Project Report

|  |  |
| --- | --- |
| Product Name | Advanced Certificate in Web Development |
| Qualification Name (NICF) | NICF-Advanced Certificate in Infocomm Technology (Software & Applications) |
| Product Name | Web Development Foundations |
| Module Name (NICF) | NICF -Web Development Foundations |

|  |  |  |  |
| --- | --- | --- | --- |
| Student name | | Assessor name | |
| **Agung Yuda Pratama** | |  | |
| Date issued | Completion date | | Submitted on |
|  |  | |  |
|  | |  | |
| Project title | Design, Develop, Implement & Document Community Portal Website | | |

|  |
| --- |
| Learner declaration |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: 05-03-2023 |

Content

1. Project background
2. Project Objectives
3. Project Requirement Specification
4. Task 1
5. Task 2
6. Task 3
7. Task 4
8. Task 5
9. Task 6
10. Task 7
11. Task 8
12. Task 9
13. Task 10
14. Task 11
15. Project Background

We have been approached by ABC Jobs Pte Ltd to develop a community portal for software developer. This project will be continued through Module 3, Module 4, Module 5 and Capstone projects. For this module the scope is Designing, Developing, Implementing & Documenting a Spring Website. Project Scope is to design a Community Portal Similar to Linkedin.com. Users will be able to register on the portal using the Registration Page. Users of the portal can search for other users using various parameters like First Name, Last Name, Company Name, City. Users will be able to see the Public Profile of the user after searching for it. Portal allow user to login, request forgotten password and Update their profile information Project scope in this module is for Spring Website development.

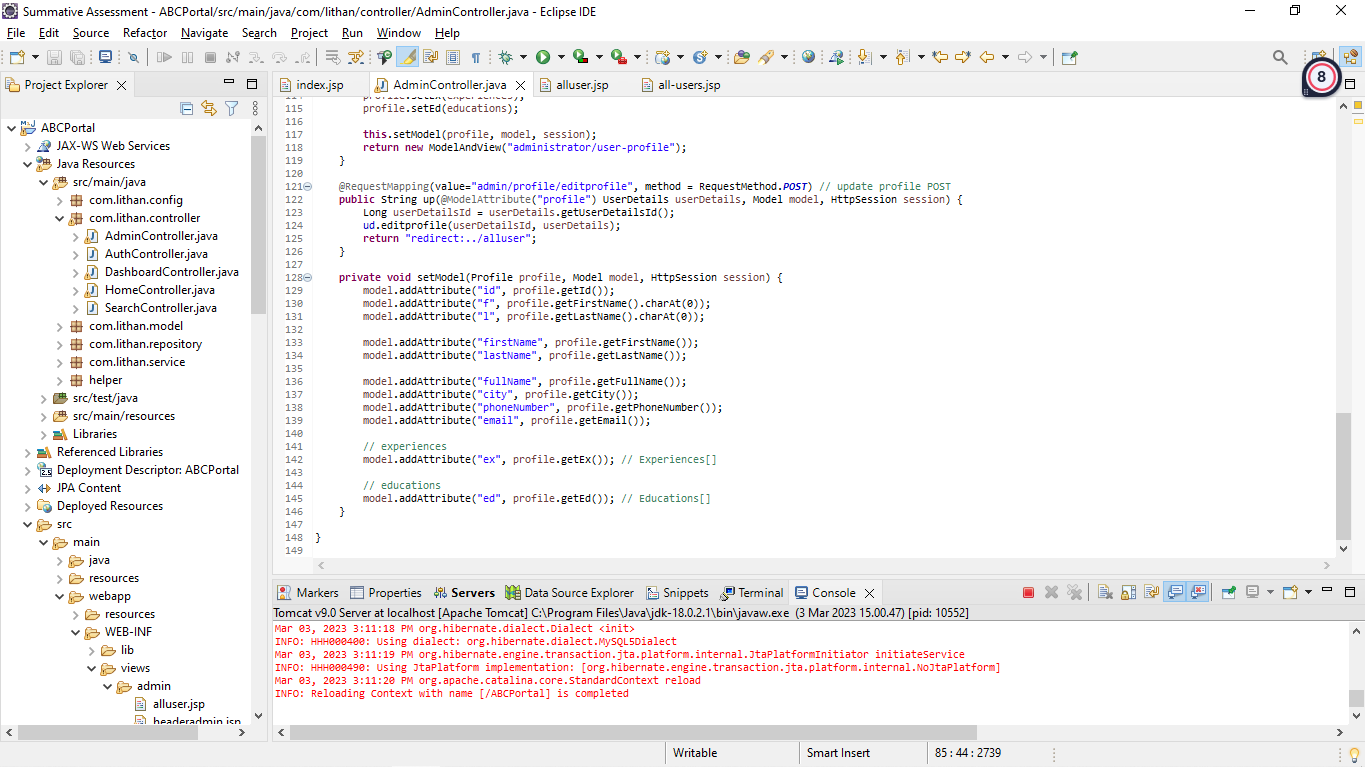
1. Project Objective

We need to design, develop, implement, test and document for ABC Jobs community

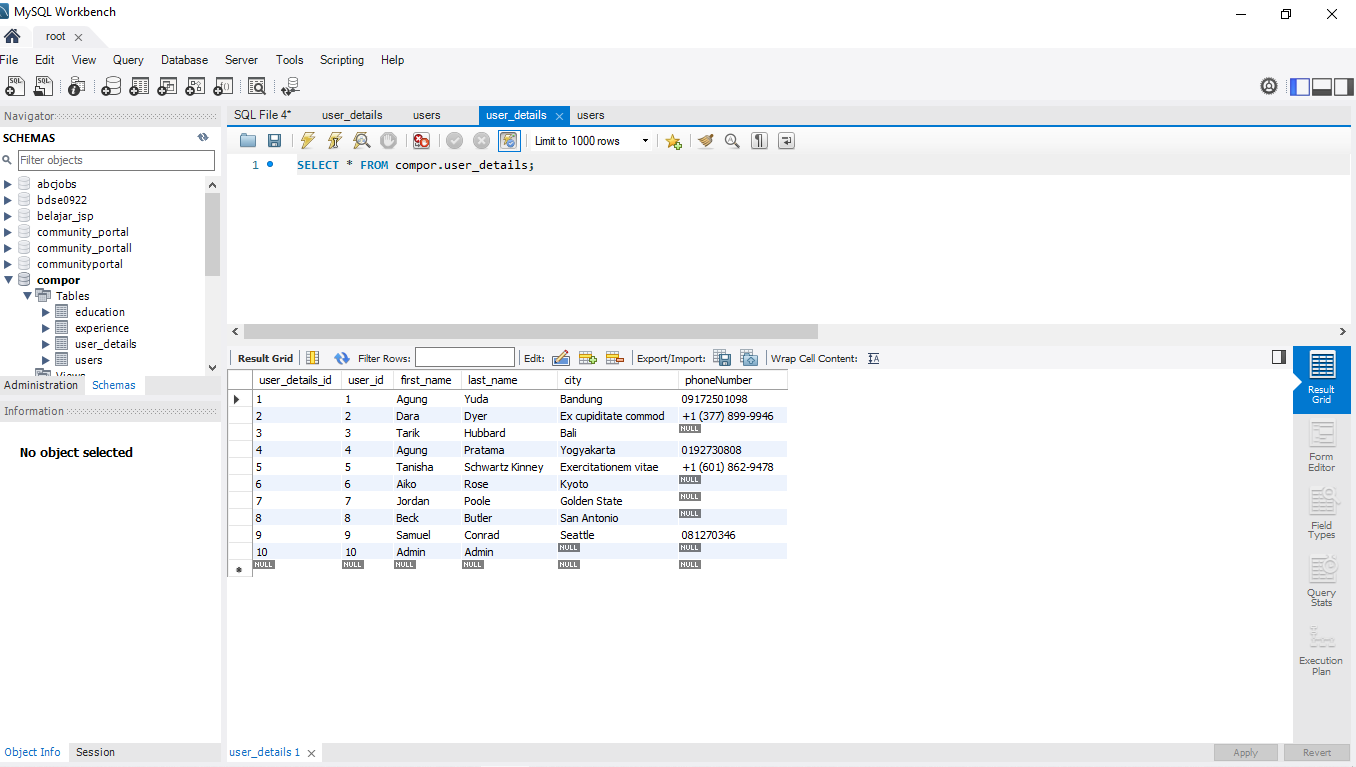
Portal using spring framework.

* To develop and design the community portal website functional requirements using Spring MVC framework
* To design the webpages using JSP
* To implement a java-based Spring website using the Eclipse
* To document the develop website using a document generator, Javadoc

1. Project Requirements Specifications
2. Eclipse



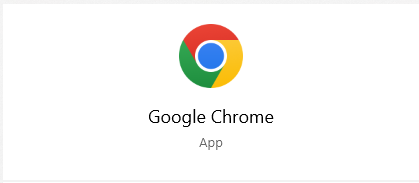
1. SQL Workbench



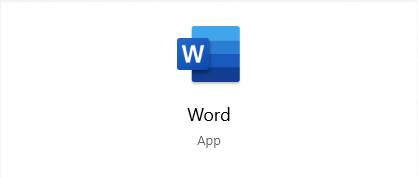
1. Apache Tomcat 9.0



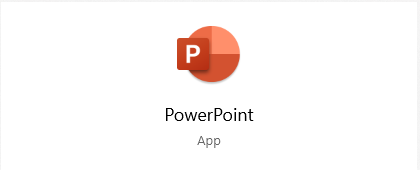
1. Google Chrome



1. Microsoft Word



1. Microsoft PowerPoint Presentation

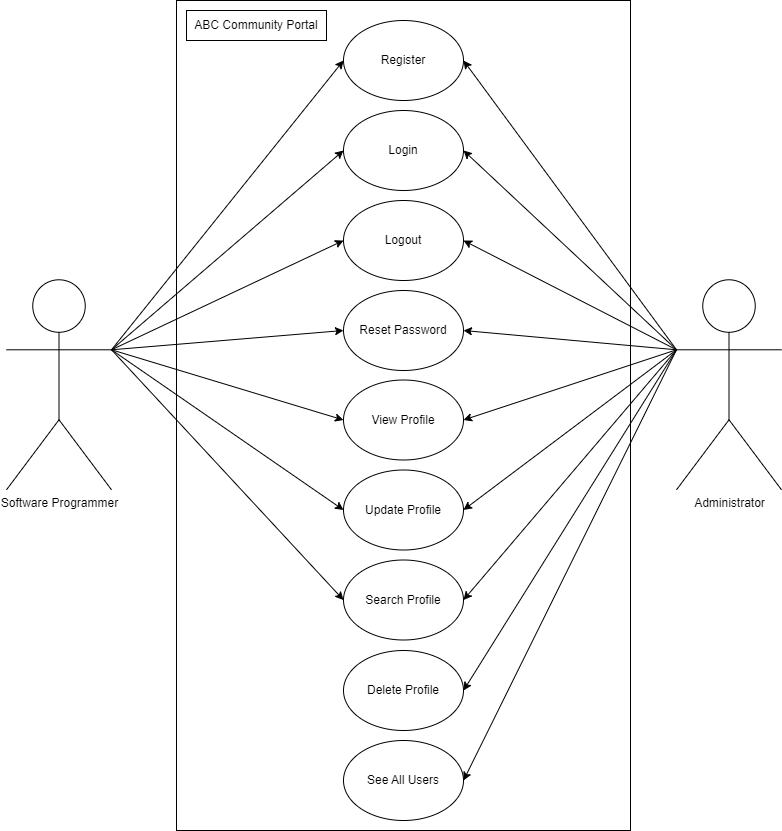


1. Task 1 Technical design
2. Task Statement (a): Block diagram of the system

Solution:

Diagram, timeline

Description automatically generated



|  |  |
| --- | --- |
| Flowchart | Steps |
| **Login** | 1. User goes to the Sign in Page. 2. User must enter the email and the password that has been made. 3. The system will check the data after the user clicks a sign-in button if the valid user continues. If false or different user will re-enter the email and password again. 4. Next, Thank you page will show,and there is a button to continue to User Dashboard. |
| **Register** | 1. User at the sign-up page.  2. User enters their first name,last name, city, email, and password, confirms the password, and clicks sign up.  3. The user will see an error state above the button if entries are invalid.  4. If valid, a verification code will be sent on the next page.  5. If the user did not get the code, the user might click the resend button.  6. User inputs the verification code.  7. When clicking Create Your Account, the user will direct to create a profile  8. The page will direct to the login page. |
| **Forgot Password** | 1. User at the sign-in page and click the forgot password  2. User enters the email that is used for sign in  3. If the email in database the process will continue, if not will show the alert  4. User enters a new password and confirms the password  5. When success directed to sign in page |
| **Update Profile** | 1. Go to the user profile 2. Click edit profile 3. We can change the first name, last name, phone number, , and city. Also, have another option for editing the education and experience 4. After we complete editing the profile, we can click save 5. System will collect and update the data |

Task Statement (b): Architecture of the system

Solution:

Diagram

Description automatically generated

Diagram

Description automatically generated

((Consider one scenario as an example and explain the request flow from controller->service->DAO->DB clearly))

|  |  |
| --- | --- |
|  | After user already registered on the website and done the verification, User will direct into the login page and the system save the data user into database. Using |
|  | The first method is a GET method with a request mapping value of "/login". This method takes a HttpSession object as a parameter and returns a new ModelAndView object with a view name of "login". This method is likely used to display the login form. |
|  | The second method is a POST method with a request mapping value of "/login". This method takes several parameters, including a Users object that is annotated with "@ModelAttribute". This means that the method expects form data to be submitted with the request that can be used to create a new Users object. The method also takes a Model object, which can be used to add attributes to the view, and HttpServletRequest and HttpServletResponse objects for handling the request and response, respectively. |
|  | The method first gets the current session object using the HttpServletRequest object. This is done in order to store information about the current user in the session. |
|  | The method then calls a method called "login" on a Users service object called "us" and passes in the Users object created from the form data. This method is likely used to authenticate the user based on their email and password. |
|  | If the login is successful (the "isLogin" object is not null), the method sets several attributes on the session object, including the user's email, userId, roleId, and a boolean value indicating that the user is logged in. The method then returns a new ModelAndView object with a view name of "redirect:/dashboard". This will redirect the user to the dashboard page. |
|  | If the login is unsuccessful, the method sets a message indicating that the credentials are incorrect as an attribute on the model object. The method then returns a new ModelAndView object with a view name of "login". This will display the login form again with an error message indicating that the login was unsuccessful. |

1. System
   1. List down all the modules to be developed

Private Module - Administrator - Manage user Data and Bulk Emailing.

