**STUDENT HANDOUT ANGULAR CHAPTER 4**

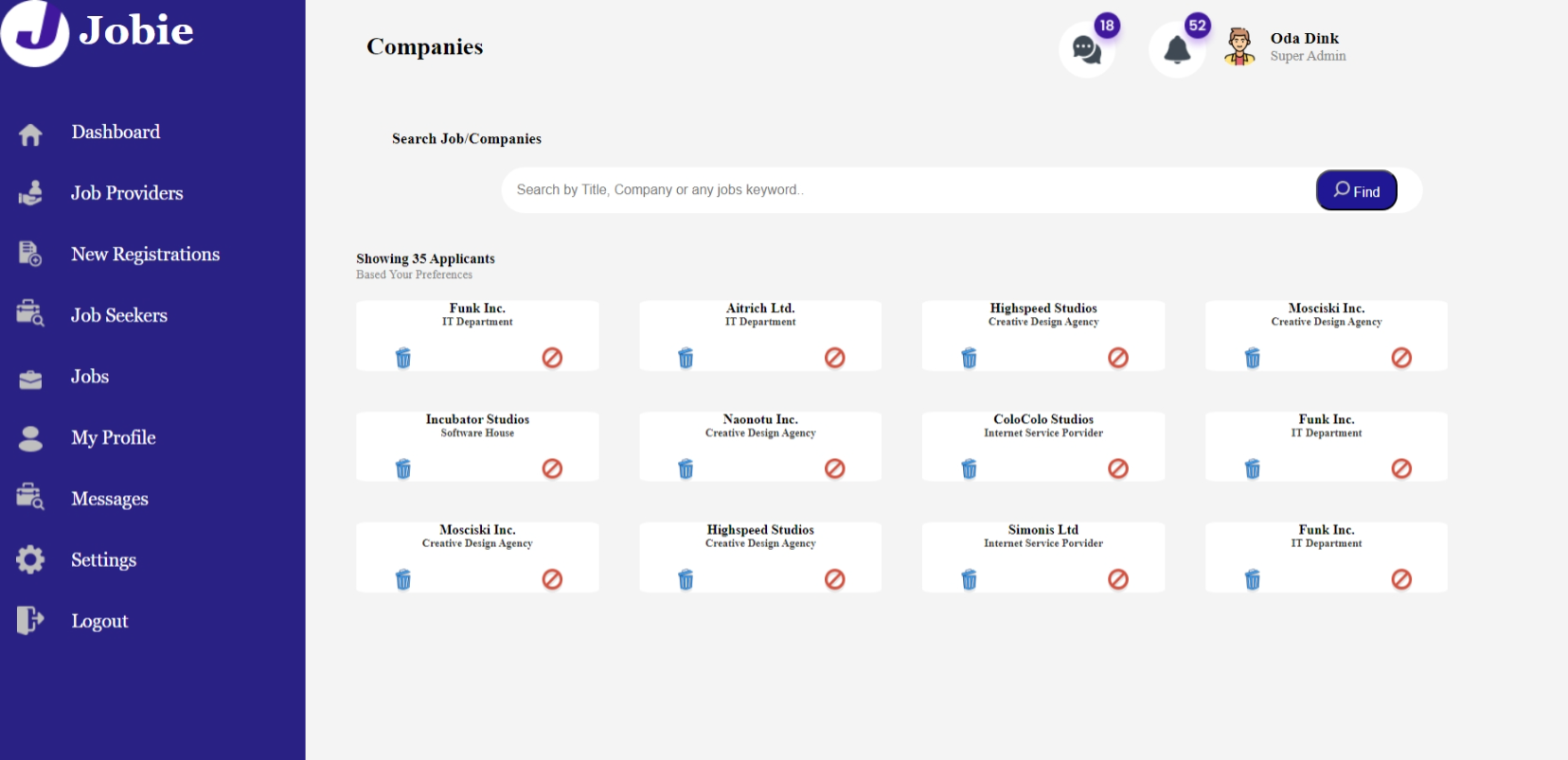
**DATABINDING**

**About Workshop :**

Data binding is a fundamental concept in Angular, a popular JavaScript framework for building web applications. It allows you to establish a connection between the model(data) and the view (UI) in your application, ensuring that changes in one are reflected in the other. There are several types of data binding in Angular, including one-way and two-way data binding.

