***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 by creating 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, MSSQL Server, and JavaScript.

**Functionality:**

There are currently three functional flows available for the website.

**A. Flow 1 (Submitting user information and String it in Database)**

**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, Sector, and Sign-in). The home tab is active (using CSS) as the present page is home.html (Image 1).



**Figure: Image 1**

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

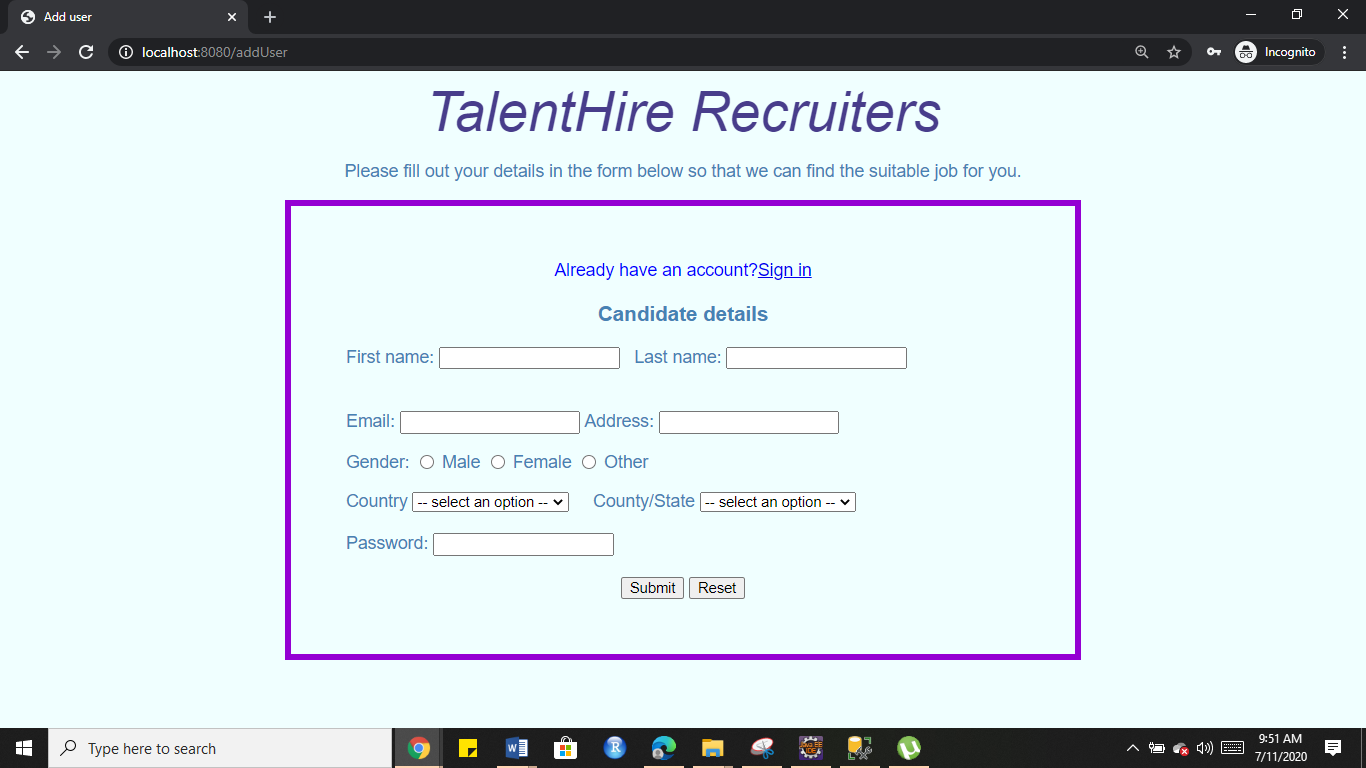


**Figure: Image 2**



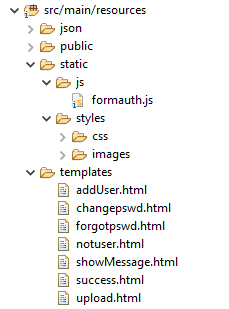
**Figure: Image 3**

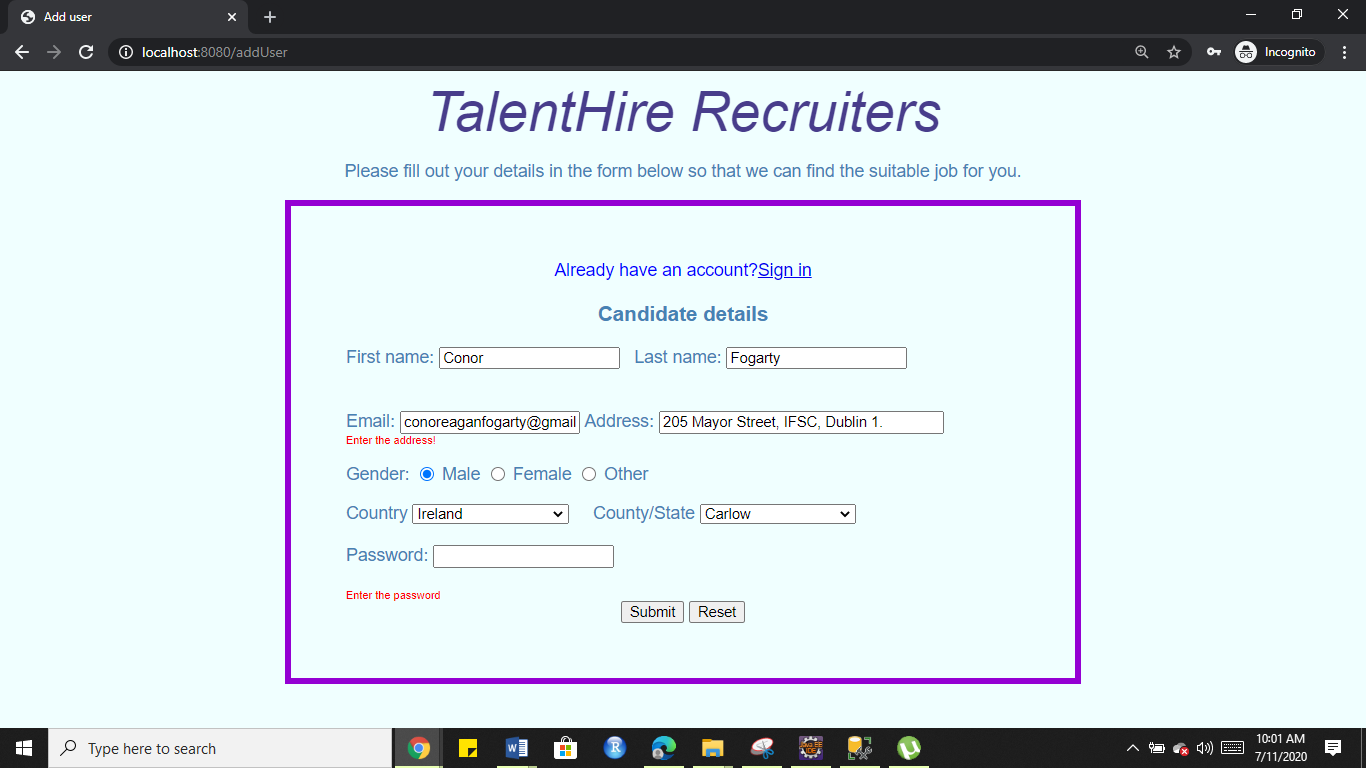
3. When Upload your CV is clicked the page is redirected to addUser.html (mapped with /addUser for Controller) that contains a form as shown below in Image 4.



