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 | |
| **I Kadek Adi Astawa** | |  | |
| Date issued | Completion date | | Submitted on |
| **11-01-2022** | 12-15-2022 | | 12-15-2022 |
|  | |  | |
| 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: 12-15-2022 |

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. The project will be carried over through Module 3, Module 4, Module 5 and Capstone project. For this module the scope is to Design, Develop, Implement & Document Spring Website.

The Scope of the Project is to design a Community Portal a place where someone can apply for a job. Users will be able to register in the portal using the Registration Page. Users of the portal can search for other users using various parameters such as First Name, Last Name, Company Name, City & Country. Users will be able to view the Public Profile of users after searching them. The portal allow users to login, request for forgotten password and Update their profile information

1. Project Objective

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

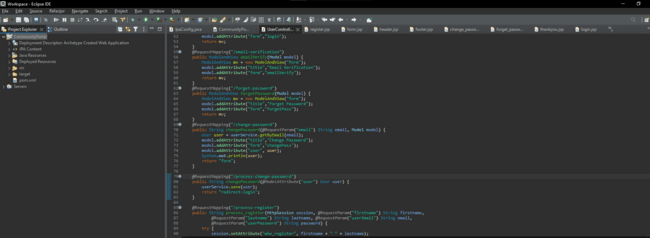
Portal using spring framework.

1. Implement the project on the project technical environment and prepare Spring Components, Javascript, Images.
2. To design the web page using jsp
3. Allow programmer to register, get email configuration and see thankyou page

Search and see other user profile

1. Retrieve the password using email.
2. Login page to login using their account.
3. Update their profile after login
4. Logout
5. Divide user into 2 types that are Software Programmer and Administrator.
6. Administration page to update and delete user data
7. And send bulk email functionality.
8. To test the website using Junit
9. Prepare Java Website Documentation
10. Project Requirements Specifications

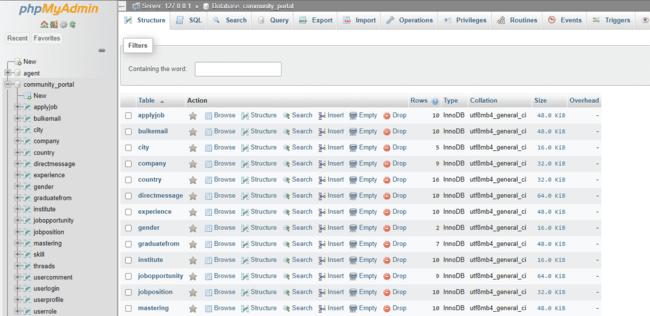
* IDE: Eclipse



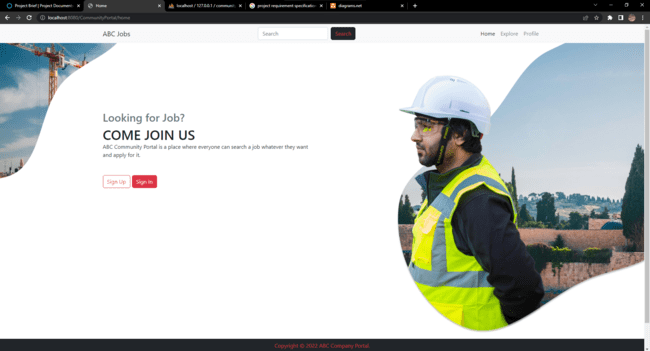
* Server: Tomcat



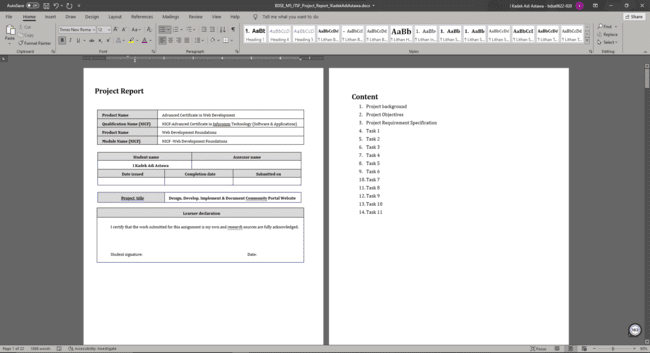
* Database: PHP



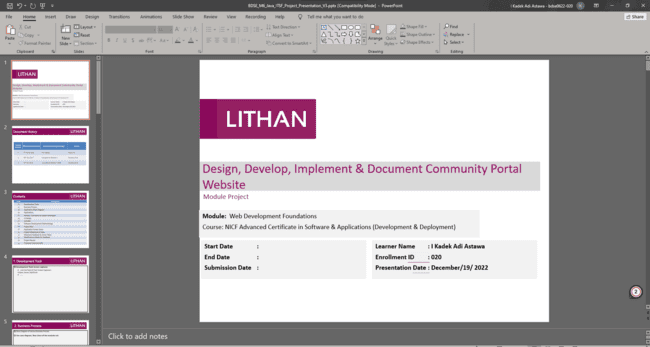
* Browser: Chrome



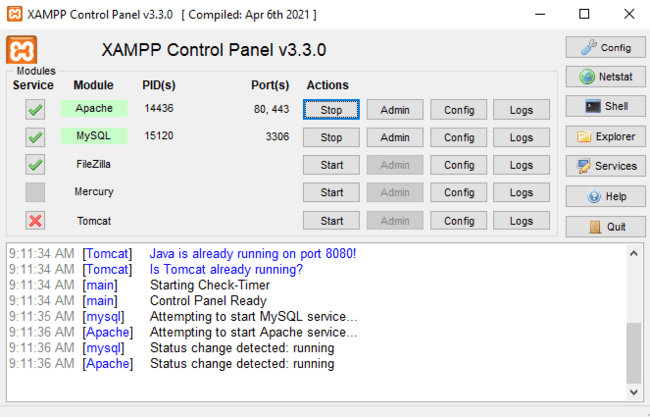
* Documentation: Microsoft Word



* Presentation: Microsoft PowerPoint



* XAMPP



1. Task 1

Task Statement:

1. Architecture of the system
2. Draw the block diagram of the system
3. Explain the Architecture of the application
4. System
5. List down all the Modules to be developed
6. List Down all the Pages in each of the module and briefly describe the purpose
7. Note down which are Create, Edit, & View Pages
8. Create the sitemap of Public & private Site
9. Technical Environment Requirements
10. System Integration Requirements
11. Portability Requirements
12. Maintainability Requirements
13. Performance Requirements
14. Security Requirements
15. Briefly explain login & logout mechanism

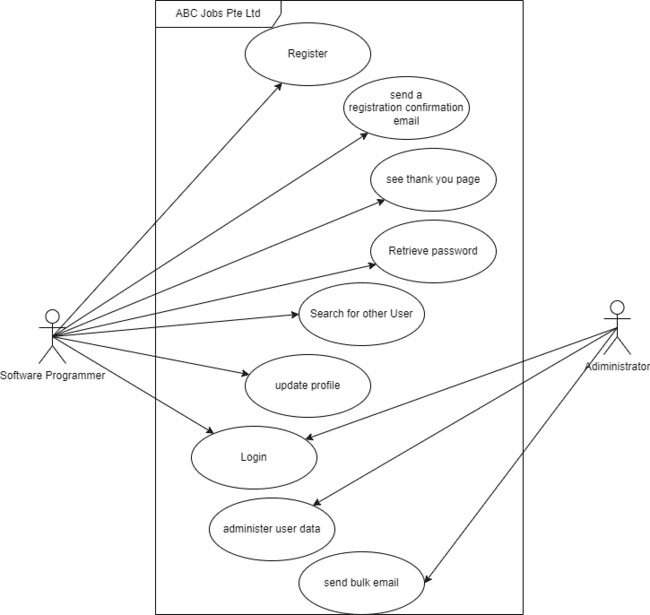
Solution:

Diagram, timeline

Description automatically generated

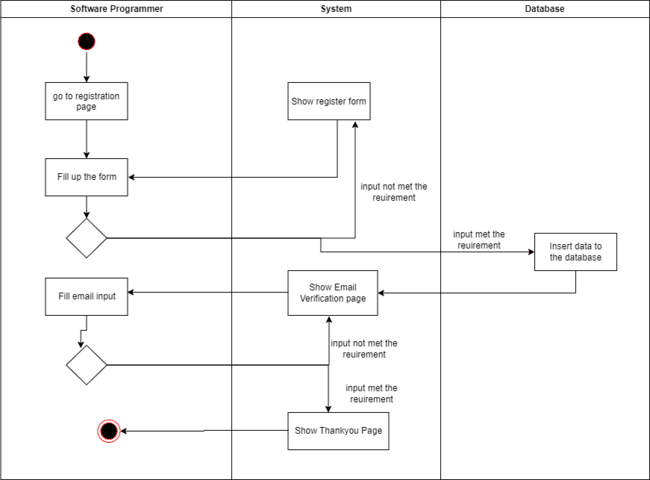
1. Architecture of the System

Use Case Diagram:

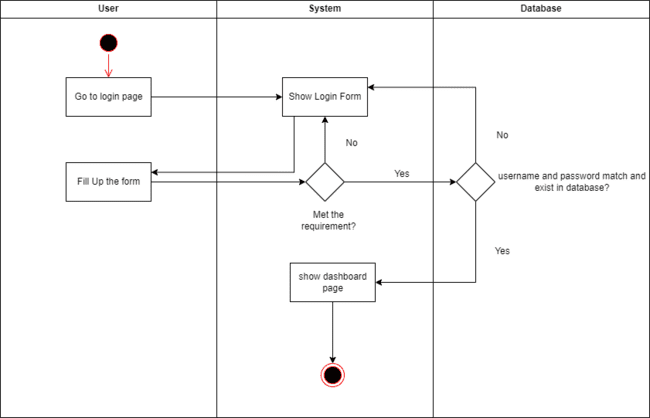


Activity Diagram:

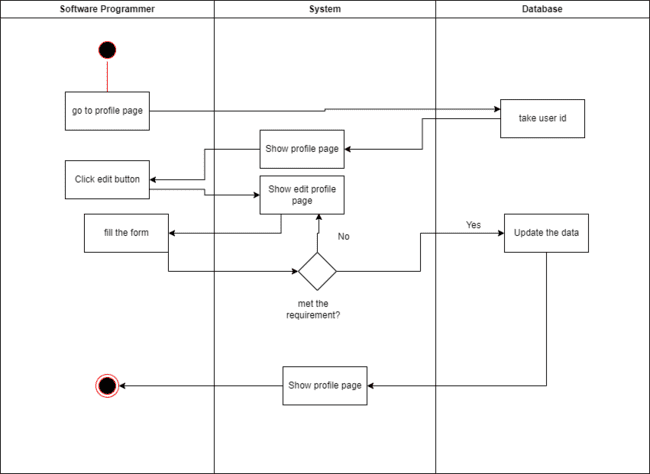
1. Registration



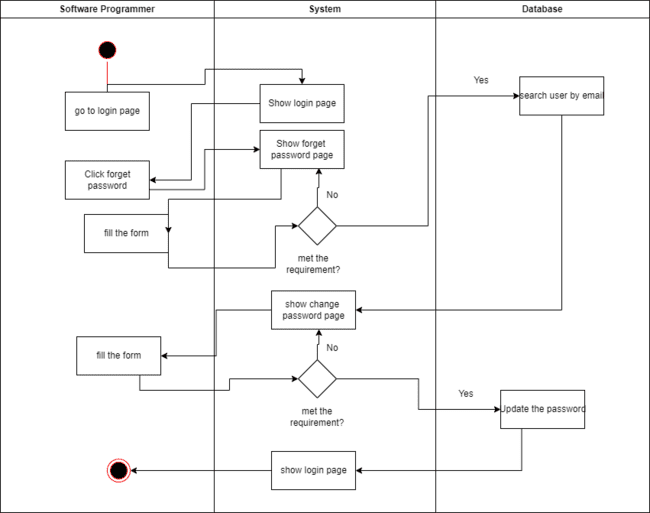
1. Login



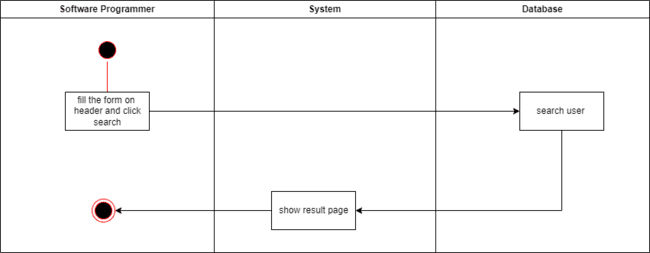
1. Update Profile



1. Retrieve Password



1. Search



Task Statement (b): Architecture of the system

Solution:

Diagram

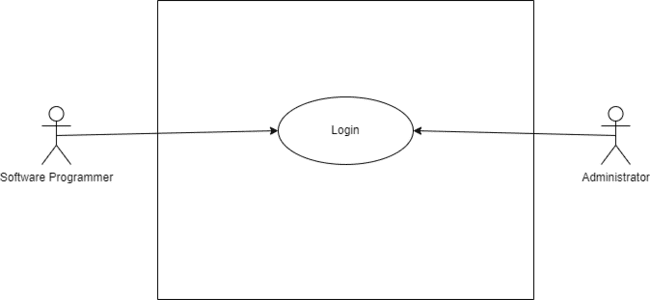
Description automatically generated

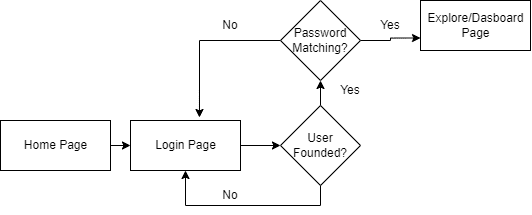
Explanation:

Every request that come from user will be handled by front controller. Front controller is a servlet called DispatcherServlet, after the request comes to front controller the request will be dispatched to the controller. The controller will give response depends on what the request is, if it need data from database controller need to connect to service to use some function in order to take data from database and send response back to front controller.

Example:

* Login Scenario





|  |  |
| --- | --- |
| Process | Description |
|  | After passing the validation of email and password user input will be transferred to controller by using @RequestParam Annotation in the process login parameter. |
|  | The email that user passes through form input will be checked if it exists or not using custom method from Service. |
|  | In the Sevice class, there are many default methods provided by the repository and custom methods created manually to use specific queries.  The custom method to find user by their username will return object, because many data will be needed in the login controller. |
|  | The default method that mentioned earlier only can be used after the repository extends to CrudRepository and the custom method will be created after crate @Query Annotation that has some value in it. |
|  | If the user not founded the program will not get stopped because the process using try catch. The catch will return a string that redirect to login page while passing error message. |
|  | If the user founded and user password match the input, user object will be stored in variable and save user object in the session. Otherwise the page will be returned to login page and passing error message. |

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

Private Module:

* Manage User Data
* Send Bulk Email

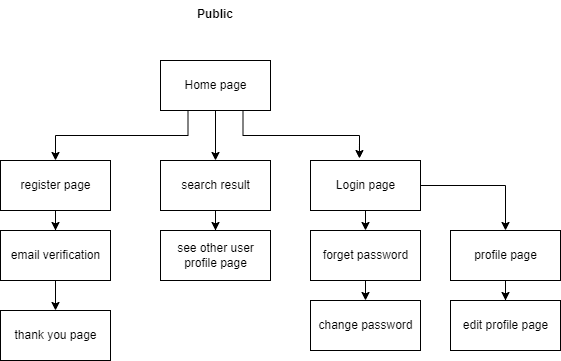
Public Module:

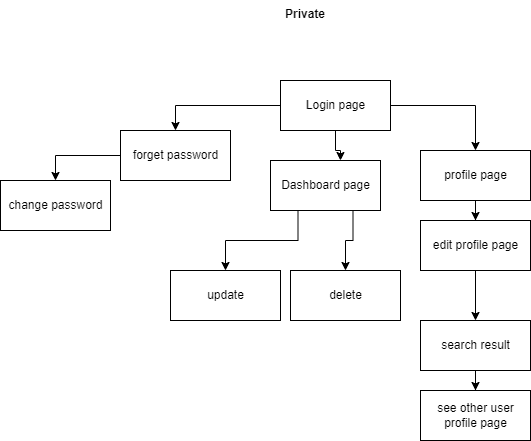
* Register
* Login
* Search
* Update Profile
* Retrieve Password
  1. List down all the pages in each of the modules and briefly describe the purpose

1. Home Page: Page that has information about ABC Jobs.
2. Register Page: User will create their account here.
3. Email Verification Page: After creating an account user will get a code through email and the user need to put the code on this page.
4. Thank you Page: The thank you page will come up after the user success verify their email.
5. Login Page: The login page is a gate for users to enter ABC Job.
6. Profile Page: The information of the user that login will be displayed here
7. Edit Profile Page: This page will help user update their profile.
8. Search Result Page: The Result of the search field on header will appear here.
9. Other User Profile Page: Other user profile will be displayed here
10. Forget Password Page: To change their password user need to go to this page first in order to send their email.
11. Change Password Page: This page where user can type their new password.
    1. Note down which are Create, Edit & View pages.

|  |  |  |
| --- | --- | --- |
| Pages | Brief description | Operations |
| footer.jsp | To display Copyright and Company name | View |
| Header.jsp | To display navigator, search form, and company logo | View |
| Home.jsp | Landing page that has company information | View |
| Register.jsp | Page to create new account | Create |
| Email-verification.jsp | This page will verify user email | View |
| Thank-you-page.jsp | To display thankyou letter for registering | View |
| Login.jsp | For login to ABC jobs | Select |
| Forget-password.jsp | To send an email that the password will be changed | Select,Edit |
| Change-password.jsp | To change the password | Edit |
| Profile.jsp | To display users profile | Select,View |
| Edit-profile.jsp | To edit user profile | Select, Edit |
| Search-result.jsp | To display search result | Select ,View |
| Other-user.jsp | To diplay other user profile | Select,View |
| Dashboard.jsp | To display all user | Select,View |

* 1. Create the Sitemap of Public & Private Site





1. Technical Environment Requirements

Hardware Requirements

* Database Server: MySQL Server
* Firewall Setting:
* Port 8080 (localhost)
* Port 3306 (database)

Software Requirements

* Development Environment: JDK 18.0.2.1
* Framework: Spring Framework 5.1.4 RELEASE
* Database: MySQL Server 10.4.24
* IDE: Eclipse

1. System Integration Requirements

* Mysql: Mysql Connector 8.0.31

1. Portability Requirements

* Multiple OS: Windows, Mac, and Linux
* Multiple Browser: Chrome, Safari, Microsoft Edge, and Firefox
* Multiple Devices: Desktop, Tablet, Mobile

1. Maintainability Requirements

* Backup the database for every 6 hours.
* Vulnerability check – periodical scanning.