In our workshop, we create a webpage for a platform admin, and we bind data from components to our view using property binding, attribute binding, and event binding.

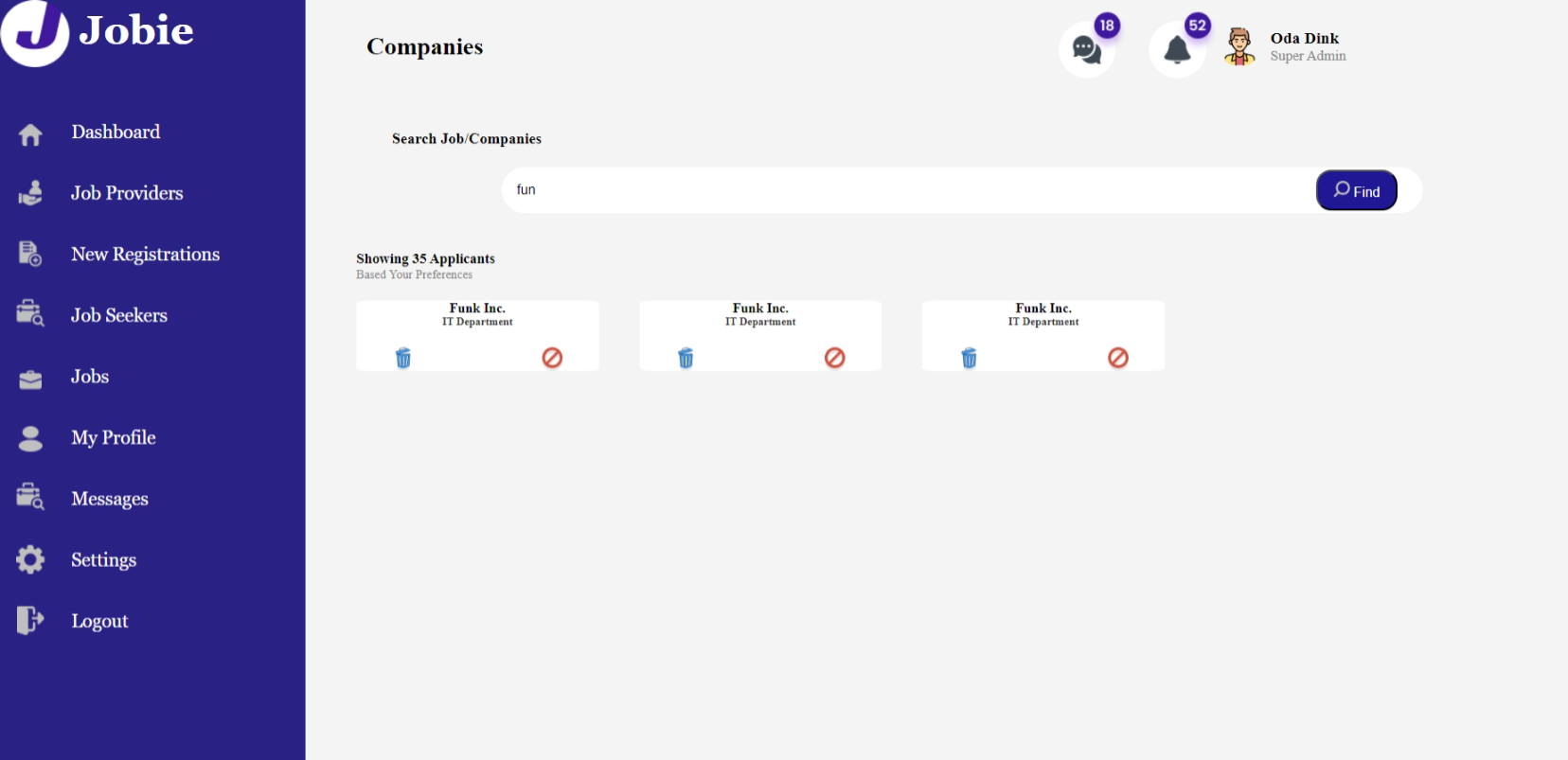
**Workshop 1:**

****

In our workshop, our task is to create a company webpage for 'PlatformAdmin.' This webpage comprises several key components, including a content section for listing the companies, a sidebar for navigation, and a header component for the top of the page. To facilitate this, we've created a model in a file named 'jobproviderlist.ts,' which declares a class representing a company.

