

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2022

Talent Jet

Yashwanth Reddy Kasarla

Follow this and additional works at: <https://opus.govst.edu/capstones>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

TALENT JET

by

YASHWANTH REDDY KASARLA

Bachelors in Computer Science and Engineering, JNTUH, 2019

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science



**Governors State University
University Park, IL 60484**

2023

ABSTRACT

In today's world, finding a job is like a rat race in every professional field. The scenario is such that finding jobs that fit the skill sets and interests takes much work for job seekers. On the other side, recruiters face similar challenges in finding the finest candidate to match their requirements.

Talent jet is a website with unique ingredients to solve all these challenges. Talent jet provides an easy and convenient search application for job seekers to find their desired job and recruiters to find suitable candidates. Talent jet will provide a platform where job seekers and recruiters can interact with each other irrespective of geographical barriers, with just a click. That is why Talent jet is an important project.

The website is developed by our team from scratch with all the required functionality to make the website user-friendly. It will contain mainly three user roles jobseeker, recruiter, and administrator. All of them will have a different console with unique functions. Jobseekers need to set up an account with the website before applying for a position. Recruiters will post job openings and contact applicants. The administrator will manage users and job postings. The home page will be the landing page after signup and signing in. It will be very interactive with different and useful sections to navigate

To develop this website, we have used reliable and robust programming language like Java, with spring boot framework for back-end. HTML, CSS, and Java script for front-end. MySQL for creating database and tables to store the data. Talent Jet will be release out and will be available to all the users from 29/11/2022, Tuesday.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements.....	1
1.5	Definitions and Acronyms	1
2	<i>Technical Description</i>	2
2.1	Project/Application Architecture	3
2.2	Project/Application Information flows	4
2.3	Interactions with other Applications	5
2.4	Capabilities	5
2.5	Risk Assessment and Management.....	6
3	<i>Project Requirements</i>	6
3.1	Identification of Requirements	6
3.2	Operations, Administration, Maintenance and Provisioning (OAM&P)	6
3.3	Security and Fraud Prevention.....	6
3.4	Release and Transition Plan.....	6
4	<i>Project Design Description</i>	6
5	<i>Project Internal/external Interface Impacts and Specification</i>	16
6	<i>Project Design Units Impacts</i>	17
6.1	Functional Area/Design Unit A	17
6.1.1	<i>Functional Overview</i>	17
6.1.2	<i>Impacts</i>	17
6.1.3	<i>Requirements</i>	17
7	<i>Acknowledgements</i>	17
8	<i>References</i>	18
9	<i>Appendices</i>	18

1 Project Description

Talent Jet is designed to provide jobseekers and recruiters with a common platform to search for jobs and find talents. This website will have three user roles: jobseeker, recruiter, and admin. Here, the job seeker can apply for the listed jobs, and the recruiter, if he/she finds the job seeker suitable, will contact for further process. The recruiter will create a job by posting a job, the job seeker will apply for the job, and the admin will have vigilance of the happening on the website; if found, any jobseeker or recruiter suspicious can delete them from the website [1].

1.1 Competitive Information

In the current scenario, there are lots of job-portal applications available for job searching, but talent jet is different and ahead of all of them with its unique features. As another job portal gives a job search feature to only registered users, talent jet has keyword and advanced search features available to all users. Any users can explore the home page and access the information available. Also, talent jet provides a contact us feature in which users can provide valuable feedback.

1.2 Relationship to Other Applications/Projects

This project has unique and independent features unrelated to available applications/projects. Here, the job seeker can apply for the job created by the recruiter. On the other hand, Recruiters have the power to select or reject a candidate based on selected parameters.

1.3 Assumptions and Dependencies

- In the entire development of the project, we have assumed that the user visiting this website is well-versed in computer handling.
- All the equipment used by the users is in good condition.
- Version control will be implemented so that if replacement is needed transition should be smooth.

1.4 Future Enhancements

Following are some future enhancements we are going to implement:

- Phase1 – we will try to improve the existing project by improving the database structure.
- Phase2 – we will implement some extra security feature like SMS integration for OTP verification.

1.5 Definitions and Acronyms

Following are the various Definitions and Acronyms used in this project:

- HTML (Hypertext markup language) - HTML stands for “Hyper Text Mark Up” language. It is one of the fundamental technologies required for the web development. It provides basic template for a web page [2].
- CSS (Cascading Style Sheet) - CSS defines the appearance of the web page. This includes fonts, page layout, colors and element positioning. The CSS makes webpage visually appealing to the end users.
- Java Script: Java Script is the most popular lightweight, compiled programming language. Java script is used to make the web page dynamic so that it will be interactive for the users.
- AJAX (Asynchronous JavaScript and XML) - AJAX is used to communicate with the servers. With AJAX, when you submit the JavaScript request, the request goes to the server, from where it gets the information that is to be displayed on the page.
- JAVA - Java is high level, class-based object-oriented language. Java provide runtime environment and API. It is also called platform independent.

- JDBC (Java Database connectivity) - JDBC defines how client will interact with the database. It provides application programming interface (APIs) for the programming language java.
- MySQL - MySQL is relational database system. MySQL provide the database and manage the data by manipulating data using SQL queries
- Bootstrap - It is an open-source CSS framework. We have used the bootstrap framework in developing the website.
- jQuery - jQuery is a fast, small, feature-rich Java script library. It makes things like HTML document traversal, manipulation, animation, event handling, and AJAX API calls very easy.
- API – Application program interface.
- JDBC – Java database connection
- JDK- Java development kit
- CRUD – Create, Retrieve, Update and Delete

2 Project Technical Description

Technologies and Languages used

For Frontend

- Hyper Text Markup Language (HTML)

HTML stands for “Hyper Text Mark Up” language. It is one of the fundamental technologies required for the web development. It provides basic template for a web page. HTML code ensures that all the code used for web page well formatted. Latest version of HTML is HTML-5 which we have used in the website development [\[2\]](#).

- Cascade Style Sheet (CSS)

CSS stands for “Cascading Style Sheet” defines the style and aesthetics used in the web page. While HTML defines the template of the web page, CSS defines the appearance of the web page. This includes fonts, page layout, colours, and element positioning. The CSS makes webpage visually appealing to the end users [\[3\]](#).

- Java Script (js)

Java Script is the most popular lightweight, compiled programming language. Java script is used to make the web page dynamic so that it will be interactive for the users. It is also known as the scripting language for the web page.

- AJAX

AJAX stands for “Asynchronous JavaScript and XML.” AJAX is used to communicate with the servers. With AJAX, when you submit the JavaScript request, the request goes to the server, from where it gets the information that is to be displayed on the page.

- jQuery

jQuery is feature rich java script libraries. jQuery help in reduces the coding and hence save the time.

- Boot strap

Boot strap is a CSS framework helps in improve the visual appearance of website also it helps in making the website responsive.

For Backend

- JAVA

For backend we have used JAVA language. Java is high level, class-based object-oriented language. As java provide runtime environment and API it is also called platform independent.

As java is class based, every line of code is inside a class. Class name should start with upper case letter. The name of java file should match the class name. And the application starts after running main class from where execution starts.

- Spring boot

For developing backend code, we have used spring boot framework. It helps developing understandable code in less time. Creating APIs using spring boot is quite easy [\[4\]](#).

For Database

- MySQL

For database in the project, we have use MySQL as the relational database. SQL is the language used to script the queries. SQL stands for “Sequential Queries Language”. MySQL provide the database and manage the data by manipulating data using SQL queries. MySQL is relational database system.

Tools Used:

For Frontend

- Visual Studio Code

We have used visual studio code as source code editor for developing frontend code. Visual studio code is preferred because it is a very light weight source code editor. It has helpful extensions in its marketplace to make programming easy [\[5\]](#).

