***DEMO PROJECT***

(<https://github.com/shreeyan1011/JavaProject>)

**Introduction:**

This project is about a recruitment website called “TalentHire Recruiters” where job seekers can submit their information (including CV and Cover Letter) on the website and create their user account so that the talented team of recruiters can contact them regarding the job opportunity based on their skills and information. It also allows users to search and apply for job opportunities in the field that they are interested in. This project is made using Eclipse (Java 14), Spring Boot, Maven, Tomcat Server, HTML, CSS, and JavaScript.

**Functionality:**

There are currently two functional flows available for the website.

**A. Flow 1 (Submitting user information)**

**1.** The homepage (<http://localhost:8080/home.html>) of the website (home.html) contains a header, footer, and a navigation tab (containing tabs: Home, About us, Upload your CV, Jobs, and Sector). The home tab is active as the present page is home.html (image 1).



**Figure: Image 1**

2. The tabs Jobs and Sector have dropdown as shown below in Image 2 and 3.

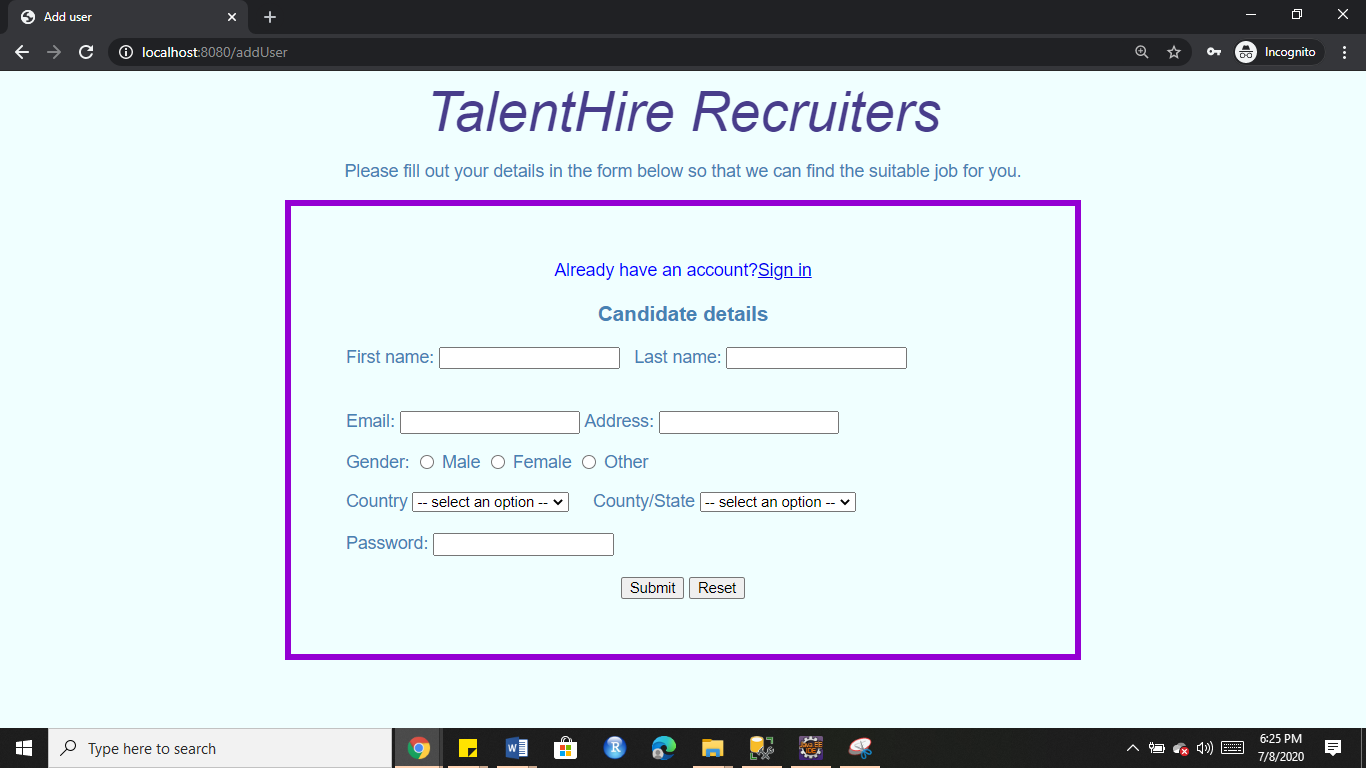


**Figure: Image 2**



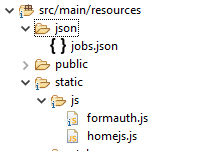
**Figure: Image 3**

3. When Upload your CV is clicked the page is redirected to addUser.html (mapped with /addUser) that contains a form as shown below in Image 4.