In the content component, we've initialized an array of the type defined in the 'jobproviderlist.ts' model class. This array is used to store the details of the various companies that will be displayed on the webpage. We accomplish this by populating the array with company information.

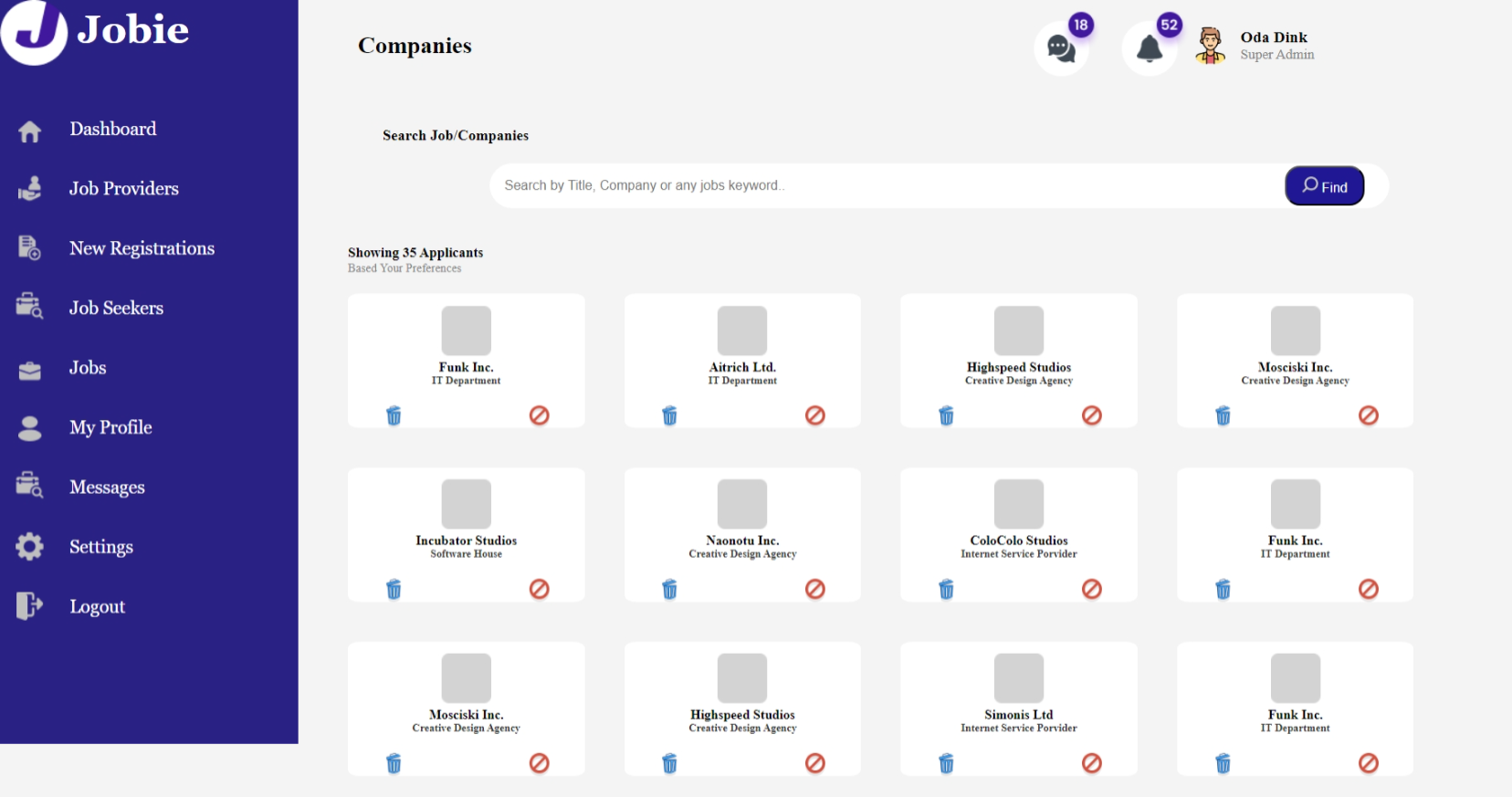
To dynamically display the company data on our content page, we utilize interpolation, ensuring that the information in the array is synchronized with the content component's template. This allows for real-time updates as needed.



