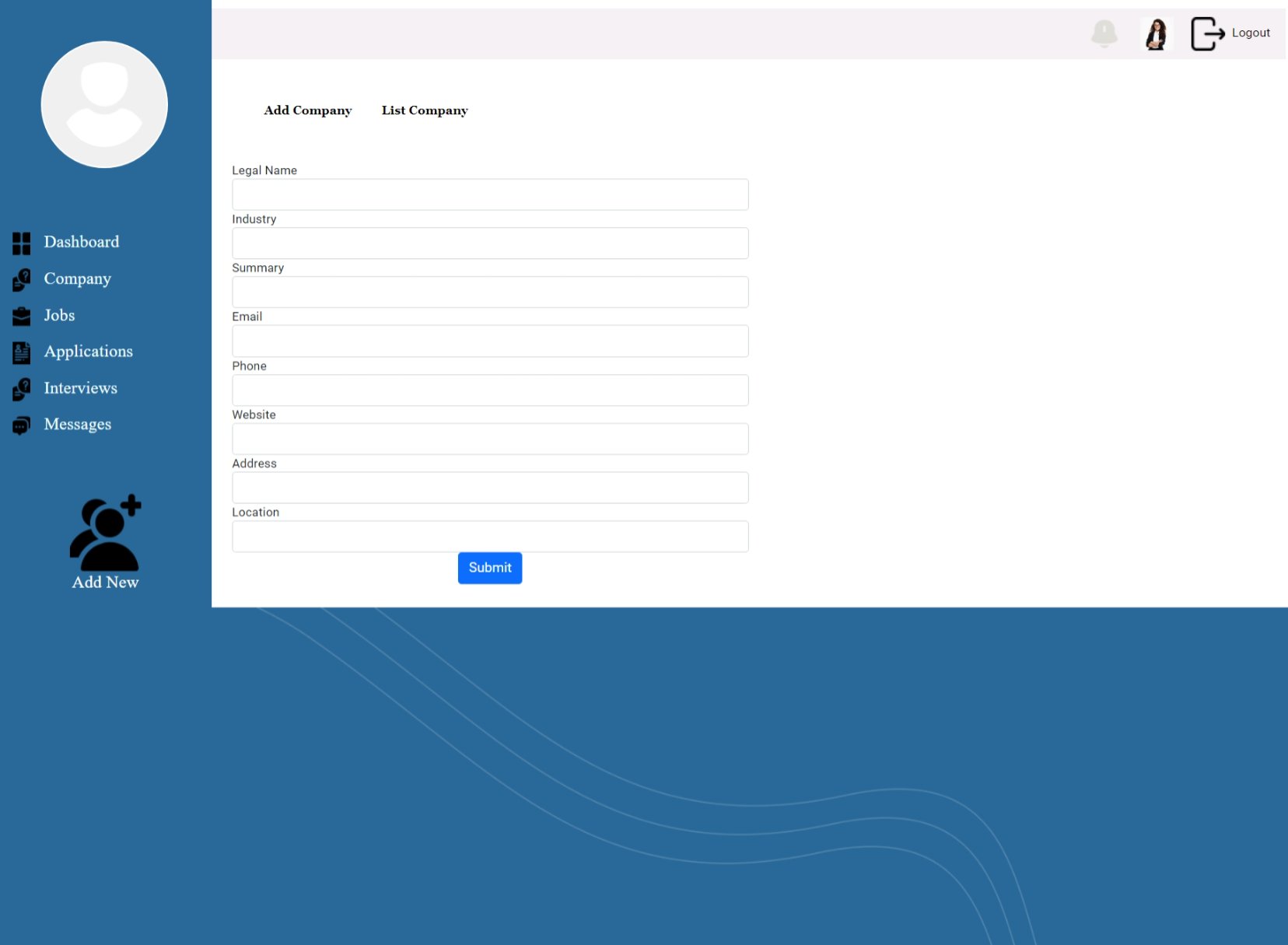
Your task is to design and implement the user interface for managing skills, allowing administrators to view, add, and delete skills. Additionally, the interface should display a list of existing skills retrieved from the backend API.