Public Module - Software Programmers –

1. Registration
2. Login
3. Forgot Password
4. Search and Search Result
5. Update Profile
   1. List down all the pages in each of the modules and briefly describe the purpose

The answer is on C

* 1. Note down which are Create, Edit & View pages.

Insert - Create Page for users to register and data are inserted into database

Select - View Page for the users can view the others profile page

Update - Update Page for the user to edit and update their profile page

|  |  |  |
| --- | --- | --- |
| Pages (.jsp) | Brief description. | Operations. |
| Header1 | Navbar, ABC Logo, button sign up and sign in | View |
| Header2 | Navbar, ABC Logo | View |
| Header3 | Navbar, ABC Logo, button dashboard, profile, search, and logout | View |
| Index | Landing page | View |
| Registration | New users can create a new account in the portal | Insert |
| Thankyou | Appreciate new user | View |
| Verified | Verifying the account | View |
| Login | Users can enter the portal by logging in with a registered username and password | Select |
| Forgotpass | Verifies the user registered email before the password can be changed | Select |
| Resetpass | Allows users to change password to new password | Update |
| Dashboard | Show the content | View |
| Profile | Show user profile and data after successfully logging in | View and Update |
| Search | Users can search for other programmers | View and Select |
| Result | See a list of search results viewing other profiles | View and Select |
| Footer1 | Logo ABC, Copyright, company name, address, and contact | View |
| Footer2 | Copyright, company name | View |
| Footer3 | Copyright, company name | View |
| Admin/Headeradmin | Logo ABC, button Dashboard, Search and Logout | View |
| Admin/index | A simple page for the admin to select options for managing users | View |
| Alluser | View all the users and can view details, delete | View, Select, Update |
| Send-bulk | Send bulk email to user | Select |

* 1. Create the Sitemap of Public & Private Site

Diagram

Description automatically generated

1. Technical Environment Requirements

Hardware Requirements

* + - * Database server – MySQL server
      * Firewall settings - Port 9091 -- Http

- Port 3306 – Database

Software Requirements

Development environment- JDK 18.0.2

Framework Spring framework 5.2.10

Database server- MySQL Server 8.0.31

1. System Integration Requirements

JDBC - JDBC -My SQL Connector 8.0.31

1. Portability Requirements

Multi OS- Windows, Linux, and Mac

Multiple Browsers - Chrome, Microsoft Edge, Mozila Firefox

Multiple Devices - Desktop, Tablet, Mobile

1. Maintainability Requirements
   * + - Backup the database for every 6 hours.
       - Vulnerability check – periodical scanning.
2. Performance Requirements

Each request should be processed with 10 seconds.

Concurrent number of users- 100

Loading time – 15secs

1. Security Requirements
2. List down the Security Access for the different kind of users

|  |  |  |
| --- | --- | --- |
| Pages | Access based on User Role | |
| Software Programmer | Administrator |
| Community portal home page | View | View |
| Registration page | View/Create | View/Create |
| Thankyou page | View | View |
| Verified email page | View | View |
| Login page | View/Login | View/Login |
| Forgot Password page | View | View |
| Forgot password confirmation page | View/Edit | View/Edit |
| Public Profile page | View | View |
| Update profile page | View/Edit | View/Edit |
| Bulk Email | N/A | View/Edit |
| Administration page | N/A | View |

|  |  |  |
| --- | --- | --- |
| Functions | Access based on User Role | |
| Software Programmer | Administrator |
| Register | Yes | Yes |
| Login | Yes | Yes |
| Change password | Yes | Yes |
| Update profile | Yes | Yes |
| Search others | Yes | Yes |
| View other user profiles | Yes | Yes |
| Update other user profiles | No | Yes |
| Delete other user profiles | No | Yes |
| Send bulk email | No | Yes |

1. Briefly explain Login & Logout mechanism

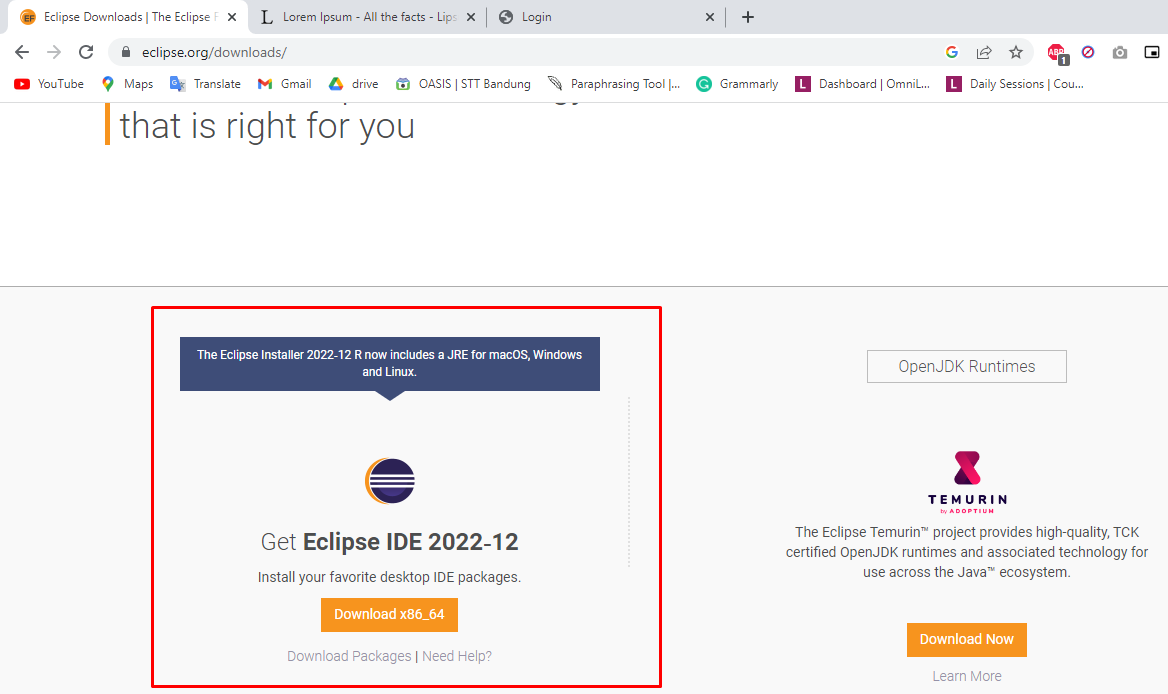
When a registered user log in to the website by entering their username and password, they will provide a unique ID to verify their identity as a key. The key or login credentials are compared to the data originally stored in the database server. If the credentials match the stored data, the user is allowed access or authenticated to their user profile or home page (Website Authentication: The Complete Guide with FAQs, 2020).

Logging out is performed the moment users clicks on the logout button which redirects the user to the login page. The other way is that the website can automatically log out by itself if the user leaves the page for a long time where the session was inactive for long periods. Logging out aids in preventing unauthorized user from accessing other people's account and therefore a crucial part of security (Log Out, 2015)

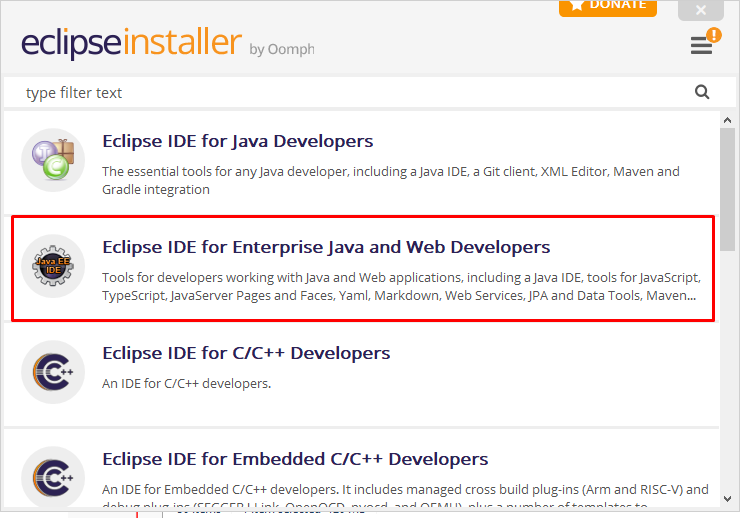
Task 2

Task Statement:

1. Install Eclipse on your laptop
2. Download eclipse on eclipse.org/downloads/
3. Open the download folder and double-click on the eclipse installer.
4. Choose the second option because we want to develop Web Development.
5. Last Thing is to click on install and wait for several moments. After that, you can run it on your desktop.
6. Provide Screen capture of eclipse IDE









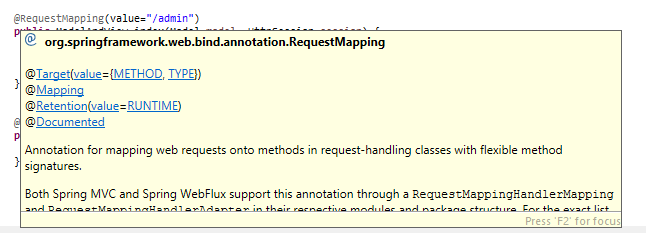
1. Task 3

Task Statement:

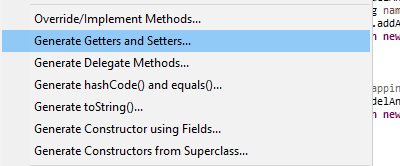
1. Provide description of 3 features in IDE along with the screen capture of their use.

Three IDE features are

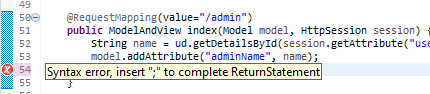
* 1. Give Us Information when we point the cursor into a code that we dont understand

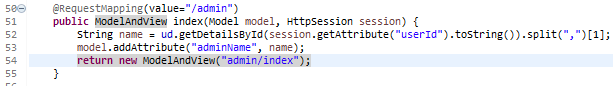


* 1. Has automatic Source Generator



* 1. Can fix the error by following the suggestion





1. Include it as part of Project Presentation
2. Task 4

Task Statement

Create the following items in “Coding & Documentation Standards” Section in Project Report

1. List 5 Coding Standards & Conventions to be used and explain shortly.

|  |  |
| --- | --- |
|  | Accessing class members privately  Class fields should inaccessible to protect the fields using a private access modifier. It is best practice to maintain encapsulation or hiding the data in preventing the information from being compromised. |
|  | Appropriate commenting  Commenting, such as a single line comment, ensures readability and describes how the code functions. This gives information that the code alone cannot perceive, allowing different people to comprehend the Java code. |
|  | Organizing a Project  The Spring MVC framework should be reflected in the project's structure by adding packages on the layer with corresponding MVC classes or interfaces. The project's controller, dao, service, and model layers are shown in detail in a separate package. |
|  | Class Declaration  Class names should be nouns, in mixed case, with the first letter of each internal word capitalized. |
|  | Naming patterns  Each class, interface, method, variable, and other name conventions should follow the correct naming conventions; for example, methods should use the PascalCase convention while variables should follow the camelCase style. |

1. Describe briefly documentation tool, you will be using and its configuration.

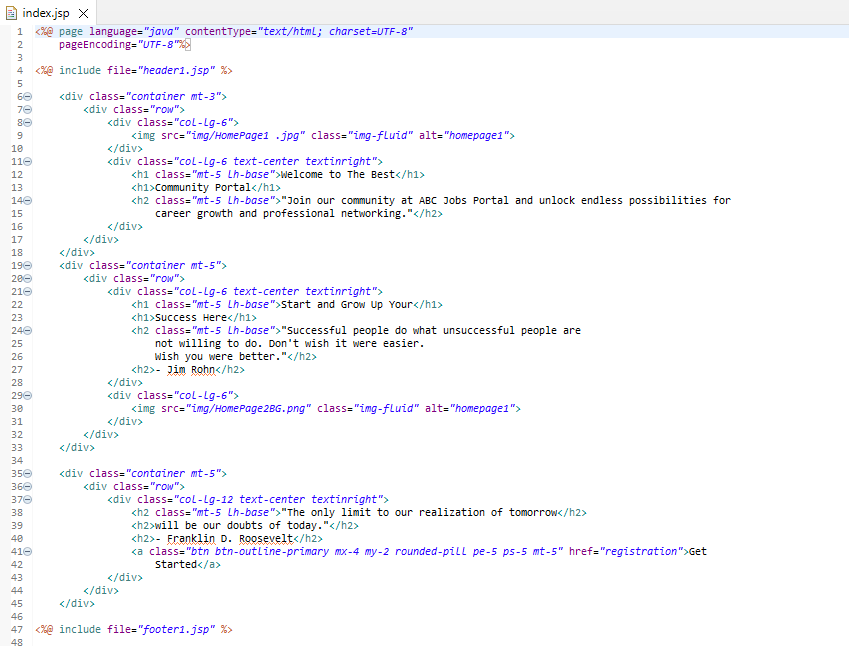
|  |  |
| --- | --- |
|  | Click on Project and Generate Javadoc... |
|  | Choose your project and the destination folder click next. |
|  | Click next on this part. |
|  | Choose JRE source compability based on your own. |
|  | Click Yes to All. |
|  | Wait until done |
|  | Open Index.html in javadoc folder |
|  | And you can see the result |
|  |  |
|  |  |

