

SKILL AND JOB RECOMMENDATION SYSTEM USING FoDRA
ALGORITHM

A PROJECT REPORT

Submitted by

SANJEEV P N **211419104234**

SANTHOSH J **211219104235**

SARATH S **211419104238**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



PANIMALAR ENGINEERING COLLEGE
(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE
(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report “**SKILL AND JOB RECOMMENDATION SYSTEM USING FoDRA ALGORITHM**” is the bonafide work of “**SANJEEV P N (211419104234), SANTHOSH J (211419104235), SARATH S (211419104238)**” who carried out the project work under my supervision.

SIGNATURE

Dr. L. JABASHEELA, M.E., Ph.D.,

HEAD OF THE DEPARTMENT,

DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE,

NAZARATHPETTAI,

POONAMALLEE,

CHENNAI-600 123.

SIGNATURE

Mr. M. KRISHNAMOORTHY, M.E., M.B.A., (Ph.D.),

ASSOCIATE PROFESSOR,

DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE

NAZARATHPETTAI,

POONAMALLEE,

CHENNAI-600 123.

Certified that the above mentioned students were examined in the End Semester project viva-voice held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **SANJEEV P N (211419104234)**, **SANTHOSH J (211419104235)**,
SARATH S (211419104238) hereby declare that this project report titled “**SKILL
AND JOB RECOMMENDATION SYSTEM USING FoDRA
ALGORITHM**”, under the guidance of **Mr. M. KRISHNAMOORTHY, M.E.,
M.B.A., (Ph.D),** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

- 1)SANJEEV P N
- 2)SANTHOSH J
- 3)SARATH S

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.**, for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHI KUMAR, M.E., Ph.D.,** and **Dr.SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.**, for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.**, who facilitated us in completing the project.

We thank the Head of the Department, **Dr. L.JABASHEELA, M.E., Ph.D.**, for the support extended throughout the project.

We would like to thank my project coordinator **Mr M.MOHAN, M.E., (Ph.D.)**, and **Project Guide Mr. M. KRISHNAMOORTHY, M.E., M.B.A., (Ph.D.)**, and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

SANJEEV P N

SANTHOSH J

SARATH S

ABSTRACT

In an increasing number of cash-rich, stable, and promising technical companies on the web which are in more demand right now, many job seekers want to apply and work for these companies. They tend to miss out on those opportunities because there is an ocean of existing recommendation systems that list millions of jobs which are generally not relevant at all to the users. There is an abundance of choices and not much streamlining. On the basis of the actual skills and qualifications of an individual, job seekers often end up receiving the inappropriate and irrelevant jobs for them. Therefore, this system approaches the idea from a data point of view, emphasizing more on the quality of the data than the quantity.

An employment portal is an application that facilitates the search for employment and ranges from small to large, generalized sites to specialized working groups for work categories such as Engineering, Law, Insurance, Social work, Teaching, and developing Mobile applications. Job seekers can usually submit their resumes to the employers and recruiters for review, whereas employers and recruiters can post job postings and search for employees in the job portal. The Job Portal System will recommend accurate jobs which minimizes the time. The main aim of this system is to allow communication between the job seeker and employer.

Job recommendation system plays a vital role in connecting job seeker and employers. So, we use FoDRA which is an efficient and effective algorithm compared to other existing recommendation algorithms. This system reduces the stress and frustration of the Job seeker in finding a job and recommends the accurate job based on the qualification and other related parameters.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	V
	LIST OF TABLES	VI
	LIST OF FIGURES	XI
1	INTRODUCTION	
	1.1 INTRODUCTION	1
	1.2 OBJECTIVE	1
2	LITERATURE SURVEY	
	2.1 LITERATURE SURVEY	2
3	SYSTEM ANALYSIS	
	3.1 EXISTING SYSTEM	4
	3.2 PROPOSED SYSTEM	4
4	REQUIREMENT AND SPECIFICATION	
	4.1 SOFTWARE SPECIFICATION	5
	4.2 HARDWARE SPECIFICATION	5

5	SYSTEM DESIGN	
5.1	USER MODULE	6
5.2	ADMIN MODULE	7
5.3	RECRUITER MODULE	8
6	SOFTWARE DESCRIPTION	
6.1	JAVA	10
6.1.1	USES OF JAVA	11
6.2	JSP	11
6.2.1	USES OF JSP	13
6.3	HTML	13
6.3.1	USES OF HTML	14
6.4	CSS	14
6.4.1	USES OF CSS	15
6.5	SQL	15
6.5.1	USES OF SQL	17
6.6	NETBEANS	17
6.6.1	USES OF NETBEANS	18
7	DIAGRAMS	
7.1	USECASE DIAGRAM	19