1. Performance Requirements

* Concurrent number of users: 100
* Loading time: 15 seconds
* Each request should be processed within 10 seconds.

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

|  |  |  |
| --- | --- | --- |
| Pages | Access based on User Role | |
| Software Programmer | Administrator |
| Home Page | View | View |
| Register Page | View/Create | View/Create |
| Email Verification Page | View | View |
| Thankyou Page | View | View |
| Login Page | View/Select | View/Select |
| Forget Password Page | View/Select | View/Select |
| Change Password Page | View/Edit | View/Edit |
| Search User & Other User Profile Page | View/Select | View/Select |
| Public Profile Page | View/Select | View/Select |
| Update Profile Page | View/Edit | View/Edit |
| Dashboard Admin Page | N/A | View/Select |
| Bulk Email | N/A | View/Create |

|  |  |  |
| --- | --- | --- |
| Functions | Access based on User Role | |
| Software Programmer | Administrator |
| Register | Yes | Yes |
| Login | Yes | Yes |
| Change Password | Yes | Yes |
| Update Profile | Yes | Yes |
| Search User | Yes | Yes |
| View Other User Profile | Yes | Yes |
| Update Other User Profile | No | Yes |
| Delete Other User Profile | No | Yes |
| Send Bulk Email | No | Yes |

1. Briefly explain Login & Logout mechanism

Login:

First User need to fill the form on login page and click login. User input will be checked at database if they were exist or not, after email founded system will check if their email and password match or not, if the email and password match system will create a session with user id as session value.

Log out:

A button to log out will appear after user login, this button have function to delete the session. If system doesn’t find any session in browser, system will redirect the page to login page.

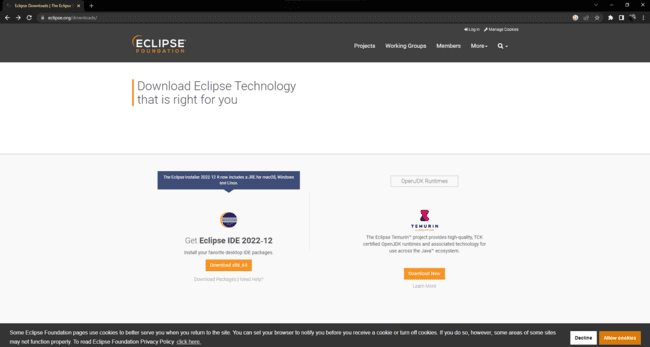
Task 2

Task Statement:

1. Install Eclipse in your laptop
2. Provide Screen capture of eclipse IDE

Solution:

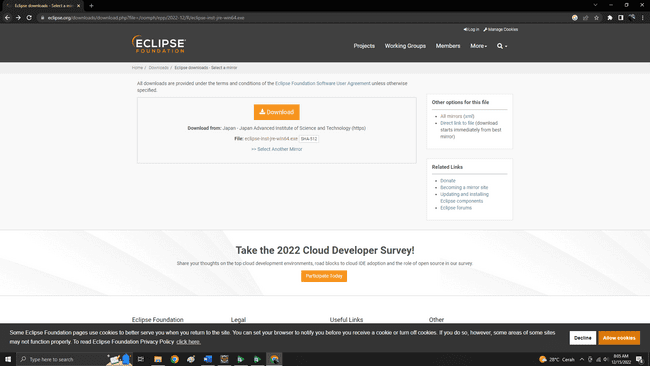
1. Go to [eclipse.org/downloads/](https://www.eclipse.org/downloads/)



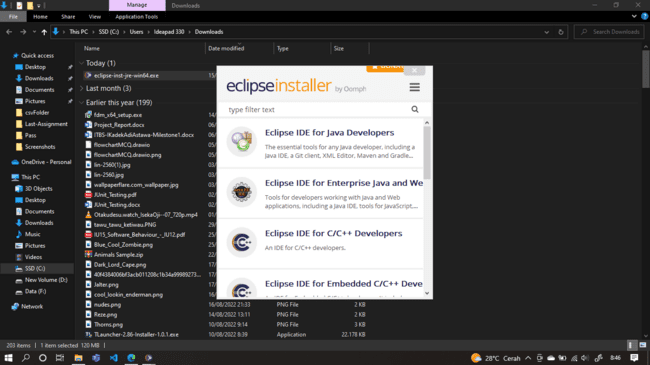
1. Click Download x86\_64



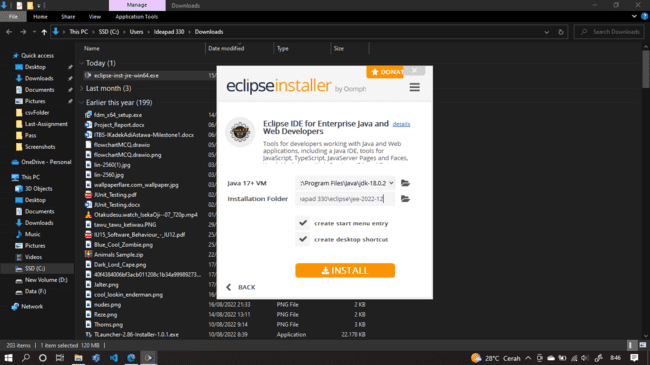
1. Wait until download done



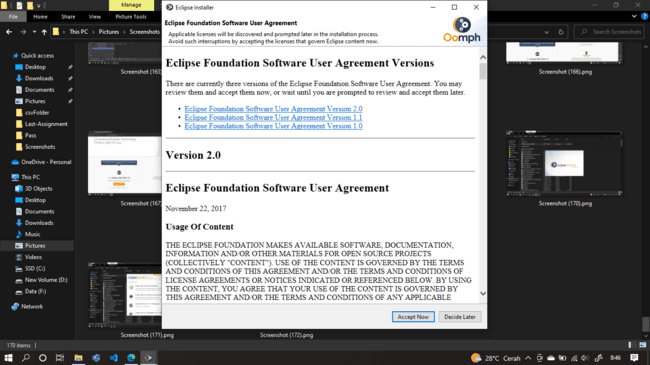
1. Select Eclipse IDE for Enterprise



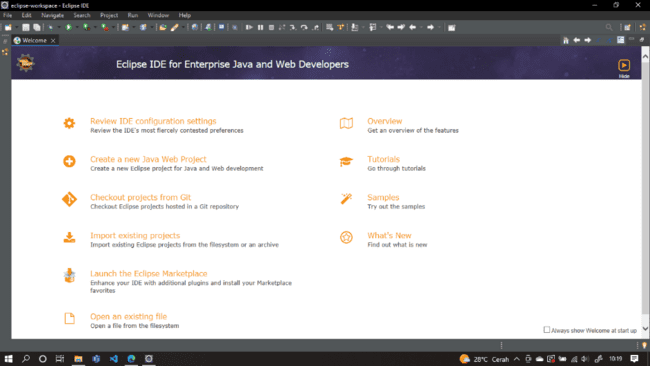
1. Select instalation folder and click install



1. Accept user agreement



1. Screen Capture of Eclipse IDE



1. Task 3

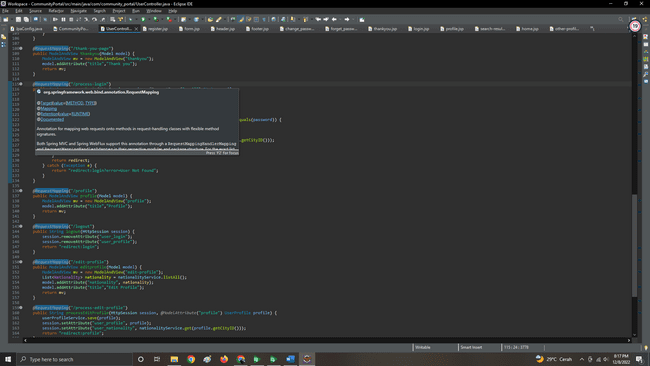
Task Statement:

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

Solution:

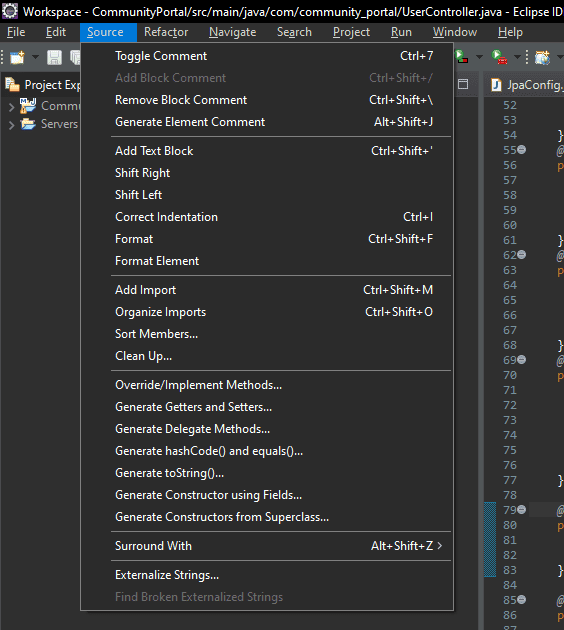
1. Explanation Features

In eclipse programmer can get information about some function just by hover it and the explanation about it will appear.



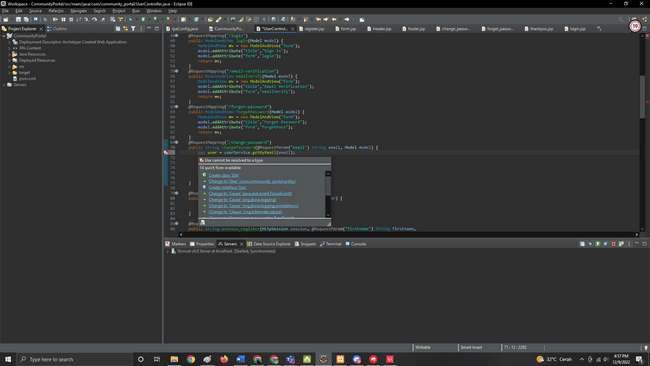
1. Auto Generate Features

This feature can generate some code in instant, by go to source and click what to generate, there are many option from generate gettter setter to wrap some code using try catch block.



1. Error Handler

This feature can be used when the cursor hovering the line that has the error.



1. Task 4

Task Statement

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

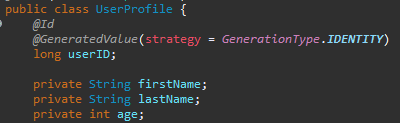
1. List 5 Coding Standards & Conventions to be used and explain shortly.
2. Describe briefly documentation tool, you will be using and its configuration.

Solution

1. Coding Standards

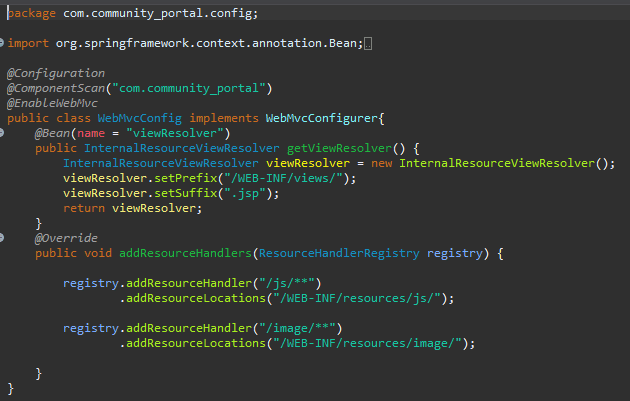
* Naming Convention

Using upper camel case to class name and lower camel case to variable.



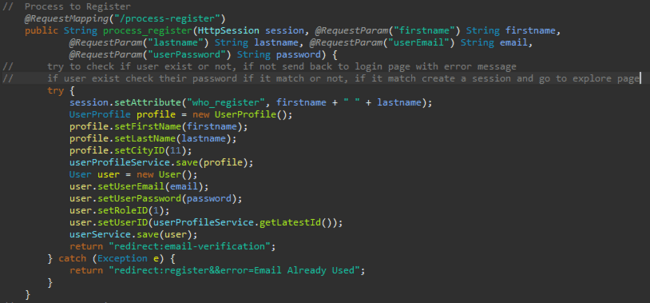
* Use Spring Java Configuration

Using java as web configuration.

****

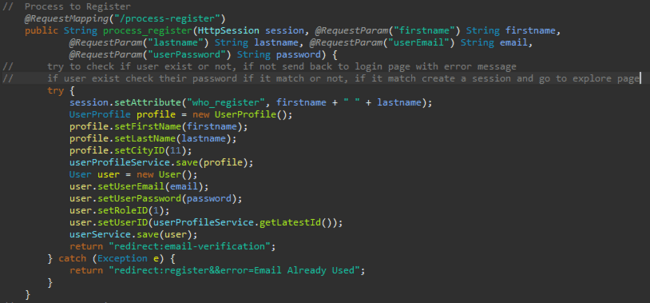
* Do Not Make one line long

If typing the code has a long length try to type it on next line to make it easier to read.



* Using Comment

Using comment to avoid confusion to another person who read the syntax.



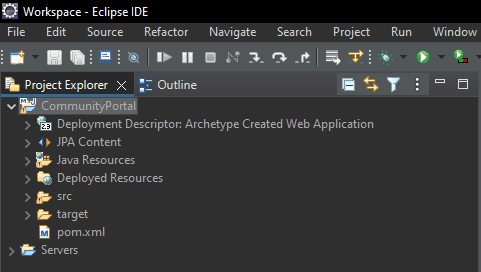
* Private class member

Keep member class private to reduce confusion when using same name as class member to another class.