1. Task 5 (Views)

Task Statement

1. Develop the following pages in HTML (Developed in Module 3) (source code)(no java page)
   1. Home Page

<<Source code>>

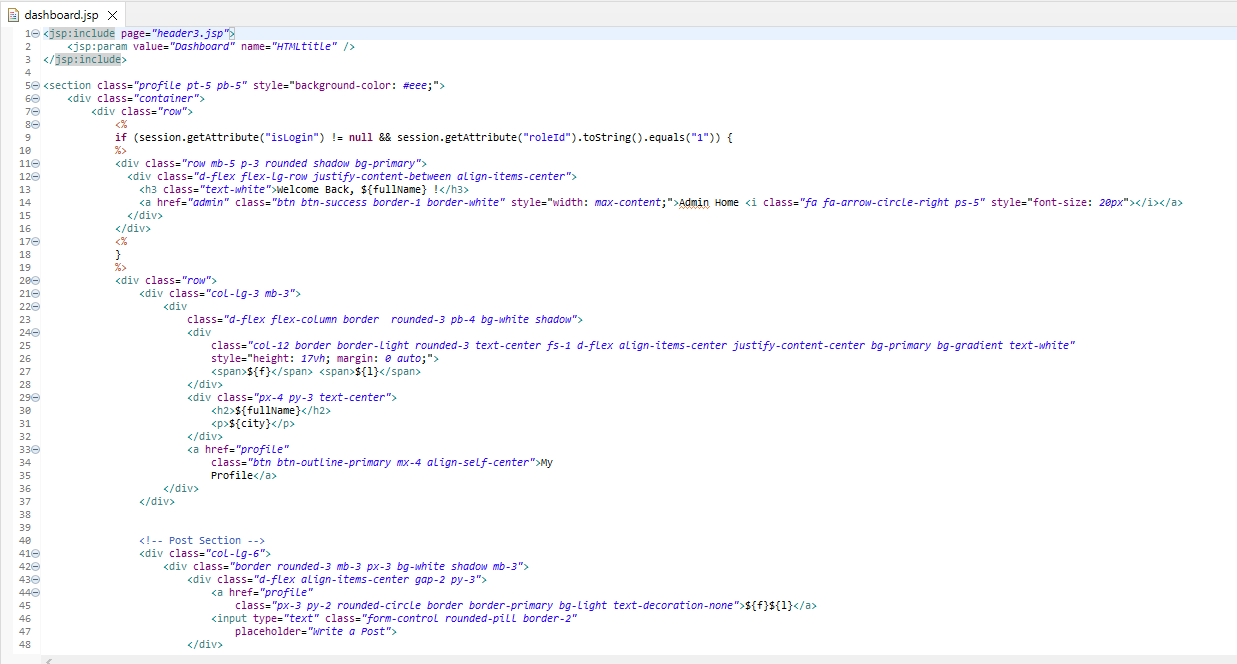


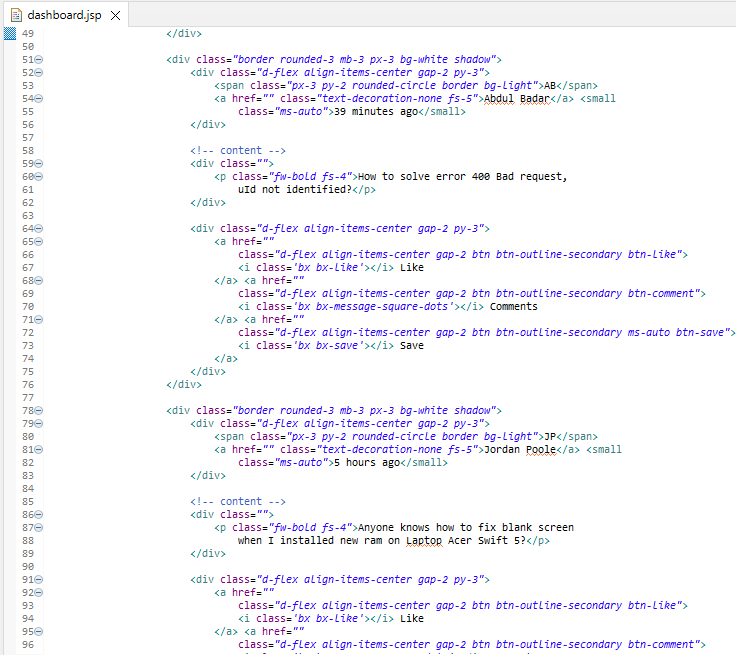
* 1. Registration Page

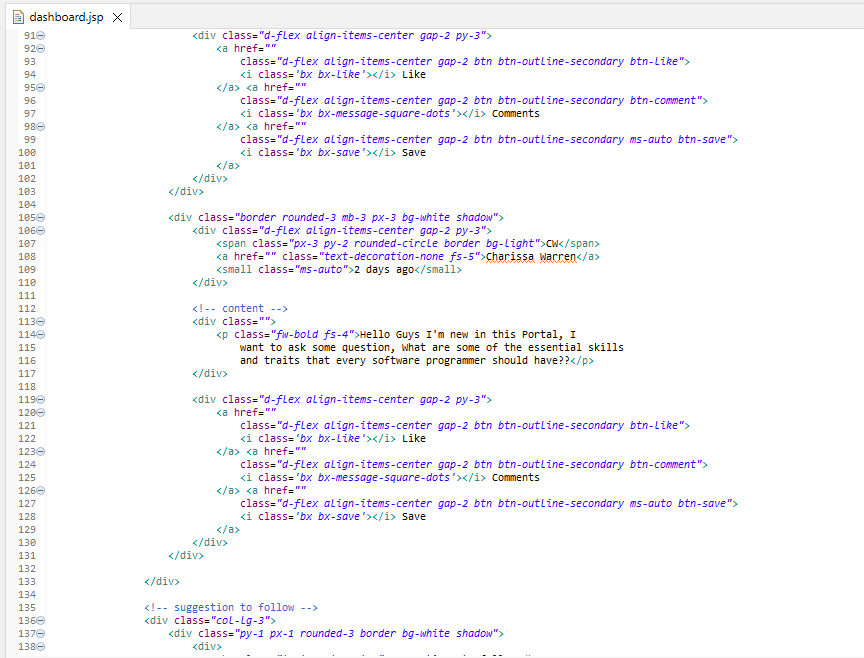




* 1. Dashboard Page

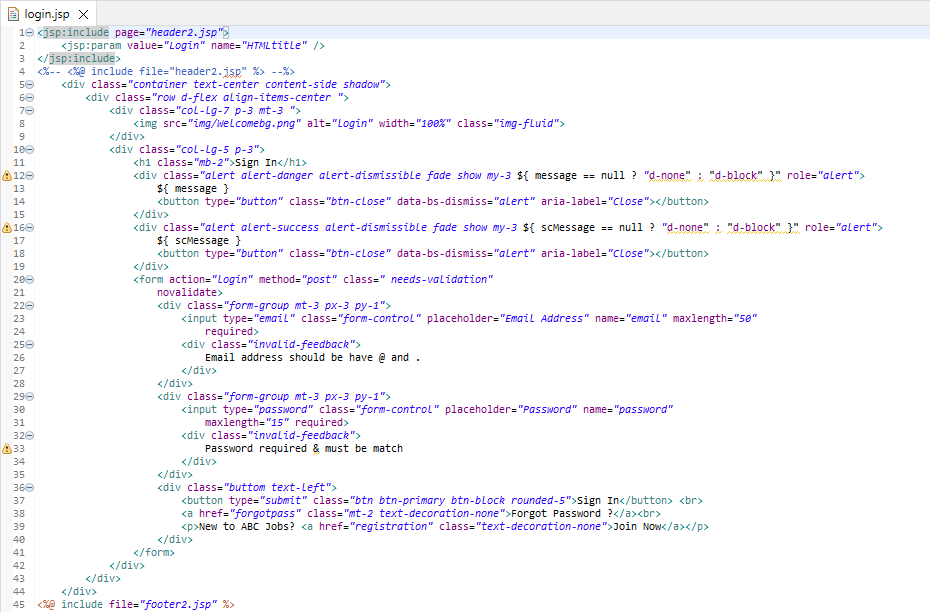




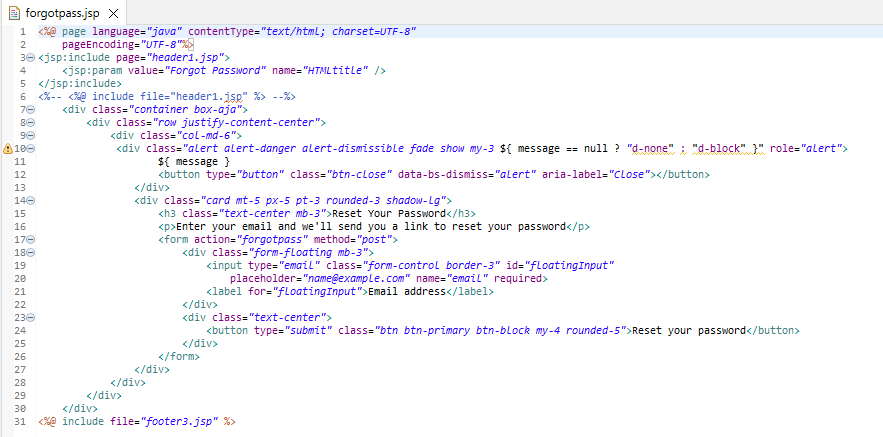




* 1. Login/Logout

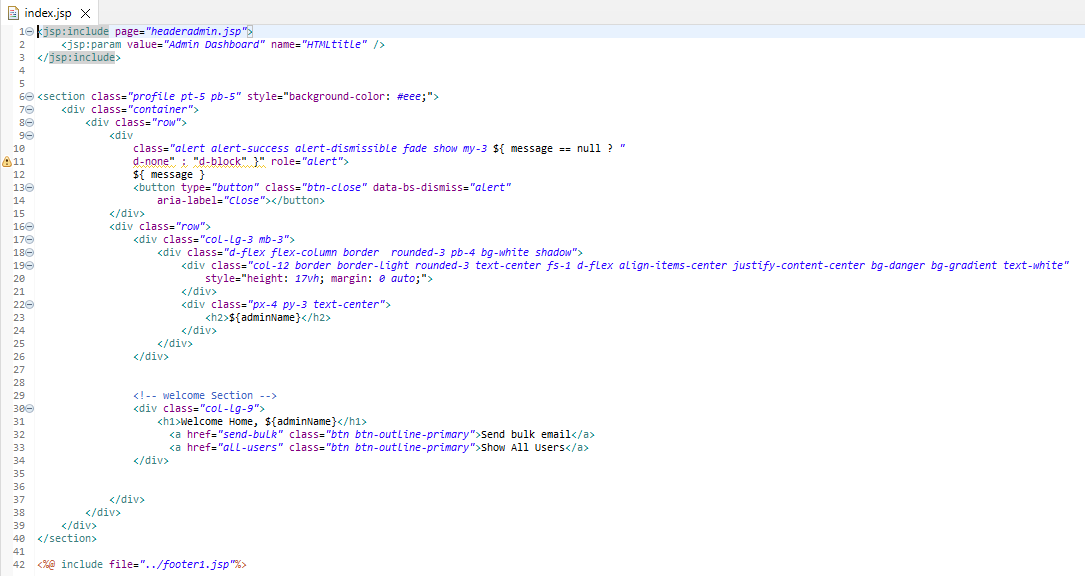


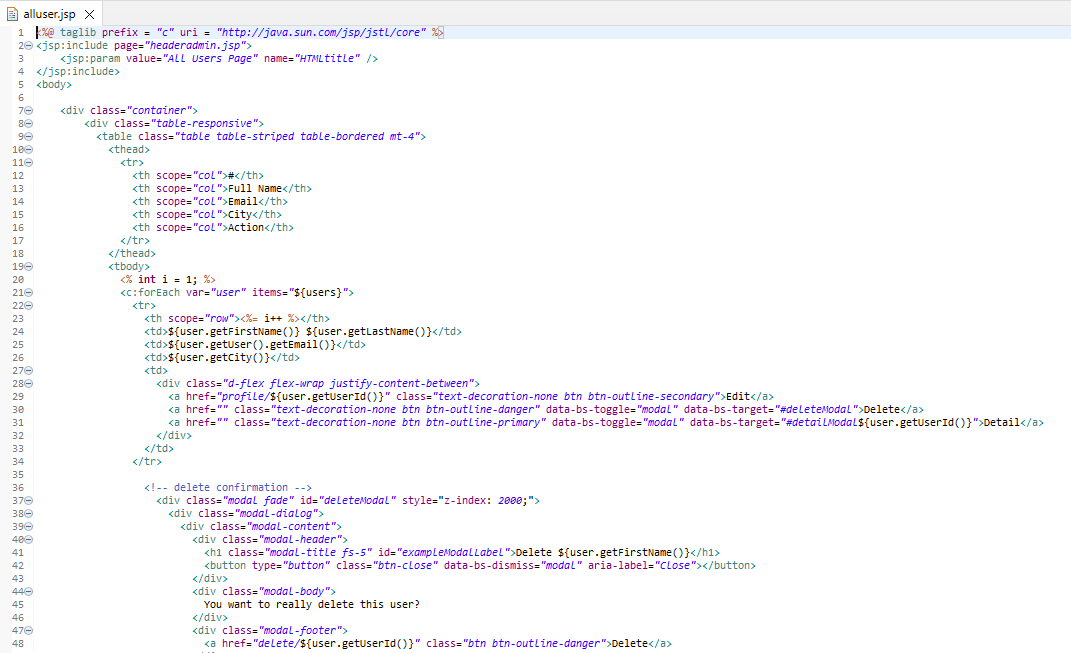
* 1. Forget Password

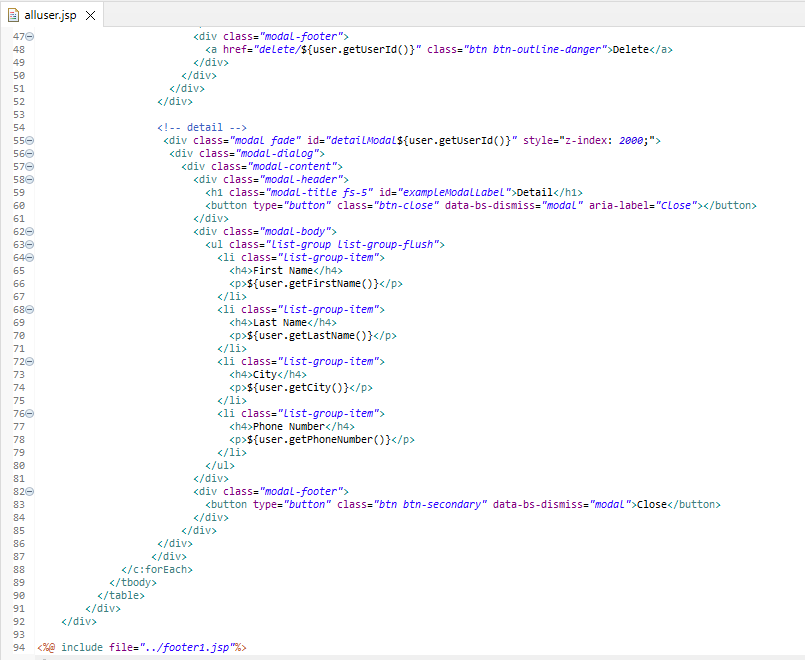




* 1. Administration Page



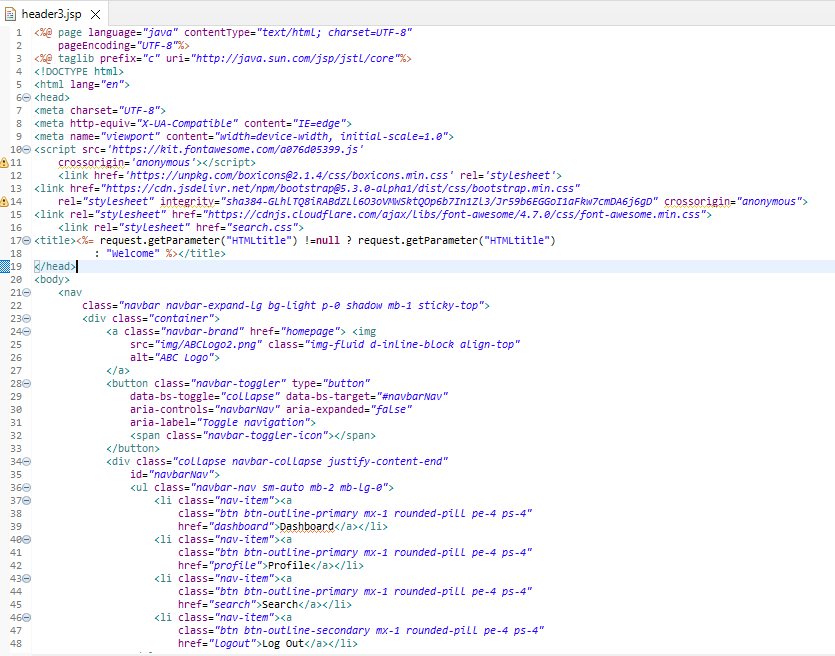


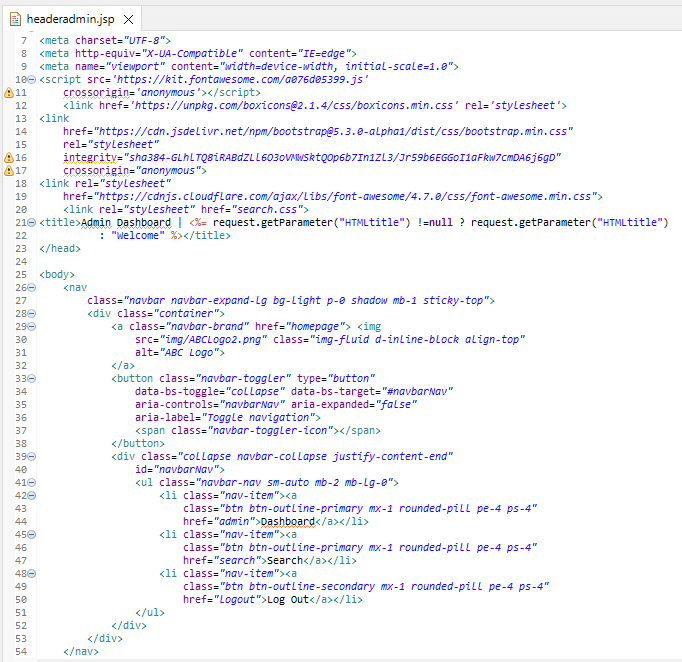


1. Template them by separating Headers/ Footers and including them using Server Side includes.
   1. Header

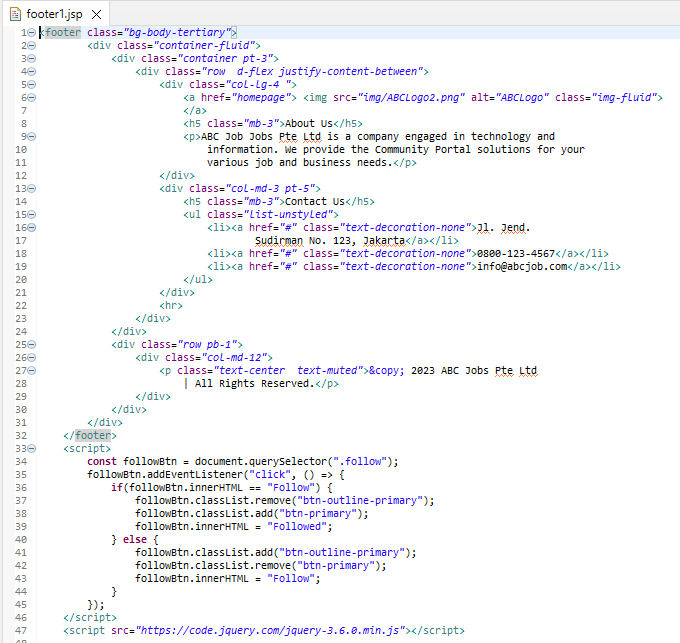