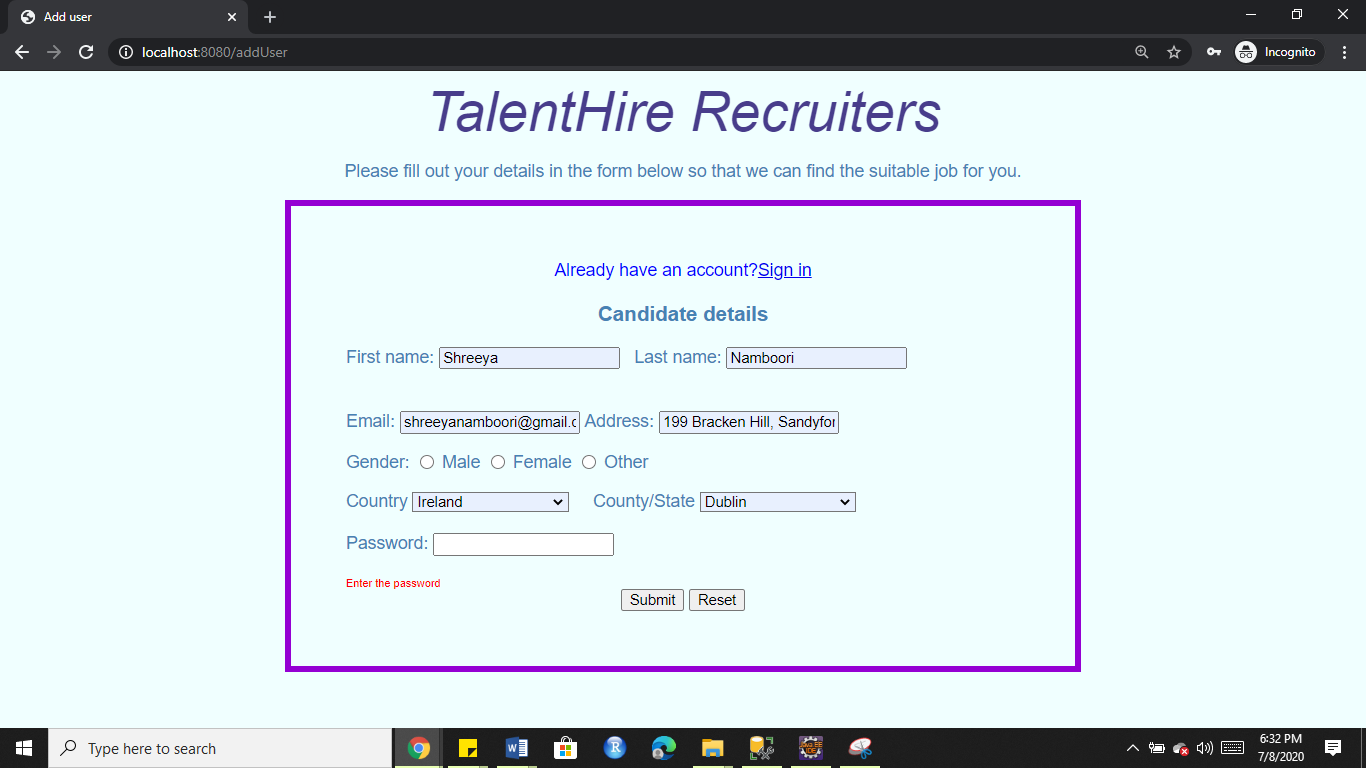


**Figure: Image 4**

4. The form is validated using JavaScript (formauth.js) that checks if the form fields are null or not and if null a message to enter a value on those fields is displayed (Image 6 and 7).



**Figure: Image 5 ( JS folder loacation)**

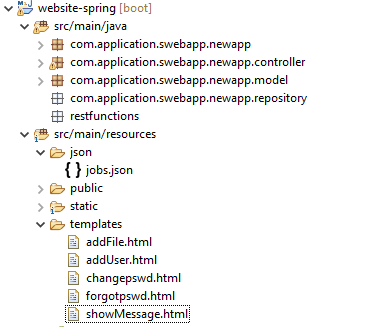


**Figure: Image 6**



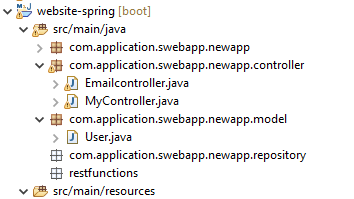
**Figure: Image 7**

4. Once all the required fields are filled the form is successfully submitted and a new page (showMessage.html also mapped with /addUser) is visible that shows the details that have been filled in the previous form for confirmation (using backend java programming).