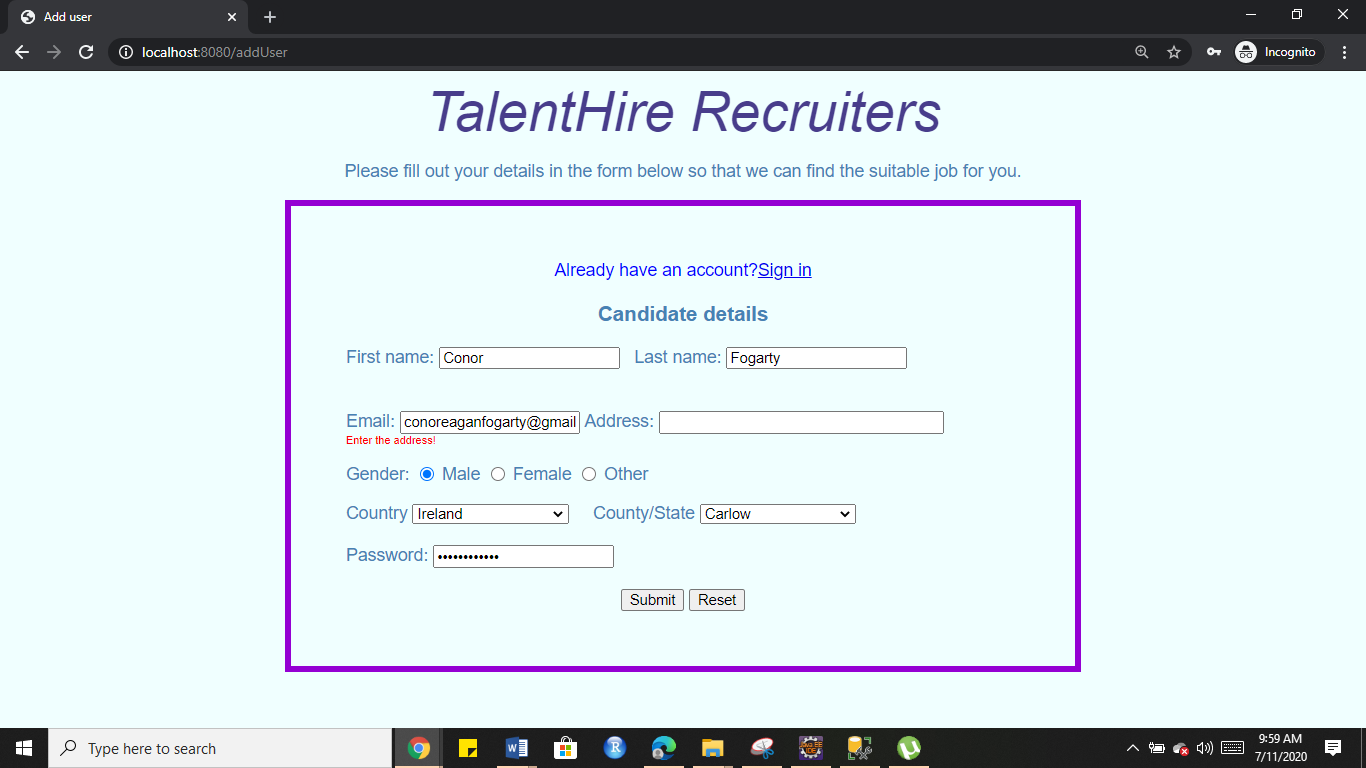
**Figure: Image 4**

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

**Figure: Image 5 ( Static resources location)**

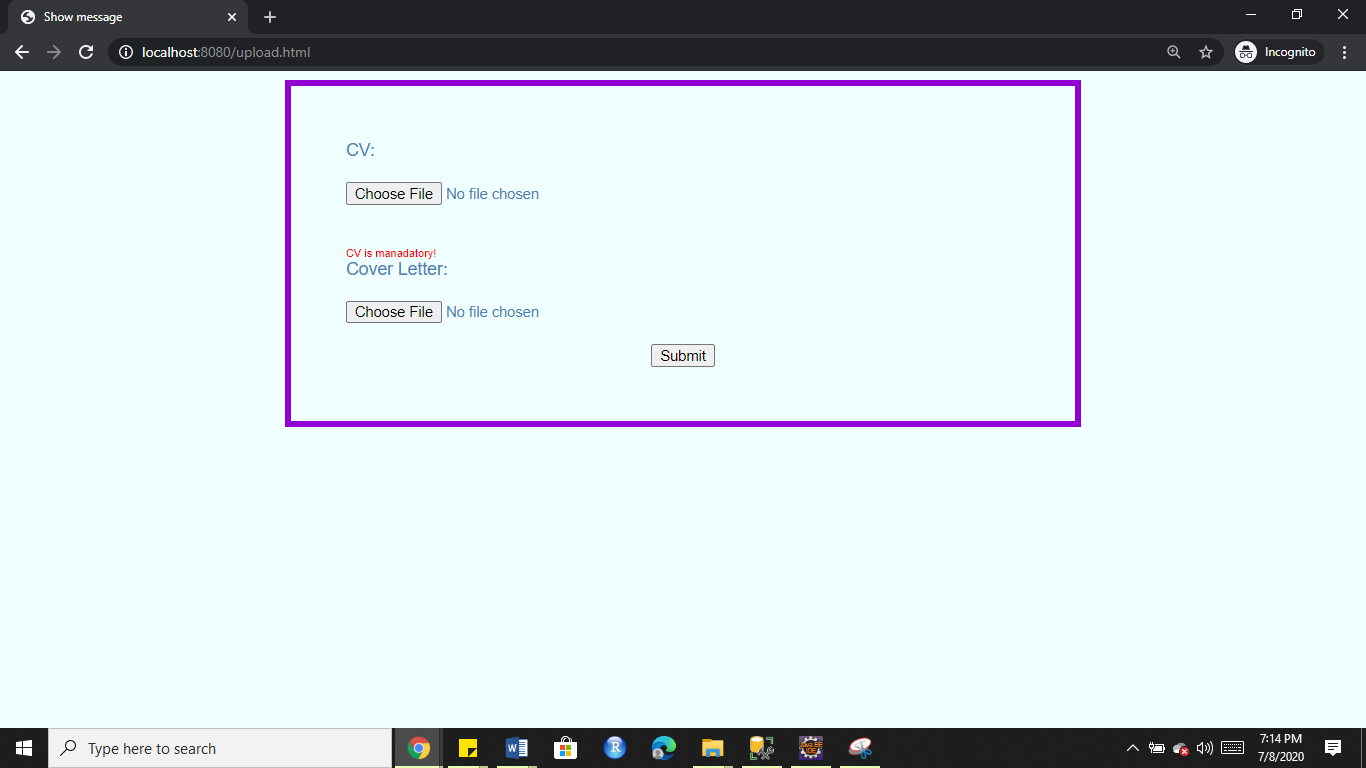


**Figure: Image 6**



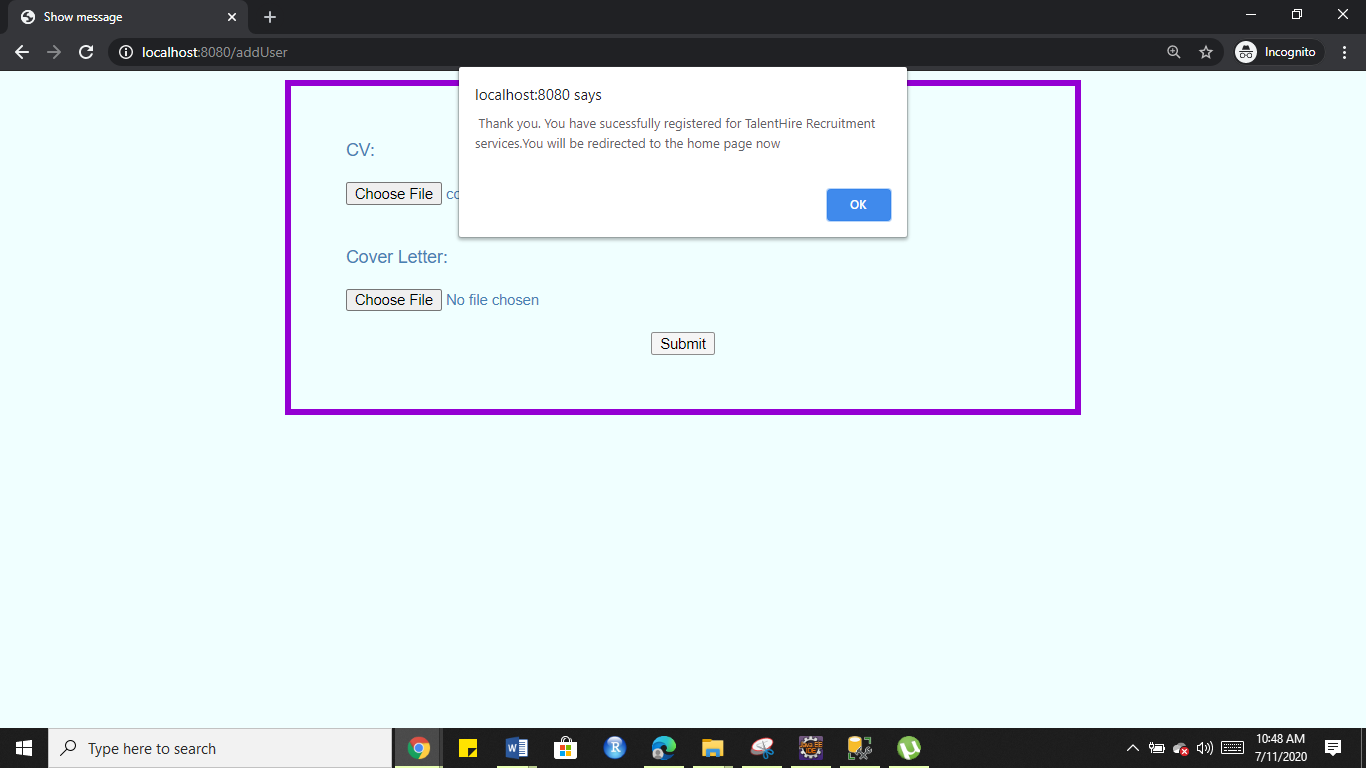
**Figure: Image 7**

4. Once all the required fields are filled the form is successfully submitted and the user is redirected to a new page (upload.html) where