- Google Chrome Browser

Google chrome browser is used to launch the project. It also has inbuilt developer tool which makes debugging the front-end code easy.

For Backend

- Eclipse IDE

We have used eclipse IDE as a source code editor for developing backend code. It provides in-built tomcat server for project. Also, its debugging perspective view makes debugging of backend code quite easy [\[6\]](#).

- Postman

Post man is a tool to test the API. Before sending the API for frontend integration we tested each APIs in postman so, that correct API is sent for front-end integration.

For Database

- MySQL workbench

We have used MySQL workbench as a relational database management system. Database used in our project is created using workbench [\[7\]](#).

2.1 Application Architecture

Here, this is our architecture diagram which we have used in development of project. In this architecture all the front-end is loaded at once in the client side. This enable client to send and receive data which is required. The data is sent asynchronously using AJAX. Instead of receiving whole HTML page as response client here receive response as a JSON representation of data. It then updates those part of page with new data without having to reload the entire page.

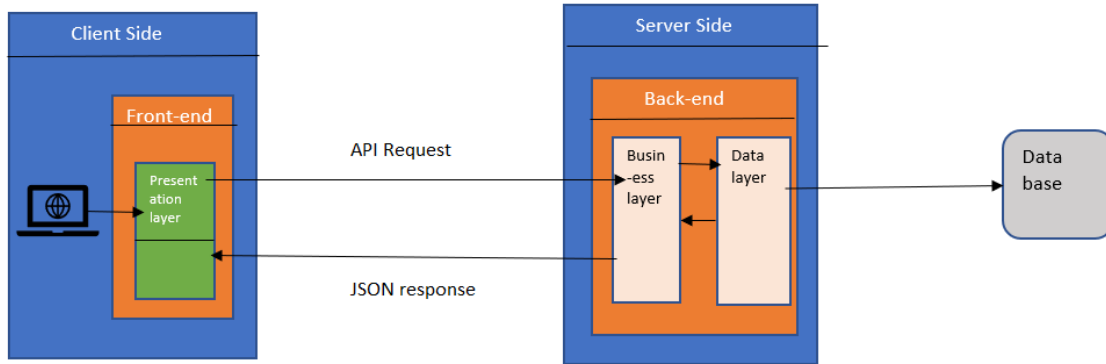


Figure 1: Architecture Diagram

2.2 Application Information flows

On the talent jet website, information flow starts from the home page, which any user who visits the website will be able to access. After that, there will be three users: jobseeker, recruiter, and admin login page. If a user has already registered as a jobseeker or recruiter, he/she will access the respective home page. Moreover, it will redirect them to the signup page if they still need to register. Once the user is registered, the user will be able to log in with an email id and password and access the home page. Furthermore, the user will end the information flow by clicking the logout button.

Admin is the super user, so they have a separate login page without a signup option. Admin login using username and password, access the admin home page and click logout to end the information flow.

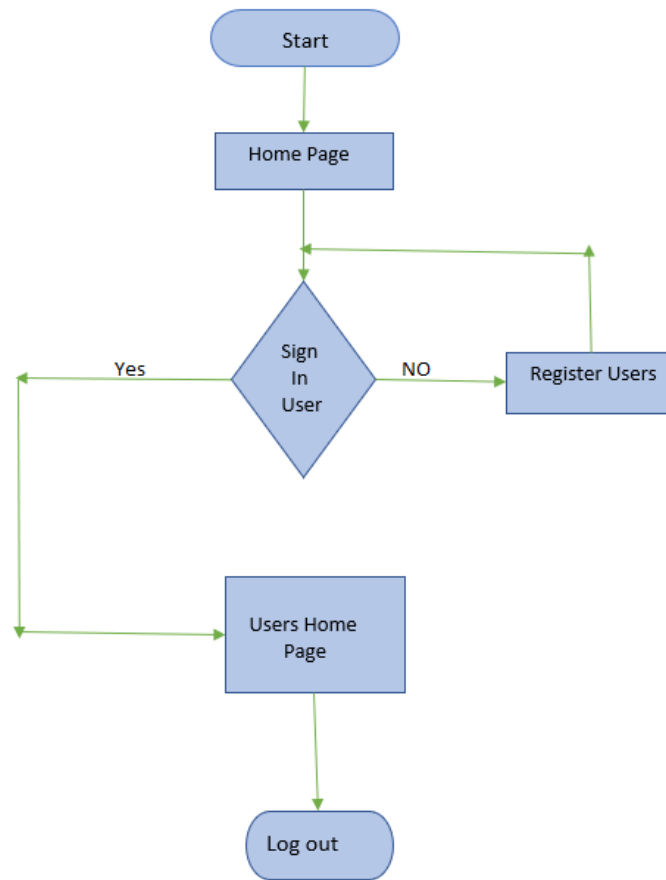


Figure 2: Information flow diagram

2.3 Interactions with other Applications

In this project, when a jobseeker applies for a job, the recruiter will contact the applicant through email id. Here, most of the interaction will take place through email id.

2.4 Capabilities

Following is the capabilities

- Recruiters can create, view, edit and update jobs.
- Admin can view and delete the jobseekers and recruiters.
- Jobseekers can create profile and update profile.
- Jobseeker can do keyword search and advance search.

2.5 Risk Assessment and Management

- Resource – As we all are novice developers; we all need some guidance for execution of project. So, to improve our resources skill we assign some training session.
- Time – As time is crucial so from starting itself, we kept meeting on regular basis to check the status of development and decide steps required to tackle the situation.

3 Project Requirements

3.1 Identification of Requirements

<Talent-Jet_2022-1 User-Capability-000100>

The project must allow three user roles: Jobseeker, Recruiter, and Administrator.

<Talent-Jet_2022-1 User-Capability-000101>

Console for each role.

<Talent-Jet_2022-1 Jobseeker-Keyword search-000102>

Jobseeker are allowed for keyword search with listing searched items.

<Talent-Jet_2022-1 Jobseeker-Advanced Search-000103>

Jobseeker are allowed for advance search of jobs.

<Talent-Jet_2022-1 Users-Registration-000104>

All the users must have registration, login, profile and account management.

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

The talent jet web application also provide facility to search jobs and explore certain informational pages to any visitor of the website. Registration and login are required when user want to apply for any job or want to post any job. The jobseeker and recruiters can register providing their educational and professional information to the website. This information will be stored in the very secured database which we have created using MySQL. Keeping information in database keep the website light weighted so that it does not take much time in loading. To safeguard the integrity of the website and its data the website will regularly backup data and keep maintain the database on regular interval.

3.3 Security and Fraud Prevention

Security and fraud preventions is an important issue to handle. It is best to include web application security at the designing and coding phase. Following steps, we are taking to keep our website secure and fraud proof:

- At designing phase, itself we have decided to use secured and robust programing language like Java, Java Script, and MySQL to design our website.
- We will train our developers on open web application security to handle any security breech.
- All the sensitive data on the website will be in encrypted form.
- Used role-based access to resources to ensures that only privileged user having authority to login can access the resource.

3.4 Release and Transition Plan

Release and transition plan includes all the steps, activates, and process required to make the application available to the end users. Following steps, we have planned to release the web application:

- Preparation:
In this stage we gathered all the required codes along with configuration files, libraries, or resource required to function the application. Moreover, ensures that the host server is running smoothly.
- Testing:
Before an updated code is released, it should be deployed to test server, where it goes through pre-configures set of automated testing. Developers will go through the test case and fix the bugs and errors before deployment.
- Release:

For releasing the code, we will create AWS account where we will create S3 bucket for front-end code, RDS for database connection and EC2 for backend code. Final step is to check bug at live server level so that end users experience hassle free service.