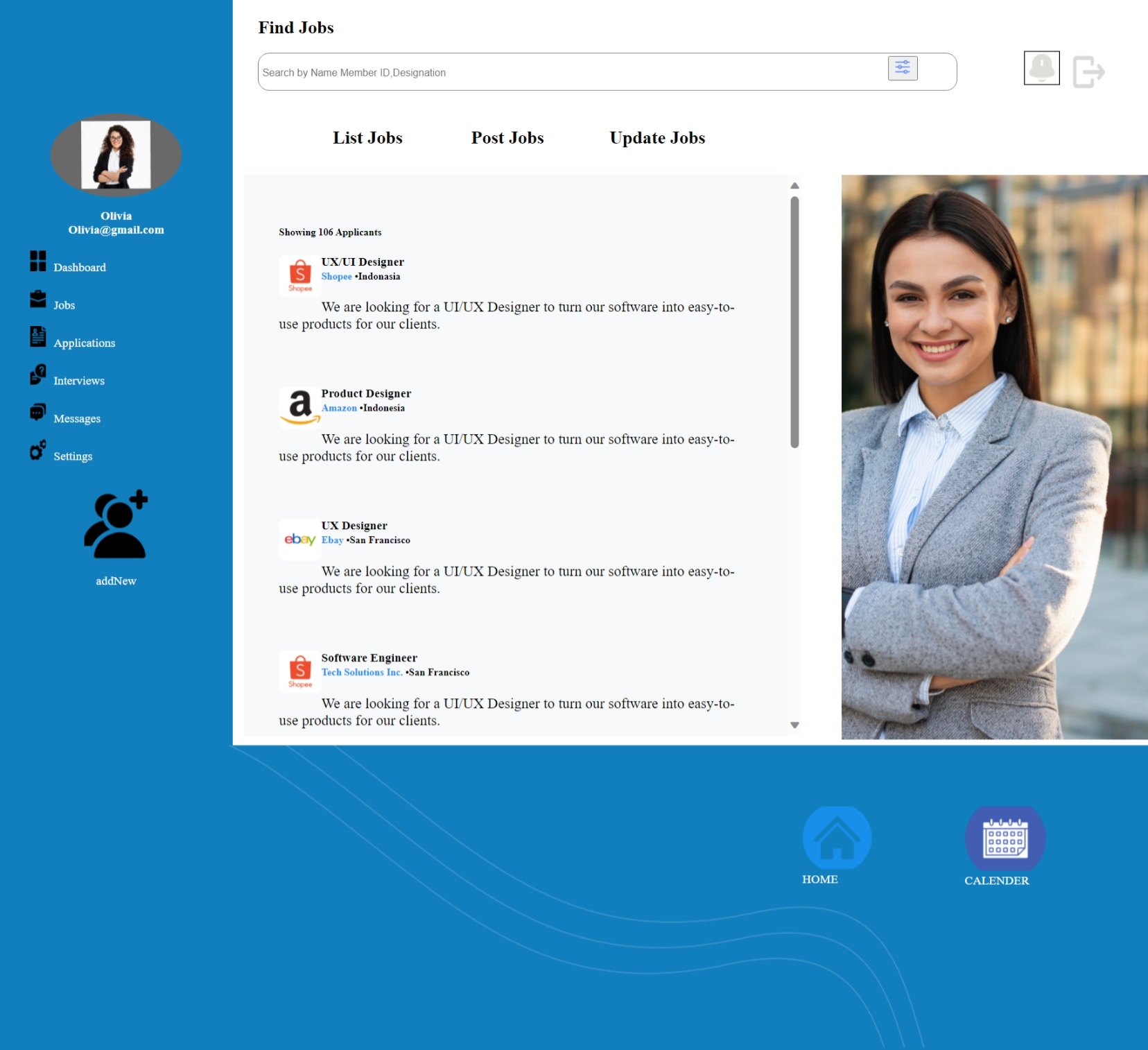
Furthermore, we've implemented a filtering function that enables users to search and filter the displayed companies based on their names, providing a user-friendly and efficient way to interact with the listed companies.

Additionally, we have added company logos to cards using property binding.