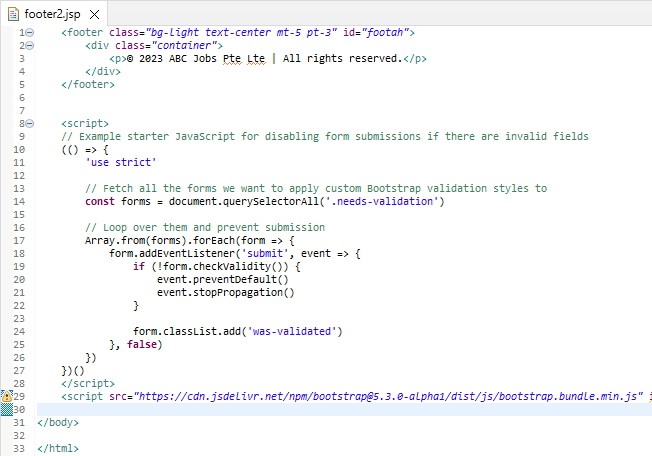






* 1. Footer







example





1. Provide the screen capture in Project Presentation

1. Task 6 (java classes)
2. Package wise

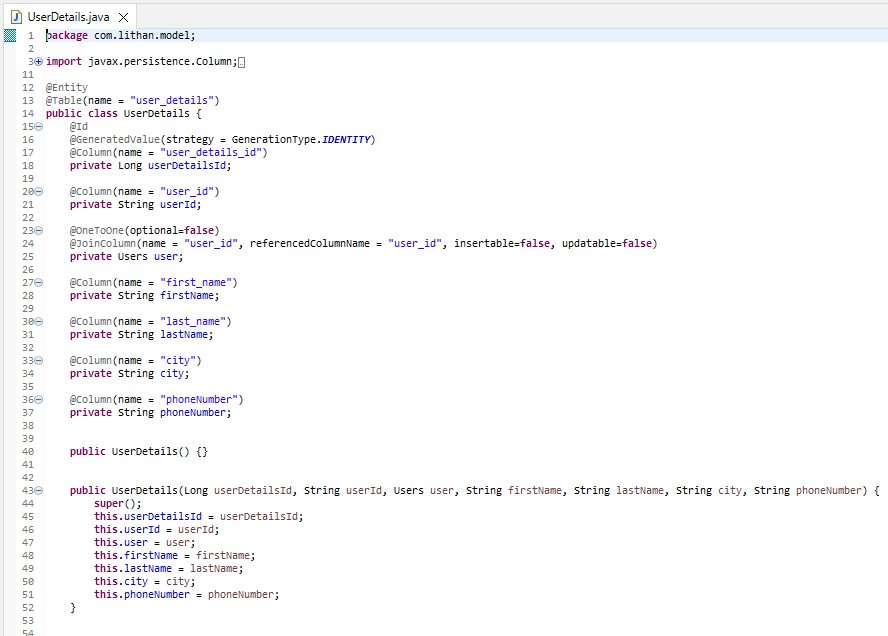


1. MainCotroller.java

Source code

Task Statement:

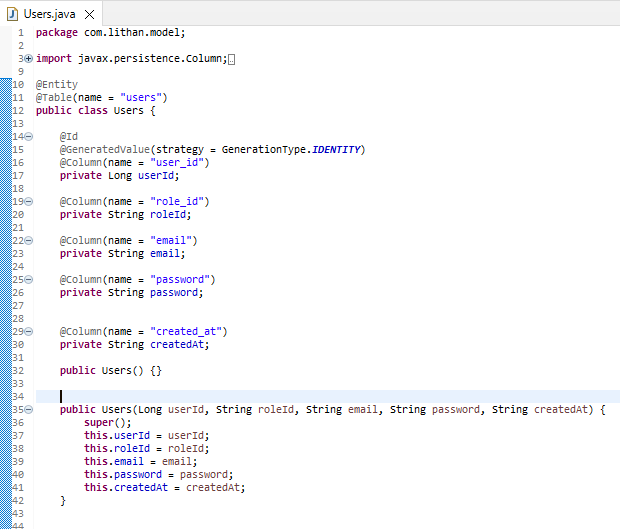
1. Create the required classes in Java
   1. UsersDetails.java



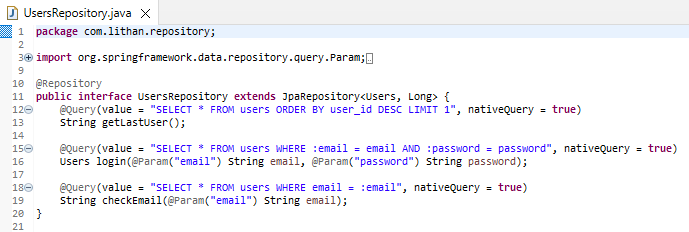
* 1. UserDetailsService.java



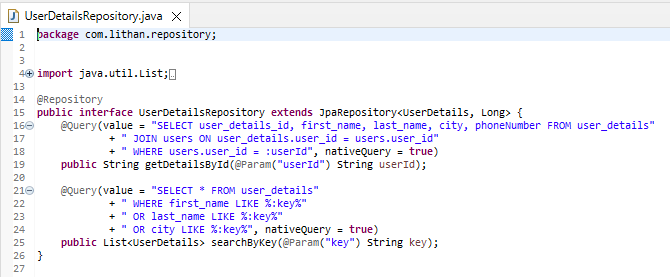
* 1. Users.java



1. Include the Data Access Layer classes
   1. UsersRepository.java



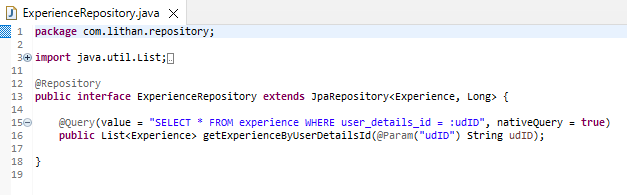
* 1. UserDetailsRepository.java



* 1. EducationRepository.java



* 1. ExperienceRepository.java



1. Include samples as part of Project Report

1. Task 7

Task Statement:   
Provide screen capture of developed pages and hosted application in Project Presentation