4 Project Design Description

Talent Job is a job searching and talent hunting platform. In this website jobseeker can search and apply for a job and recruiter can find a right talent. For this project we have created three users' roles such as Admin, Jobseeker and Recruiter. Also, each user will have their own console with different functionality.

Following is the description of each of the users.

1. Workflow of unregistered user

Any users who have not registered yet on the website can also explore the certain features of the website. So, any user can land on the home page when launch the website. In the home page user can navigate to different pages through navigation bar. Following are the features available to unregistered users:

- Any user can search job through keyword search and advance search features. In response the user will get list of jobs which user will be able to apply only when user is registered on the website.
- Unregistered user can access blog page from the navigation menu. Blog page is the place where user can learn various tips and treks on available topics.
- Any users can also give valuable feedback and ask their queries in contact us section.
- In about us section user can find all detail about Talent Jet website.

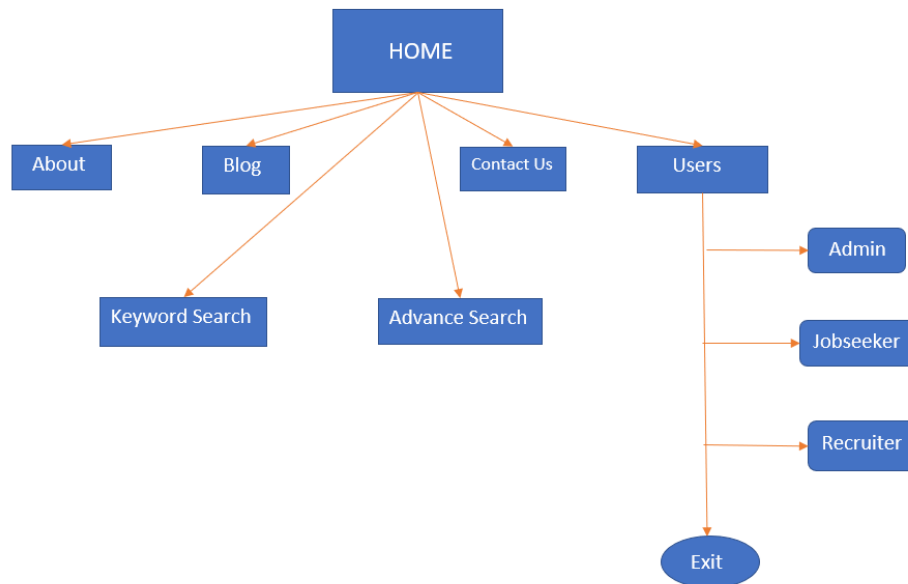


Figure 3: Workflow of Unregistered user.

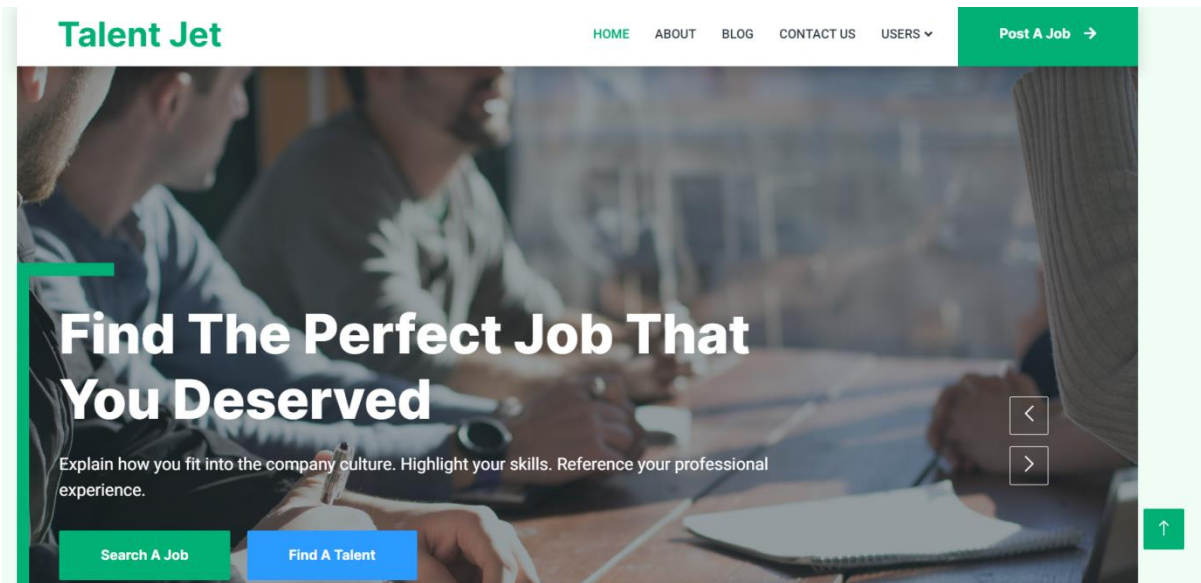


Figure 4: Home Page

2. Workflow of Administrator

Administrator or Admin is the super user of the website. Admin will have all right to access all the data available on the website.

Following are the features available to the admin:

- Admins have unique console where admin perform certain task. Admin will access the admin home page after login as admin.
- After login admin will land on the admin home page. In the home page admin can navigate to different available pages through navigation menu.
- Admin in Jobseeker list page can be able to see the list of jobseekers who have registered on the website. And if admin find any jobseeker suspicious admin have power to delete the jobseeker from the website.
- Admin in Recruiter list page can be able to see the list of recruiters who have registered on the website. And if admin find any recruiter suspicious admin have power to delete the recruiter from the website.
- Similarly, in job list page admin can view the detail of job and if admin find any suspicious job admin have power to delete that job from website.