they can upload their CV and Cover Letter. ( having JavaScript validation for CV as uploading CV is mandatory for finding a job (refer Image 8))

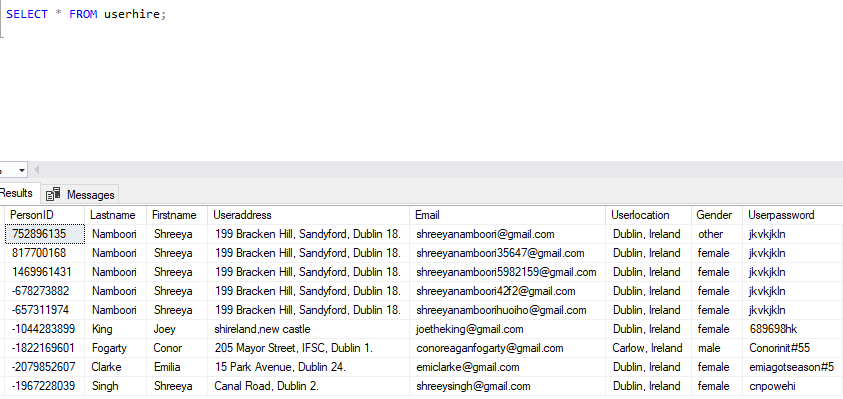


**Figure: Image 8**

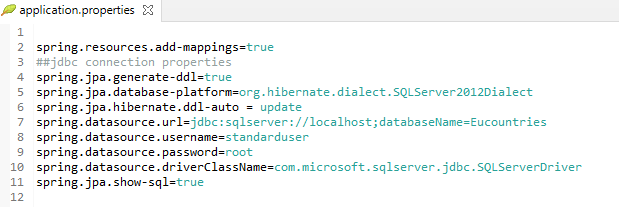
Once the CV is selected and submitted (Cover Letter is not mandatory) a popup confirming the submission appears and the page is redirected to the home page again (Image 9).