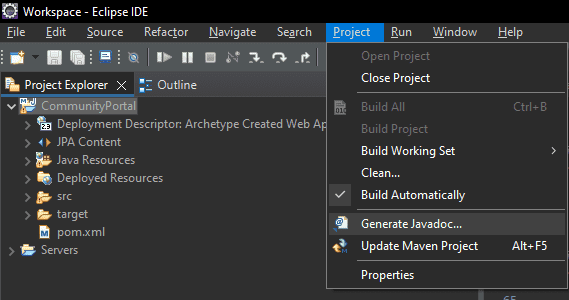


1. Documentation Tool

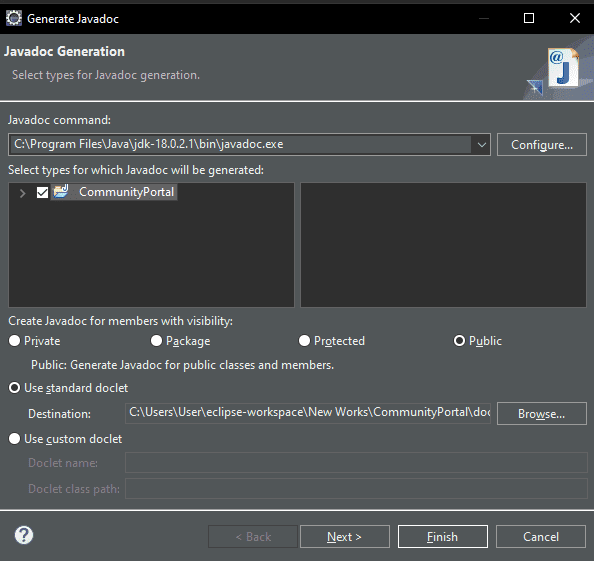
* Go to Project



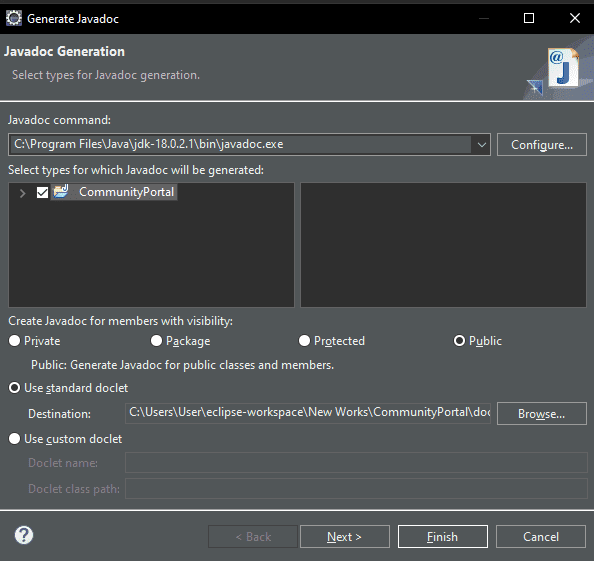
* Click generate Javadoc

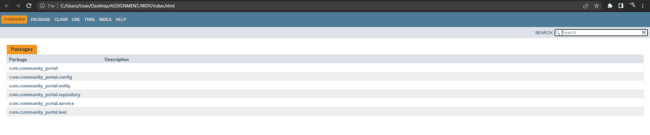


* Choose the location of Javadoc command



* Click Finish





1. Task 5 (Views)

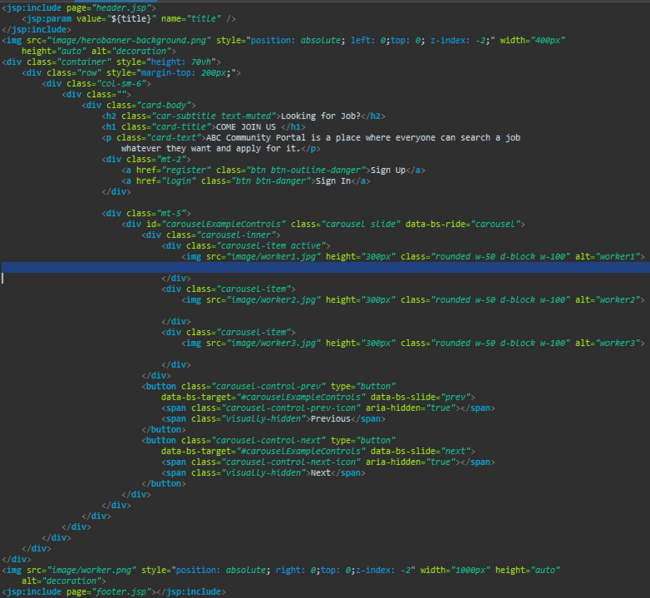
Task Statement

1. Develop the following pages in HTML
   1. Home Page
   2. Create/Edit Page
   3. Listing Page
   4. Login/Logout
   5. Forget Password screen
   6. Administration Page
2. Template them by separating Headers/ Footers and including them using Server Side includes.
3. Provide the screen capture in Project Presentation

Solution:

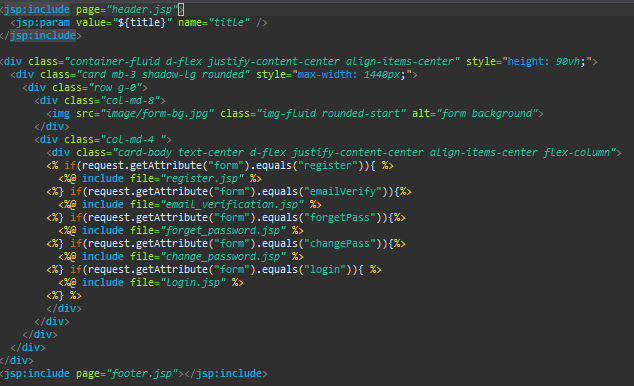
1. Home Page

Index.jsp

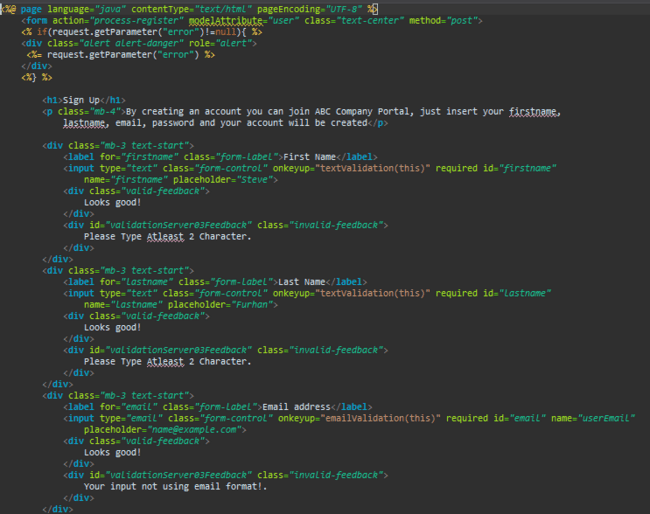


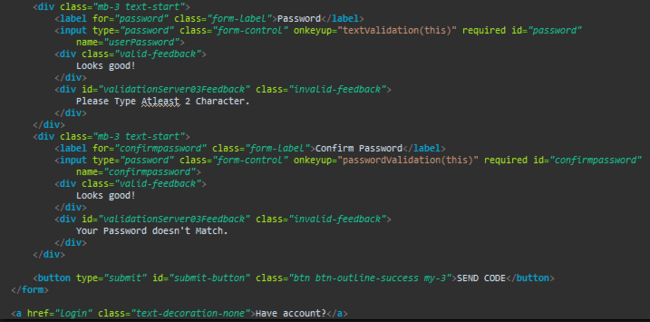
1. Registration Page

Form.jsp



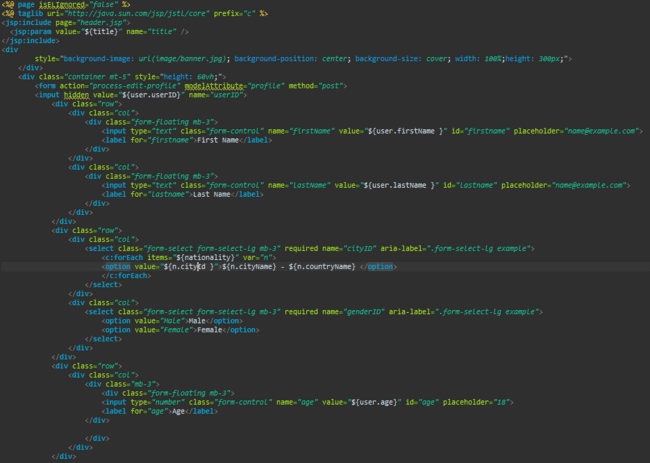
Register.jsp

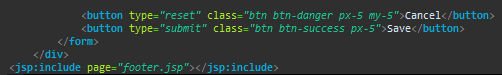




1. Edit Profile Page

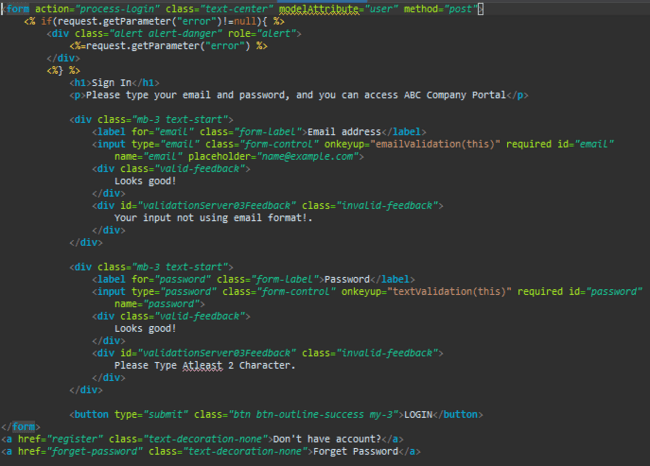
Edit-profile.jsp





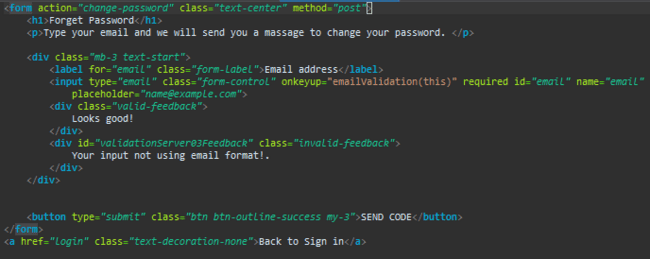
1. Login and Logout

Login.jsp

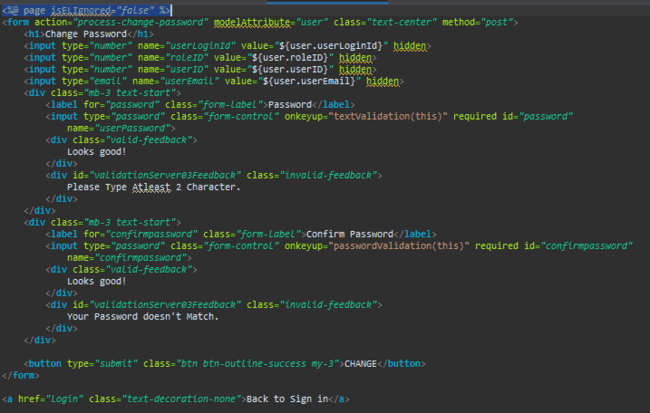


1. Forget Password Page

Forget\_password.jsp

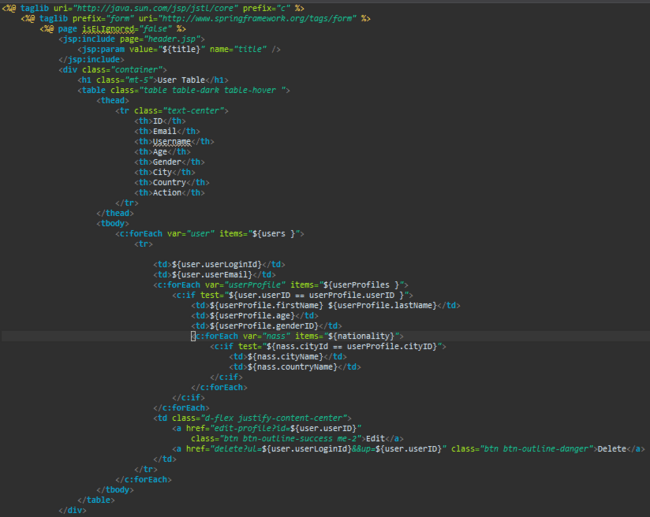


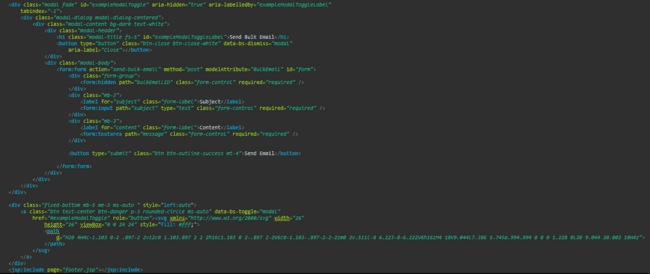
Change\_password.jsp



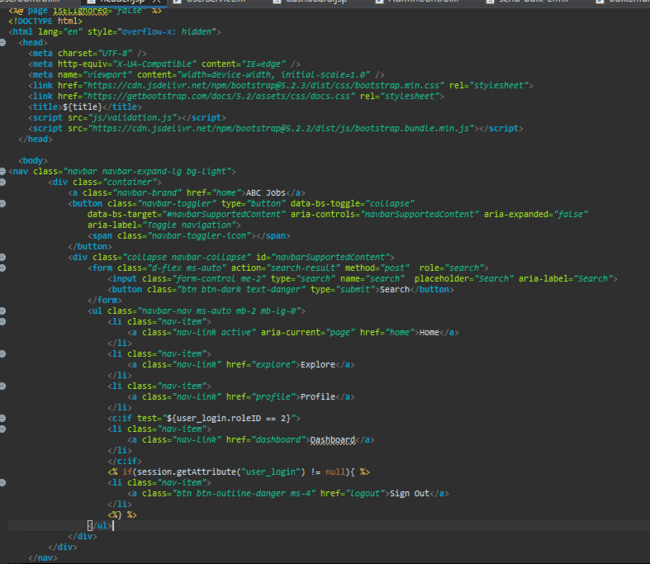
1. Administration Page

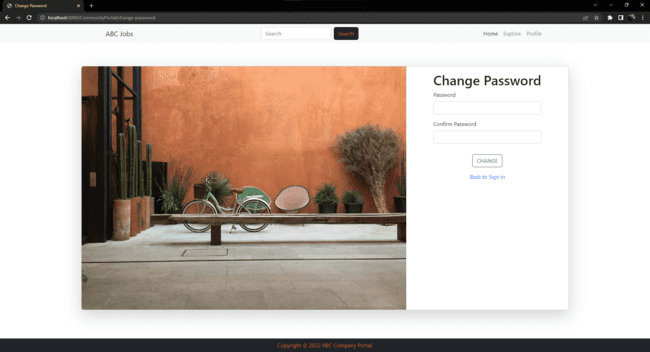
Dashboard.jsp



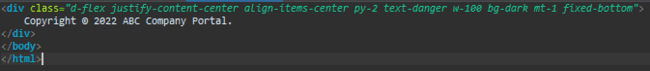


1. Header





1. Footer





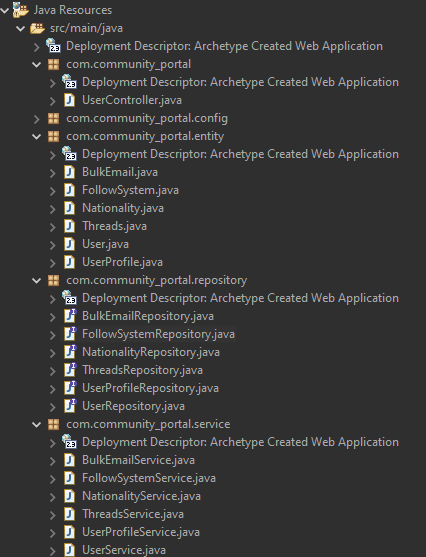
1. Task 6 (java classes)

Task Statement:

1. Create the required classes in Java
2. Include the Data Access Layer classes
3. Include samples as part of Project Report

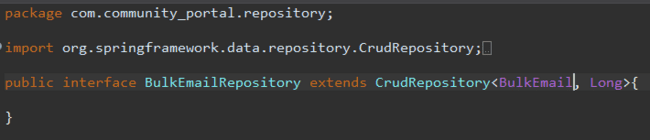
Solution:

1. Required Class

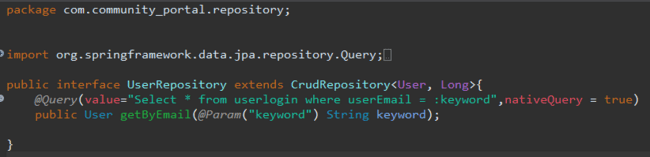


1. Data Access Layer Class

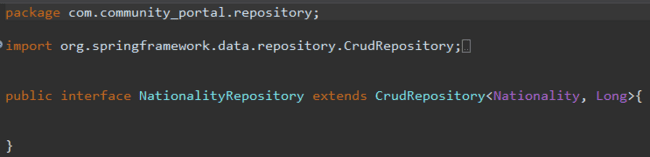
* BulkEmailRepository



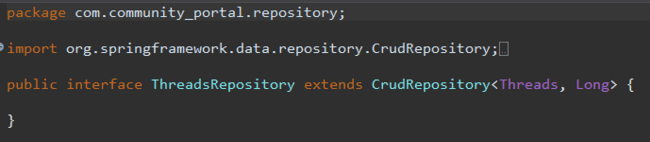
* UserRepository



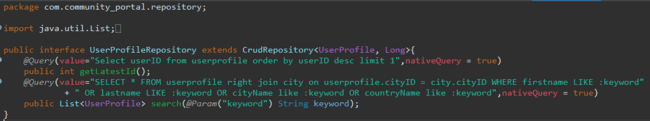
* NationalityRepository



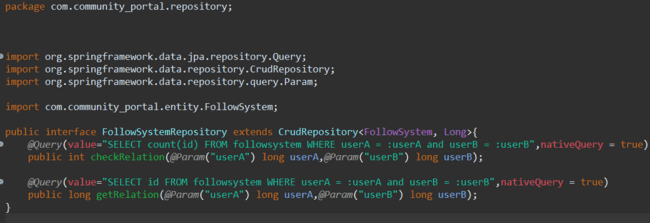
* ThreadsRepository



* UserProfileRepository



* FollowSystemRepository



1. Task 7

(screen capture of your working website – web pages)

Task Statement:

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

1. Use Queries, Procs Developed in Module 4
2. Develop the Models, Views, Controller (at least for one of the functionality, better if you can use for all the functionality)
3. Develop Home Page & Integrate with Login System
4. Develop Registration Form & Registration Thank You
5. Develop Forget Password mechanism
6. Develop Landing Page post login
7. Develop Search Box for Users
8. Develop View Search Results
9. Develop View Profile Page
10. Develop Edit Profile Functionality

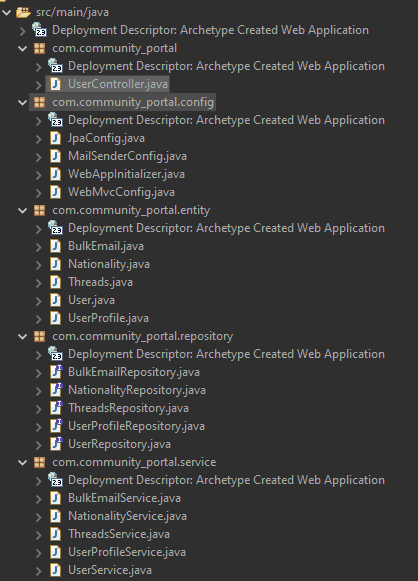
Solution:

1. Queries used

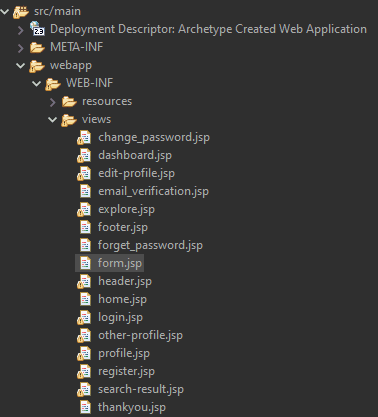
|  |  |
| --- | --- |
| Web Pages | Queries/Procedures |
| Registration.jsp | userProfileService.save(profile);  userProfileService.getLatestId()  @Query(value="Select userID from userprofile order by userID desc limit 1",nativeQuery = true)  public int getLatestId();  userService.save(user); |
| Login.jsp | userService.getByEmail(email);  @Query(value="Select \* from userlogin where userEmail = :keyword",nativeQuery = true)  public User getByEmail(@Param("keyword") String keyword);  userProfileService.get(userFounded.getUserID()); |
| Change\_password.jsp | userService.save(user); |
| Edit-profile.jsp | userProfileService.save(profile); |
| Search-result.jsp | userProfileService.search(search));  @Query(value="SELECT \* FROM userprofile right join city on userprofile.cityID = city.cityID WHERE firstname LIKE :keyword OR lastname LIKE :keyword OR cityName like :keyword OR countryName like :keyword",nativeQuery = true)  public List<UserProfile> search(@Param("keyword") String keyword); |
| Other-user.jsp | userProfileService.get(id); |
| Dashboard.jsp | userService.listAll();  bulkEmailService.sendEmail(user.getUserEmail(), bulkEmail.getSubject(), bulkEmail.getMessage());  userProfileService.listAll();  userService.listAll();  nationalityService.listAll();  userService.delete(ul);  userProfileService.delete(up); |
| Other-profile.jsp | followSysService.save(follSys);  followSysService.delete(id);  @Query(value="SELECT count(id) FROM followsystem WHERE userA = :userA and userB = :userB",nativeQuery = true)  public int checkRelation(@Param("userA") long userA,@Param("userB") long userB);    @Query(value="SELECT id FROM followsystem WHERE userA = :userA and userB = :userB",nativeQuery = true)  public long getRelation(@Param("userA") long userA,@Param("userB") long userB); |

1. Models,Views, and Controller

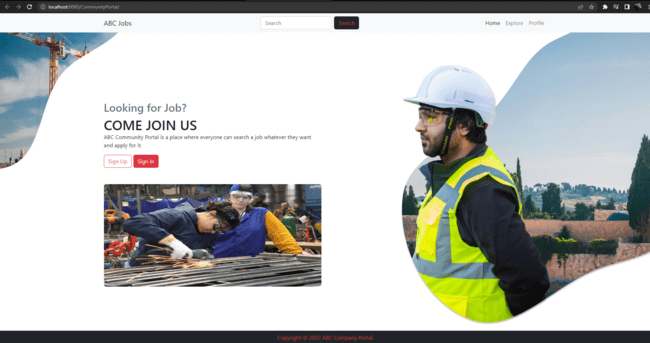
Model and Controller



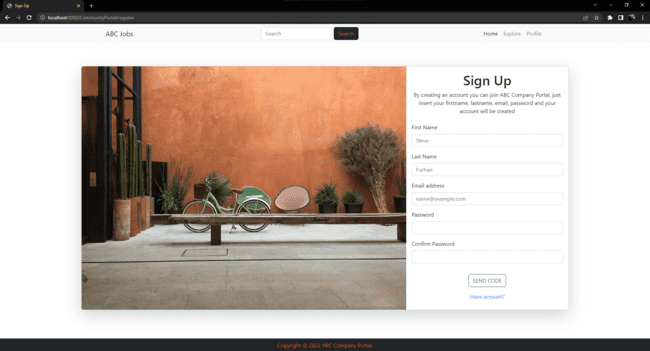
Views

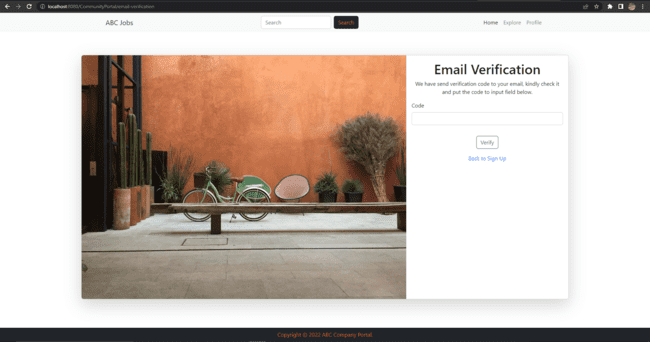


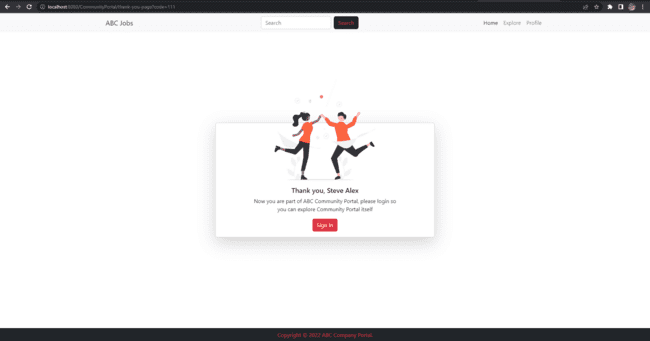
1. Home page



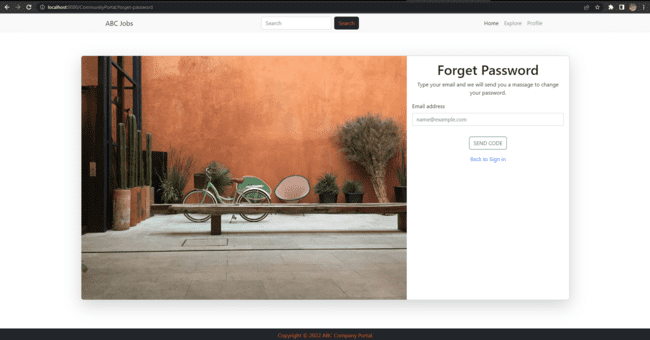
1. Registration, Email Verification, and Thankyou Page

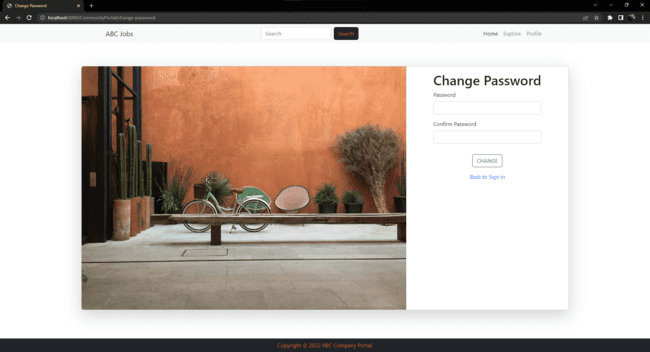




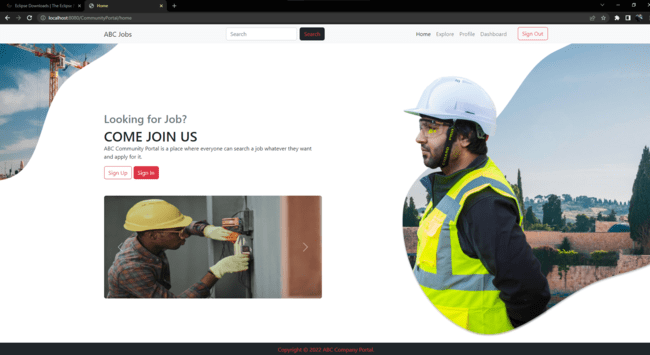


1. Forget Password





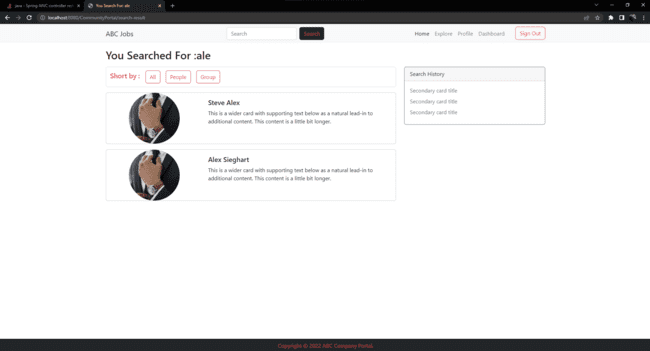
1. Landing Page Post Login



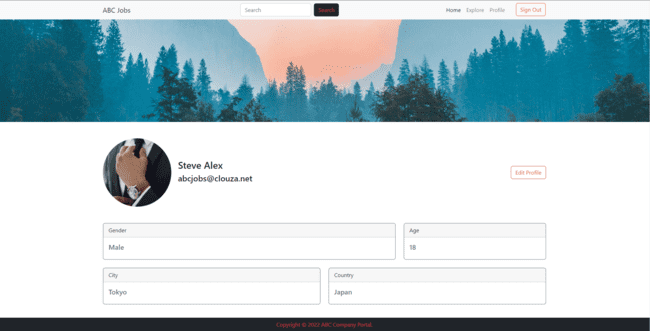
1. Search Box



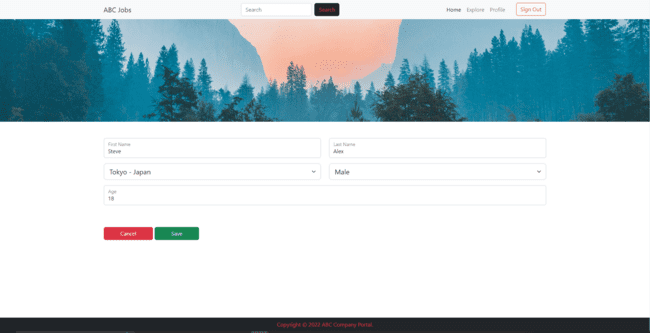
1. Search Result



1. Profile Page



1. Edit Profile Page



1. Task 8

Task Statement:

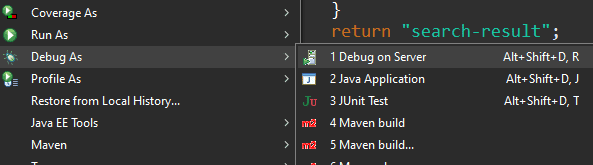
1. Debug the application in Eclipse
2. Provide the screen capture of your application code in Eclipse Debug Mode include it as part of Project Presentation

**Solution:**

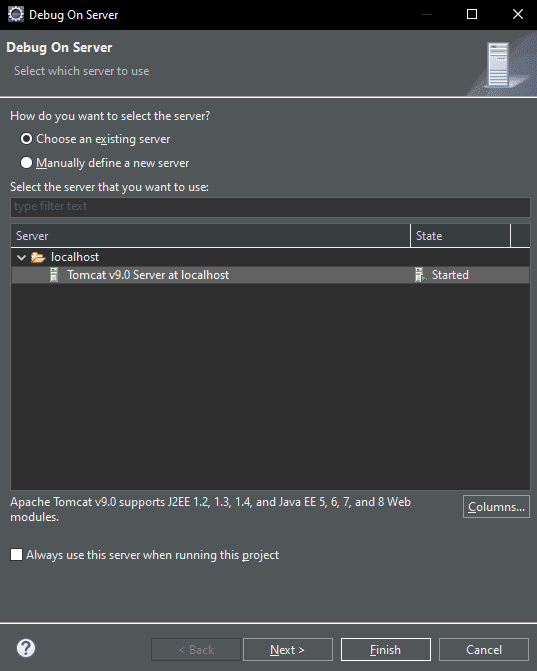
1. **Set breakpoint by clicking on left side on line number**

****

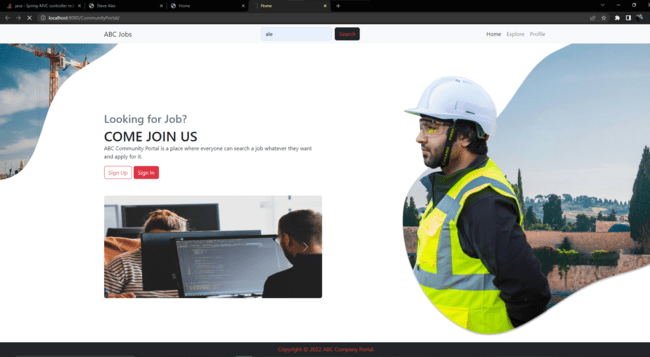
1. **Right click at the project name Debug As > Debug on Server**

****

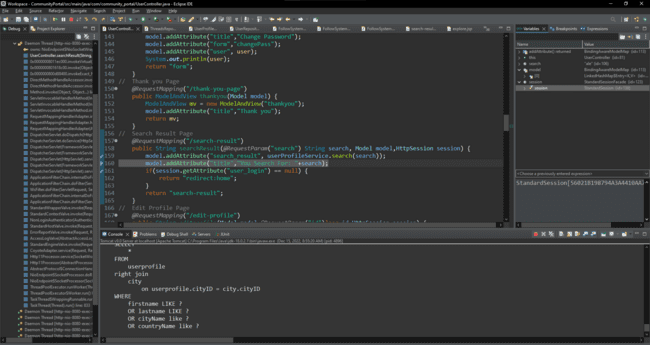
1. **Select the server and click finish**

****

1. **Now go to your page that you want to debug**

****

1. **Go to eclipse to see the debug mode**

****

1. **Click step over button to continue process**

****

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

**Solution:**

|  |  |
| --- | --- |
| **Error** | **Error handling (Solution)** |
|  | -add:  <resource>  <directory>src/main</directory>  <includes>  <include>META-INF/persistence.xml</include>  </includes>  </resource>  -change the directory of META-INF to src/main/META-INF  -update project maven |
|  | - check repository  - check service  - check controller |
|  | -make sure mysql server is strated  -rerun the server  -try new workspace  -update maven project |

1. Task 10

Task Statement:

1. Propose & Document a New feature to the application
2. Develop the new feature and provide a report.
3. Include it as part of Project Report

**Solution:**

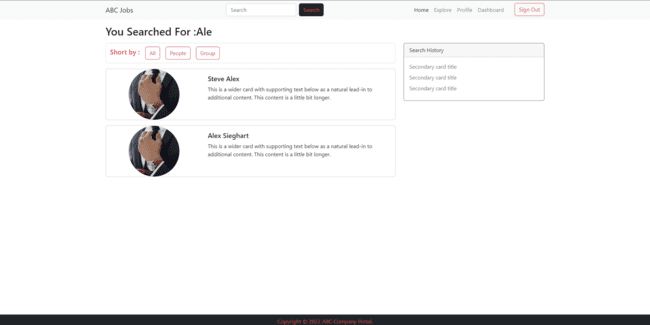
New feature that added is Follow System, by clicking follow button on other user profile page user will follow other user. The purpose of this feature is to follow other user, user threads will show the threads by user follow.

Step by step to use this feature:

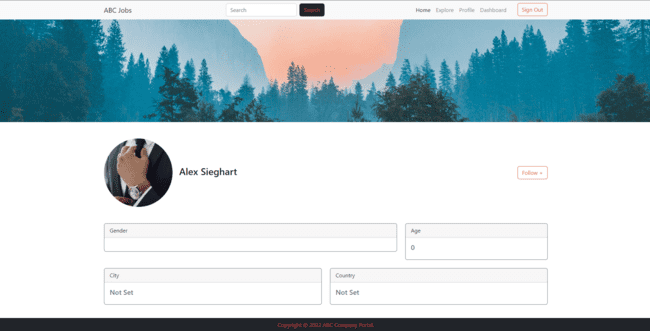
1. After sign in search someone in search box



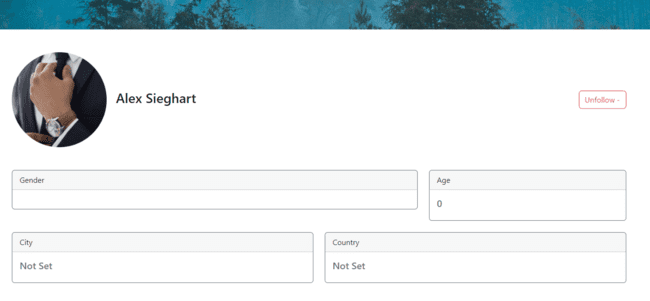
1. Click the card to open the profile



1. Click follow to follow the user



1. Unfollow button will appear after follow someone



1. Task 11

Task Statement:

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

Solution:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | | | Follow Other User | | | | |
| TSUT01 | | |
| Test Cases | | | Following other user account in ABC Jobs | | | | |
| TCUT01 | | | Validating controller if function in service working or not | | | | |
| Test Scenario ID | Test Case ID | Test Priority | | Pre-condition | Expected Result | Actual Result | Pass/Fail |
| TSUT01 | TCUT01 | Extreme | | ABC Jobs website is completed | The return that passed save method from FollowSystemService must return true | Both method returning true | Pass |
| Test Step & Test Data |  | | | | | | |
| Evidence |  | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | | | Register | | | | |
| TSUT02 | | |
| Test Cases | | | Registering new account in ABC Jobs | | | | |
| TCUT02 | | | Validating controller if function in service working or not | | | | |
| Test Scenario ID | Test Case ID | Test Priority | | Pre-condition | Expected Result | Actual Result | Pass/Fail |
| TSUT02 | TCUT02 | Extreme | | ABC Jobs website is completed | The return that passed save method from both service UserService and UserProfile Service mustn’t null | Both method returning true | Pass |
| Test Step & Test Data |  | | | | | | |
| Evidence |  | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | | | Change Password | | | | |
| TSUT03 | | |
| Test Cases | | | Changing the Password of the user | | | | |
| TCUT03 | | | Validating controller if function in service working or not | | | | |
| Test Scenario ID | Test Case ID | Test Priority | | Pre-condition | Expected Result | Actual Result | Pass/Fail |
| TSUT03 | TCUT03 | Extreme | | ABC Jobs website is completed and user exist | The return must be true | Save method returning true | Pass |
| Test Step & Test Data |  | | | | | | |
| Evidence |  | | | | | | |