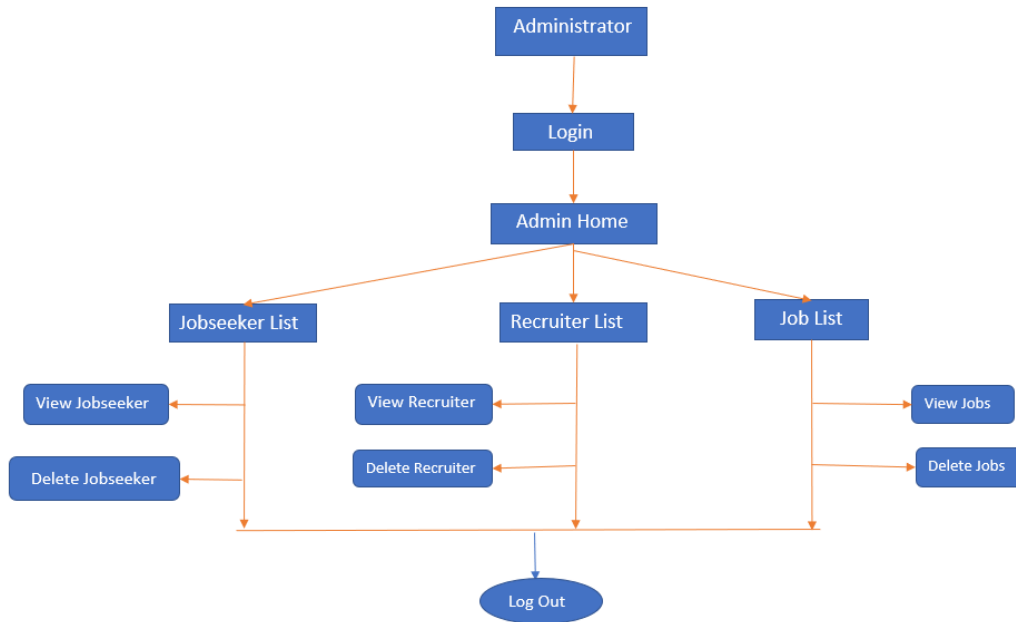


Figure 5: Workflow Diagram of admin

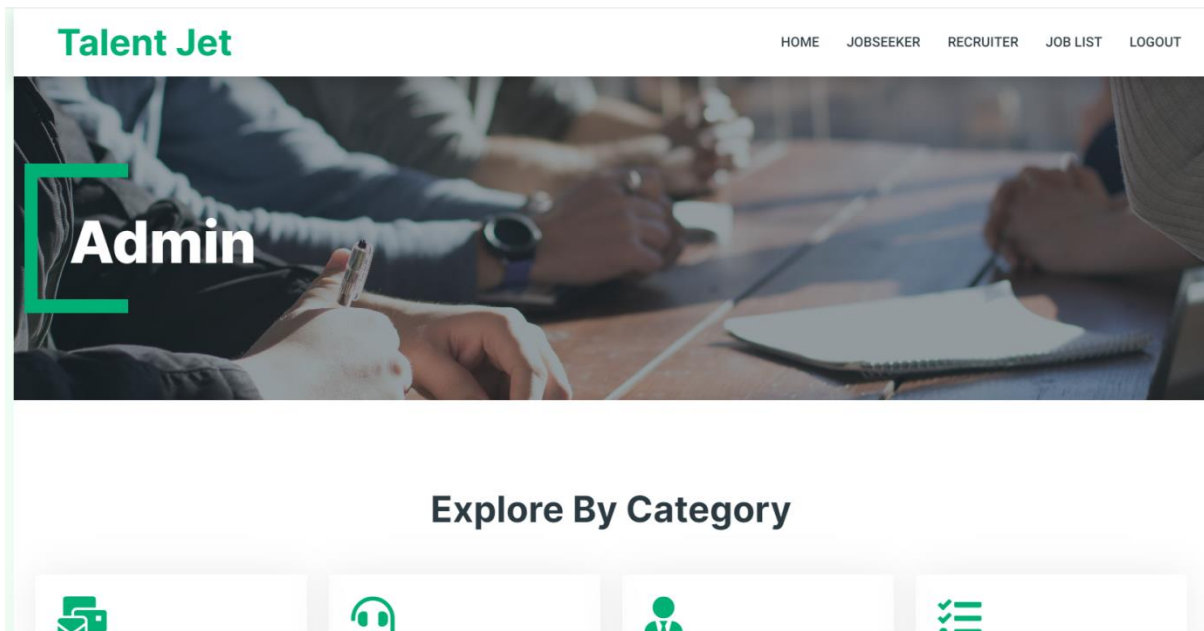


Figure 6: Admin home page

Recruiter List

Name	Company	Location	Email	View	Delete
Dharmenda	Microsoft	New York	microsoft@gmail.com	View	delete

Company	Quick Links	Contact	Newsletter
<ul style="list-style-type: none">> About Us> Contact Us> Our Services> Privacy Policy> Terms & Condition	<ul style="list-style-type: none">> About Us> Contact Us> Our Services> Privacy Policy> Terms & Condition	<p>123 Street, New York, USA</p> <p>+012 345 67890</p> <p>info@example.com</p> <p>Twitter Facebook YouTube LinkedIn</p>	<p>Dolor amet sit justo amet elittr clita ipsum elittr est.</p> <p>Your email SignUp</p>

Figure 7: Admin's Recruiter list

3. Workflow of Jobseeker

Jobseekers are the user who will come on the website to search the jobs and to apply for the job. Jobseekers are the registered user so to avail all the functionality of jobseeker user must registered as jobseeker.

Following features are available to the jobseeker:

- After registration and login using email id and password. Jobseeker will land on the jobseeker home page. In the home page jobseeker can navigate to the different pages through navigation menu.
- In profile page jobseeker can view profile and can be able to edit and update it.
- In job list page jobseeker will find list of various available jobs. Jobseeker can browse jobs here and if find any suitable job can apply for that job. Here jobseeker can also delete jobs if he/she find it not suitable.
- Applied job section jobseeker can see the list jobs that jobseeker has applied for. In applied job list page jobseeker can see the status of his application.
- Through jobseeker home page jobseeker can also do keyword search and advance search and find filtered job list.