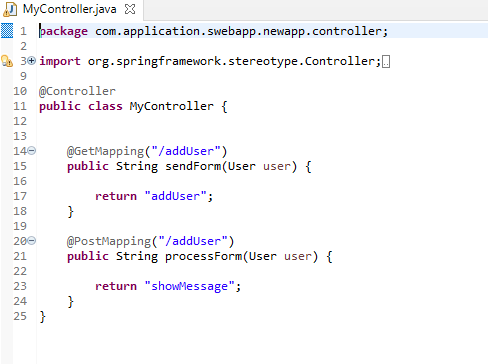


**Figure: Image 8**

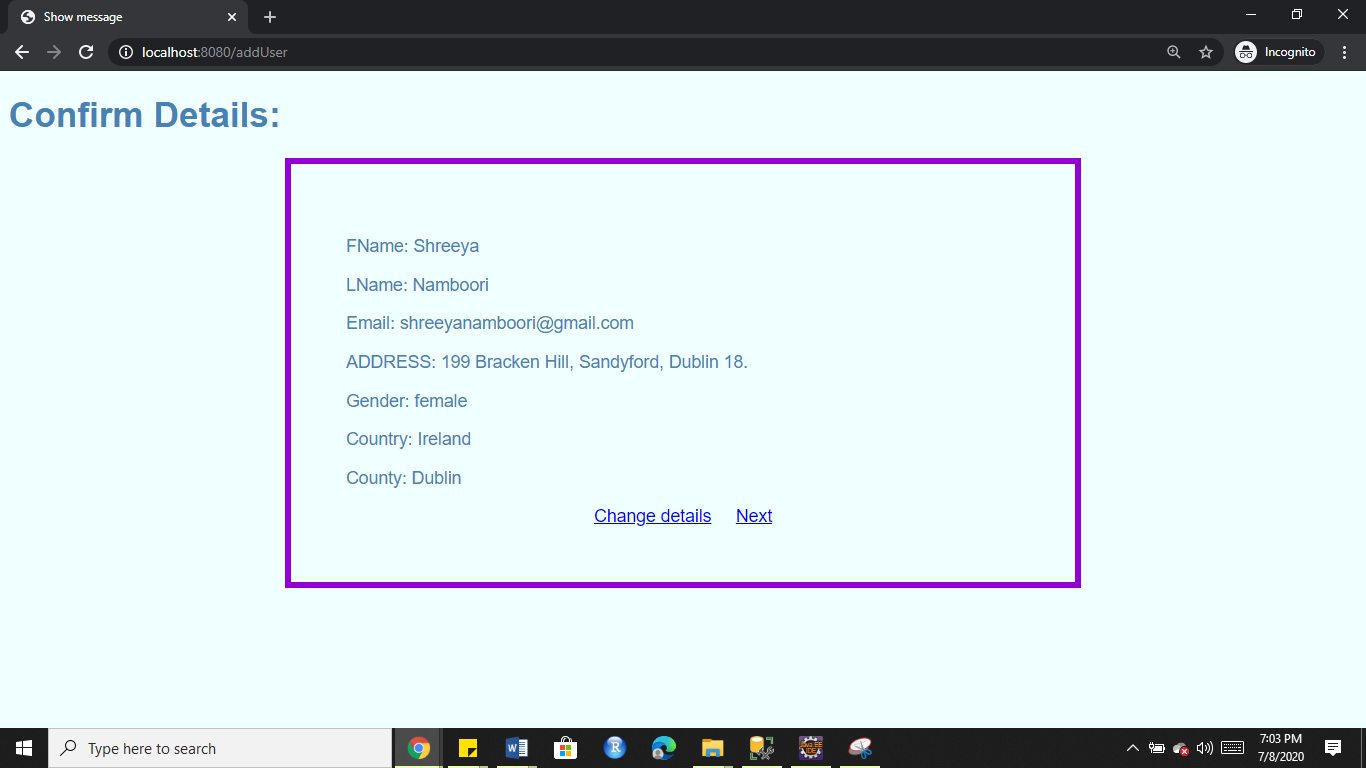
Note: MyController.java is responsible for setting the form data from addUser.html in User.java (bean class) and then passing those values to showMessage.html with the help of User.java.