**Expected Output:**

****

**Worksop 2:**

****

In our workshop, our task is to create a jobList webpage for 'PlatformAdmin.' This webpage comprises several key components, including a job component for listing the jobs, a sidebar for navigation, and a header component for the top of the page. To facilitate this, we've created a model in a file named 'jobList.ts,' which declares a class representing a job

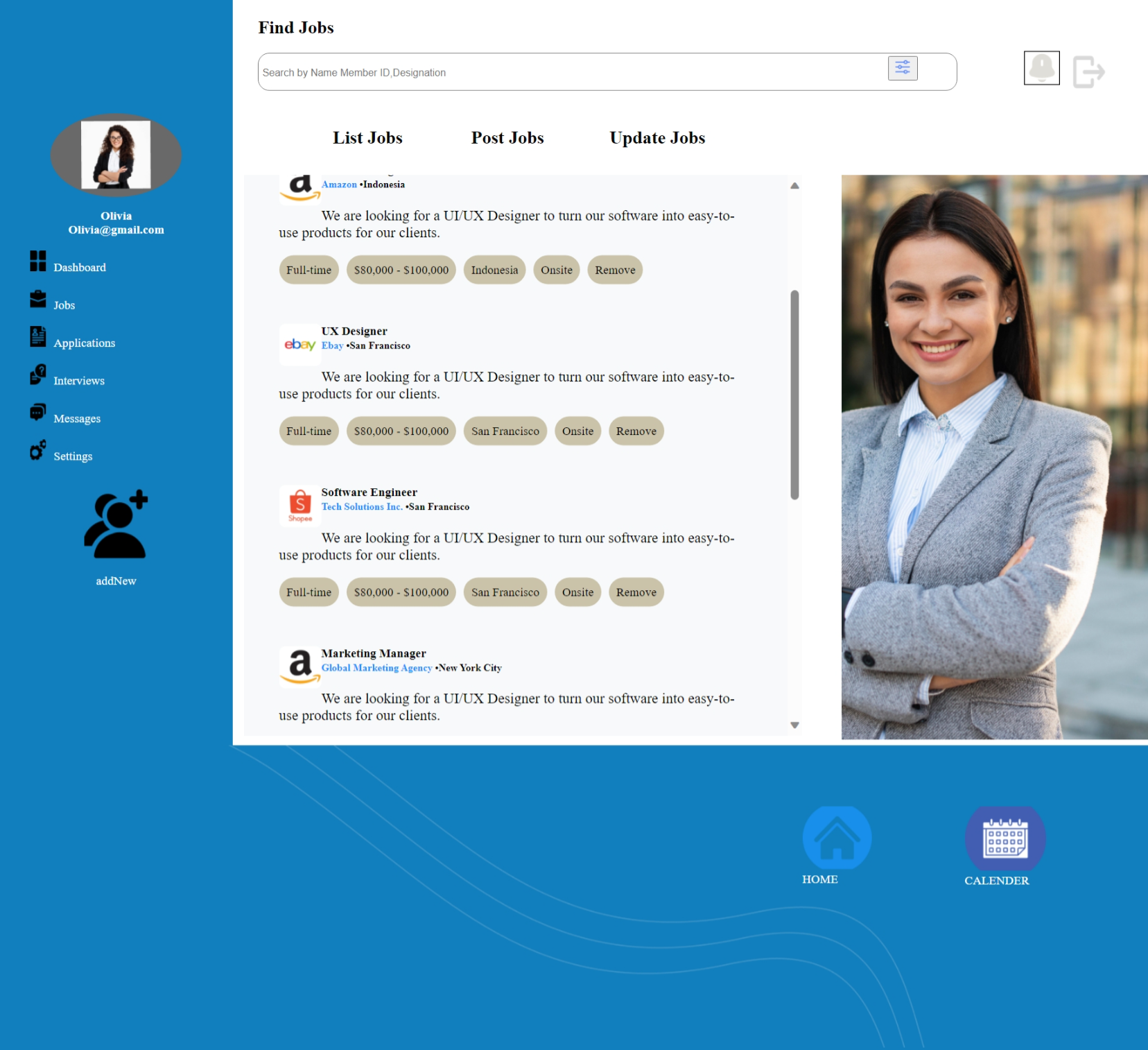
In the job component, we've initialized an array of the type defined in the 'jobList.ts' model class. This array is used to store the details of the various jobs that will be displayed on the webpage. We accomplish this by populating the array with job information.

To dynamically display the job data on our job page, we utilize interpolation, ensuring that the information in the array is synchronized with the job component's template. This allows for real-time updates as needed.