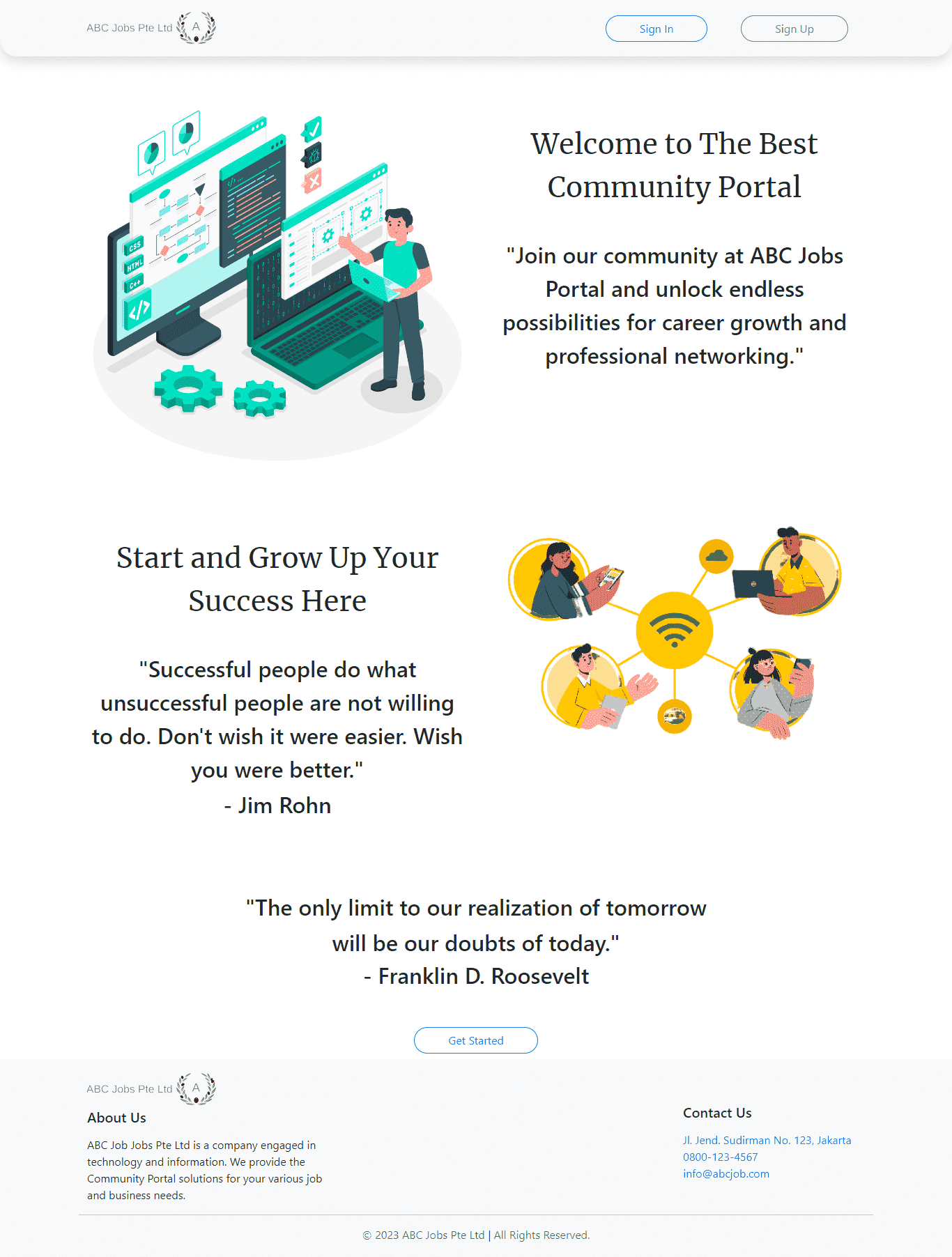
1. Queries and code

|  |  |
| --- | --- |
| Registration | public Users register(Users user) {  return repo.save(user);  } |
| Viewprofile other user | @Query(value = "SELECT user\_details\_id, first\_name, last\_name, city, phoneNumber FROM user\_details"  + " JOIN users ON user\_details.user\_id = users.user\_id"  + " WHERE users.user\_id = :userId", nativeQuery = true) |
| search | @Query(value = "SELECT \* FROM user\_details"  + " WHERE first\_name LIKE %:key%"  + " OR last\_name LIKE %:key%"  + " OR city LIKE %:key%", nativeQuery = true) |
| Login | public Users login(Users user) {  try {  Users login = repo.login(user.getEmail(), user.getPassword());  return login;  } catch (Exception e) {  System.out.println("Credential is null " + e);  }  return null;  } |
| Update Prorfile | public UserDetails editprofile(Long userDetailsId, UserDetails ud) {  UserDetails userDetails = repo.findById(userDetailsId).get();  // update  userDetails.setCity(ud.getCity());  userDetails.setFirstName(ud.getFirstName());  userDetails.setLastName(ud.getLastName());  userDetails.setPhoneNumber(ud.getPhoneNumber());;  // save  return repo.save(userDetails);  } |
| Forget Password | public Users updatePassword(String password, String email) {  Long userId = Long.parseLong(repo.checkEmail(email).split(",")[0]);  Users user = repo.findById(userId).get();  // update  user.setPassword(password);    // save  return repo.save(user);  } |

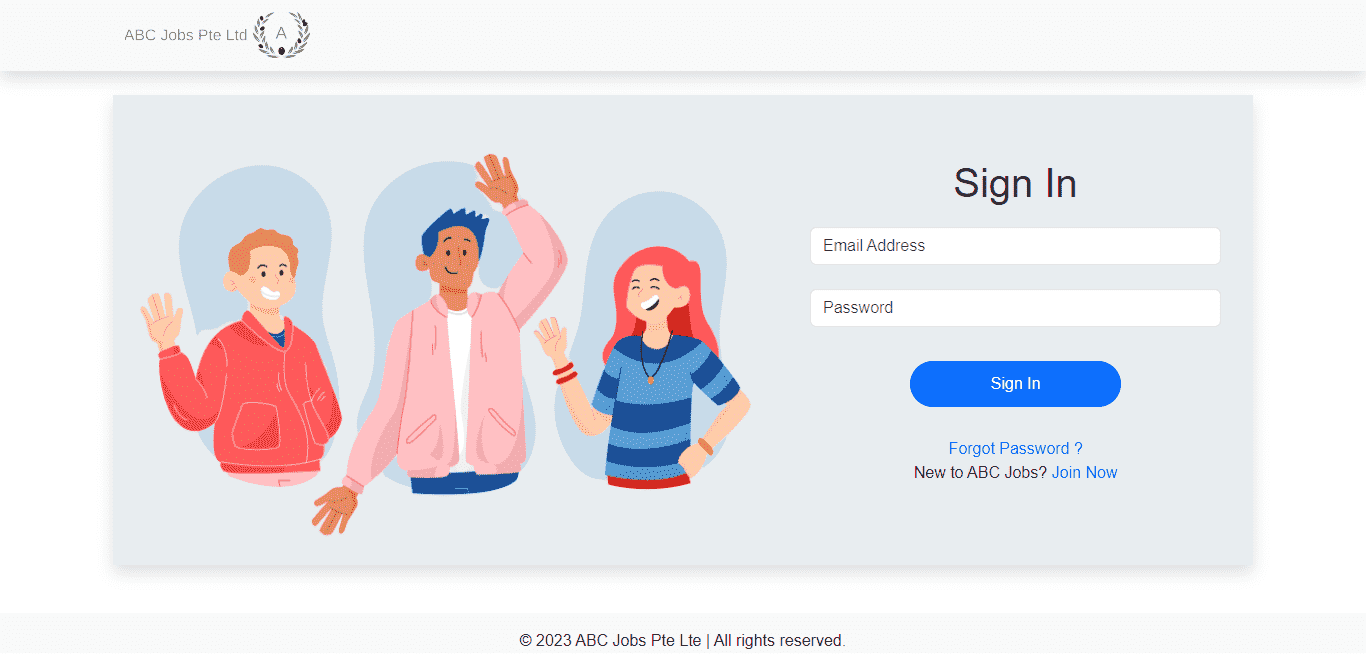
1. Develop the Models, Views, Controller (at least for one of the functionality, better if you can use for all the functionality)

|  |  |
| --- | --- |
|  | The Java Spring Controller defined by this code manages user registration requests. The controller displays an empty registration form when a user reaches the "/registration" URL. The controller processes the data once the user accepts the registration form, creates a new user account, and stores the user data in a database. If the registration process is successful, a "thank you" page is displayed to the user. An error warning appears on the registration page if the user's email address has already been registered with the system.  Two @RequestMapping methods are used in the code: one handles GET requests to display the registration form and the other handles POST requests to handle the form's input data. |

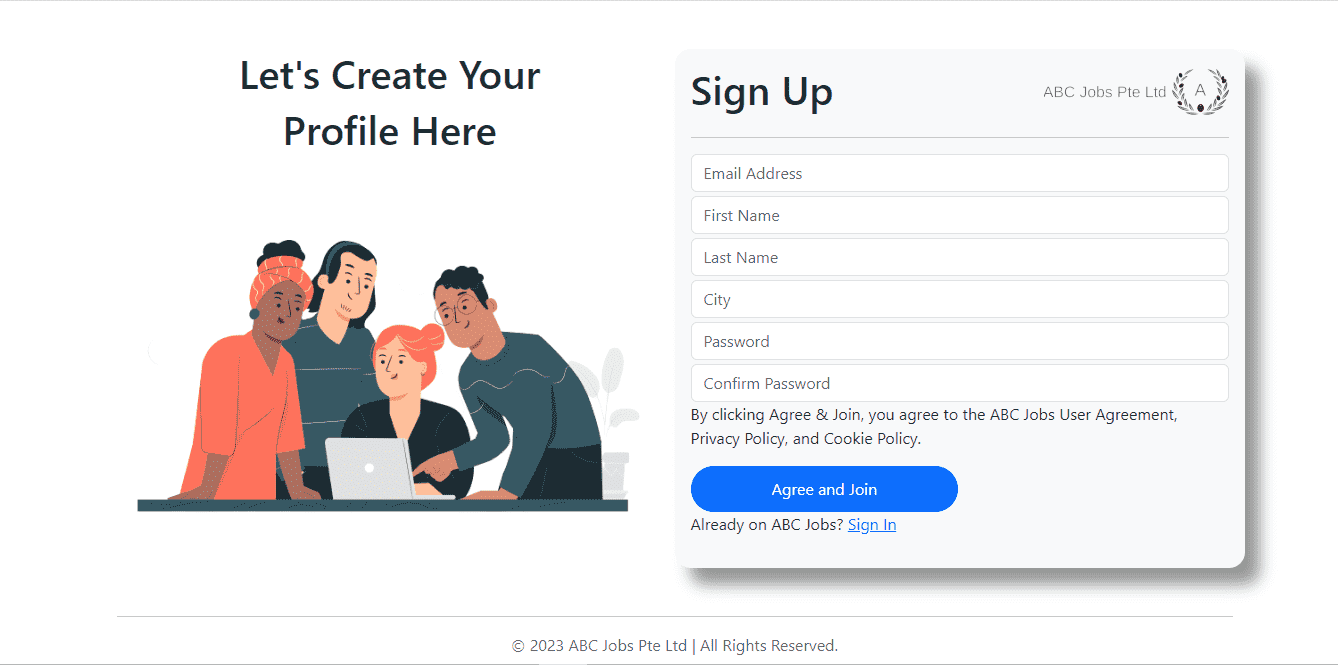
1. Develop Home Page & Integrate with Login System
   1. Home Page



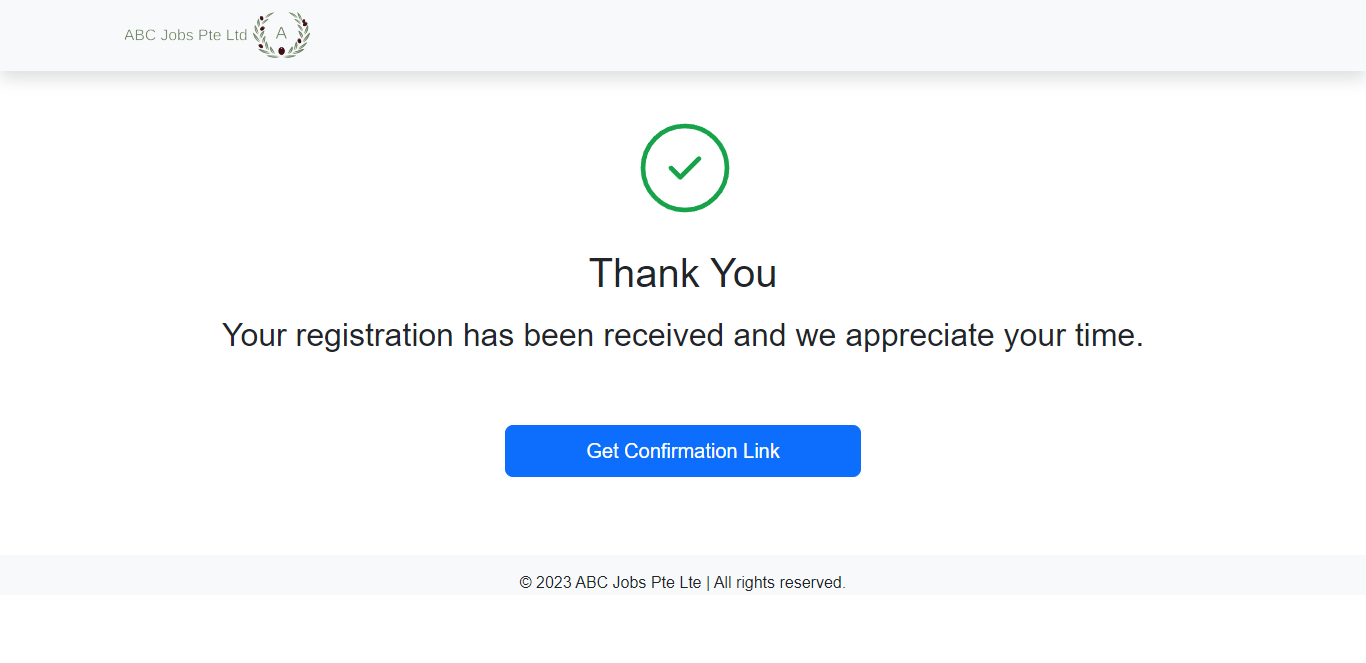
* 1. Login



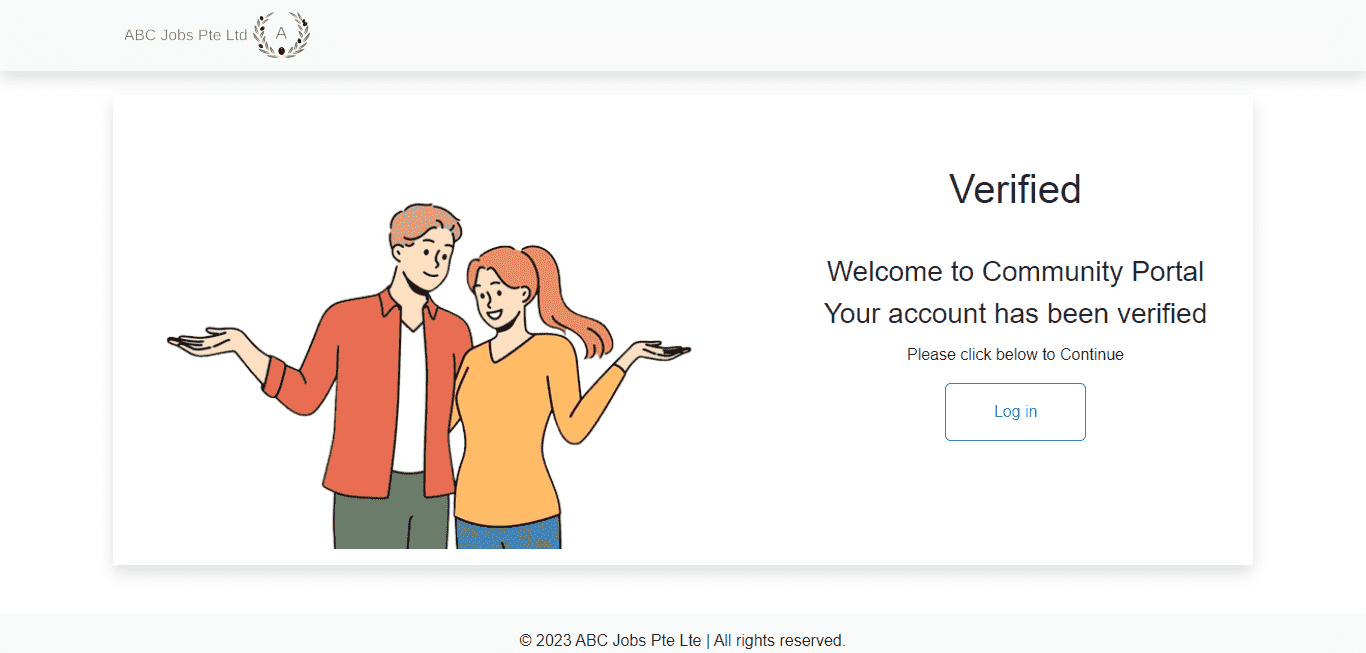
1. Develop Registration Form & Registration Thank You
   1. Registration Form