**Figure: Image 9**

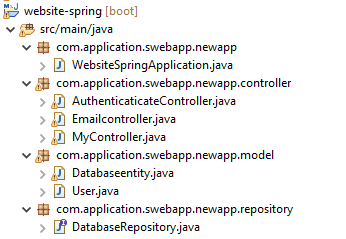


**Figure: Image 10 (data stored in the database from HTML form)**

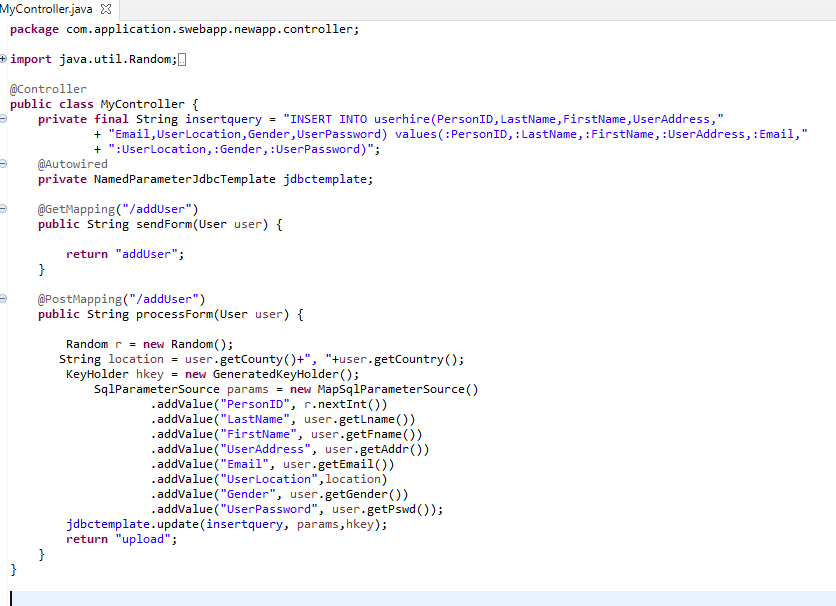


**Figure: Image 11 (SQL properties set in application.properties file for JDBC connection)**

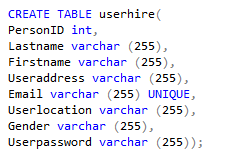
Note: MyController.java is responsible for setting the form data from addUser.html to User.java (bean class) and then passing those values to be inserted in MSSQL server database table (named userhire) through established JDBC connection by setting properties in application.properties.



**Figure: Image 12 (src folder structure in Eclipse)**

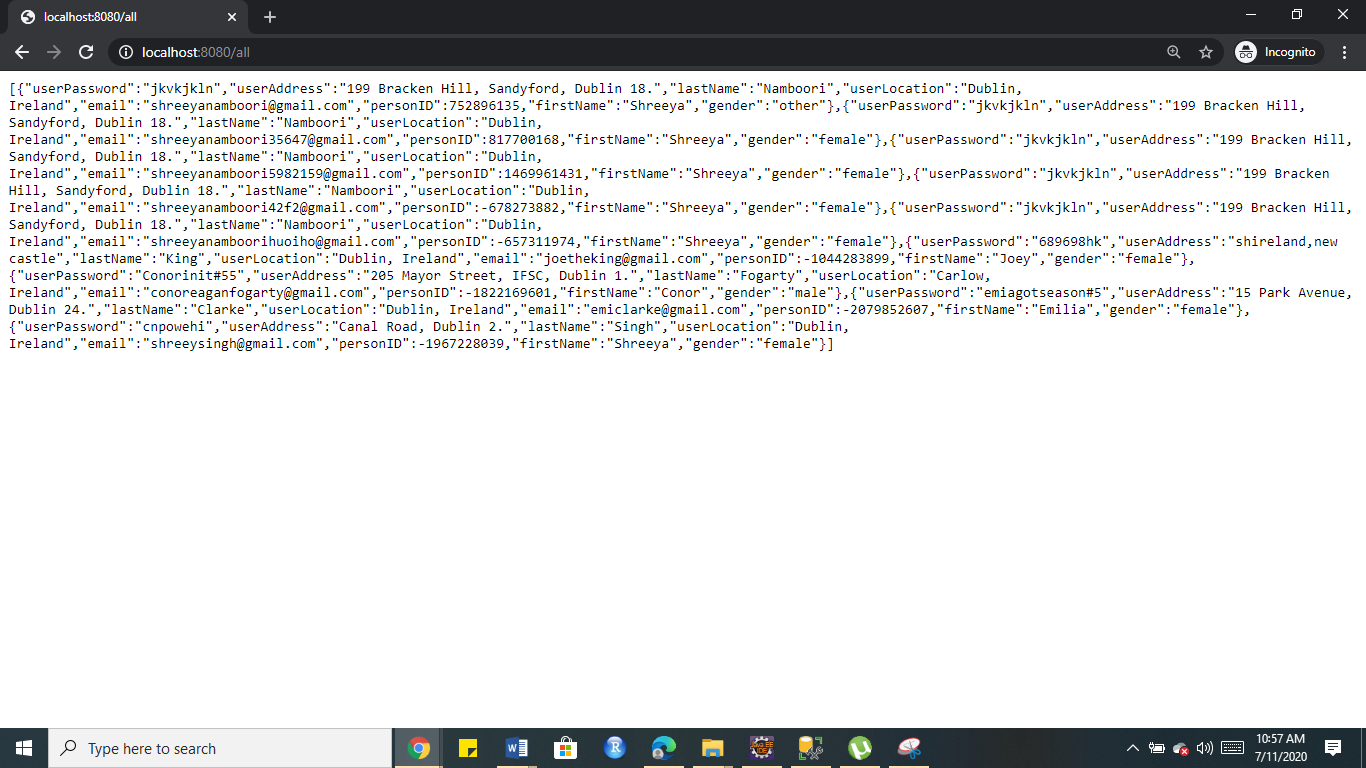


**Figure: Image 13 (MyController.java class in Eclipse)**



**Figure: Image 14 (SQL query for creating table “userhire”)**

Alternatively, all the table entries could be seen on the browser too (using DatabaseRepository Interface (extending Crud repository), Databaseentity (@Entity), and AuthenticateController(@Controller)). The link <http://localhost:8080/all> gives all the entries in the table “userhire” on the browser in JSON format therefore there is no need to open and check the database every time (refer to Image 15).