**Instructions**

* Configuration of JSON Server to serve mock data for job providers and skills.
* Modification of the Angular service class (jobService) to fetch data from JSON Server for job providers.
* Creation of a new service class for managing skills and integrating it with JSON Server.
* Implementation of Angular components to display job providers and skills.
* Integration of job provider and skill management components into the admin panel.
* Creating Angular components to display skills and provide forms for adding new skills.
* Utilizing the SkillService to fetch existing skills, add new skills, and delete existing skills.

**Exercise: 3**

**Job Provider Module: company**



**Objective**

You are tasked with implementing features in an Angular application for the job provider module. Specifically, you need to develop functionality to add companies associated with a job provider and to list the companies that belong to the logged-in job provider. You have a service class named CompanyService which communicates with the backend API to perform operations related to companies. The service class provides methods such as addCompany(data: company) to add a new company and getCompany() to fetch the list of companies associated with the logged-in job provider.

Your task is to simulate a backend API using JSON Mock Server and integrate it with the Angular application to support adding and listing companies.

**Instructions**

* Setting up JSON Mock Server to serve mock data for the backend API.
* Modifying the Angular service class (CompanyService) to interact with JSON Mock Server for adding and fetching companies.
* Implementing Angular components to provide forms for adding companies and to display the list of companies.
* Integrating the company management components into the job provider module of the Angular application.
* Feel free to use any additional Angular libraries or tools that you find suitable for achieving this task. Additionally, consider implementing user-friendly features such as form validation for adding companies and error handling for failed API requests.

**Exercise: 4**

**Job Provider Module: Add company member**



**Objective**

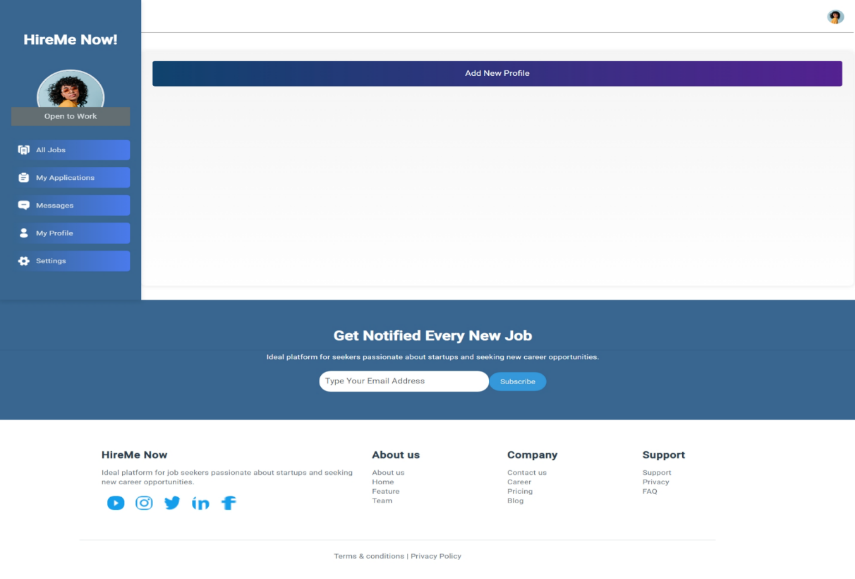
You are working on the development of a job provider module for an Angular application. As part of this module, you need to implement functionality to manage company members associated with a specific company. You have a service class named CompanyMemberService which interacts with the backend API to perform operations related to company members. The service class provides methods such as addCompanyMember(data: companyMember) to add a new company member, listCompanyMember() to fetch the list of company members, and removeCompanyMember(companyMemberId: string) to remove a company member by their ID.

Your task is to simulate a backend API using JSON Mock Server and integrate it with the Angular application to support adding, listing, and removing company members.