7.1.1 DESCERIPTION	19
7.2 E-R DIAGRAM	20
7.2.1 DESCRIPTION	20
7.3 SEQUENCE DIAGRAM	21
7.3.1 DESCRIPTION	21
7.4 ACTIVITY DIAGRAM	22
7.4.1 DESCRIPTION	22
7.5 DATABASE DIAGRAM	23
7.5.1 DESCRIPTION	23
7.6 INPUT AND OUTPUT DIAGRAM	24
7.6.1 DESCRIPTION	24

8 SYSTEM TESTING

8.1 WHITE BOX TESTING	25
8.2 BLACK BOX TESTING	27
8.3 TEST PROCEDURES	27
8.4 TEST CASES	27
8.5 TEST DESIGN TECHNIQUES	28
8.6 PERFORMANCE TESTING	28
8.7 UNIT TESTING	28
8.8 SOFTWARE TESTING	30

8.9 FUNCTIONAL TESTING	31
8.10 TESTING TYPES	31
8.10.1 LOAD TESTING	31
8.10.2 STRESS TESTING	32
8.10.3 SOAK TESTING	32
8.10.4 SPIKE TESTING	32
8.10.5 CONFIGURATION TESTING	33
8.11 ISOLATION TESTING	33
8.12 INTEGRATION TESTING	33
8.13 SYSTEM TESTING	34
8.14 STRUCTURE TESTING	34
8.15 OUTPUT TESTING	34
8.16 USER ACCEPTANCE TESTING	35
9 CODING	
9.1 JSP CODE	36
9.2 JAVA CODE	38
9.3 HTML CODE	40
10 SCREENSHOTS	
9.1 WELCOME PAGE	42
9.2 USER MODULE	43
9.3 ADMIN MODULE	46
9.4 RECRUITER MODULE	48

11 CONCLUSION

11.1 CONCLUSION 50

11.2 ADVANCEMENT 50

REFERENCES 51

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
Fig 5.1.1	USER MODULE WORKFLOW	6
Fig 5.2.1	ADMIN MODULE WORKFLOW	7
Fig5.3.1	RECRUITER MODULE WORKFLOW	8
Fig.6.1.1	FRAMEWORK OF JAVA	11
Fig.6.2.2	FRAMEWORK OF JSP	13
Fig.6.5.1	FRAMEWORK OF SQL	16
Fig.6.6.1	ARCHITECTURE OF NETBEAN	18
Fig.7.1.1	USECASE DIAGRAM	19
Fig.7.2.1	E-R DIAGRAM	20
Fig.7.3.1	SEQUENCE DIAGRAM	21
Fig.7.4.1	ACTIVITY DIAGRAM	22
Fig.7.5.1	DATABASE DIAGRAM	23
Fig 7.6.1	INPUT AND OUTPUT DIAGRAM	24
Fig 10.1.1	WELCOME PAGE	42
Fig 10.1.2	USER PAGE	42
Fig 10.2.1	USER REGISTER PAGE	43
Fig 10.2.2	USER LOGIN PAGE	43
Fig 10.2.3	HOME PAGE	44

FIGURE NO.	FIGURE NAME	PAGE NO.
Fig 10.2.4	TOP JOBS	44
Fig 10.2.5	SEARCH JOB PAGE	45
Fig 10.2.6	RESULT PAGE	45
Fig.10.3.1	ADMIN LOGIN PAGE	46
Fig.10.3.2	USER PROFILE VIEWING PAGE	46
Fig.10.3.3	JOB VIEWING PAGE	47
Fig.10.3.4	JOB DETAIL PAGE	47
Fig.10.4.1	RECRUITER PAGE	48
Fig.10.4.2	RECRUITER REGISTRATION	48
Fig.10.4.3	RECRUITER LOGIN	49
Fig.10.4.4	JOB POSTING PAGE	49

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

To develop an enhanced web application, using web services for both online job and candidate recommendation system. By using Four Dimensions Recommendation Algorithm (FoDRA) and Text field filtering the recommendation of jobs and candidates will be classified. Three tier architecture designs have been implemented for efficient data retrieval and data transfer. They are Job seeker interface, Candidate recruitment interface and Admin will be the architecture taken for developing this application.

The primary architecture will be the job seeker interface, in followed with candidate recruitment interface and Admin will be interconnected. The professional social recommender will work as a third-party agent and the agent will retrieves all the recommended job and candidate profiles. A panel will be designed for displaying the recommended job and candidate details. All the displayed jobs will be more relevant to the user's profile.