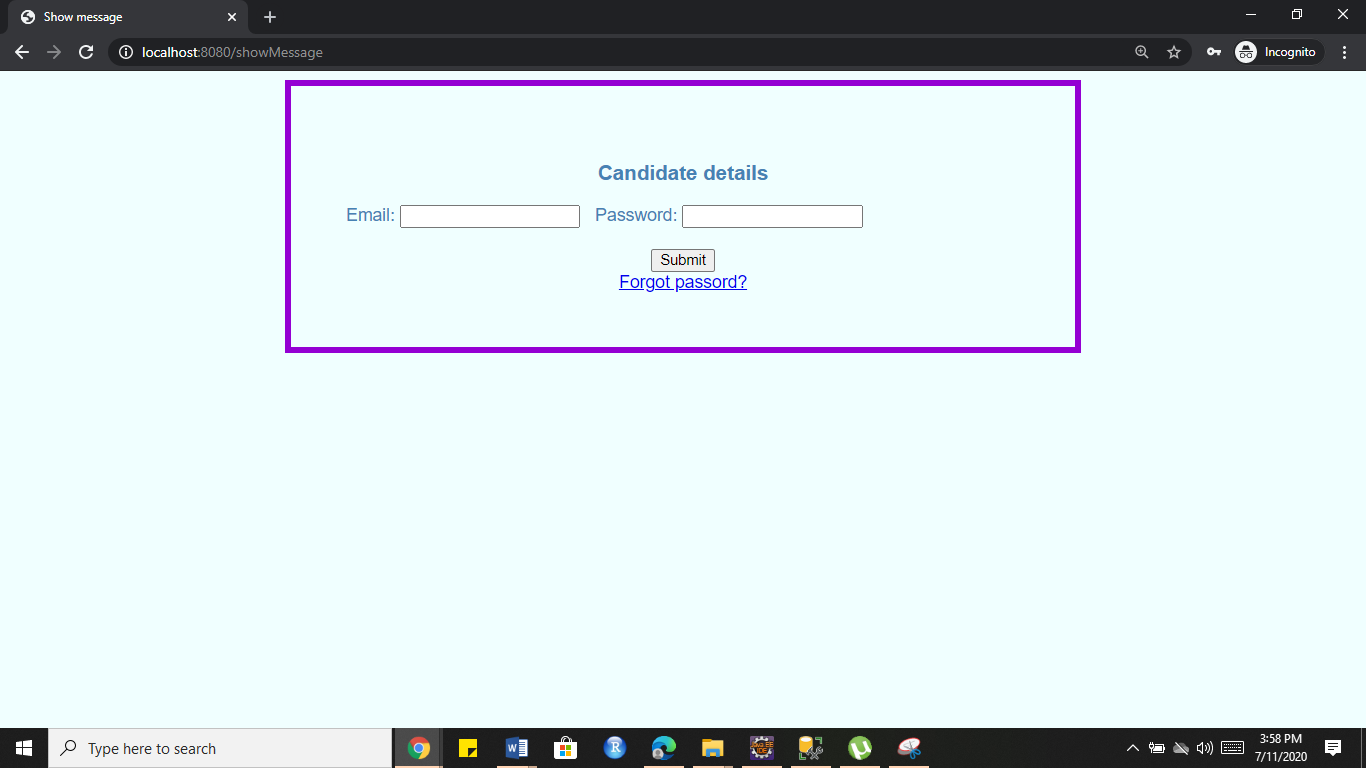
**Figure: Image 15**

5. This concludes the Flow 1 of the website.

**B. Flow 2 (Authenticating user and password during Sign-in)**

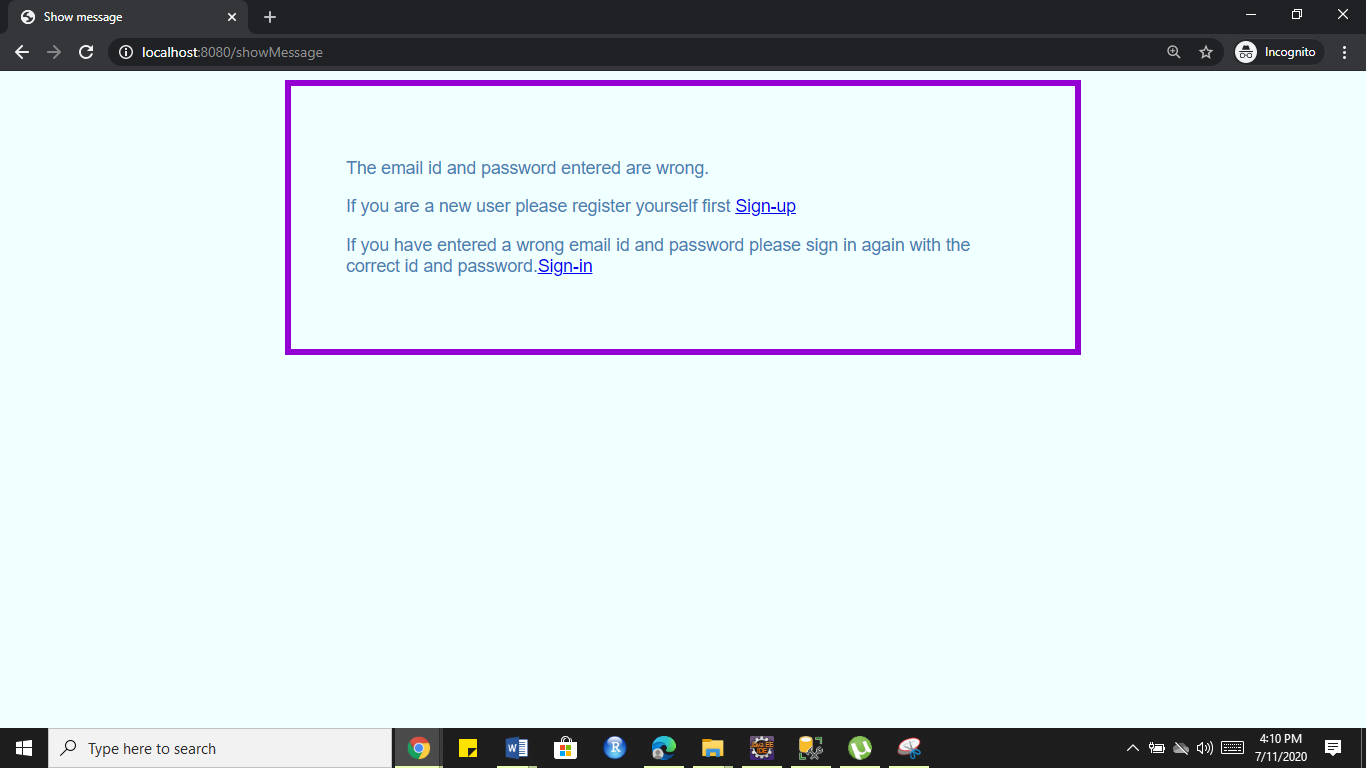
1. Users can Sign-in either from the clicking the link sign-in on the form that appears after clicking upload your CV tab or directly clicking the sign-in tab on the homepage (refer Image 1 and Image 4).

2. Once the Sign-in link or tab is clicked the users are redirected to the sign-in page (showMessage.html mapped with /showMessage) where they are asked to enter their registered email id and password (Image 16).



