**Roster Application for scheduling Interview with Tech Panel**

**Introduction**

It is web app for the Talent Acquisition team to intimate the panel list about the upcoming interviews and to know their availability.

**Business use case**

Presently for conducting an Interview for a particular skill, the Talent Acquisition team needs to contact the Service Line .The Service Line will then share the Panel List with the required skills and later need to check their available time. After that Panel list has to respond back with their availability. The whole process is very time consuming, with multiple back and forth of emails, the intend of this use case is to build an application which includes all the functionalities and can be completed with a short span of time.

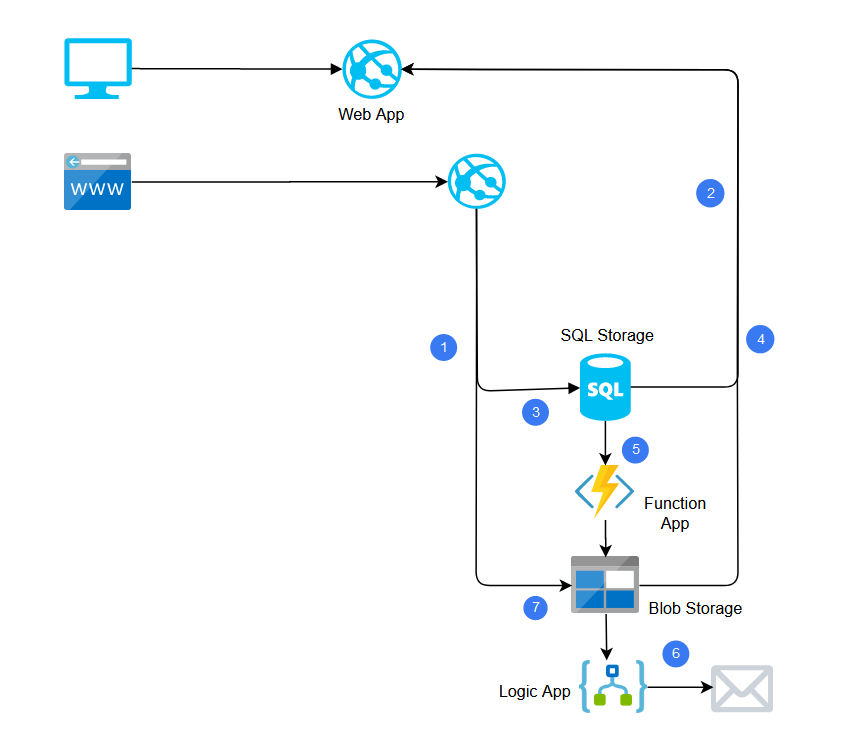
**Solution outline**

* A web application is made which has three main roles( Talent Acquisition, Service Line and Panel list)
* Based on the roles, the users will be pointed to their corresponding form to fill details.
* Talent Acquisition team will fill the Interview details and intimate the panel List based on the skills(mail is triggered using Logic app)
* Panel Lists will login , check the Interview details and then respond back based on their availability by selecting the slots.
* Service Line team will be assigning panel List to the corresponding skill pool.
* Azure Functions are used to fetch and save data to SQL Database.

**Technologies used**

* Azure Subscription
* Front End- Angular
* Backend- Azure SQL Db
* Azure Web App
* Azure Logic Apps
* Azure Functions

**High Level Diagram**



**Logic App**

Logic apps are basically used to design workflows. It would need a trigger to work and  a set of actions are configured to perform a task.  
All these configurations are done in Logic app designer.

Once the logic app is saved with the configuration, for each trigger of the logic app, a logic app instance is created, and the list of actions are performed.

For Roster application, two logicapps are configured.

LogicApp1-SendMail-PL  
When the Talent Acquisition team logged into the application, submit the interview name and interview date to check the availability of Panel, this logic app will be triggered.  
Once the logic app is triggered, a mail is sent to all the panel of the respected skills. The mail includes a url for panel to respond their availability.

LogicApp2-SendMail-TA  
Once the Panel receive a mail, they login and submit their availability, this logic app is triggered.  
Based on the availability response,  a mail is sent back to Talent Acquisition team about the Panel availability to take interview.