**Figure: Image 9**

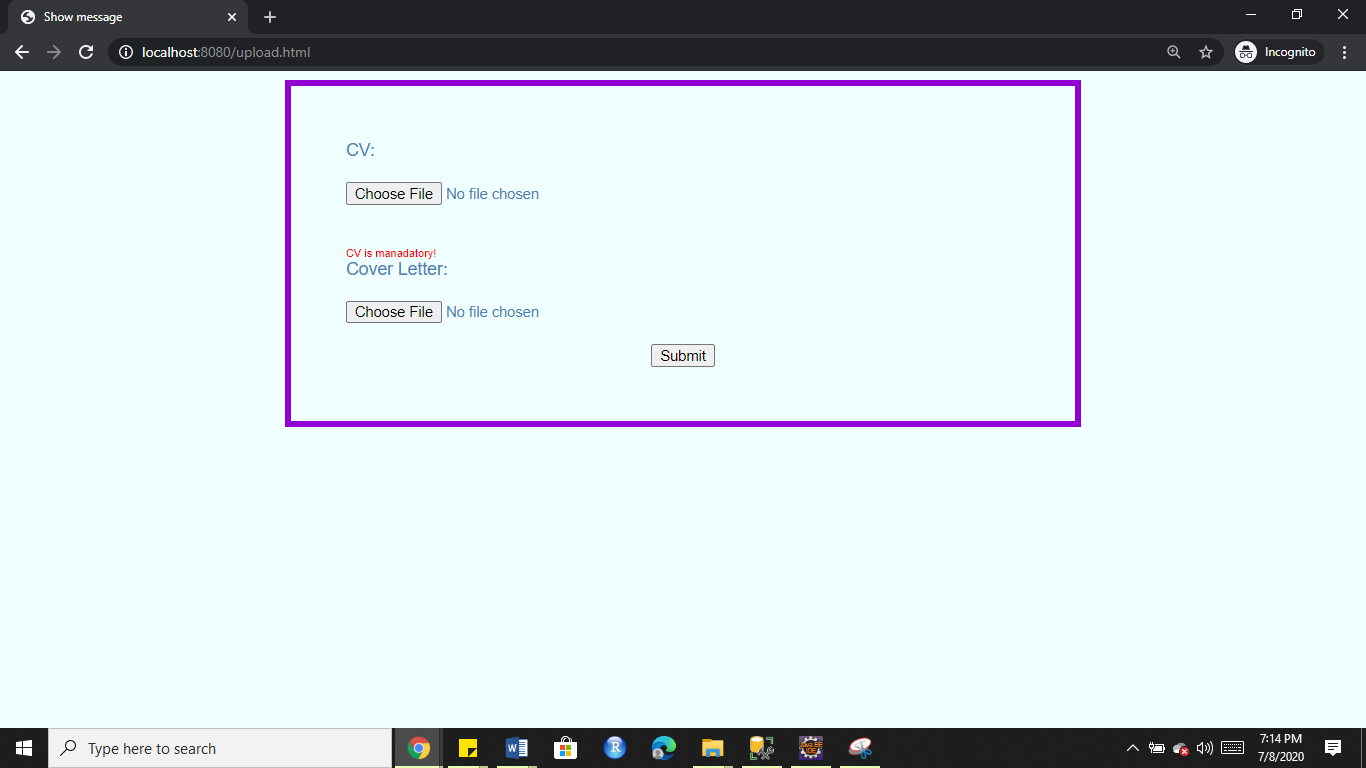


**Figure: Image 10 (MyController.java class in Eclipse)**



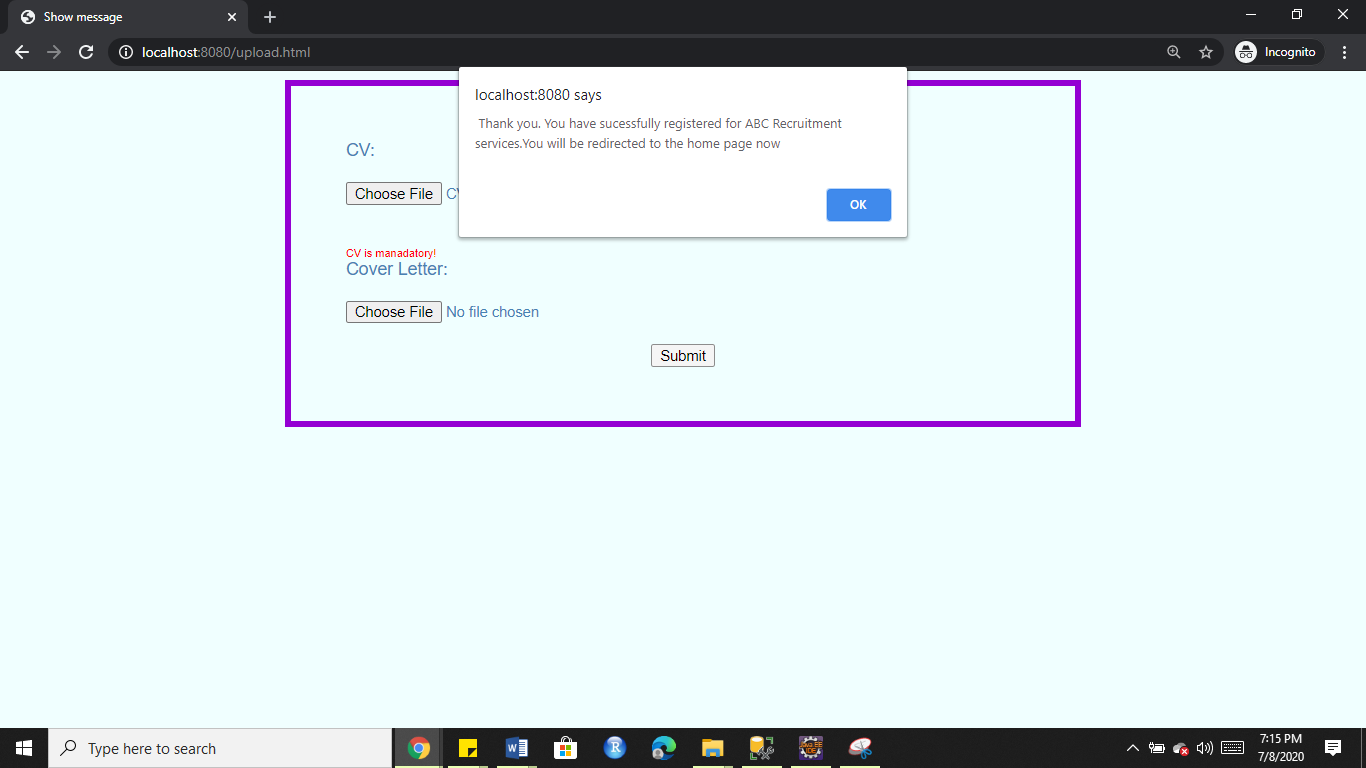
**Figure: Image 10**

5. If the details are incorrect then the user can click the change details link which will take them to the form again and if the details are correct the user can continue forward with the next link. After next is clicked the new page for uploading a CV and Cover Letter appears ( having JavaScript validation for CV as uploading CV is mandatory for finding a job)



**Figure: Image 11**

Once the CV is selected and submitted (Cover Letter is not mandatory) a popup confirming the submission appears and the page is redirected to the home page again (Image 12).