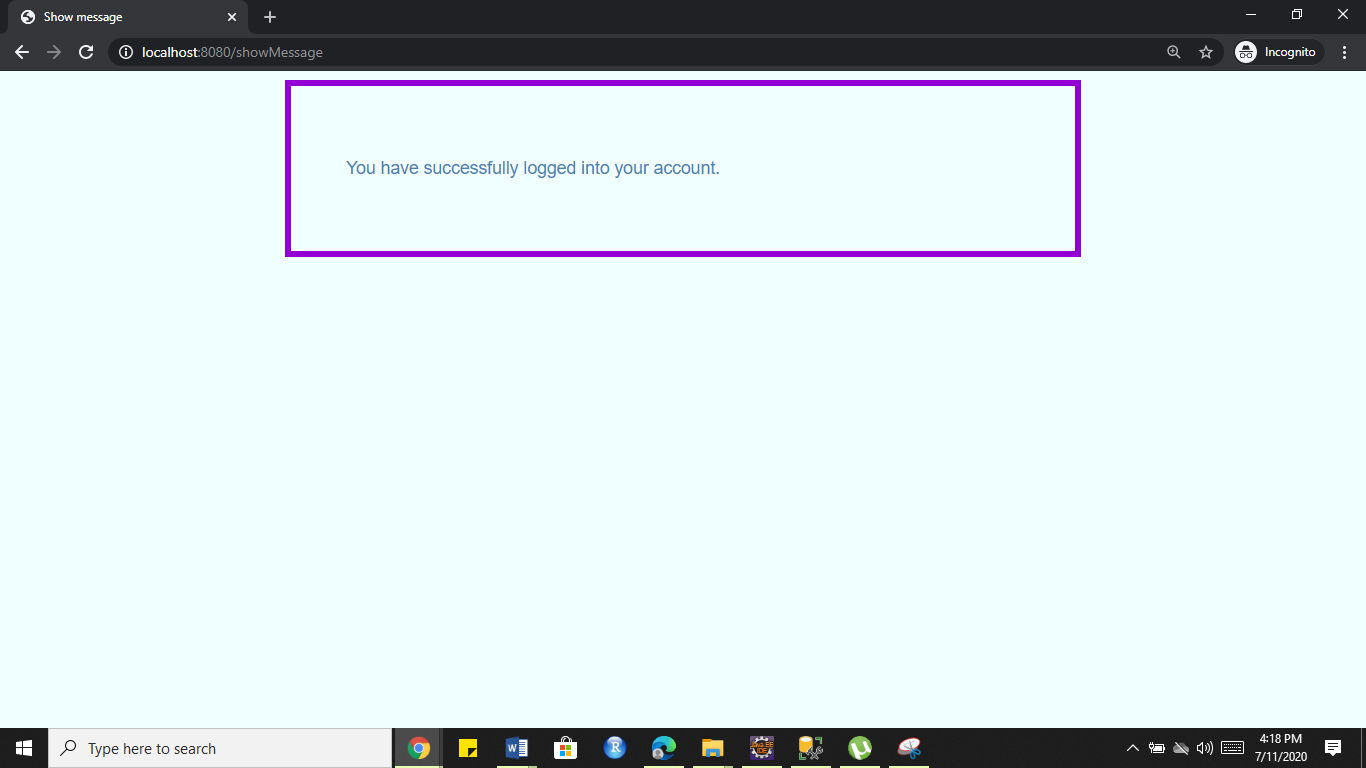
**Figure: Image 16**

3. When the users enter a password (wrong password) or email id (which is not present in the database), they are redirected to a new page (notuser.html) which asks them to sign up if they haven’t or sign-in using correct mail id or password (refer Image 17).



**Figure: Image 17**

4. When the user enters the correct mail id and password (as stored in the database during sign-up) they are successfully logged in (authenticatication done by AuthenticateController.java) and redirected to a success page (success.html) as shown in Image 18.