This saves a lot of potential time and money both, on the industrial as well as the job seeker's side. Moreover, as the candidate gets a fair chance to prove his talent in the real world in a lot more efficient fashion. The basic agenda of every algorithm used in today's world, be it a traditional algorithm or a hybrid algorithm, is to provide a suitable job that the user actually seeks a right job. So this project uses FoDRA to satisfy the above criteria.

1.2 OBJECTIVE

To find a perfect job for the Job seeker based on Job Role, Qualification, Location and Salary and to provide a perfect Job seeker with right skill and ability to the recruiter.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

- **AUTHOR :** Maria Jose Gomez Torres, Javier Rodriguez Santer and Javier Gil Flores.
- **TITLE :** Job search strategies of individuals at risk of poverty and social exclusion
- **YEAR :** 2019

This review distills available empirical research about the process and experience of a job. Job search varies according to several dimensions, including intensity, content and temporality/persistence. Our review examines how these dimensions employment are at risk for long-term social and economic exclusion.

- **AUTHOR :** Connie R. Wanberg, Abdiftah A. Ali and Borbala Csillag
- **TITLE :** Job search strategies of individuals at risk of poverty and social exclusion
- **DATE :** JANUARY 11TH 2019

According to previous literature, the economic poverty that is characteristic of this group may accompany poverty competency. In particular, the working poor resort to informal and poorly developed job-search strategies. This study addresses the job search methods used by people at risk of poverty and social exclusion. It provides evidence on the subject and serves as a basis for the adaptation of socio-labor intermediation programmers to this group.

- **AUTHOR :** Punithavathi.
- **TITLE :** Online Job Online Job and Candidate Recommendation System.
- **YEAR :** 2019

To develop an enhanced web application, using web services for both online job and candidate recommendation system. By using Professional Social

Recommender (PSR) and Text field filtering the recommendation of jobs and candidates will be classified. Three tier architecture designs have been implemented for efficient data retrieval and data transfer. They are Job seeker interface; Candidate recruitment interface and Recommendation database will be the architecture taken for developing this application.

- **AUTHOR** : Shivraj Hulbatte, Amit Wabale, Suraj Patil and Nikhil Kumar Sathe.
- **TITLE** : Enhanced Job Recommendation System.
- **YEAR** : 2018

We address the problem of recommending suitable jobs to people who are seeking a new job. Dealing with the enormous amount of recruiting information on the Internet, a job seeker always spends hours to find useful ones. To reduce this laborious work, we design and implement a recommendation system for online job hunting.

- **AUTHOR** : Rahul Pradhan, Jyothi Varshney, Kartik Goyal and Lathesh Kumari
- **TITLE** : Job Recommendation System Using Content and Collaborative Based Filtering
- **YEAR** : 2021

Dealing with the big amount of information on the web, employment seeker constantly spends hours discover useful ones. We do all this process in an easy manner. Recommendation systems usually consist of exploiting relations among understood features and content that describes services and products (content-based filtering) or the overlap of comparable users who interacted with or rated the goal item (collaborative filtering). We reveal a comparison between content filtering and based that is collaborative.

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The currently existing system widely uses collaborative algorithm which produces all the jobs in one field to the users. Due to this the users may get different jobs that are not required to them. The different jobs that are being recommended to them may have variance in job location, salary and users skill set that the user expects. The user may see anyone of the attribute being sufficient and may accept the job but later may quit the job due to other two attributes not suitable for the user i.e. if the user is fine with the salary of the job and accepts it then due to its far location and not having the required ability to complete the task he may quit the job and vice versa. Thus most of the job recommendation websites focuses on giving a large number of jobs to the user rather than giving a right job to the users.

3.2 PROPOSED SYSTEM

In our proposed system we address the problem of recommending a large set of unwanted or not suitable jobs for the user. Our proposed system will recommend the right job to the user based on suitable job title, salary and users skill set. Thus FoDRA will be used to have these four information as its primitives and when these four abilities are given and searched the right job for the user will be recommended. This will also help the recruiter to recruit the best man with the right skill set to complete the job. This a large time saving and less frustration process comparing to the existing system of job recommendation.

CHAPTER-4

REQUIREMENT AND SPECIFICATION

4.1 SOFTWARE SPECIFICATION

- LANGUAGE : JAVA
- IDE : JAVA NETBEANS
- DATABASE : SQL
- FRONT END : HTML,CSS
- CONNECTIVITY : JSP,SQL
- BACK END : JAVA

4.2 HARDWARE REQUIREMENTS

- PROCESSOR : INTEL I3
- HARD DISK : MINIMUM 120 GB
- RAM : MINIMUM 4GB
- OS : WINDOWS 8 OR HIGHER

CHAPTER-5

SYSTEM DESIGN

5. SYSTEM DESIGN

Job Recommendation System helps in recommending the accurate jobs to the Users and also helps in minimizing the search time for the users.