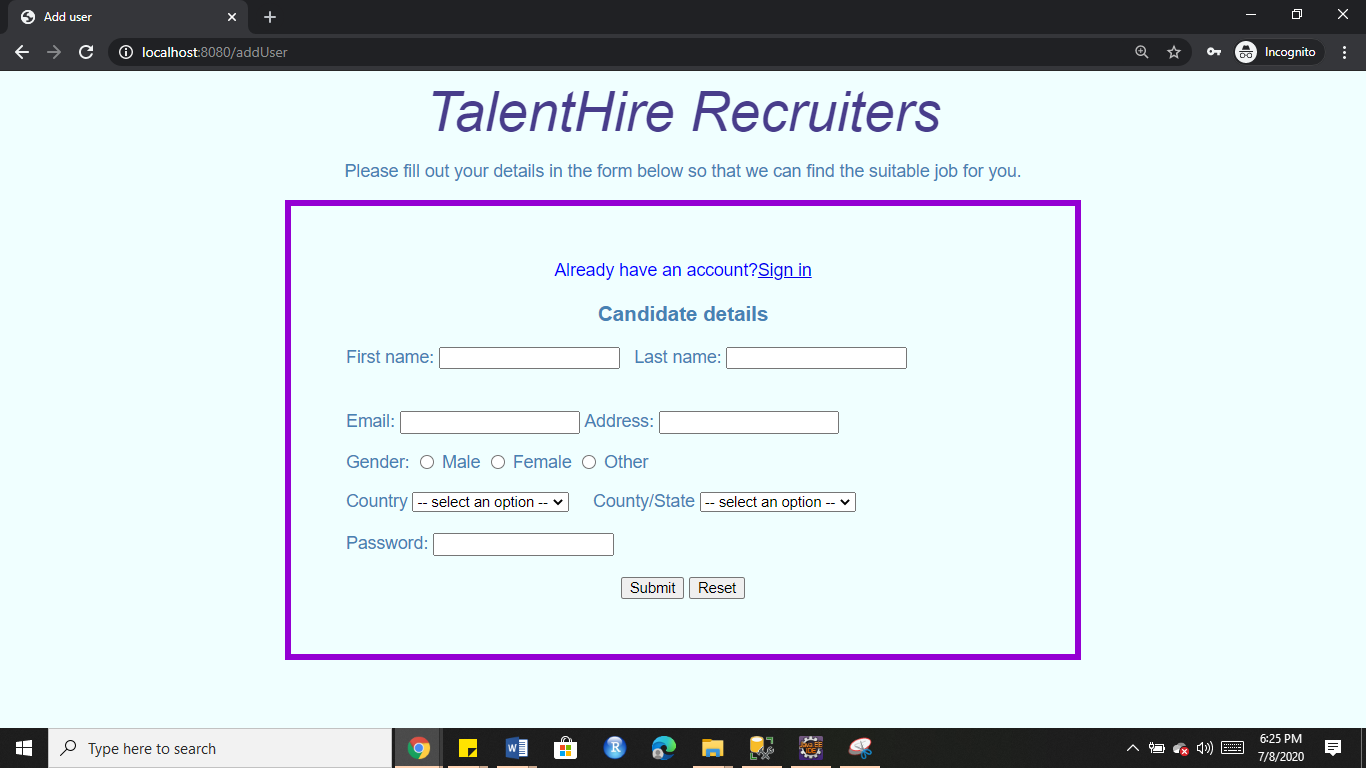


**Figure: Image 12**

5. This concludes the end of Flow 1 of the website.

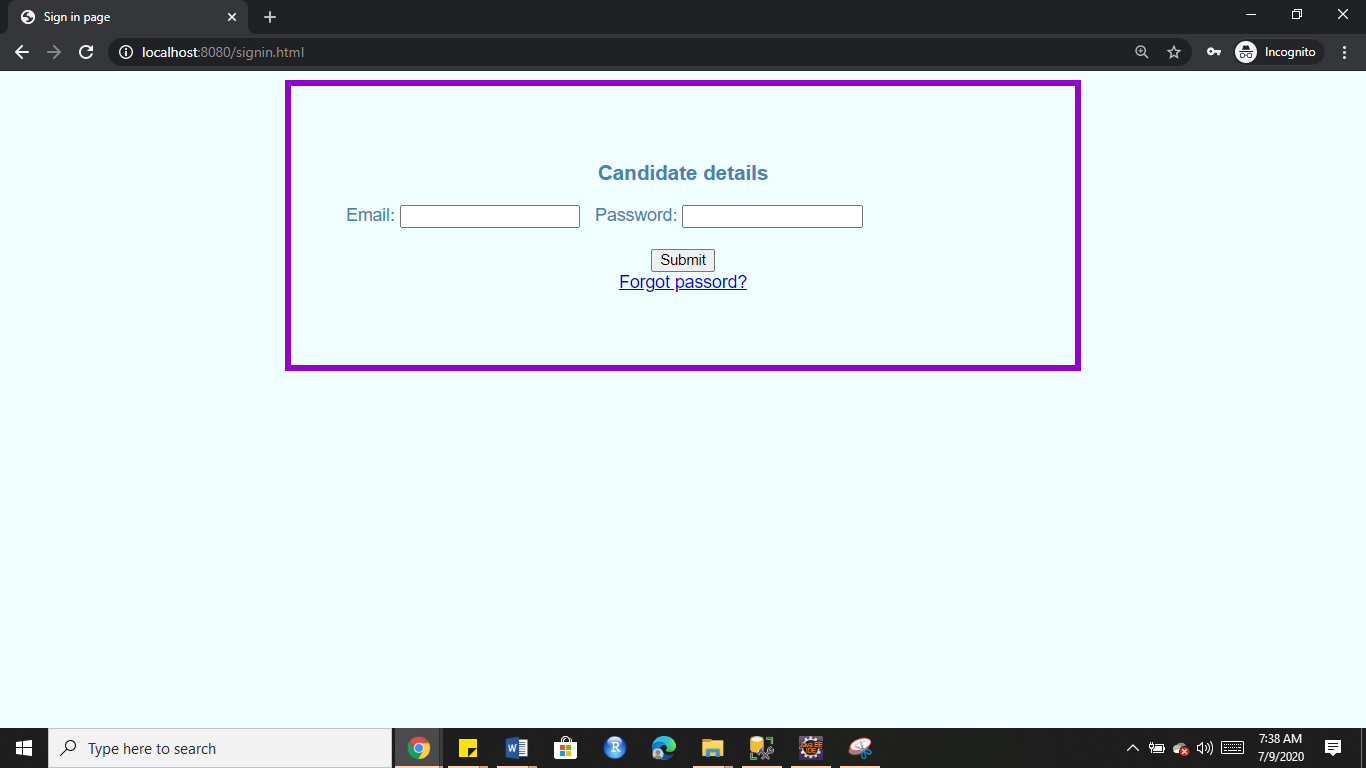
**B. Flow 1 (Sending mail using Java mail service)**

1. When the user has clicked on the tab Upload your CV and reached the form but they don’t want to signup as they had already done it before, in that case, they can click on the Sign-in link given at the top of the form shown below (Image 13).