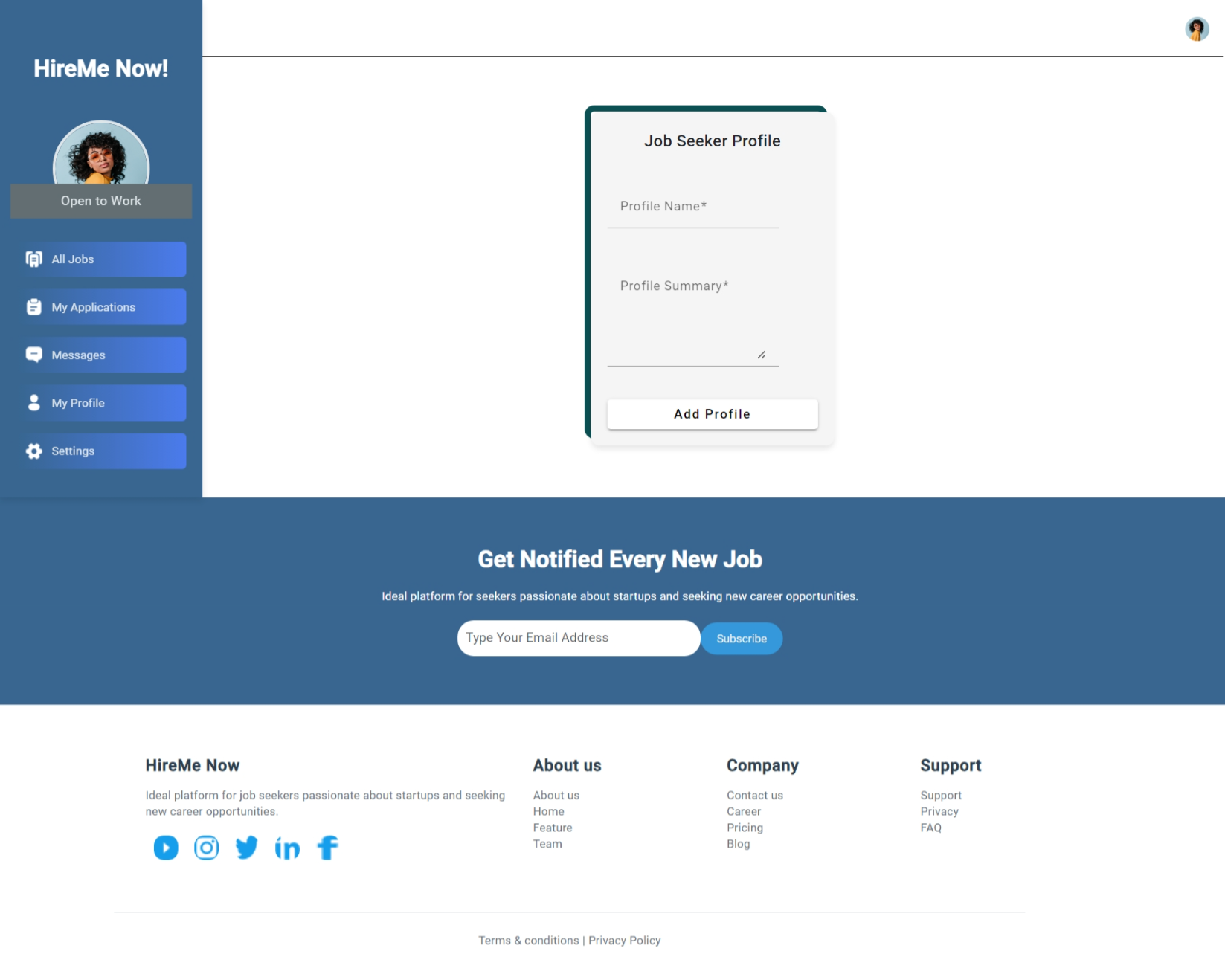
**Instructions**

* Setting up JSON Mock Server to serve mock data for the backend API.
* Modifying the Angular service class (CompanyMemberService) to interact with JSON Mock Server for adding, listing, and removing company members.
* Implementing Angular components to provide forms for adding company members and to display the list of company members.
* Integrating the company member management components into the job provider module of the Angular application.