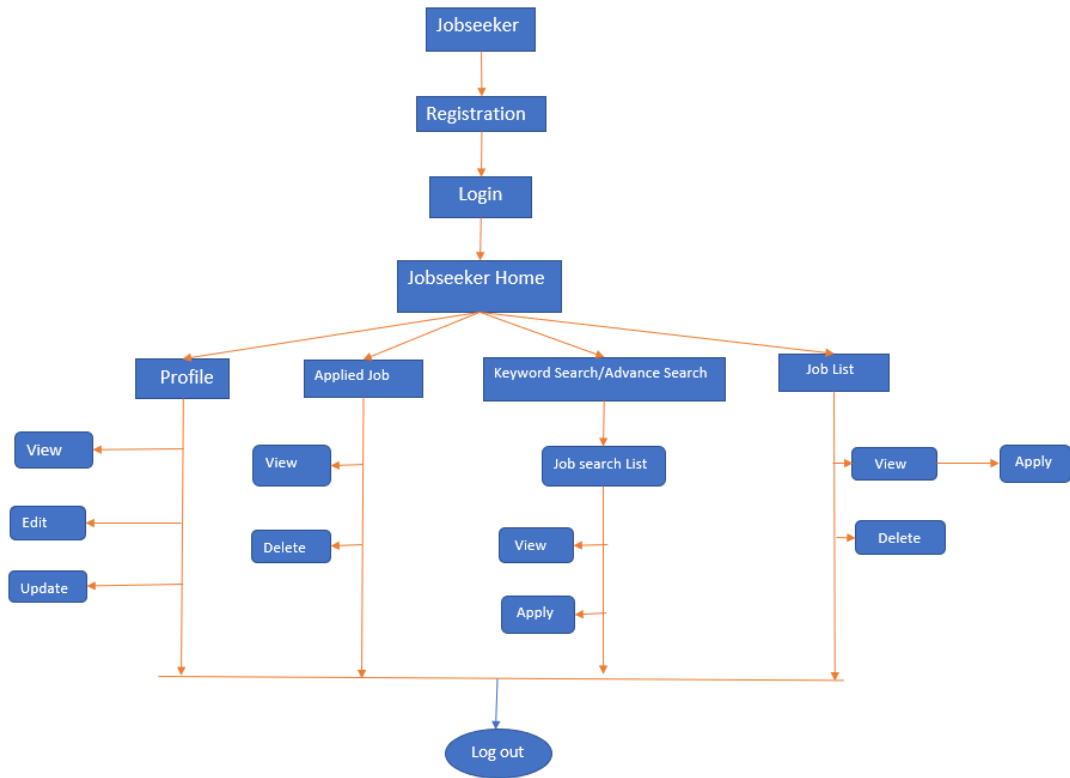


Figure 8: Jobseeker Workflow

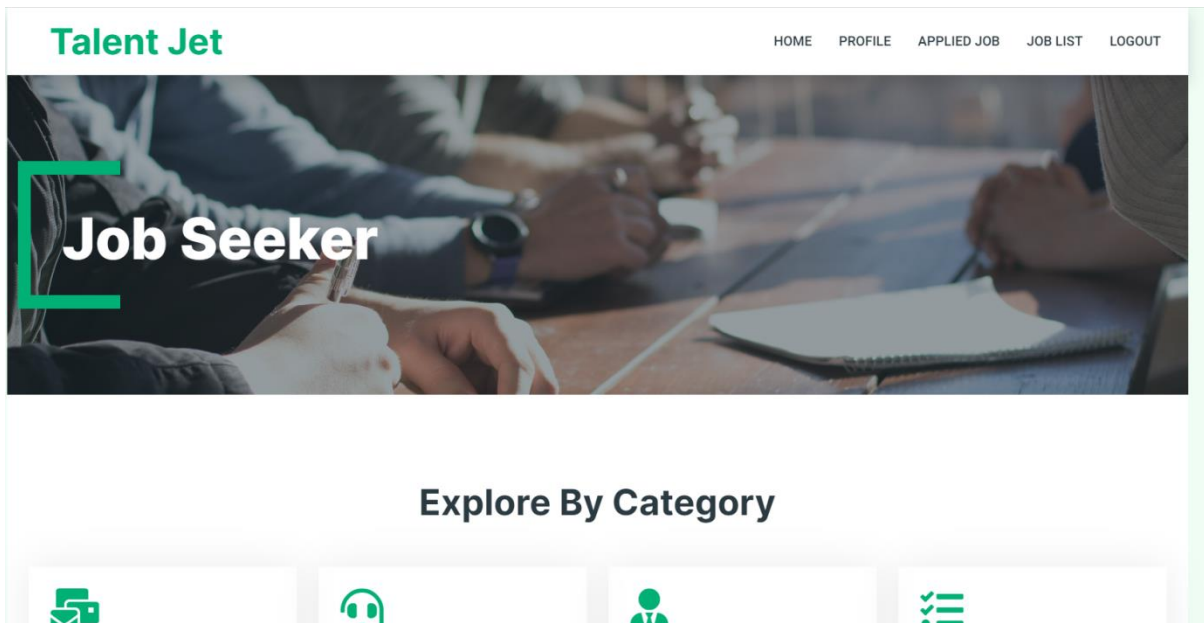




Figure 9: Jobseeker Homepage



Job List

Jobs	Location	Type	Last date	Company	View	Delete
Java Developer	New York	Full Time	2022-11-30	microsoft1@gmail.com	View	Delete
PHP developer	Canada	Freelancer	2022-12-02	oracle@gmail.com	View	Delete

Figure 10: Job List



Andy

📍 Mumbai ✉ andy@gmail.com

Introduction

Experienced python developer.

[Edit Profile](#)

Current Job detail

- Skills : Python
- Job Location : Mumbai
- Total Experience : 6
- Salary: 5

Figure 11: Profile

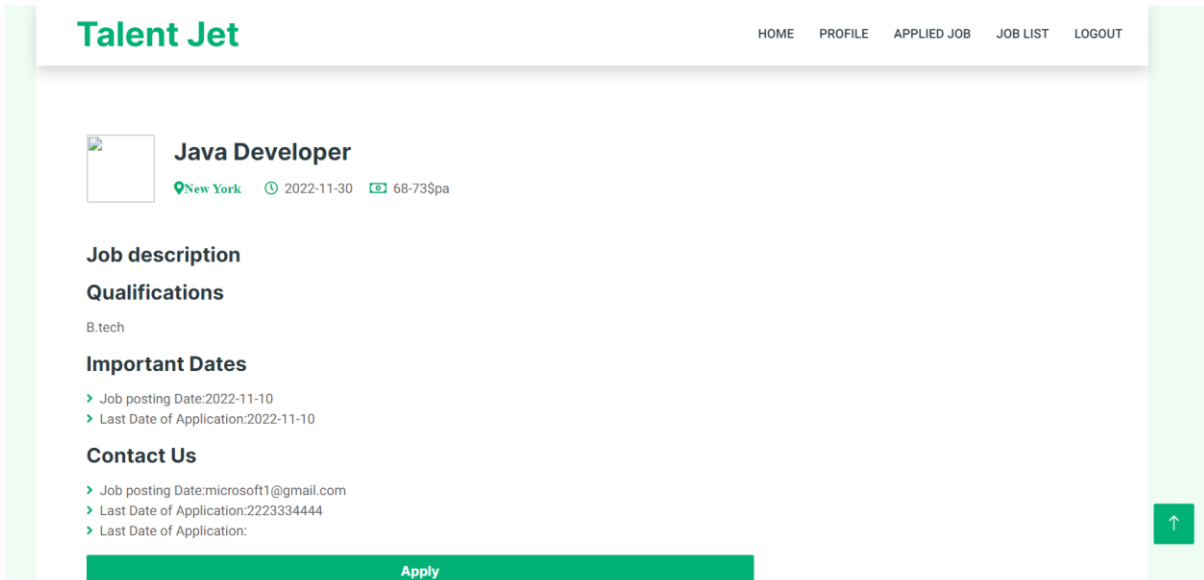


Figure 12: Job Detail

4. Workflow of Recruiter

Recruiters are the user who visit the website to find the talent. Recruiters are also registered user so to avail the functionality of recruiter user have registered and login as recruiter.

Following are the available features of recruiter:

- After registration and login using email id and password on login page, recruiter will land on the recruiter home page. In home page recruiter can navigate to different pages through navigation menu.
- In create job page recruiter can be able to post a job. Here recruiter will get form, recruiter must fill that form and post it to create a job.
- Job list page give all the detail of jobs which recruiter have created. Recruiter can manipulate the job details here. Recruiter can view, delete and edit the existing jobs.
- In applicant list page recruiter can see the list of jobseekers who have applied for the jobs.