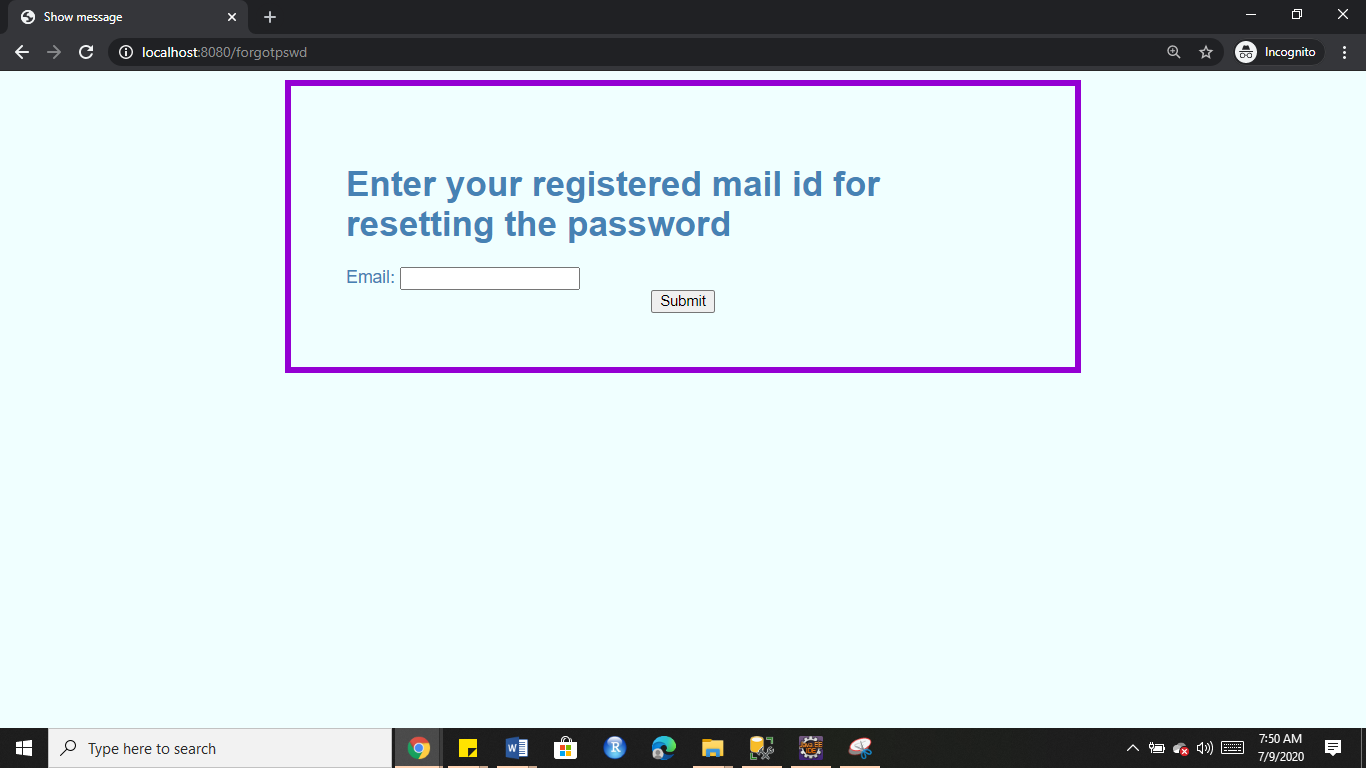
**Figure: Image 13**

2. The Sign-in link redirects to a Sign-in page (signin.html) which consists of two fields email and password. If the user forgets the login password they can click on the ”Forgot password?” link (Image 14).



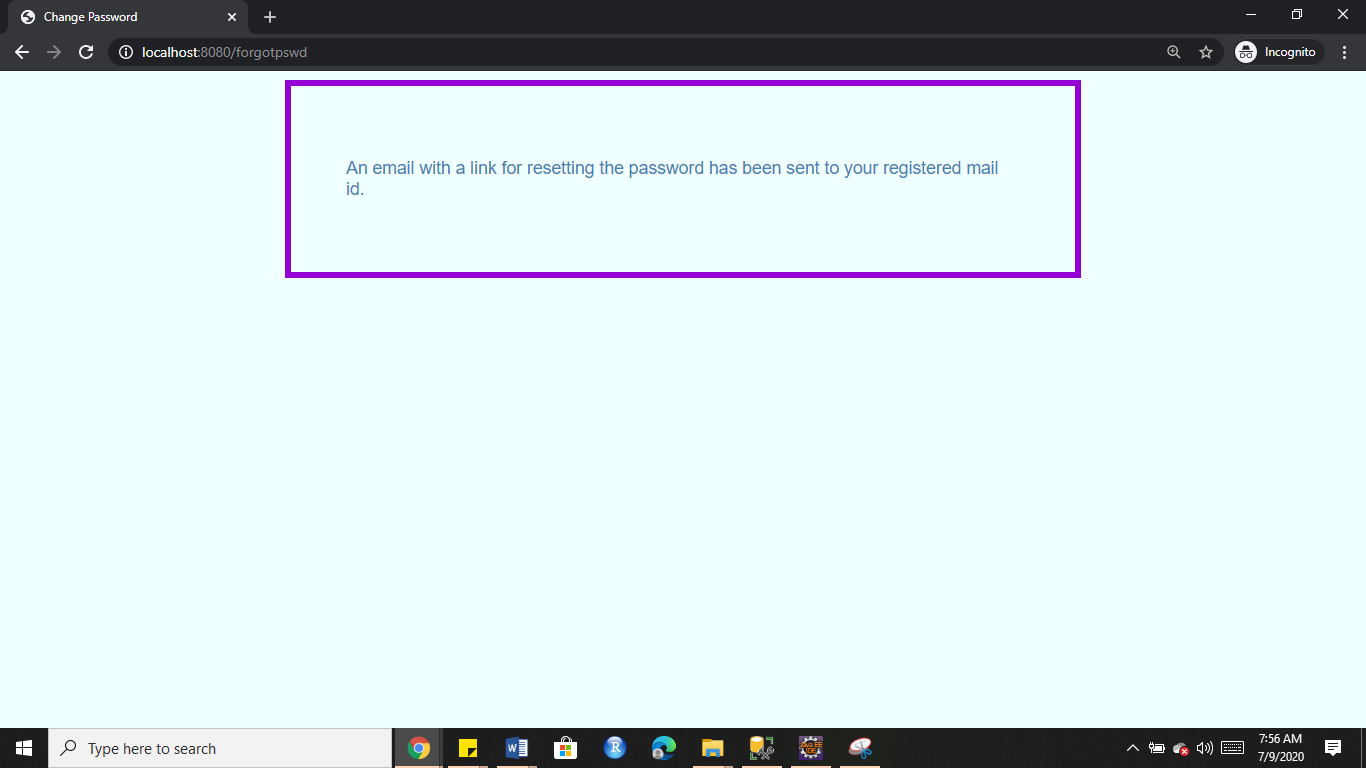
**Figure: Image 14**

3. The “Forgot password?” link redirects to a new page (forgotpswd.html mapped with /forgotpswd) which asks the user to enter their registered mail id (Image 15).