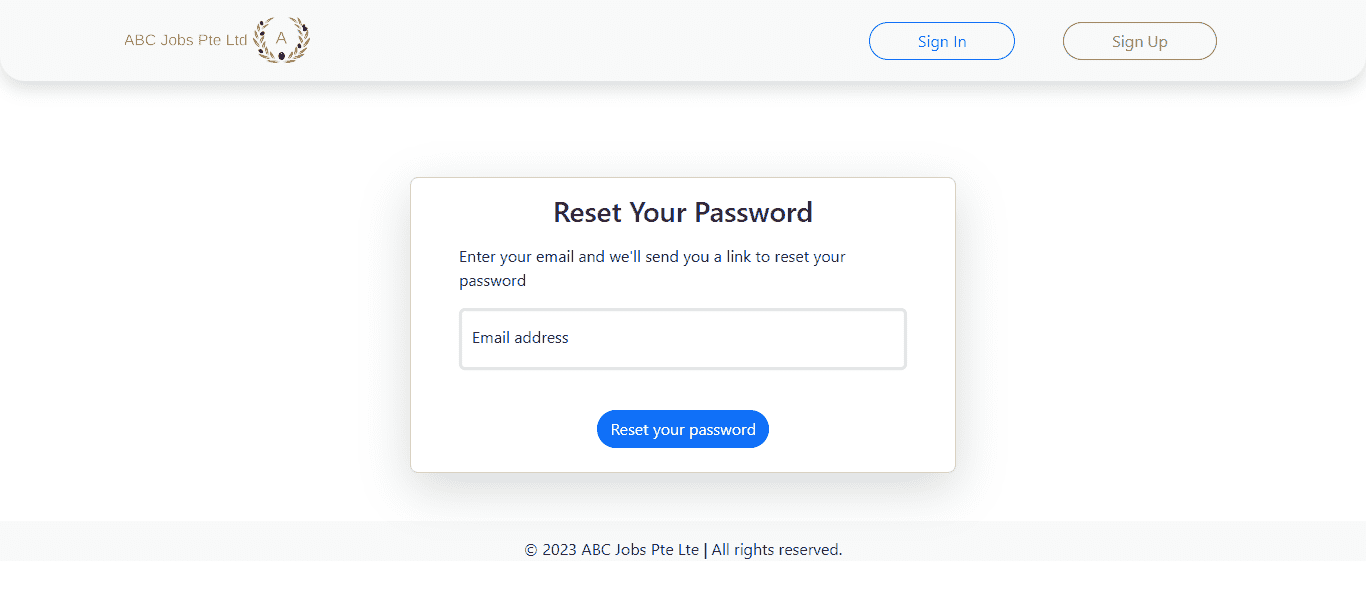
* 1. Thank You



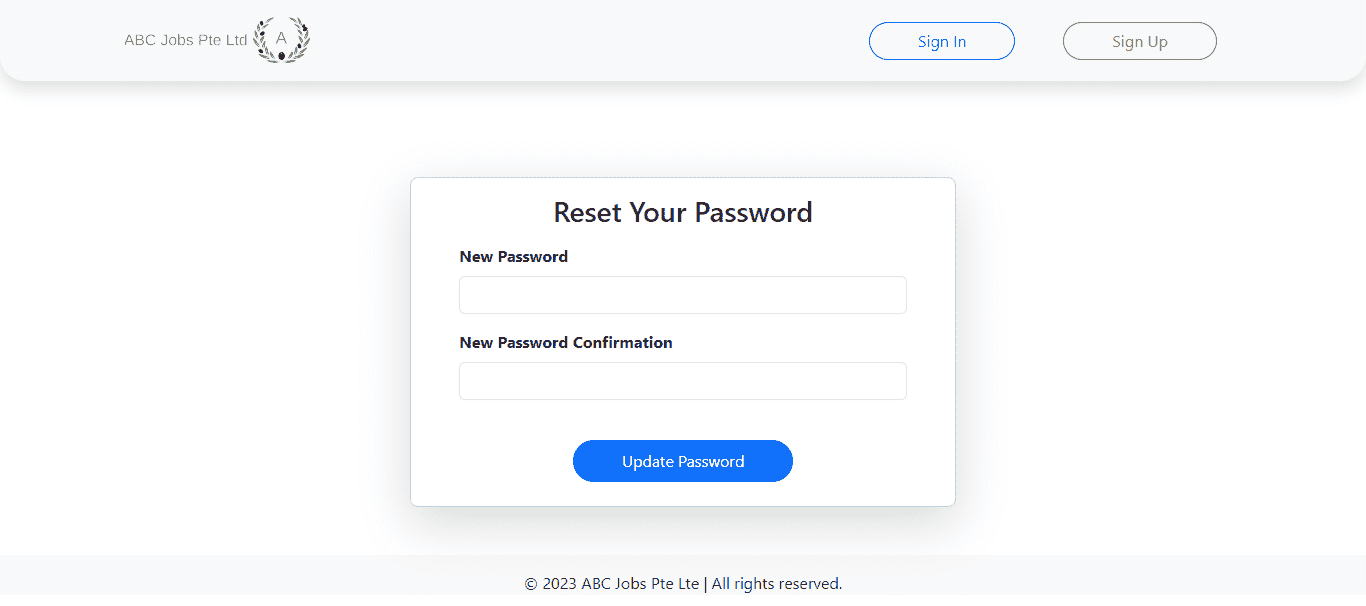
* 1. Verified Email



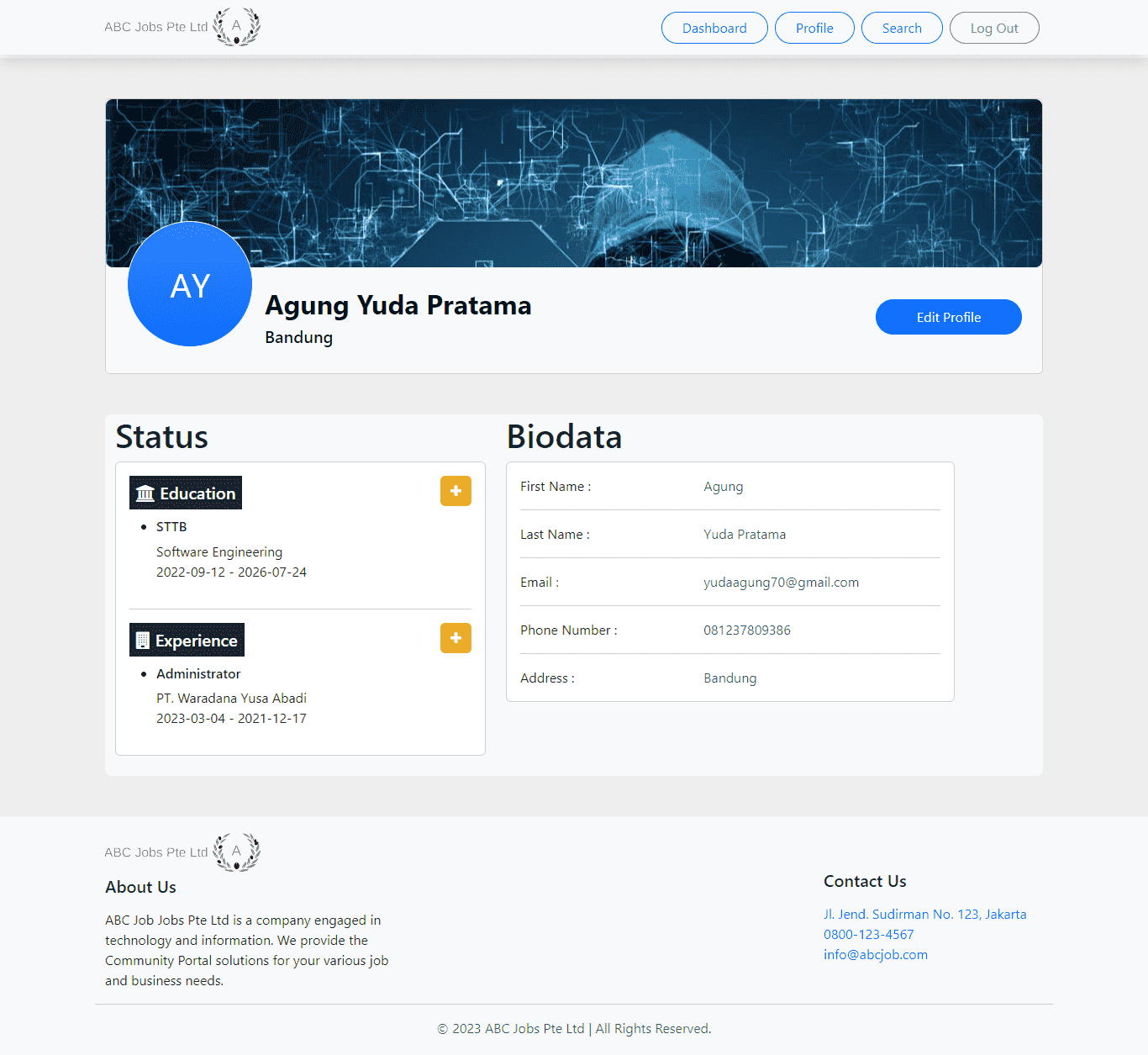
1. Develop Forget Password mechanism
   1. Forgot Password