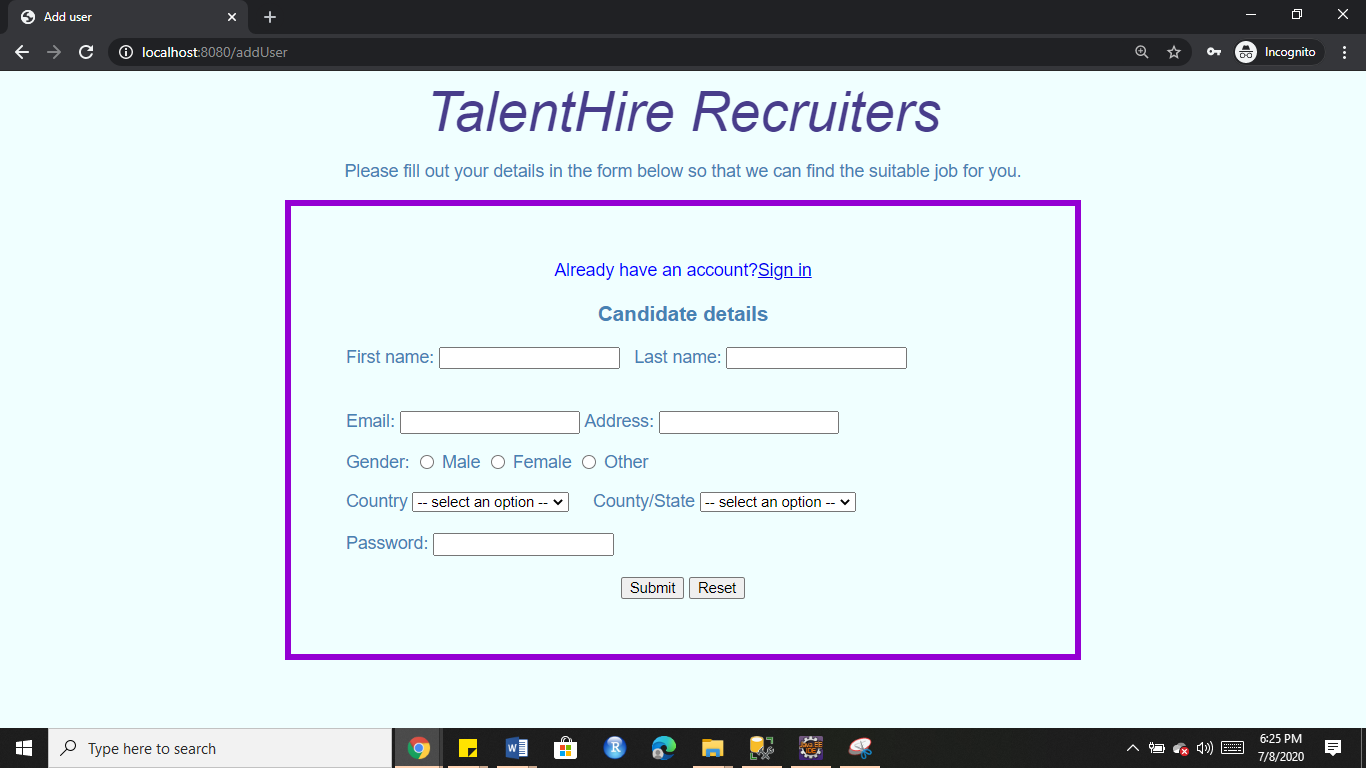


**Figure: Image 18**

5. This concludes the Flow 2 of the Website.

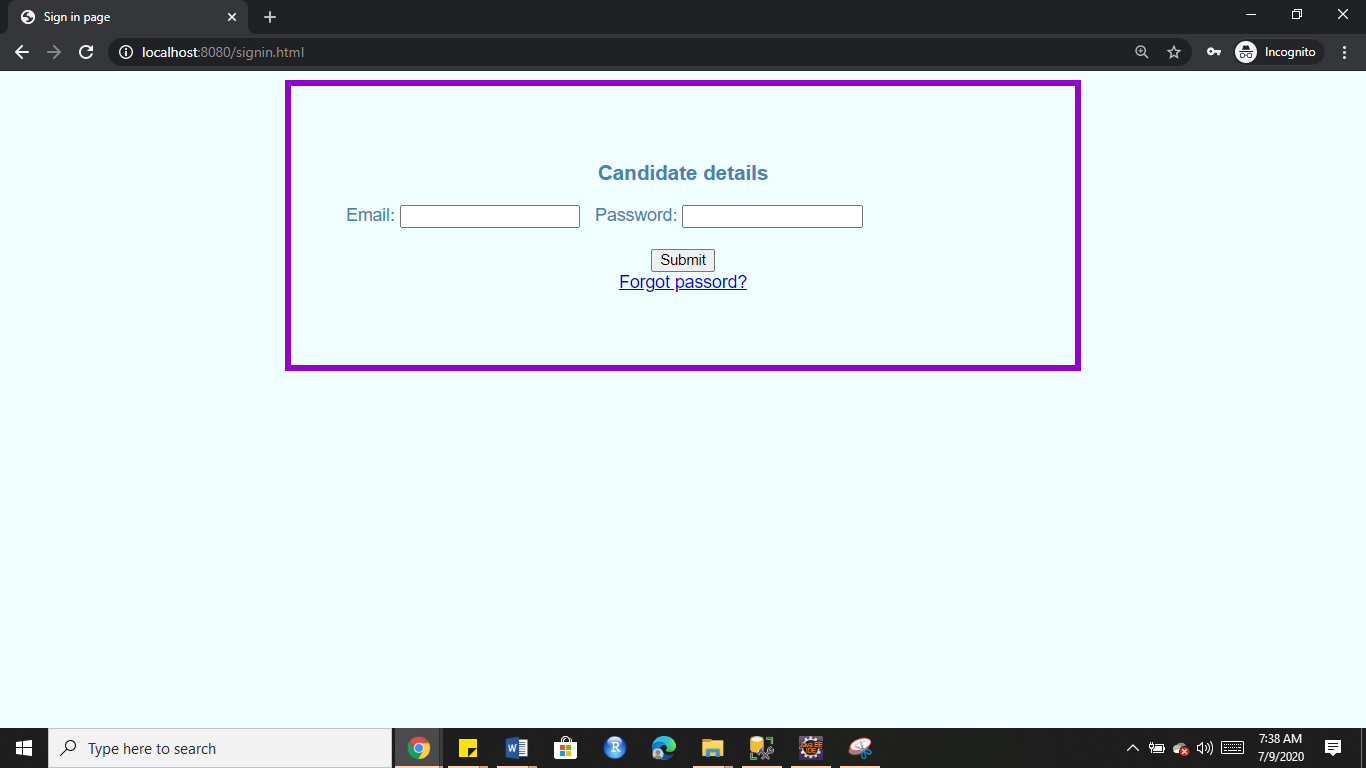
**C. Flow 3 (Sending mail using Java mail service)**

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 19).



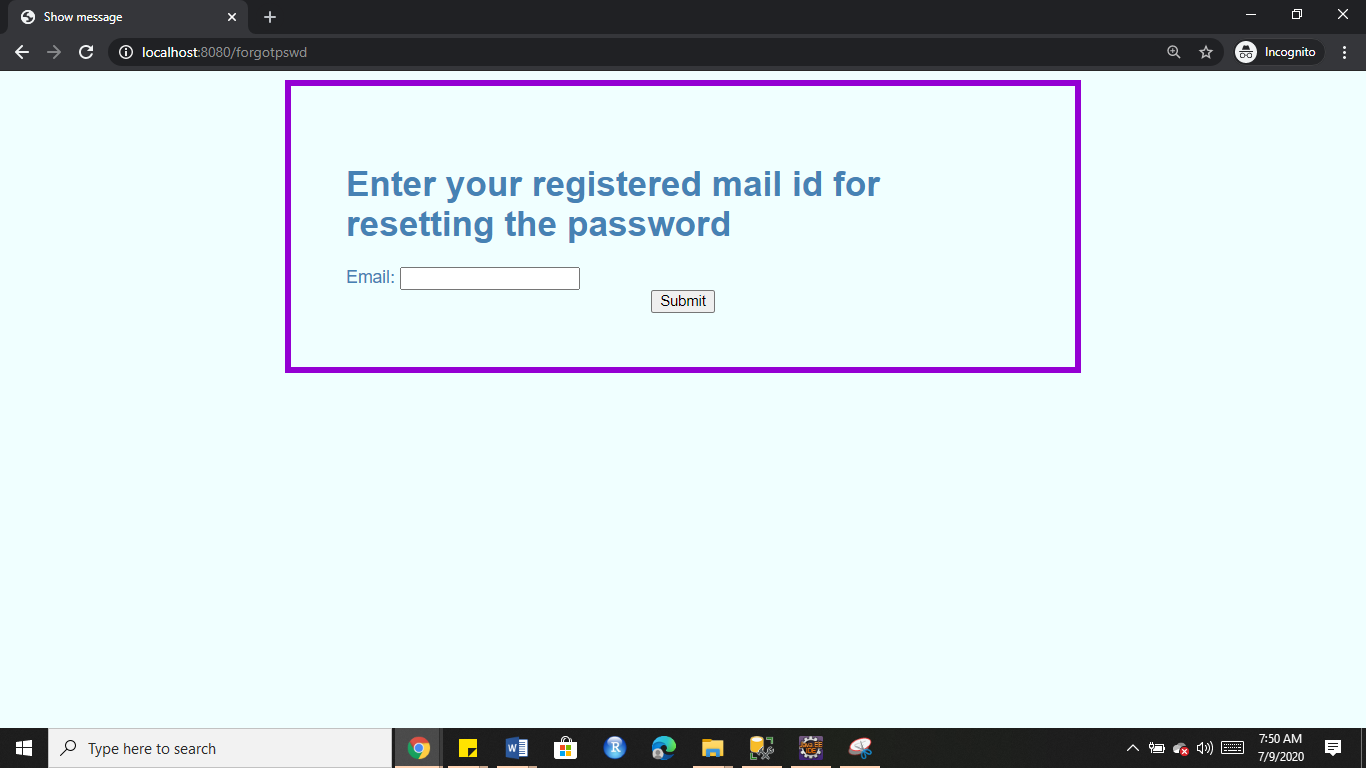
**Figure: Image 19**

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 20).