**CI/CD deployment of Functions through Github**

1.

# Continuous delivery by using GitHub Action

* To define a workflow to automatically build and deploy code to the function app in Azure Functions.

2. /\* For an Azure Functions workflow, the file has three sections:

| **TABLE 1** | |
| --- | --- |
| **Section** | **Tasks** |
| **Authentication** | Download a publish profile. Create a GitHub secret. |
| **Build** | Set up the environment. Build the function app. |
| **Deploy** | Deploy the function app. |

The recommended way to authenticate with Azure Functions for GitHub Actions is by using a publish profile

After saving the publish profile credential as a [GitHub secret](https://docs.github.com/en/actions/reference/encrypted-secrets), we'll use this secret within our workflow to authenticate with Azure.

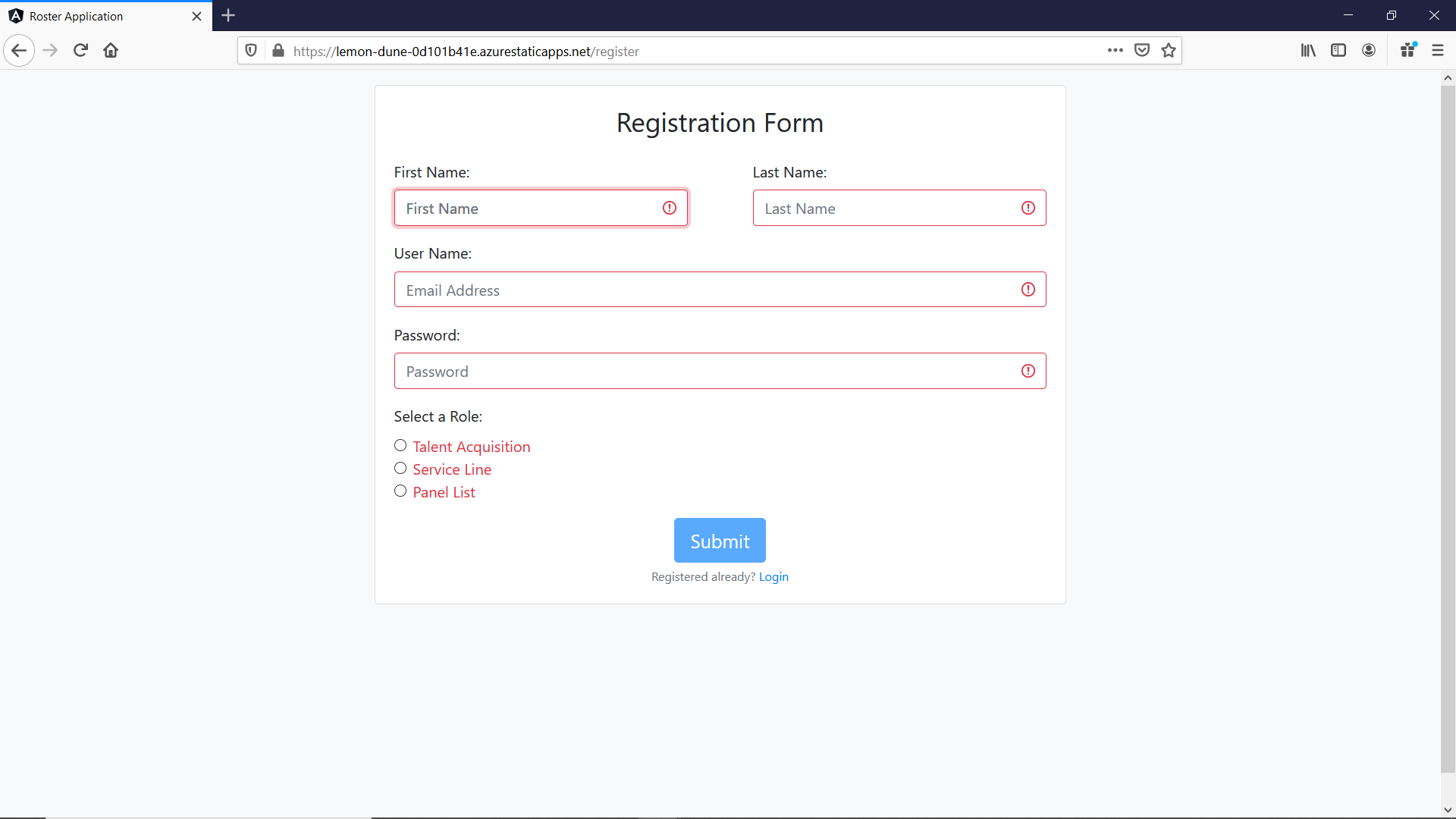
\*/

Deployment of the Function App

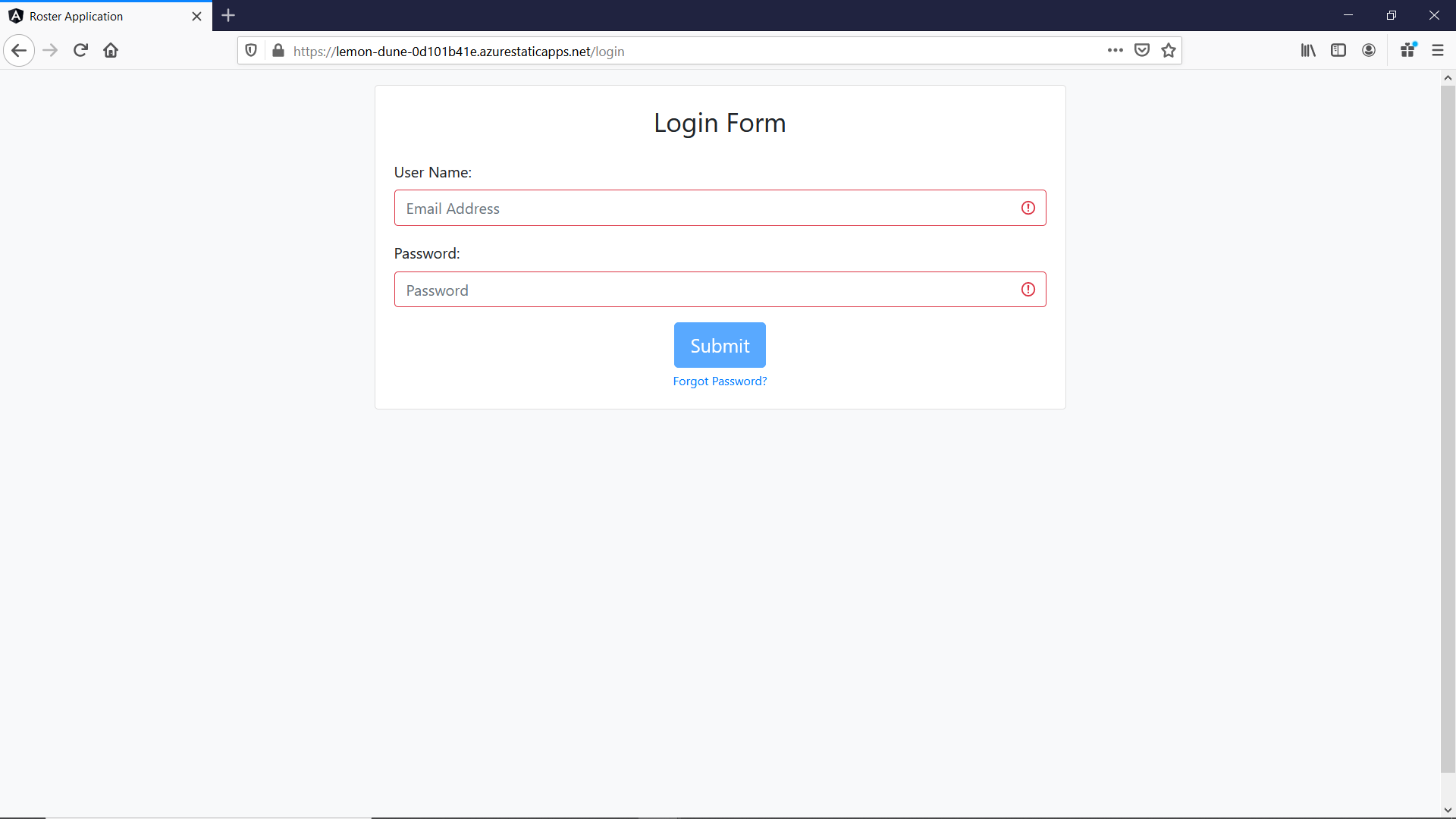
* Used the Azure/functions-action action to deploy the code to a function app.
* Parameters used are :App-name, Slot-name, Publish-profile