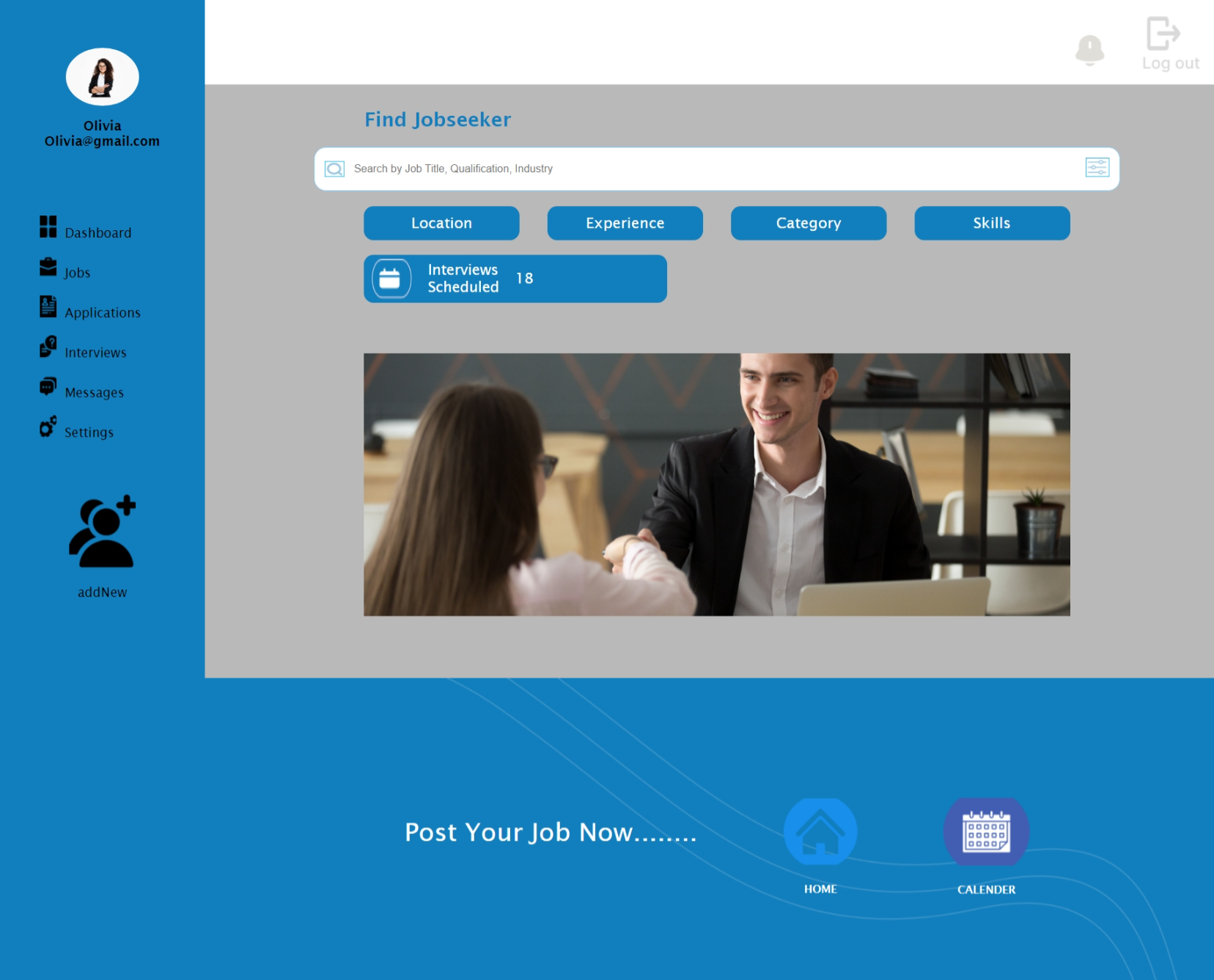
Furthermore, we've implemented a filtering function that enables users to search and filter the displayed jobs based on their names, providing a user-friendly and efficient way to interact with the listed jobs.

Additionally, we have added company logos to cards using property binding.