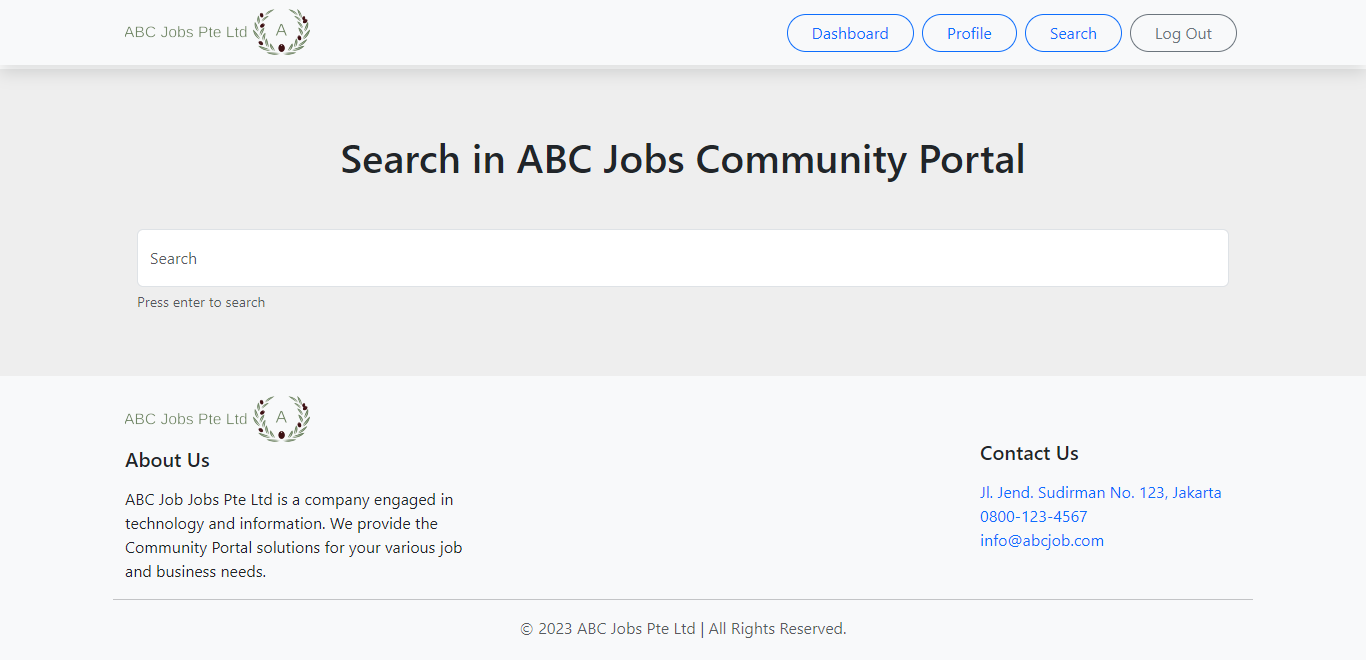
* 1. Reset Password



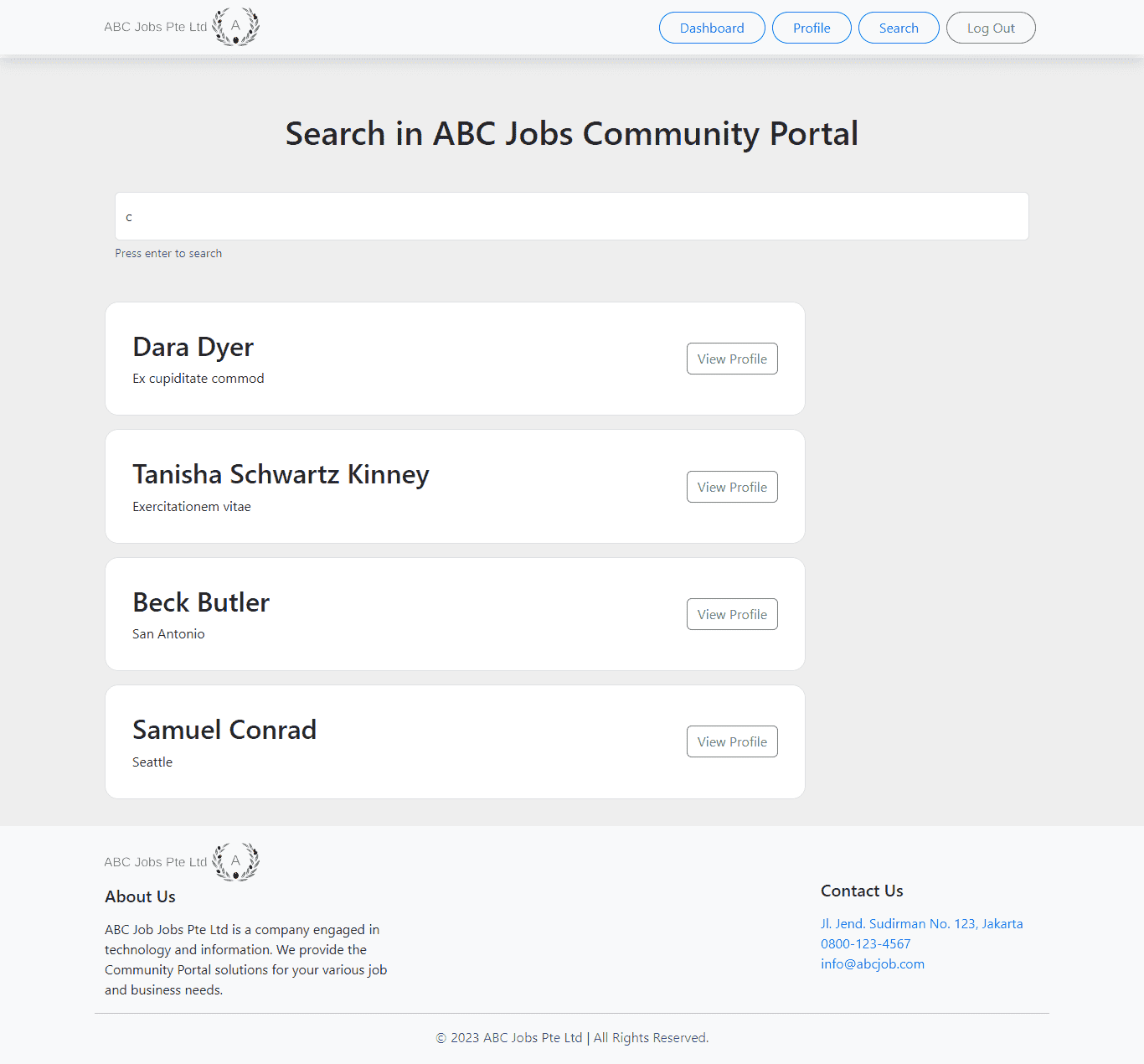
1. Develop Landing Page post login



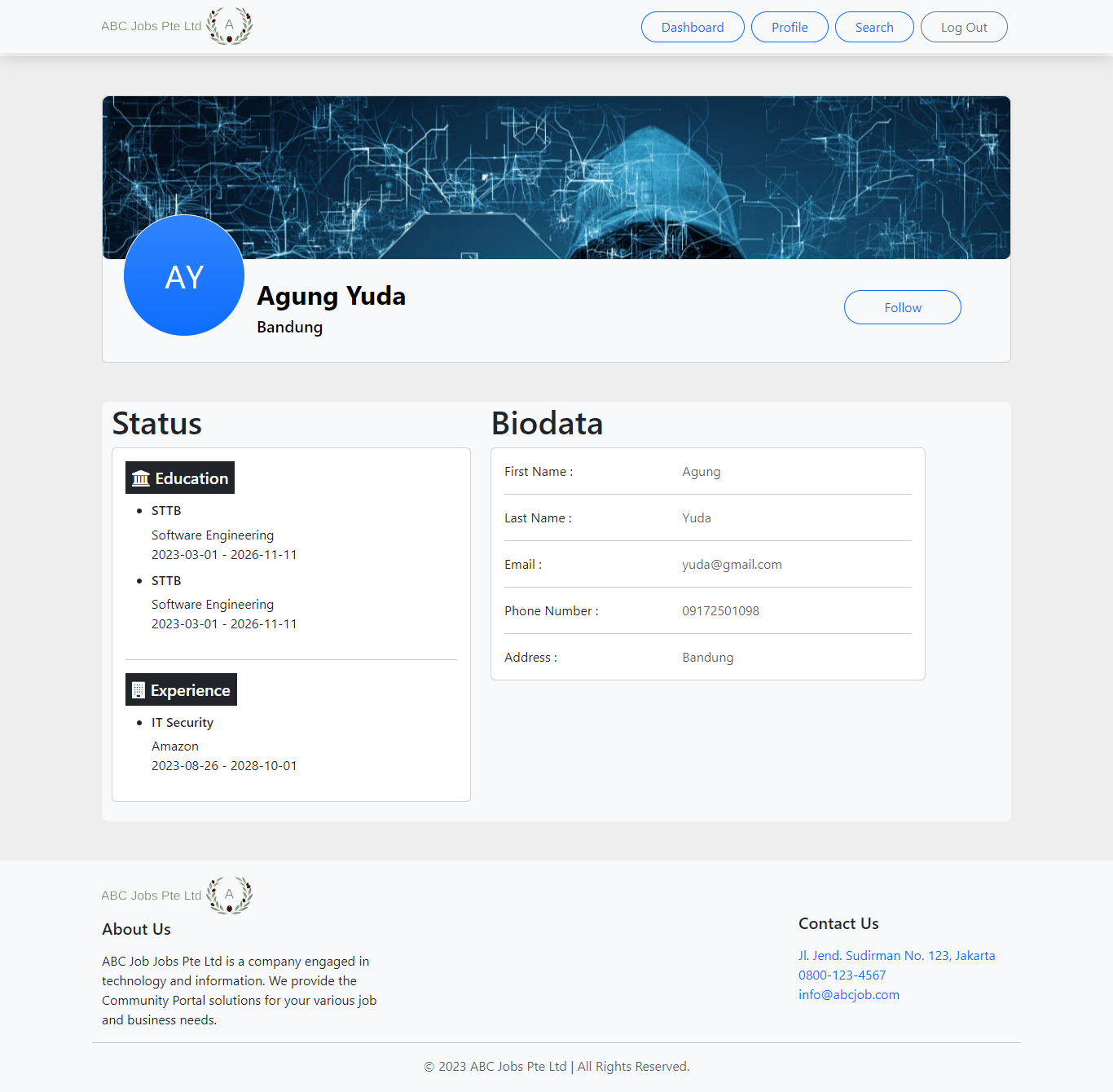
1. Develop Search Box for Users



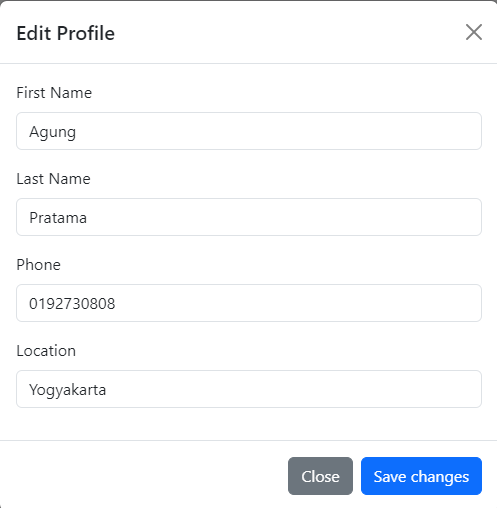
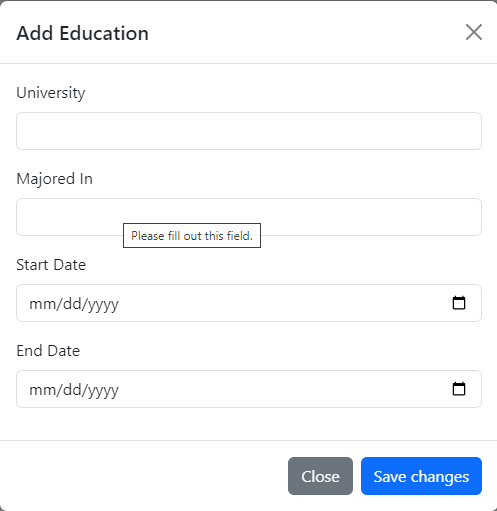
1. Develop View Search Results

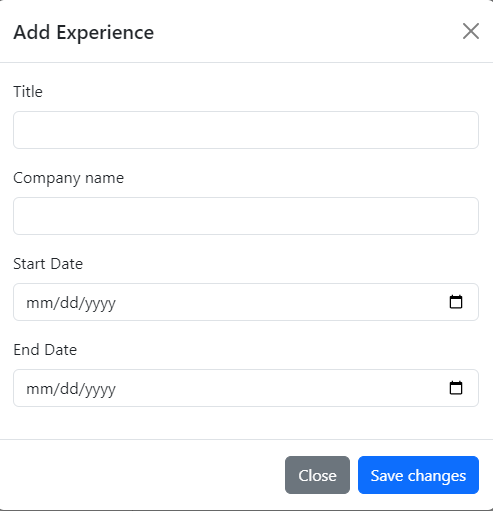


1. Develop View Other Profile Page



1. Develop Edit Profile Functionality





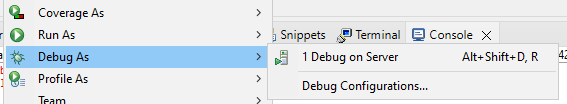
1. Task 8

Task Statement:

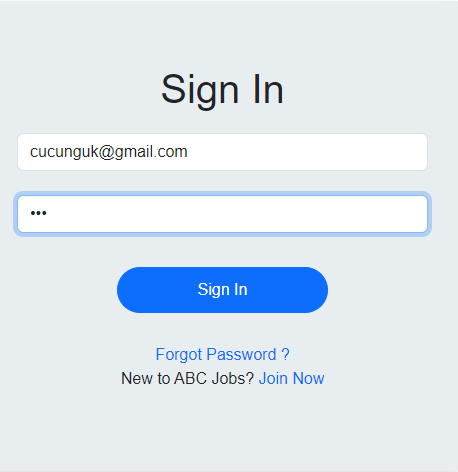
1. Debug the application in Eclipse. Provide the screen capture of your application code in Eclipse Debug Mode include it as part of Project Presentation
   1. Set Toggle Breakpoint



* 1. Debug on server

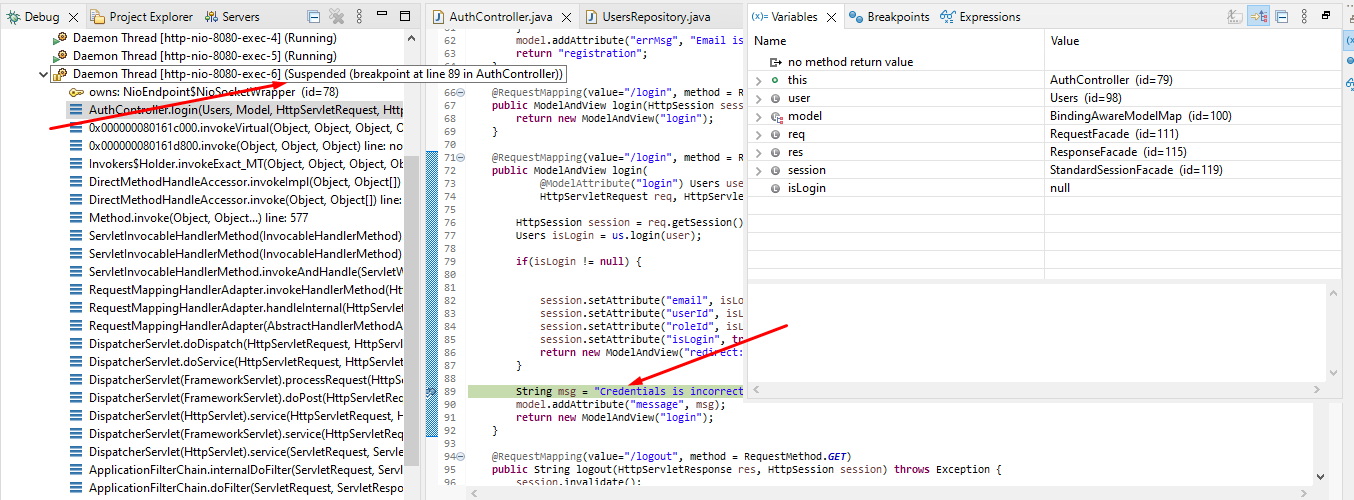


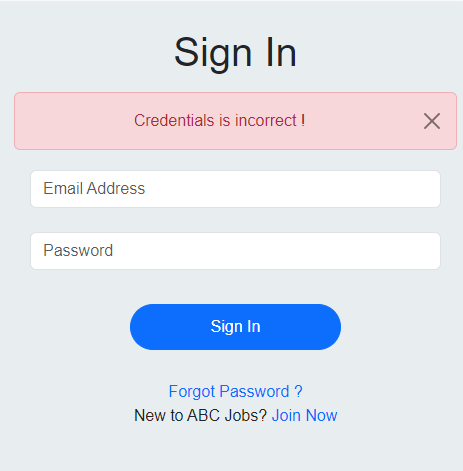
* 1. Enter Wrong Credentials



* 1. In Debug perspective will show the result

The debug stop at the breakpoint because we input wrong credential





1. Task 9

Task Statement:

1. Provide screen capture of 3 Errors you encountered while developing the application.
2. Include it as part of Project Presentation

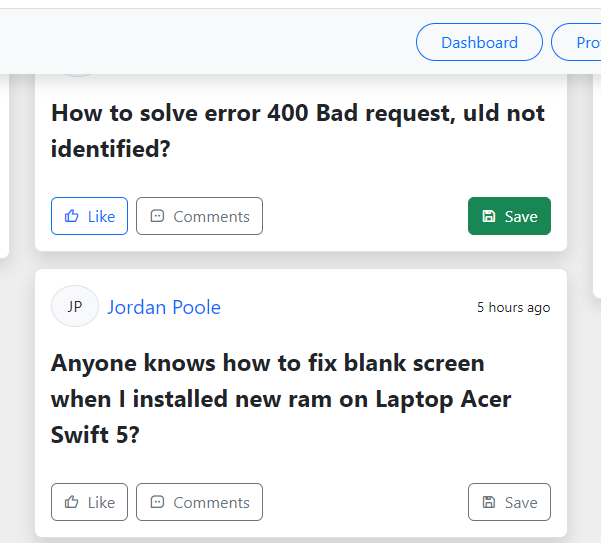
|  |  |
| --- | --- |
| **Error** | **Error handling (Solution)** |
|  | * If you run this program but this error is shown * Look at persistance.xml * Change the database name into your database * Change the password with your password |
|  | - Look at search.jsp  - Compare variable and parameter with search controller  - Make sure you have the value, param same with controller, because if different the data will not show |
|  | * Look in repository * Concern in search method * Do you have a tittle or not, if not change with other data that you have like city |

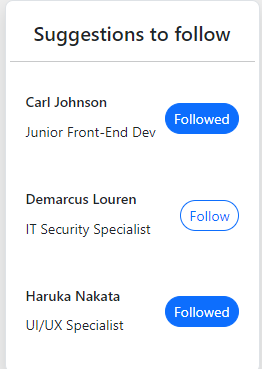
1. Task 10

Task Statement:

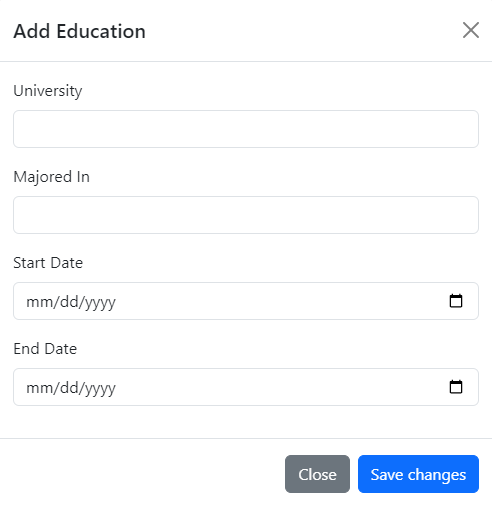
1. Propose & Document a New feature to the application
   1. Button to like, comment and save
   2. Button and suggestion to follow
   3. Add more detail about education.
   4. Add more detail about experience

.