**Screen Shots:**

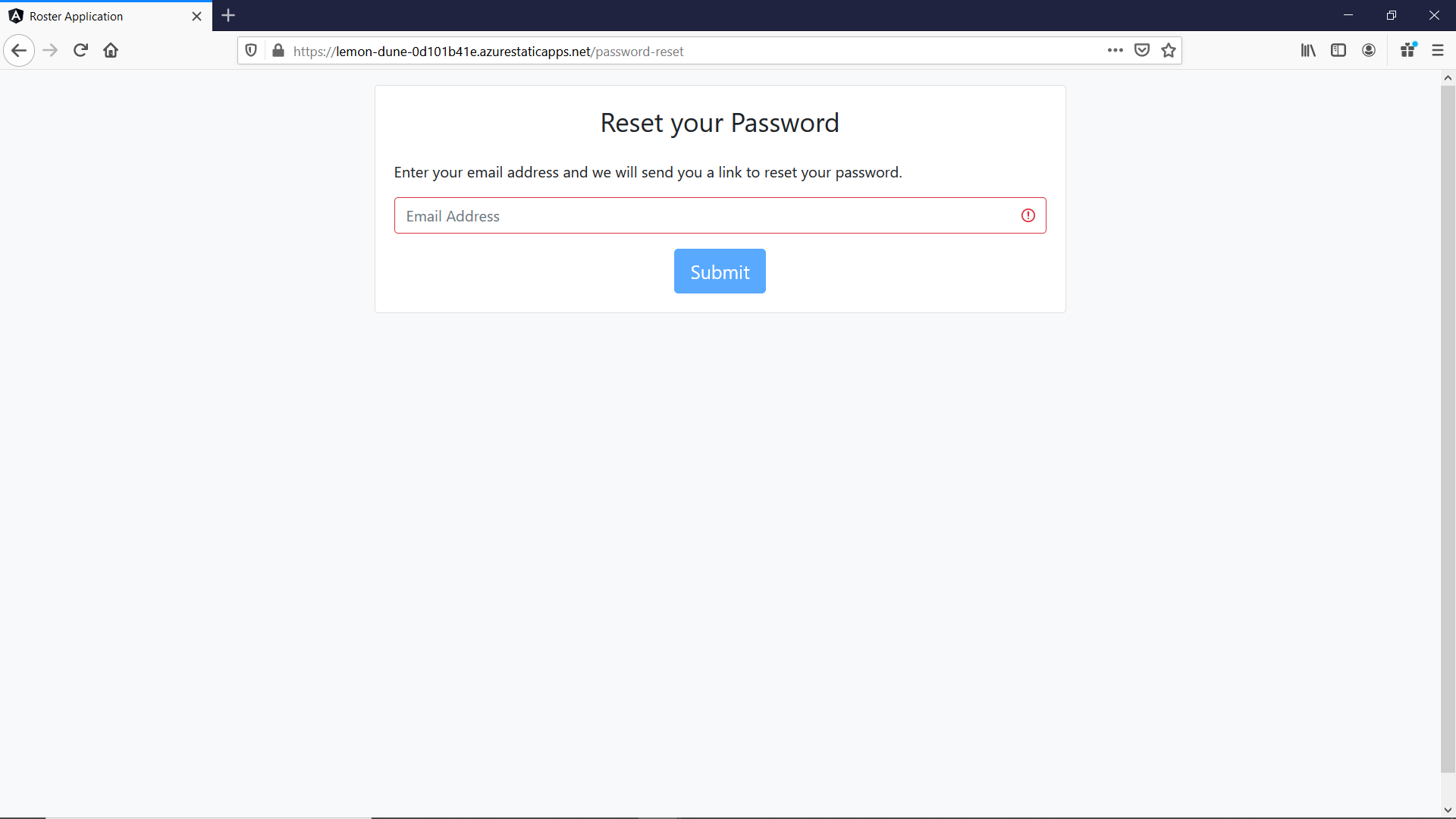
1. **User Registration form**