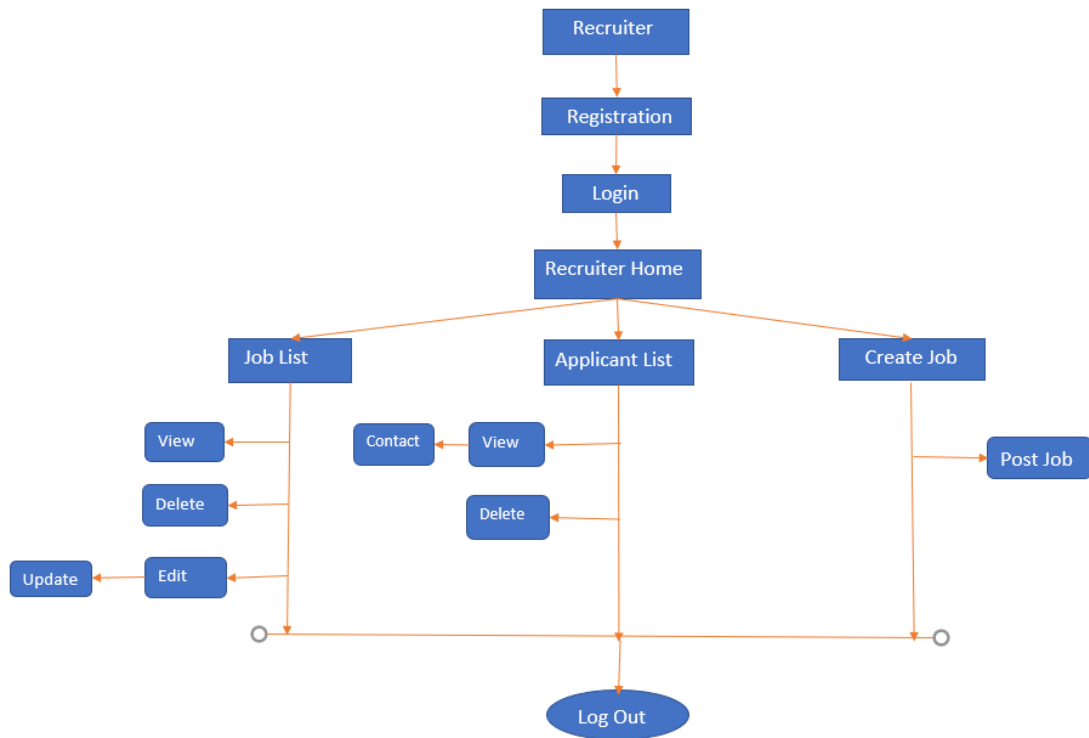


Figure 13: Recruiter Workflow

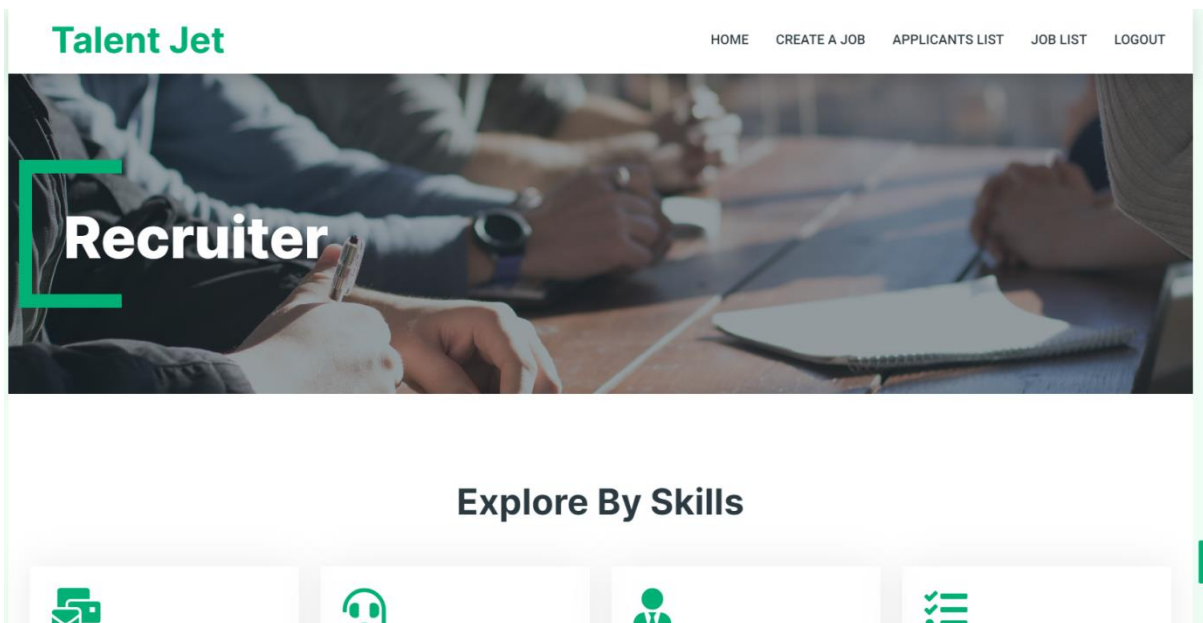


Figure 14: Recruiter Home Page

Post A Job


Job Title	<input type="text"/>
Job Type	<input type="text"/>
Official Email address	<input type="text" value="example@email.com"/>
Contact number	<input type="text" value="Enter Contact Number"/>
Registered Office Address	<input type="text" value="Enter your Address"/>
Job Posting Date	<input type="text" value="dd-mm-yyyy"/> 

Figure 15: Create Job

Recruiter

Job List



Jobs	Location	Type	Posting date	Edit	View	Delete
Java Developer	New York	Full Time	2022-11-10	Update	View	
PHP developer	Canada	Freelancer	2022-11-27	Update	View	

Figure 16: Job list

Recruiter

Edit Posted Job

Job Id	<input type="text" value="3"/>
Job Title	<input type="text" value="Java Developer"/>
Job Type	<input type="text" value="Full Time"/>
Official Email address	<input type="text" value="microsoft1@gmail.com"/>
Contact number	<input type="text" value="2223334444"/>
Registered Office Address	<input type="text" value="Enter your Address"/>

Figure 17: Edit Job

5 *Internal/external Interface Impacts and Specification*

This project need database to store data and manipulate the data. So, we are using MySQL relational database management system to create data base. And to perform basic CRUD operation we are using java programing. In backend code we have created four packages: controller, entities, repository, and services.

In entities class we defined all the tables needed for the project and decided various constraint here. So, for project we created users' entity class, jobseeker entity class, recruiter entity class, jobs entity class, and contact us entity class. Here we mapped the table. Entity class is linked with the repository class so for each entity class repository class is created. Repository class help in storing and accessing data in the database. Each entity class contain service class where methods for creating, retrieving, updating and deleting data is defined. This service class is linked to the controller class where the APIs is created, and path is mapped to connect with the front-end.

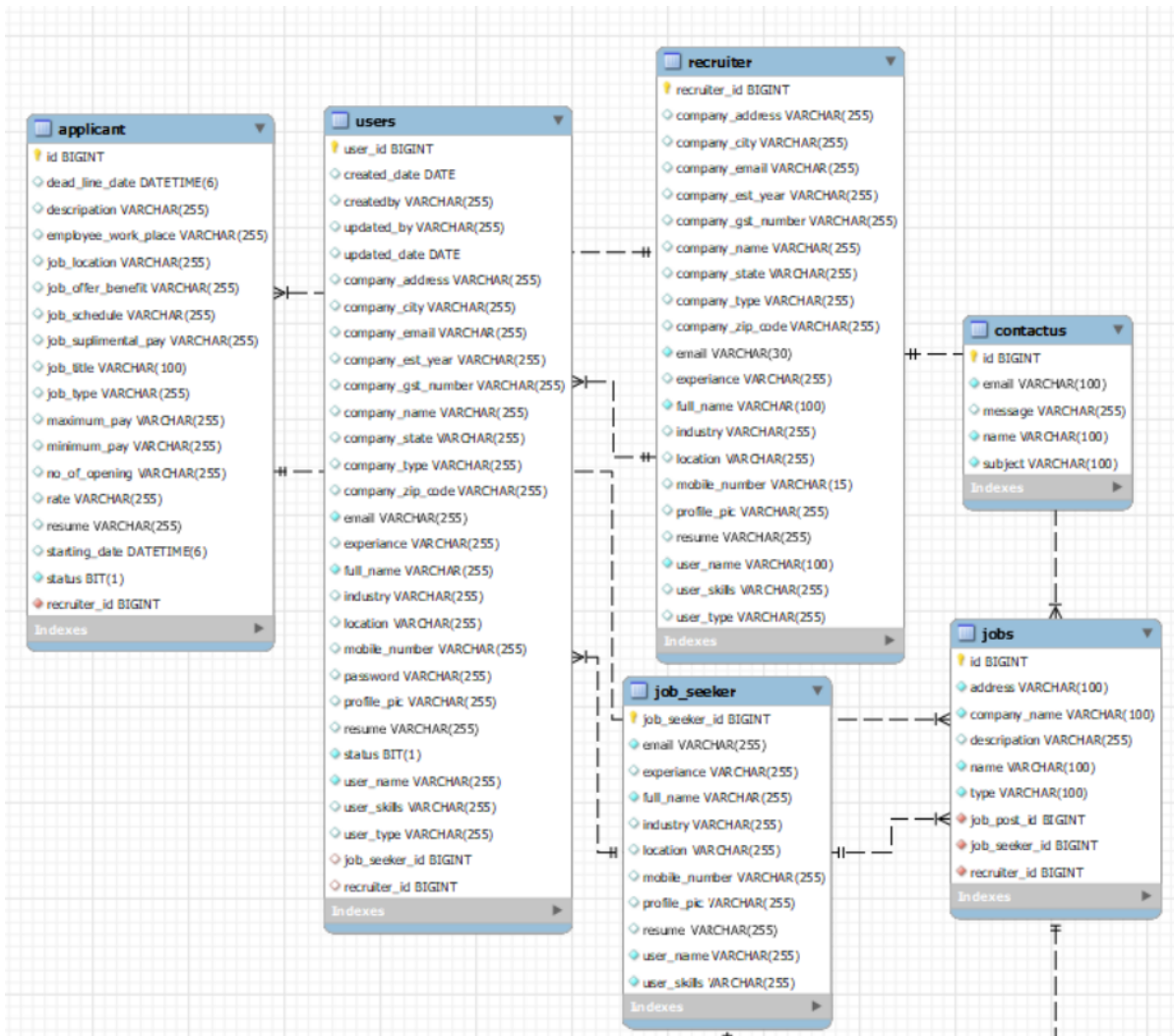


Figure 18: Database EER diagram

6 Design Units Impacts

Talent jet application is created to do specific activities so that the end users' needs is satisfied. Registration and login feature we have added.