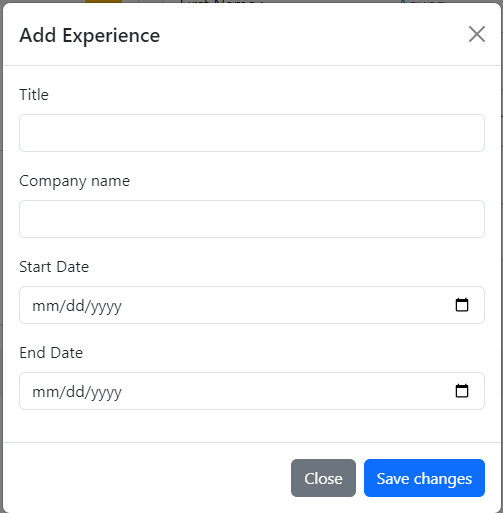
1. Develop the new feature and provide a report.
   1. Button to like, comment and save  
      
   2. Button to follow



* 1. Add more detail about education.



* 1. Add more detail about experience



1. Include it as part of Project Report
2. Task 11

Task Statement:

Provide Unit Testing code & screen capture of Eclipse IDE in debug mode.

* Test Scenario

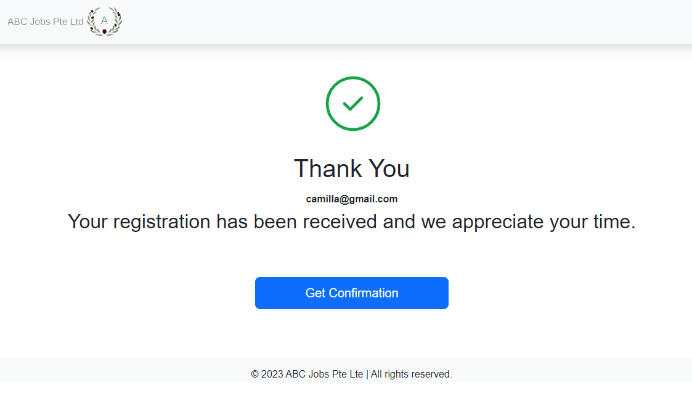
|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Test Scenario ID | Test Scenario | Number of test cases |
| Functional Testing | TS001-UAT testing | 1. Verify that users can validate their email and register their information. | 1 |
| TS002-UAT testing | 1. Verify that after logging in, the user may search for, locate, and browse the profiles of other software engineers. | 1 |
| TS003-UAT testing | 1. Verify the website's ability to change passwords. | 1 |

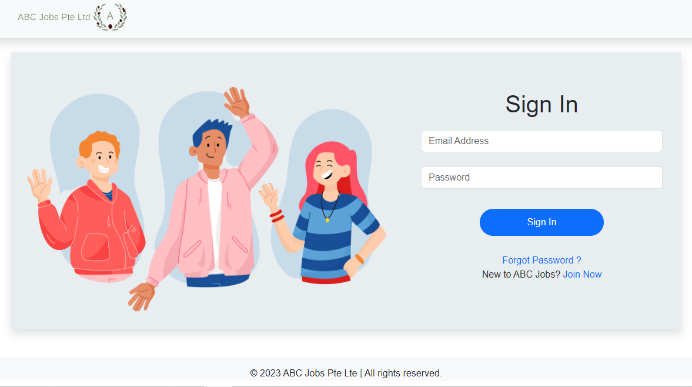
|  |  |
| --- | --- |
| Test Scenario | Verify that users can validate their email and register their information. |
| TS001 |
| Test Cases | UAT Testing |
| UATC01 | Verify that the signup and email confirmation are functional. |

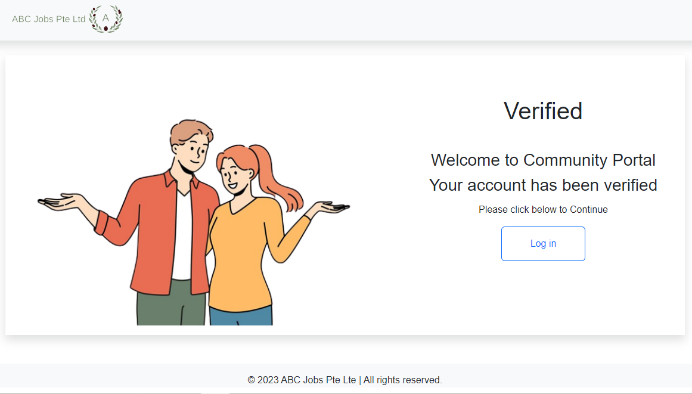
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario ID | Test Case ID | Test Case | Preconditions | Test Step | Test Data | Expected Results | Actual Results | Pass / Fail |
| TS001 | UATC01 | Verify that the signup and email confirmation are operational. | 1. The website is designed and developed  2.Unit testing and system testing is done  3.All the major errors has resolved and documented  4.The website is configured to Google Chrome by Apache Tomcat Server. | 1. Go to Google Chrome. 2. Enter URL www.ABCPortal.com 3. Click “Sign Up”in website home page 4. Enter all fields and submit 5. Click confirmation | First Name= Camilla  Last Name= Calderon  City = Seattle  Email= camilla@gmail.com  Password=”qwerty123” | User should land to login page | As expected, | Pass |





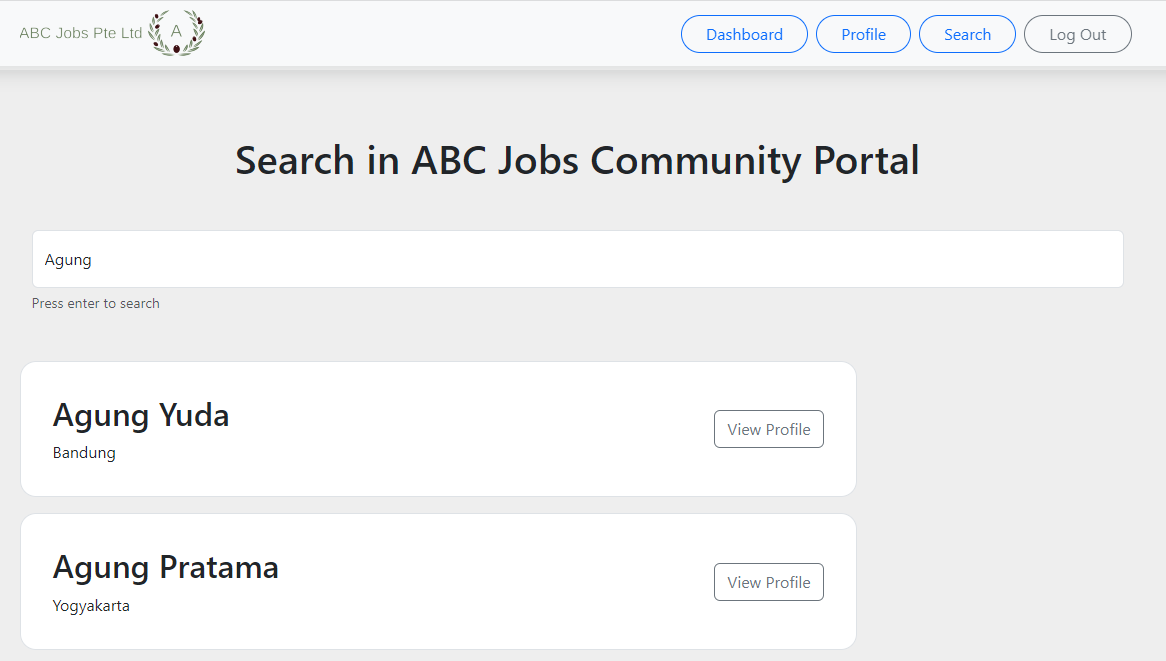


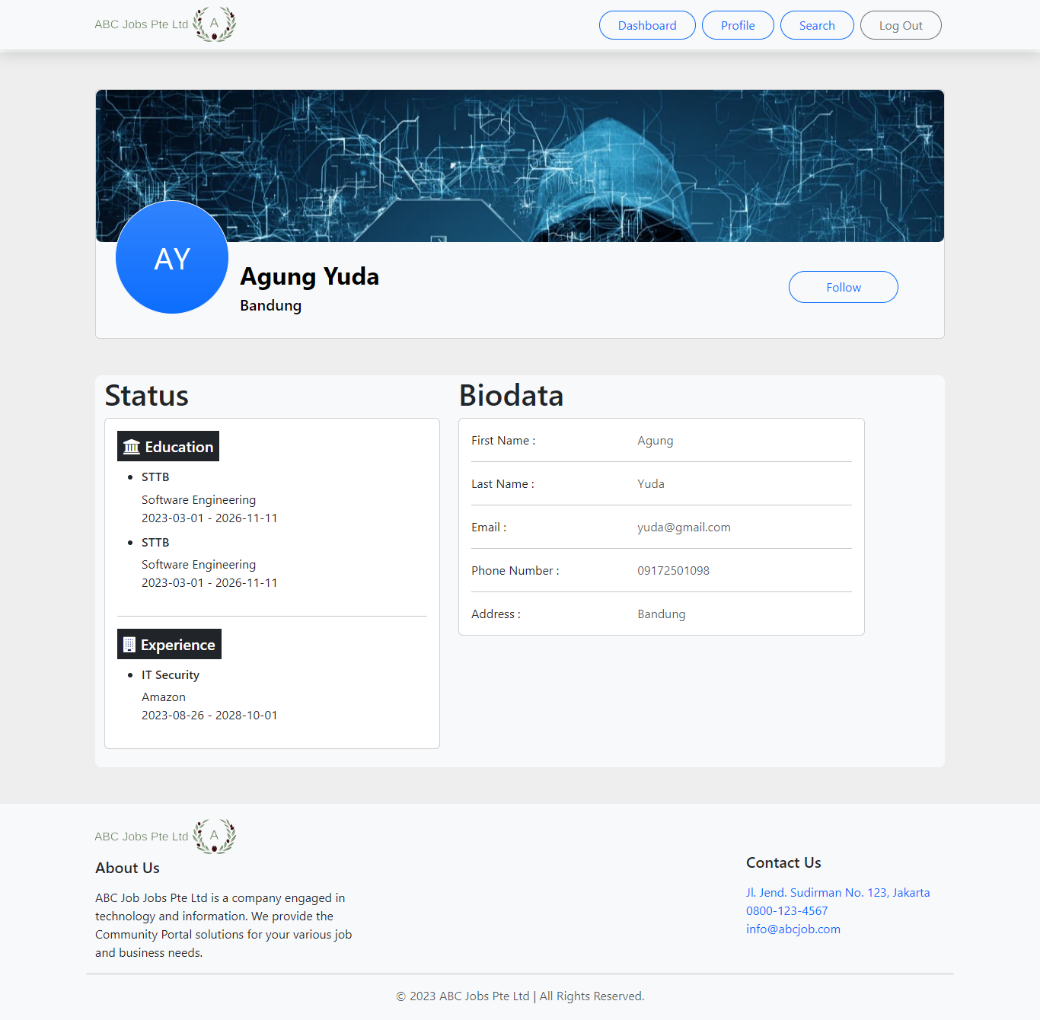




|  |  |
| --- | --- |
| Test Scenario | Verify that after logging in, the user may search for, locate, and browse the profiles of other software engineers. |
| TS002 |
| Test Cases | UAT Testing |
| UATC01 | Verify that the functions for searching and browsing profiles are functional. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario ID | Test Case ID | Test Case | Preconditions | Test Step | Test Data | Expected Results | Actual Results | Pass / Fail |
| TS002 | UATC01 | Check searching and viewing profile function is working correctly | 1. The website is designed and developed  2.Unit testing and system testing is done  3.All the major errors has resolved and documented  4.The website is configured to Google Chrome by Apache Tomcat Server.  5. User has successfully registered and login to the website | 1. Go to Google Chrome. 2. Enter URL www.ABCPortal.com 3. Click “Sign in” in website home page 4. Sign In with registered credentials 5. Click “Search” on the navigation bar 6. Search for “Agung” and click on the “View Profile” button | Search keyword = Agung | User should be able to view Agung ‘s profile | As expected, | Pass |





|  |  |
| --- | --- |
| Test Scenario | Check change password functionality in the website |
| TS003 |
| Test Cases | UAT Testing |
| UATC01 | Check that password functionality is working correctly |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario ID | Test Case ID | Test Case | Preconditions | Test Step | Test Data | Expected Results | Actual Results | Pass / Fail |
| TS003 | UATC01 | Check that password functionality is working correctly | 1. The website is designed and developed  2.Unit testing and system testing is done  3.All the major errors has resolved and documented  4.The website is configured to Google Chrome by Apache Tomcat Server.  5. User has successfully registered and login to the website | 1. Go to Google Chrome. 2. Enter URL www.ABCPortal.com 3. Click “Sign in” in website home page 4. Click “Forgot Password ?” in the Login Page 5. Enter registered email 6. Enter new password | Registered email =  camilla@gmail.com Old password =  “qwerty123”  New password = abc123 | User should land to login page and login with new password | As expected, | Pass |