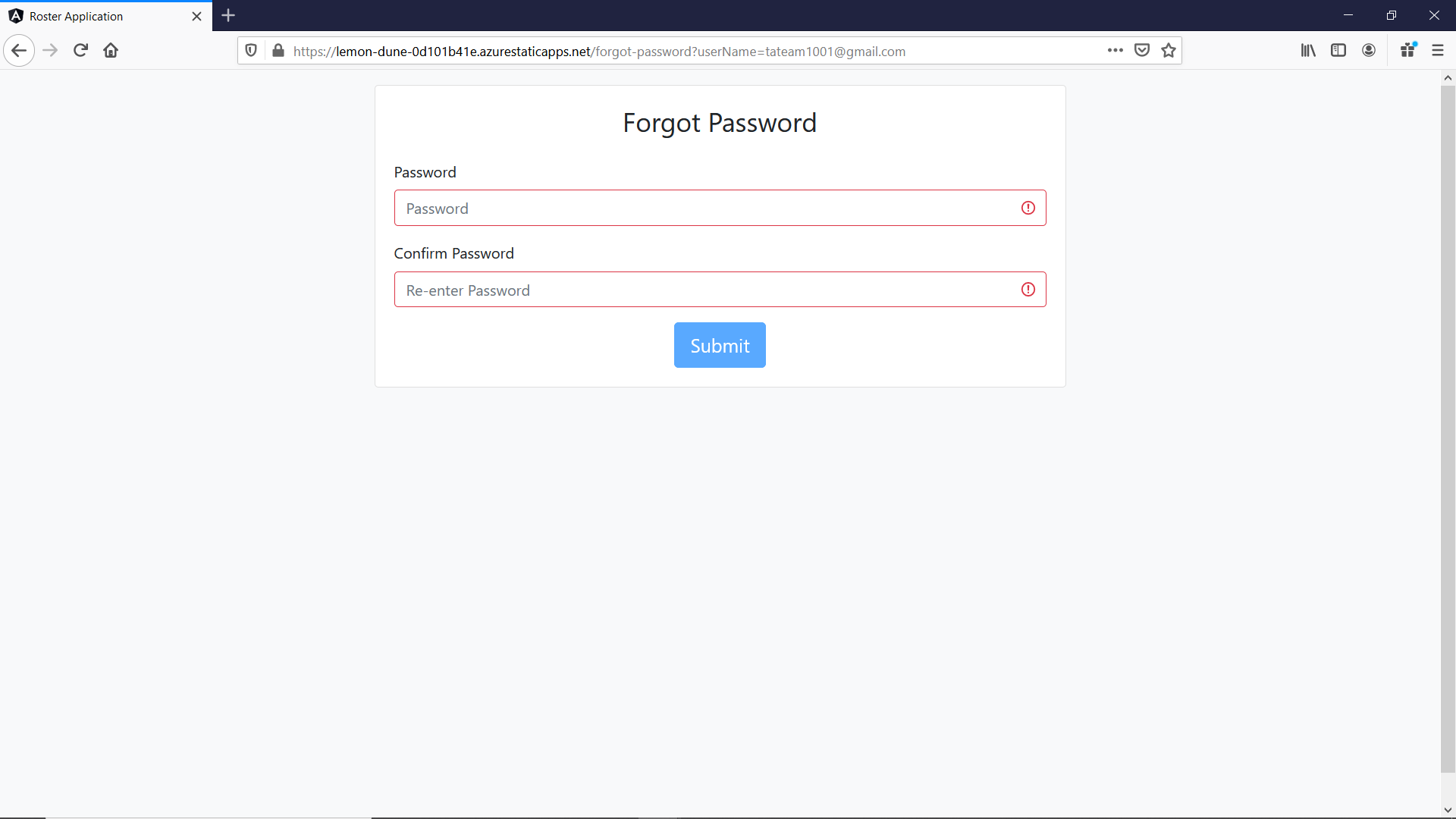
1. **Login Form**



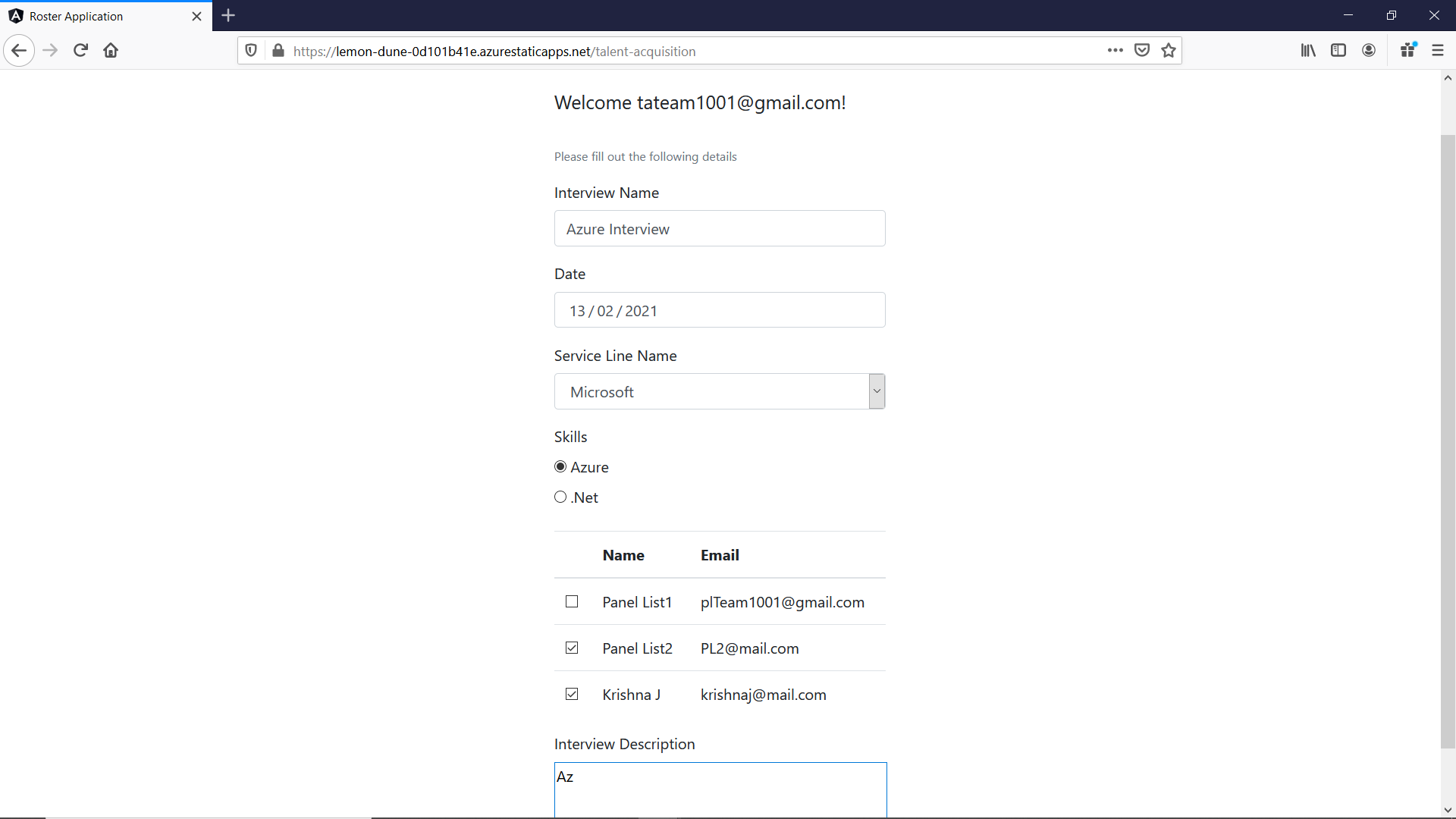