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

With the registration and login feature only authenticated user can access the features of the website.

6.1.2 Impacts

The design unit will impact the workflow of jobseeker, recruiter and admin. Without authentication no user will be avail the feature of user's home page. Update of user information will also be impacted.

6.1.3 Requirements

All the three users will require to provide email id and password for authentication process. Login and registration are required to access the home page of the users. We have audited the internal and external interface so that it provides authentication.

7 Acknowledgements

We would like to thank "Dr. Dae Wook Kim", Assistant Professor of the Division of Science, Mathematics, and Technology at Governors State University, for his contributions to the completion of this project. It was a great learning experience. I want to

take this golden opportunity to convey my appreciation to all the group members for their cooperation and input, without whom the project would not have been successful.

I would like to take this opportunity to express my gratitude to all my group members Madhav, Siddharth, Sahithi, and Vamsi. The project would not have been successful without their cooperation and inputs.

8 References

- [1] Mansourvar M, Yasin NB. Development of a job web portal to improve education quality. International Journal of Computer Theory and Engineering. 2014 Feb 1;6(1):43. Available: https://www.researchgate.net/profile/Norizan-Mohd-Yasin/publication/269838390_Development_of_a_Job_Web_Portal_to_Improve_Education_Quality/links/555c1ed608ae8f66f3ade3d1/Development-of-a-Job-Web-Portal-to-Improve-Education-Quality.pdf
- [2] Goodman, Danny. *Dynamic HTML: The definitive reference: A comprehensive resource for HTML, CSS, DOM & JavaScript.* " O'Reilly Media, Inc.", pp. 3-20,2002, Available: [URL](#)
- [3] Goodman, Danny. *Dynamic HTML: The definitive reference: A comprehensive resource for HTML, CSS, DOM & JavaScript.* " O'Reilly Media, Inc.", pp. 42-69,2002, Available: [URL](#)
- [4] VMware, Spring Framework, Version 5.3.23, 2022 Available: <https://spring.io/projects/spring-framework>
- [5] VisualStudioCode.October2022. (Version1.73). Microsoft, Available: <HTTPS://visualstudio.microsoft.com/downloads/>
- [6] Eclipse IDE 2022-09 IBM, Available: <https://www.eclipse.org/downloads/>
- [7] MySQL.2022 (Version8.0.31) Oracle, Available: <https://dev.mysql.com/downloads/installer/>

9 Appendices

```
44     </div>
45     <!-- Spinner End -->
46
47
48     <!-- Navbar Start -->
49     <nav class="navbar navbar-expand-lg bg-white navbar-light shadow sticky-top p-0">
50         <a href="index.html" class="navbar-brand d-flex align-items-center text-center py-0 px-4 px-lg-5">
51             <h1 class="m-0 text-primary">Talent Jet</h1>
52         </a>
53         <button type="button" class="navbar-toggler me-4" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
54             <span class="navbar-toggler-icon"></span>
55         </button>
56         <div class="collapse navbar-collapse" id="navbarCollapse">
57             <div class="navbar-nav ms-auto p-4 p-lg-0">
58                 <a href="index.html" class="nav-item nav-link active">Home</a>
59                 <a href="about.html" class="nav-item nav-link">About</a>
60                 <a href="blog.html" class="nav-item nav-link">Blog</a>
61                 <a href="contact.html" class="nav-item nav-link">Contact Us</a>
62                 <div class="nav-item dropdown">
63                     <a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Users</a>
64                     <div class="dropdown-menu rounded-0 m-0">
65                         <a href="adminLogin.html" class="dropdown-item">Admin</a>
66                         <a href="jobseekerLogin.html" class="dropdown-item">Job Seeker</a>
67                         <a href="recruiter-login.html" class="dropdown-item">Recruiter</a>
68                     </div>
69                 </div>
70             </div>
71         </div>
72         <a href="recruiter-login.html" class="btn btn-primary rounded-0 py-4 px-lg-5 d-none d-lg-block">Post A Job<i clas
73     </div>
74 </nav>
75 <!-- Navbar End -->
76
```

Figure 19: Index Page

```

<!-- Footer Start -->
<div class="container-fluid bg-dark text-white-50 footer pt-5 mt-5 wow fadeIn" data-wow-delay="0.1s">
  <div class="container py-5">
    <div class="row g-5">
      <div class="col-lg-3 col-md-6">
        <h5 class="text-white mb-4">Talent Jet </h5>
        <p>Talent Jet is easy to search job portal. The website will provide efficient search for job opening and talent
      </div>
      <div class="col-lg-3 col-md-6">
        <h5 class="text-white mb-4">Quick Links</h5>
        <a class="btn btn-link text-white-50" href="about.html">About Us</a>
        <a class="btn btn-link text-white-50" href="contact.html">Contact Us</a>
        <a class="btn btn-link text-white-50" href="blog.html">Blog</a>
        <a class="btn btn-link text-white-50" href="privacyPolicy.html">Privacy Policy</a>
        <a class="btn btn-link text-white-50" href="">Terms & Condition</a>
      </div>
      <div class="col-lg-3 col-md-6">
        <h5 class="text-white mb-4">Contact</h5>
        <p class="mb-2"><i class="fa fa-map-marker-alt me-3"></i>2851 S King Drive, Apt-1302 Chicago,illinois,60616</p>
        <p class="mb-2"><i class="fa fa-phone-alt me-3"></i>+248-861-4183</p>
        <p class="mb-2"><i class="fa fa-envelope me-3"></i>info@gmail.com</p>
        <div class="d-flex pt-2">
          <a class="btn btn-outline-light btn-social" href=""><i class="fab fa-twitter"></i></a>
          <a class="btn btn-outline-light btn-social" href=""><i class="fab fa-facebook-f"></i></a>
          <a class="btn btn-outline-light btn-social" href=""><i class="fab fa-youtube"></i></a>
          <a class="btn btn-outline-light btn-social" href=""><i class="fab fa-linkedin-in"></i></a>
        </div>
      </div>
      <div class="col-lg-3 col-md-6">
        <h5 class="text-white mb-4">Newsletter</h5>
        <p>SignUp for geting Newsletter in your email.</p>
        <div class="position-relative mx-auto" style="max-width: 400px;">
          <input class="form-control bg-transparent w-100 py-3 ps-4 pe-5" type="text" placeholder="Your email">
          <button type="button" class="btn btn-primary py-2 position-absolute top-0 end-0 mt-2 me-2">SignUp</button>
        </div>
      </div>
    </div>
  </div>
</div>

```

Figure 20: Footer

```

1 <!DOCTYPE html>
2 <head>
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta charset="utf-8">
5   <title>Login / Sign Up Form</title>
6   <link rel="shortcut icon" href="/assets/favicon.ico">
7   <link rel="stylesheet" href="css/adminLogin.css">
8 </head>
9 <body>
10  <div class="container">
11    <form class="adminLoginForm" id="login">
12      <h1 class="form_title">Admin Login</h1>
13      <div class="form_message form_message--error"></div>
14      <div class="form_input-group">
15        <input type="text" class="form_input" autofocus placeholder="Username or email" name="username" id="username" va
16        <div class="form_input-error-message"></div>
17      </div>
18      <div class="form_input-group">
19        <input type="password" class="form_input" autofocus placeholder="Password" name="password" id="password" value=""
20        <div class="form_input-error-message"></div>
21      </div>
22      <input type="button" class="form_input" id="validatebtn" onclick="validate()" value="Continue" />
23      <p class="form_text">
24        <a href="#" class="form_link"></a>
25      </p>
26      <p class="form_text">
27        <a class="form_link" href="#" id="linkCreateAccount"></a>
28      </p>
29    </form>
30  </div>
31  <script src="js/adminLogin.js"></script>
32 </body>