**ExpectedOutput:**



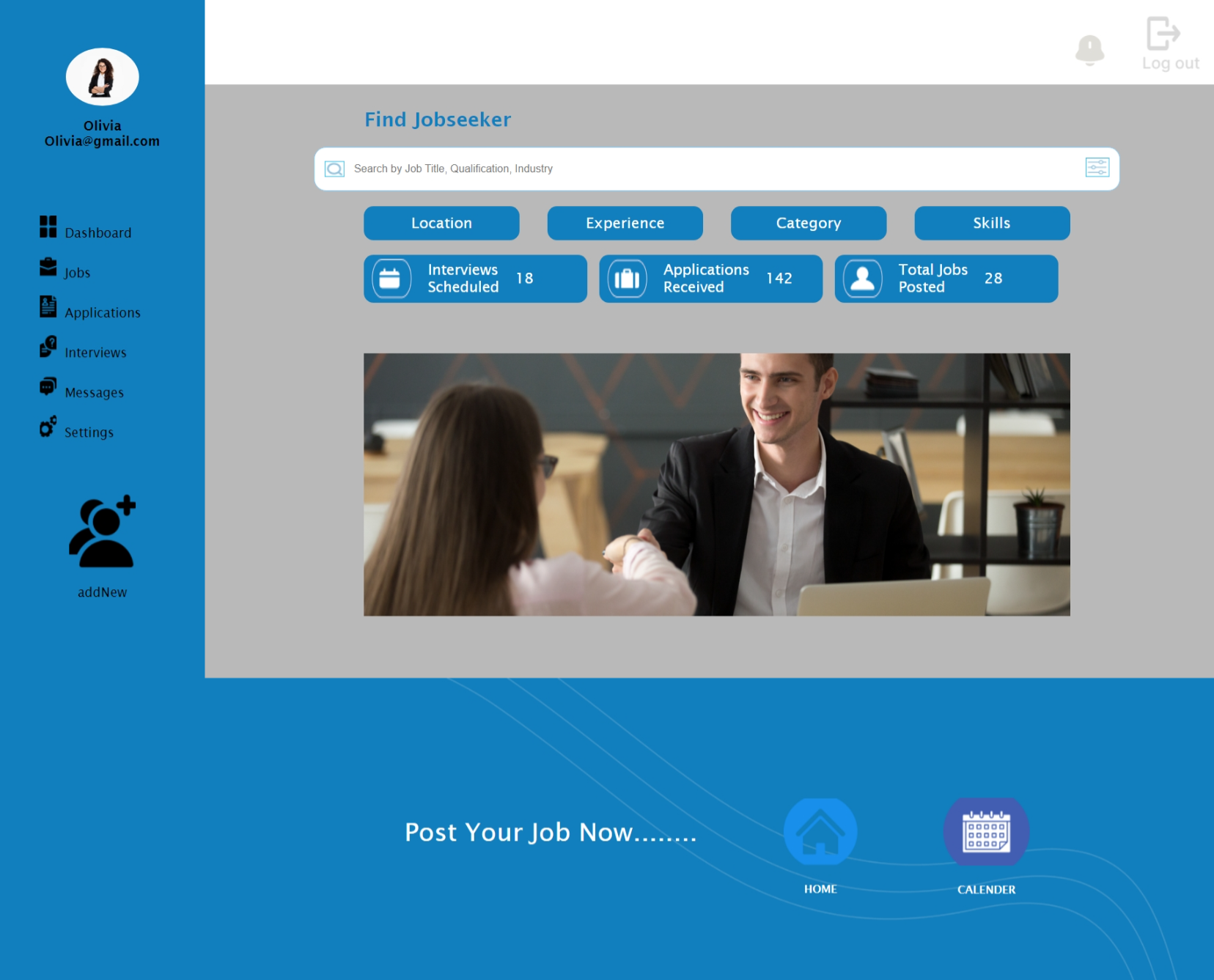
**Workshop :3**

****

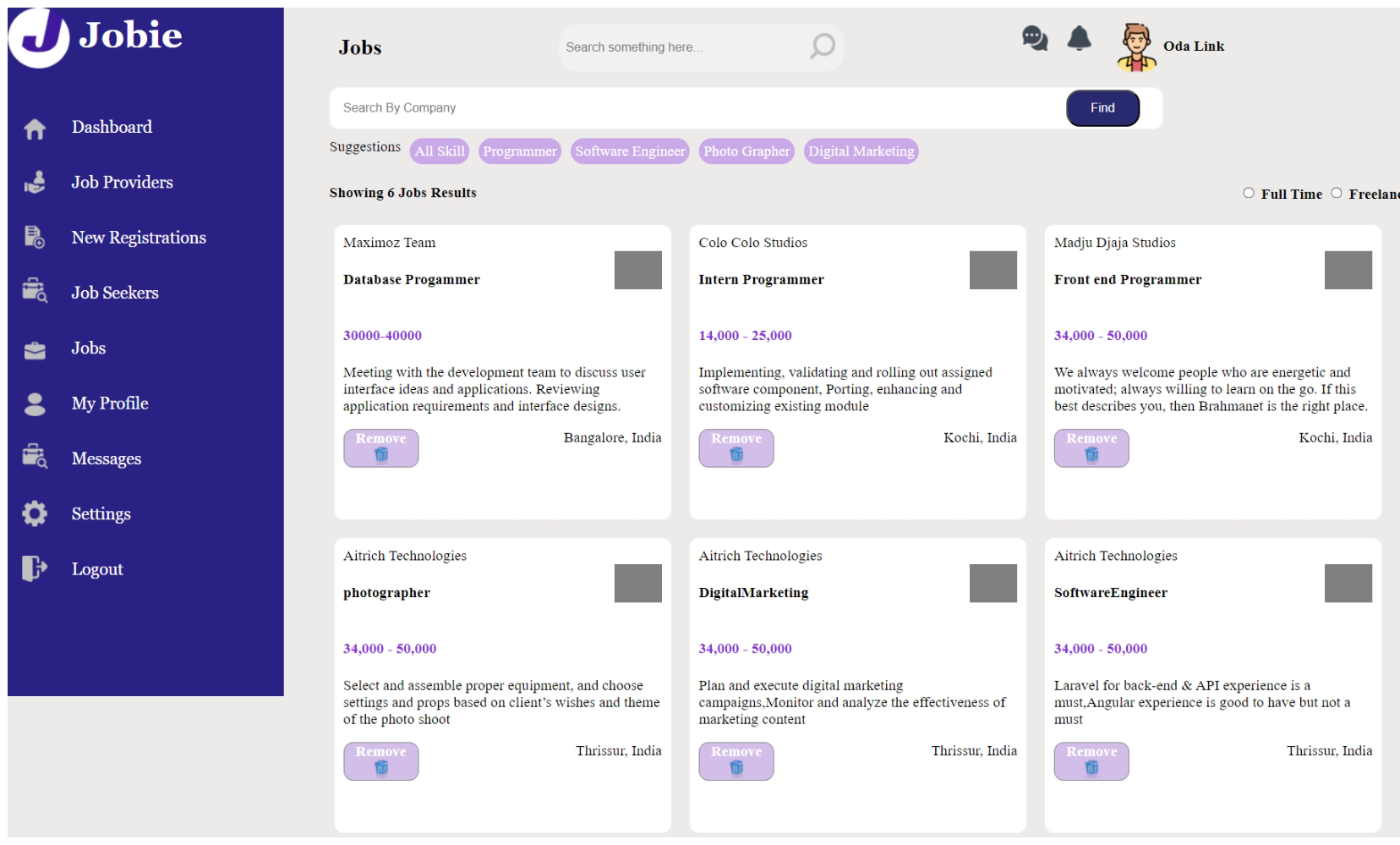
In our workshop, our task is to create a Dashboard page for 'Jobseekers.' This webpage consists of several key components, including a content component for retrieving job-related information such as scheduled interviews, a navigation sidebar, a header component at the top of the page, and a footer. In the content component, we are using data binding with interpolation to display the number of scheduled interviews, which is linked to the view from the component class.

Your task is to add two blocks for displaying the number of applications received and the total number of job postings. The values for each block should be bound to the view using data binding, as shown in the expected output.

**Expected Output:**

****

**Workshop :4**

****

In our workshop, our goal is to create a Job page for 'Jobseekers.' This webpage comprises key components, including a job component for retrieving job information, a navigation sidebar, and a navbar component at the top of the page. Within the content component, we've created an array to store job details, a function filter(jobname) for filtering jobs by job name, and a resetBackground() function to set the background color for buttons. We are utilizing interpolation, attribute data binding, and event data binding to display job details and set the background color.

Your specific task is to add a styled background color for buttons (All Skills, Programmer, Software Engineer, Photographer, Digital Marketing) upon hover, using attribute data binding and event binding, as illustrated in the expected output.

**Expected Output:**