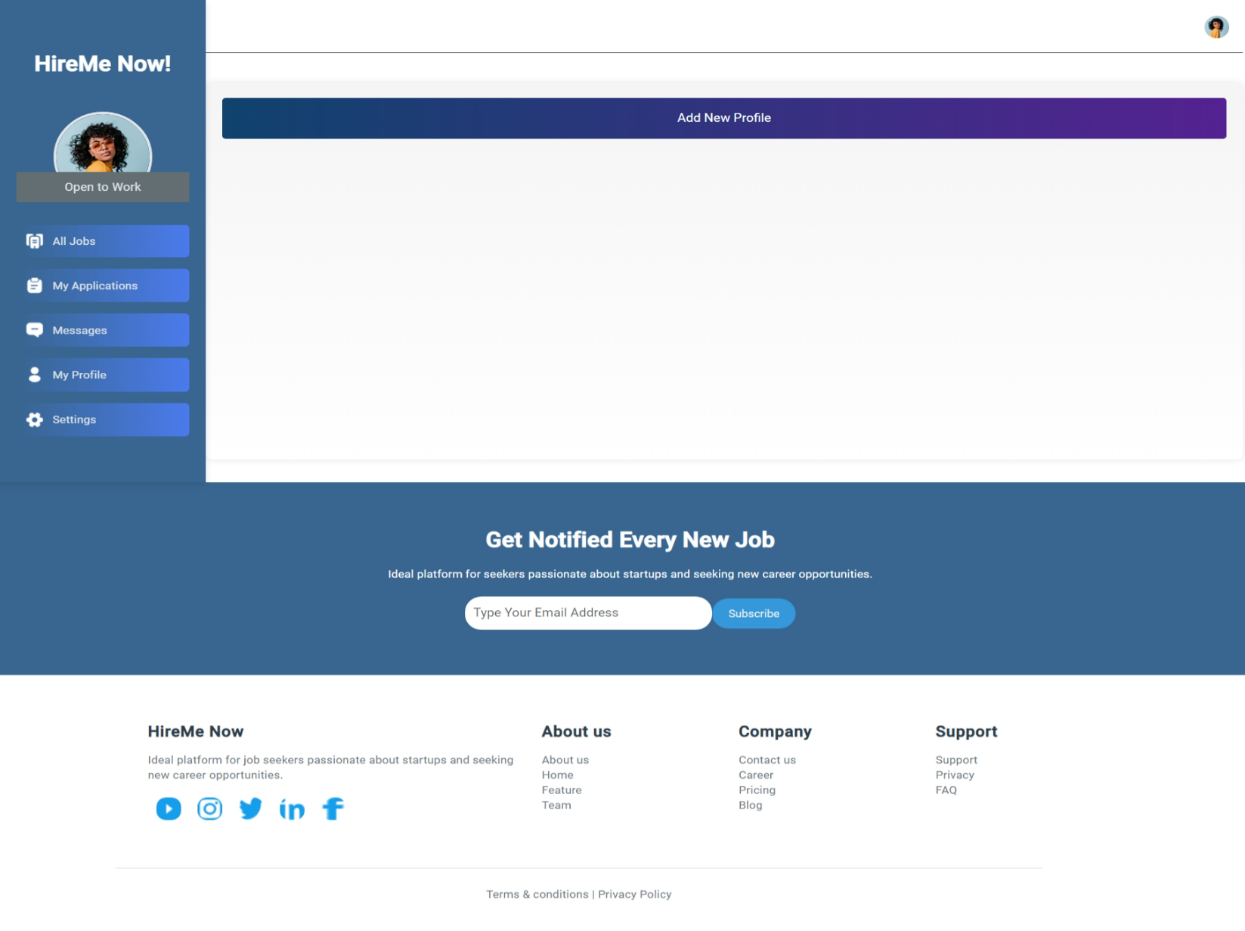
**Exercise: 5**

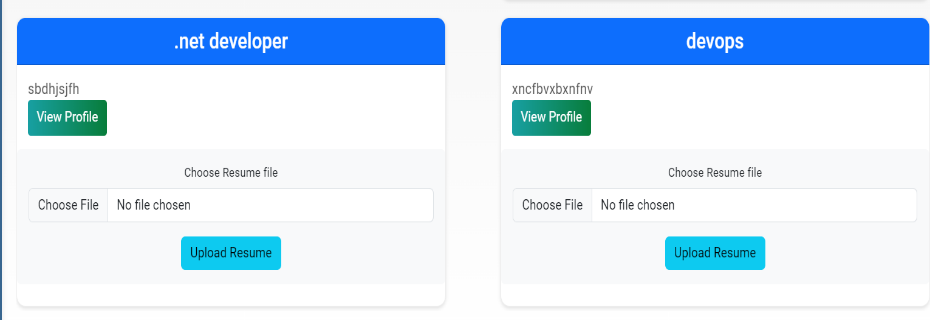
**Job Seeker Module: My Profile**

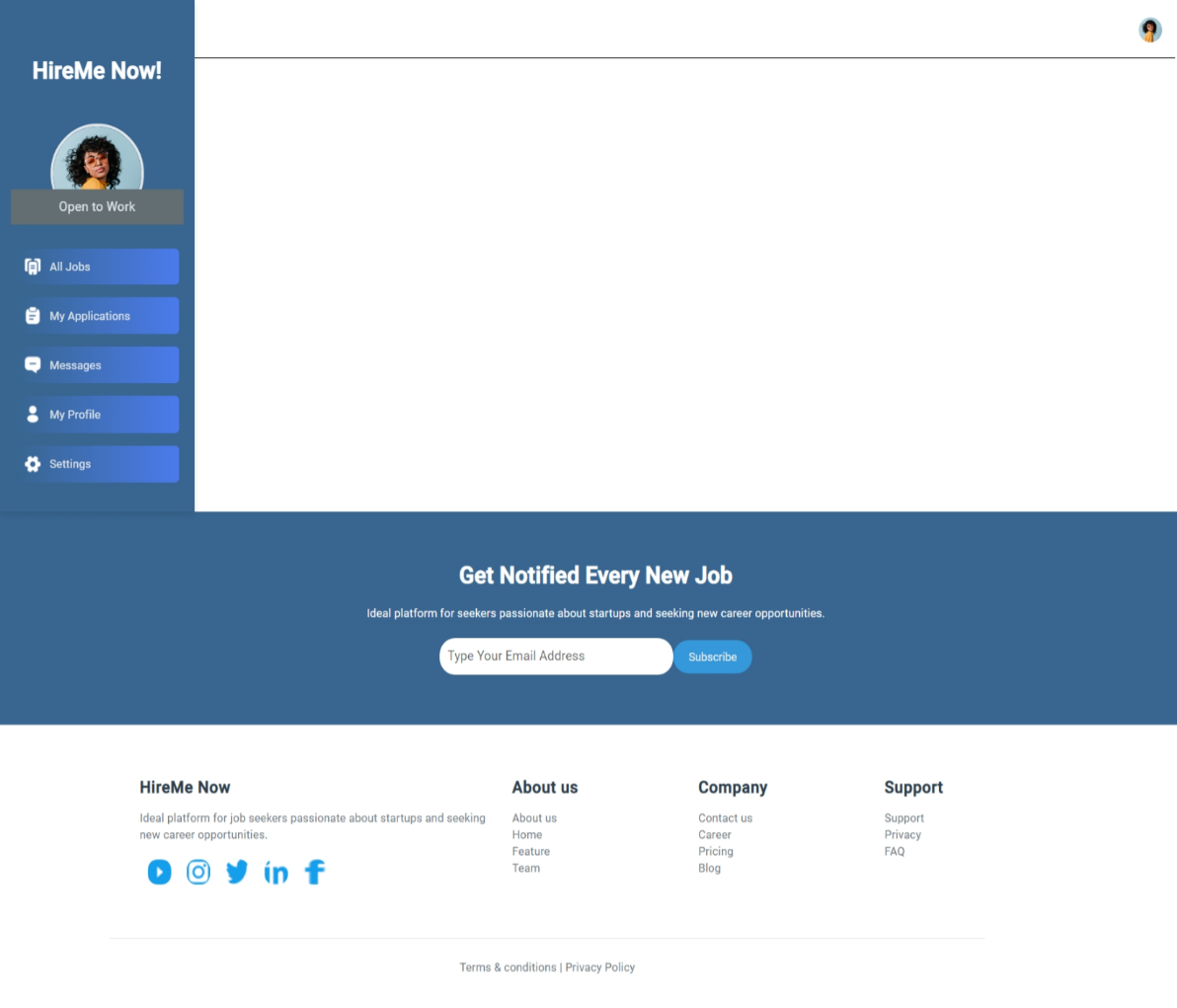
**My Profile: Add New Profile**

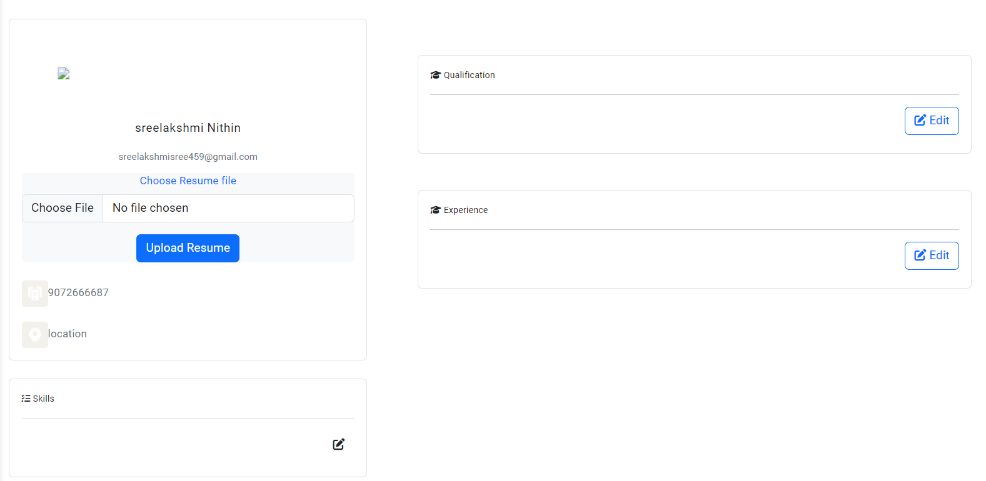


**My Profile: Multiple profile page**

****



**My Profile: View Profile**

****

**Objective**

You have been tasked with developing features for a job seeker's profile management module in an Angular application. You are provided with a service class named ProfileService which interacts with the backend API to perform operations related to profiles, skills, qualifications, experiences, and updates to user profiles. The service class offers various methods such as addNewProfile(data: any) to add a new profile, getProfile() to retrieve the user's profile, addSkill(data: any[], profileId: any) to add skills to a profile, addQualification(name: any, description: any, profileId: any) to add qualifications to a profile, and updateJobSeekerProfile(values: any) to update the user's profile information.

Your task is to simulate a backend API using JSON Mock Server and integrate it with the Angular application to support adding new profiles, viewing profiles, getting skills, and updating qualifications.

**Instructions**

Setting up JSON Mock Server to serve mock data for the backend API.

Modifying the Angular service class (ProfileService) to interact with JSON Mock Server for profile management operations.

Implementing Angular components to provide forms for adding new profiles and updating qualifications, as well as displaying profile information and skills.

Integrating the profile management components into the relevant sections of the Angular application.