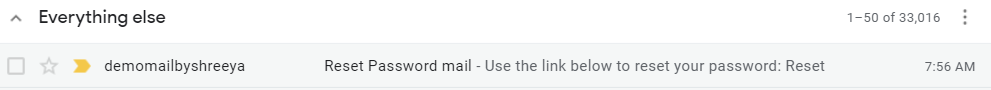


**Figure: Image 15**

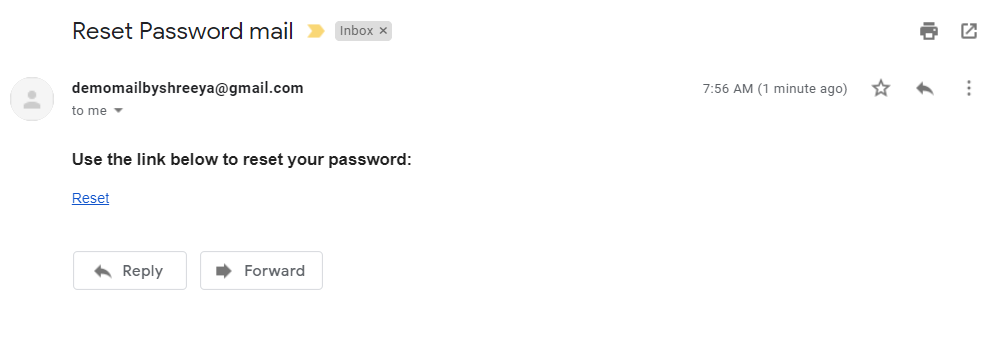
4. Once the user has submitted the mail id they then receive an email for resetting the password on their mail and the page is redirected to a new page (changepswd.html also mapped with /forgotpswd) after the email is successfully sent.



**Figure: Image 16**



**Figure: Image 17 (Email received on the user mail id by demomailbyshreeya@gmail.com (through java service))**



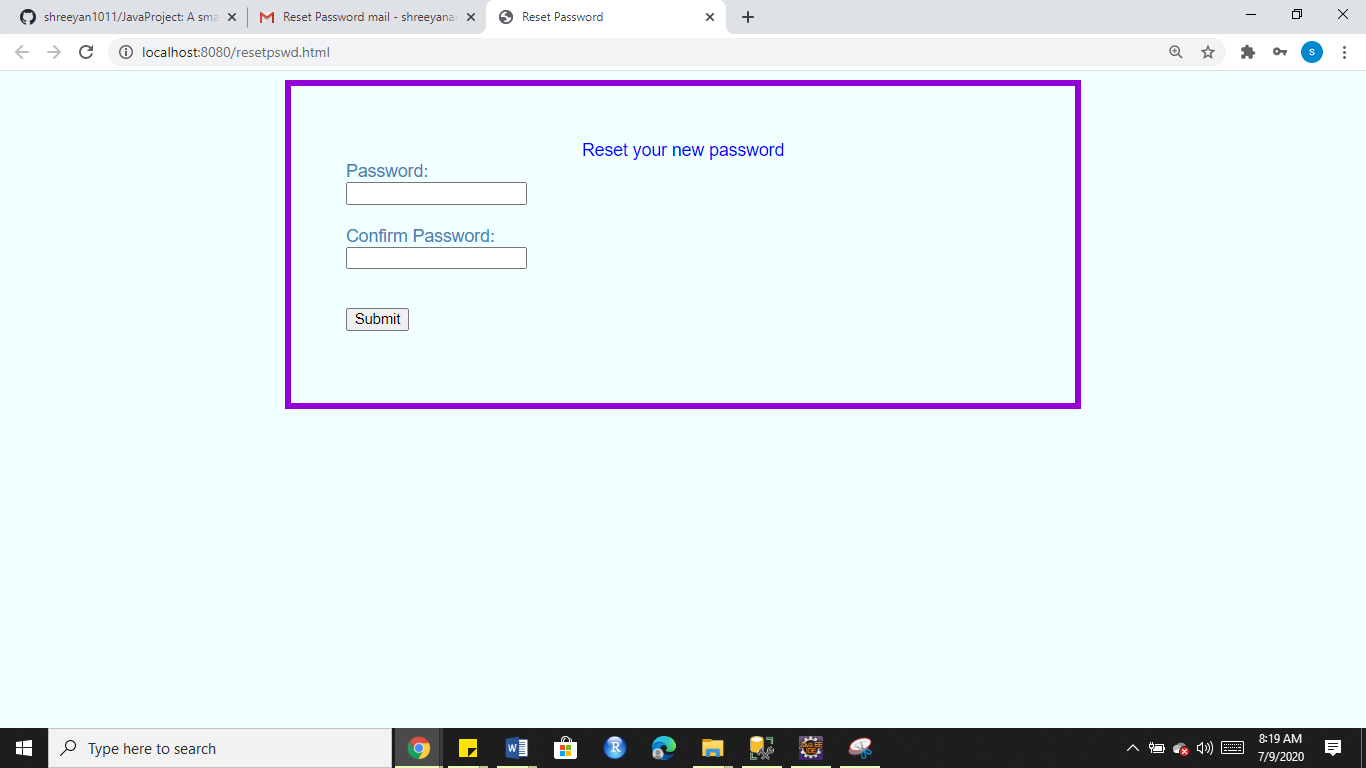
**Figure: Image 18 (Email content)**



**Figure: Image 19**

Note: Emailcontroller.java (Image 19) is responsible for getting the email id from the forgotpswd.html page and redirecting to changepswd.html after successfully sending an email.

5. Once the reset link in the mail is clicked user is redirected to a page (resetpswd.html) where they can reset their password.



**Figure: Image 20**

6. This concludes the end of Flow 2.

**Note:**

**The project is still very much in development and will be updated periodically in github along with the documentation.**