```

Figure 21: Admin login

```

js > JS admin-jobseeker-view.js > ...
Complexity is 3 Everything is cool!
1  $( document ).ready(function() {
2      let jobseekerID = sessionStorage.getItem("jobseekerID");
3      var BASEURL="http://localhost:8081/";
4      $.ajax({
5          url: BASEURL+'jobseeker/getById/'+jobseekerID, // Performed AJAX login
6          type: "GET",
7          dataType: 'json',
8          async:false,
9          processData: false,
10         headers: {
11             "content-type": "application/json",
12         },
13         success: function(response){
14             console.log(JSON.stringify(response));
15             var sandyHtml = get_html_for_jobseeker_detail_admin(response);
16             $('#adminJobSeekerDetail').empty();
17             $('#adminJobSeekerDetail').append(sandyHtml);
18         },
19         error: function(jqXHR, textStatus, errorMessage){
20             alert(errorMessage);
21             $('#loginStatus').append('<label style="color:red;">UserName dddd Password are Incorrect! </label>');
22         }
23     });
24 });
25
26 function get_html_for_jobseeker_detail_admin(response)
27 {
28     var txt="";
29
30     txt = `<div class="d-flex align-items-center mb-5">
31         <div class="text-start ps-4">
32             <h3 class="mb-3">+response.fullName+`</h3>
33             <span class="text-truncate me-3"><i class="fa fa-map-marker-alt text-primary me-2"></i>+response.currentAddr
34             <span class="text-truncate me-3"><i class="far fa-clock text-primary me-2"></i>+response.email+`</span>
35             <span class="text-truncate me-0"><i class="far fa-money-bill-alt text-primary me-2"></i>+response.dateOfBirth
36         </div>

```

Figure 22: admin-jobseeker-view.js


```

js > JS jobseekerLogin.js > on('click', '.loginJobSeeker') callback > success
1
2
Complexity is 6 It's time to do something...
3 $(document).on('click', '.loginJobSeeker', function(e){
4     "use strict";
5     var BASEURL = 'http://localhost:8081/';
6     const form = document.querySelector("#loginJobSeeker_form");
7     var mydata= new FormData(form);
8     const formData = Object.fromEntries(mydata.entries());
9
10
11     var jobSeekerJSON = JSON.stringify(formData);
12     alert(jobSeekerJSON);
13     console.log(jobSeekerJSON);
14
15     if(formData.userName==" " || formData.password==""){
16         $("#loginStatus").append('<label style="color:red;">Please Fill Username and Password! </label>');
17     }else{
18         $.ajax({
19             url: BASEURL+'jobseeker/login', // Performed AJAX login
20             type: "POST",
21             dataType: 'json',
22             async:false,
23             processData: false,
24             headers: {
25                 'userName':formData.userName,
26                 'password':formData.password,
27                 "content-type": "application/json",
28             },
29             data: jobSeekerJSON,
30             success: function(response){
31                 sessionStorage.clear();
32                 sessionStorage.setItem("jobseekerID", response.jobSeekerId);
33                 alert('Successfully logged in!');
34                 window.location.replace("http://127.0.0.1:5503/jobseeker-homepage.html");
35             },
36             error: function(jqXHR, textStatus, errorMessage){
37                 alert(errorMessage);
38             }
39         });
40     }
41 }

```

Figure 23: jobseekerlogin.js

```

Complexity is 3 Everything is cool!
$(document).on('click', '.job_seekers_delete', function(e){
    "use strict";
    if (confirm('Are You sure ?'))
    {
        var userId = $(this).attr("id");
        var myrequest = $.ajax({
            url:'http://localhost:8081/jobs/deletebyId/'+userId,
            type: "DELETE",
            dataType: 'json',
            async:false,
            processData: false,
            headers: {
                "Content-Type": "application/json",
                Accept: "application/json",
            },
        });
        alert('Thanks for Confirming.....! Your Data has been Deleted successfully!');
        window.location.replace("http://127.0.0.1:5503/recruiter-joblist.html");
    }
    else{
        alert('Thanks For Cancel.');
```

Figure 24: Jobseeker delete job.js

```

1 package com.webapp.TalentJet.controller;
2
3 import java.util.List;
20
21 // This is controller class for the Jobs.
22
23 @CrossOrigin(origins = "http://127.0.0.1:5503", maxAge = 3600)
24 @RestController
25 @RequestMapping("/jobs")
26 public class JobsController {
27
28
29     @Autowired
30     JobsService jobsService;
31
32 // Created end point for creating job.
33 @PostMapping("/create")
34 public Jobs createJobs(@RequestBody Jobs jobs) {
35
36     return jobsService.createJobs(jobs);
37
38 }
39
40 // End point for updating jobs.
41 @PutMapping("/update/{jobsId}")
42 public Jobs updateProject(@PathVariable long jobsId, @RequestBody Jobs jobs) {
43     return jobsService.updateJobs(jobsId, jobs);
44 }
45
46 // End point to get jobs by Id.
47 @GetMapping("/getById/{jobsId}")
48 public Optional<Jobs> getJobsById(@PathVariable long jobsId) {
49
50     return jobsService.getJobsById(jobsId);
51 }
52
53 // End point to get all the jobs.

```

Figure 25: Jobs Controller

```

1 package com.webapp.TalentJet.repositories;
2
3 import java.util.List;
10
11 @Repository
12 public interface JobsRepository extends JpaRepository<Jobs,Long> {
13
14
15     List<Jobs> findByJobType(String jobtype);
16
17     List<Jobs> findByLocation(String location);
18     List<Jobs> findByJobTypeAndLocation(String jobType,String location);
19

```

Figure 26: Jobs Repository

```

1 package com.webapp.TalentJet.services;
2
3 import java.util.List;
12
13 @Service
14 public class JobseekerServiceImpl implements JobseekerService {
15
16
17     /*ServiceImpl class implement the various methods Of CRUD operation and
18     perform the logic on the data stored in the entity class.*/
19
20     @Autowired
21     JobseekerRepository jobseekerRepository;
22
23     //In this create method jobseeker will be created and saved in the repository class.
24     @Override
25     public Jobseeker createJobseeker(Jobseeker jobseeker) {
26         // TODO Auto-generated method stub
27         return jobseekerRepository.save(jobseeker);
28     }
29
30
31     @Override
32     public Optional<Jobseeker> getJobseekerById(long jobseekerId) {
33         // TODO Auto-generated method stub
34         return jobseekerRepository.findById(jobseekerId);
35     }
36
37     @Override
38     public List<Jobseeker> getAllJobseekerList() {
39         // TODO Auto-generated method stub
40         return jobseekerRepository.findAll();
41     }
42
43     @Override
44     public List<Jobseeker> getallJobseekerBySkillSet(String skillSet) {

```

Figure 27: Jobs ServiceImpl class

```

1 # This is the application.properties file which contain the database details.
2 # here we have created database of name "talent"
3
4 server.port=8081
5 spring.datasource.url=jdbc:mysql://localhost:3306/talent
6 spring.datasource.username=root
7 spring.datasource.password=****
8 spring.jpa.hibernate.ddl-auto=update
9
10
11
12

```

Figure 28: Database detail