1. Forgot Password Screen



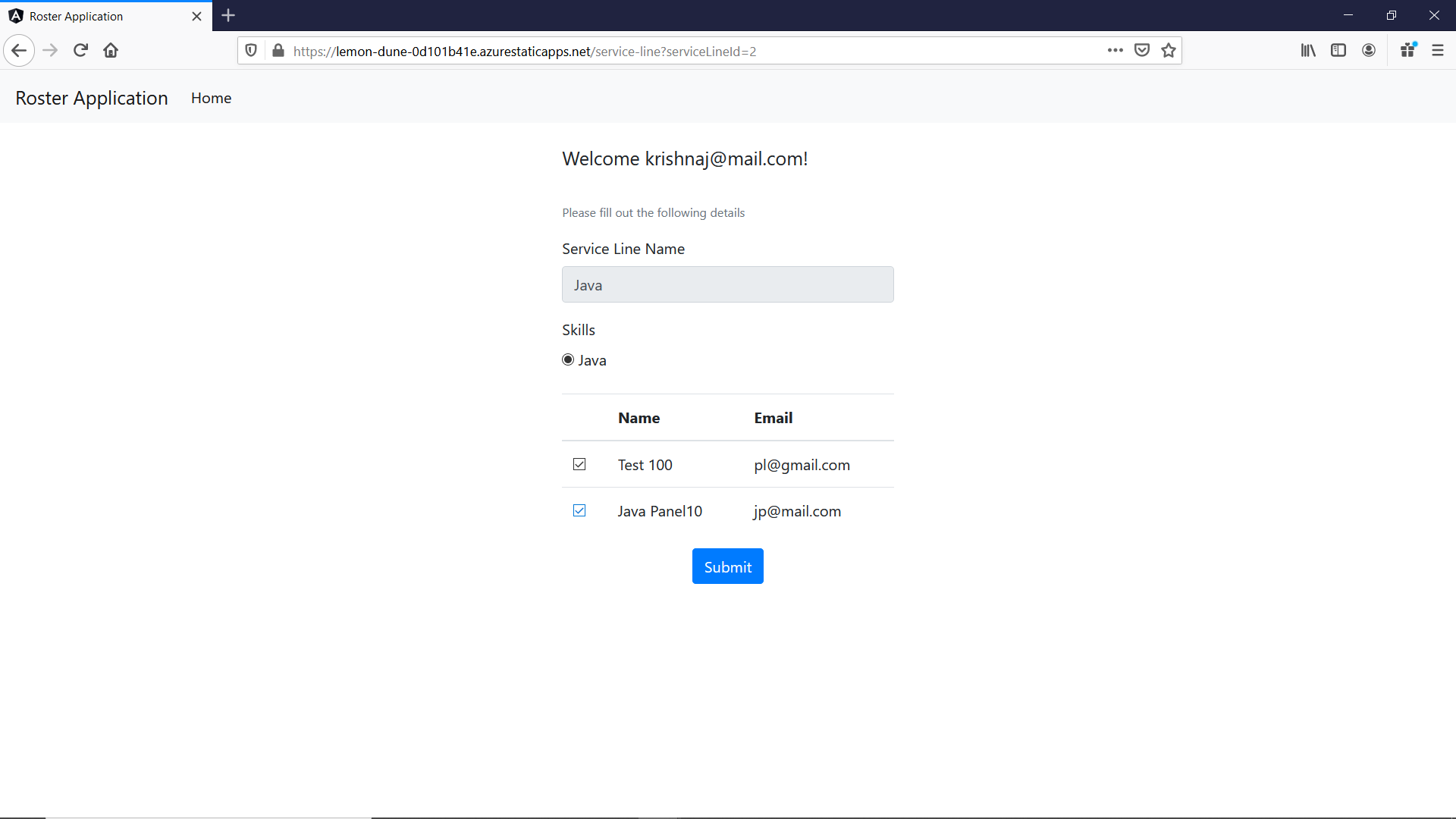
1. Reset Password Screen



1. Talent Acquisition team form



1. Service Line team form



1. Panel List entry form