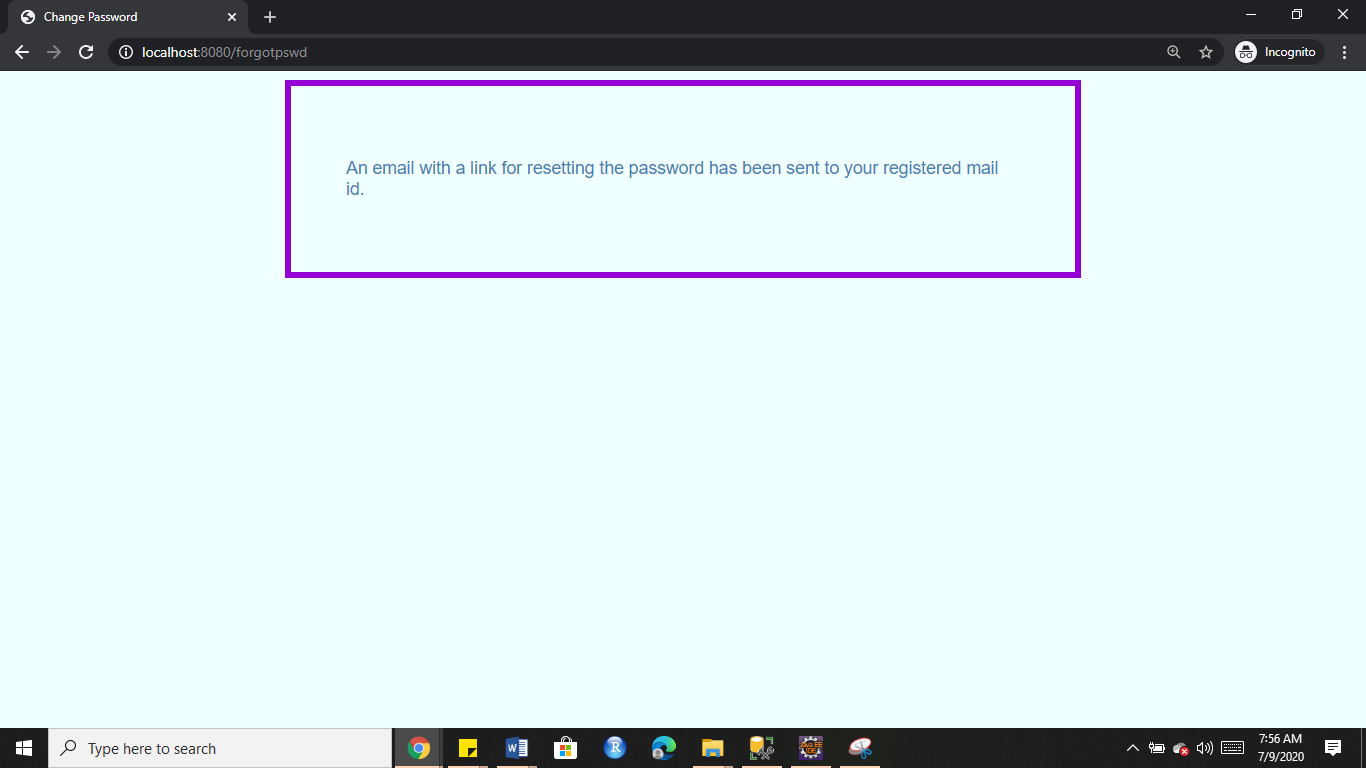
**Figure: Image 20**

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 21).

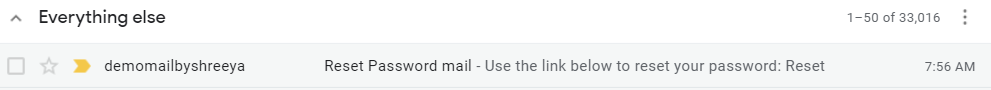


**Figure: Image 21**

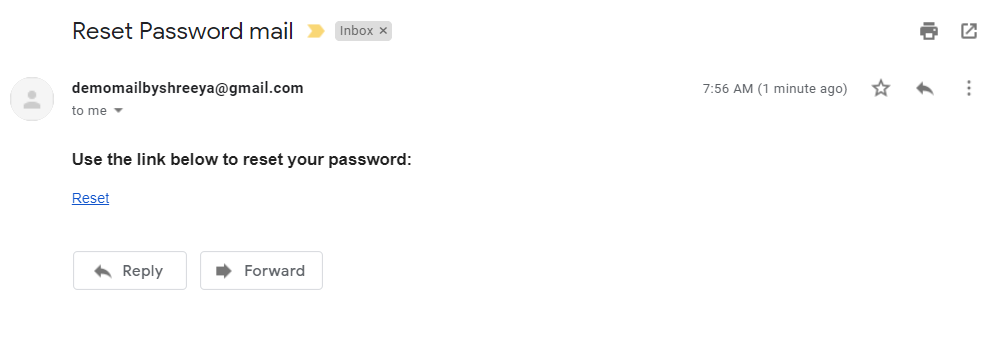
4. Once the user has submitted the mail id they then receive an email for resetting the password on their mail and the page is redirected to a new page (changepswd.html also mapped with /forgotpswd) after the email is successfully sent (Image 22).



**Figure: Image 22**



**Figure: Image 23 (Email received on the user mail id by demomailbyshreeya@gmail.com (through java service))**



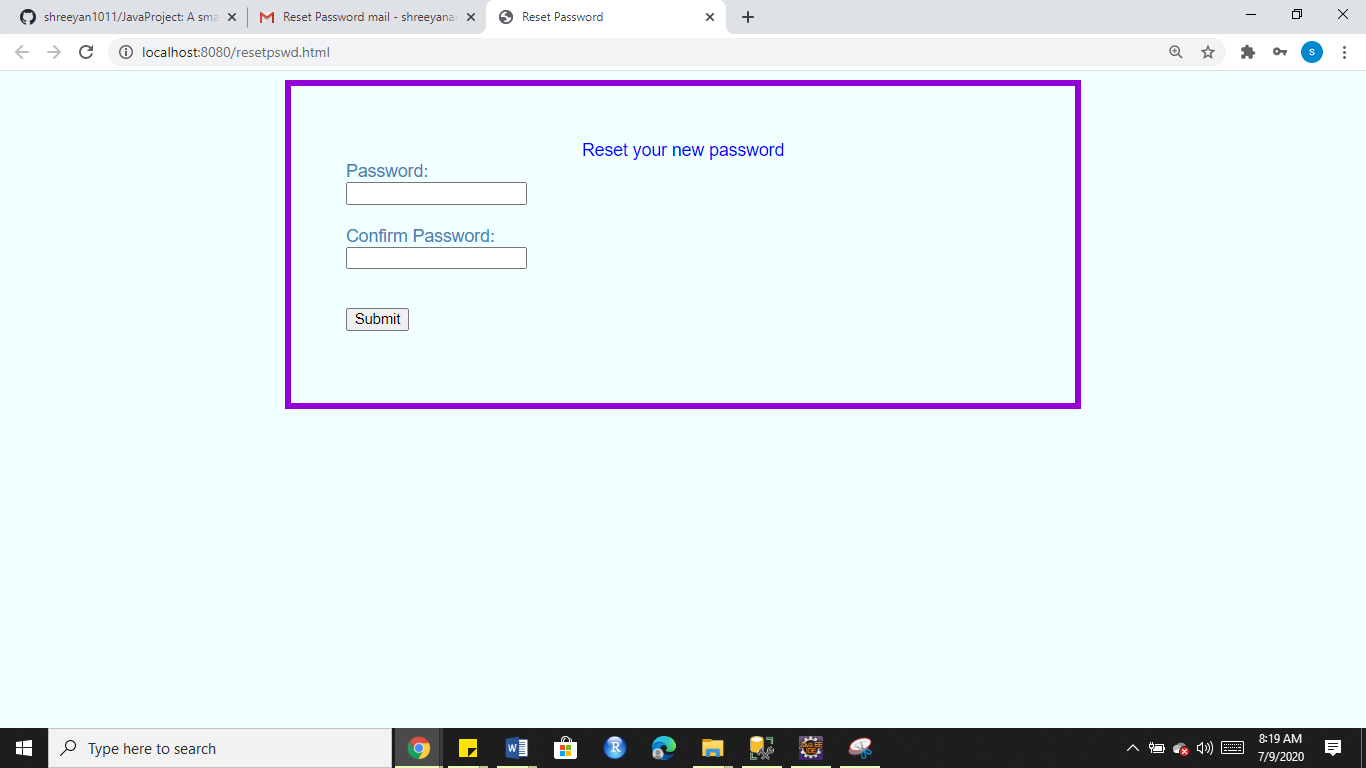
**Figure: Image 24 (Email content)**



**Figure: Image 25 (EmailController.java)**

Note: Emailcontroller.java (Image 19) is responsible for getting the email id from the form in 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 26**

6. This concludes the end of Flow 3.

**Note:**

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