5.1 USER MODULE

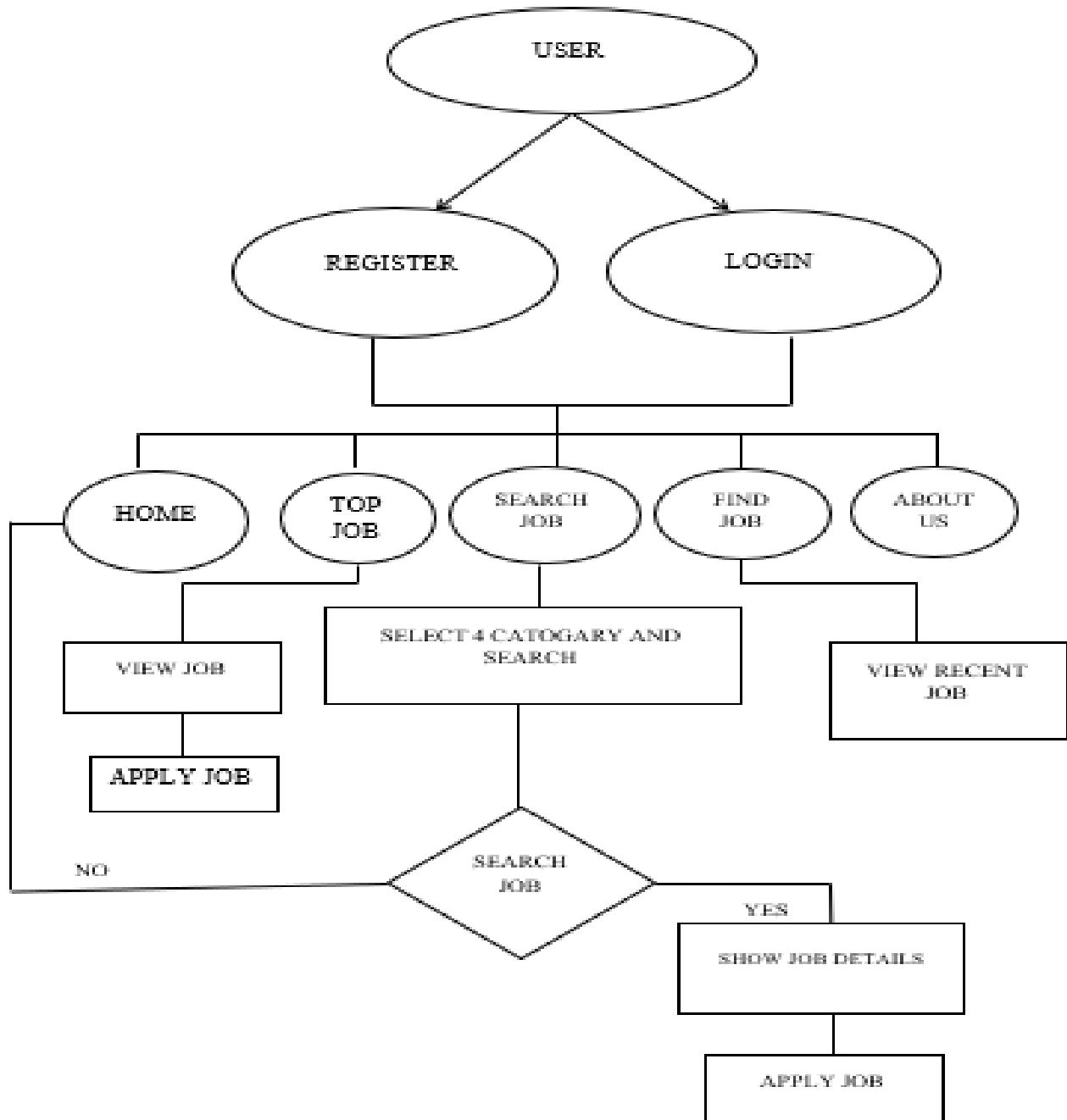


Fig 5.1.1 USER MODULE WORKFLOW

The Job Recommendation System starts with the User or Job seeker module where the job seeker have to register first to enter into the web application. After registration, the job seeker will witness the home page which describes about the process. The job seeker can also see other options such as Top Jobs, Search Job, Find Recent Job and about us. Here when the job seeker search for the job by selecting the four categories mentioned, the Algorithm will be used to recommend the accurate job to the job seeker or user. An effective addition to this module is that job seekers can see the recent jobs posted by the recruiters. The About Us option describes about the details of the developers of this web application. This page is common for all the 3 modules.

5.2 ADMIN MODULE

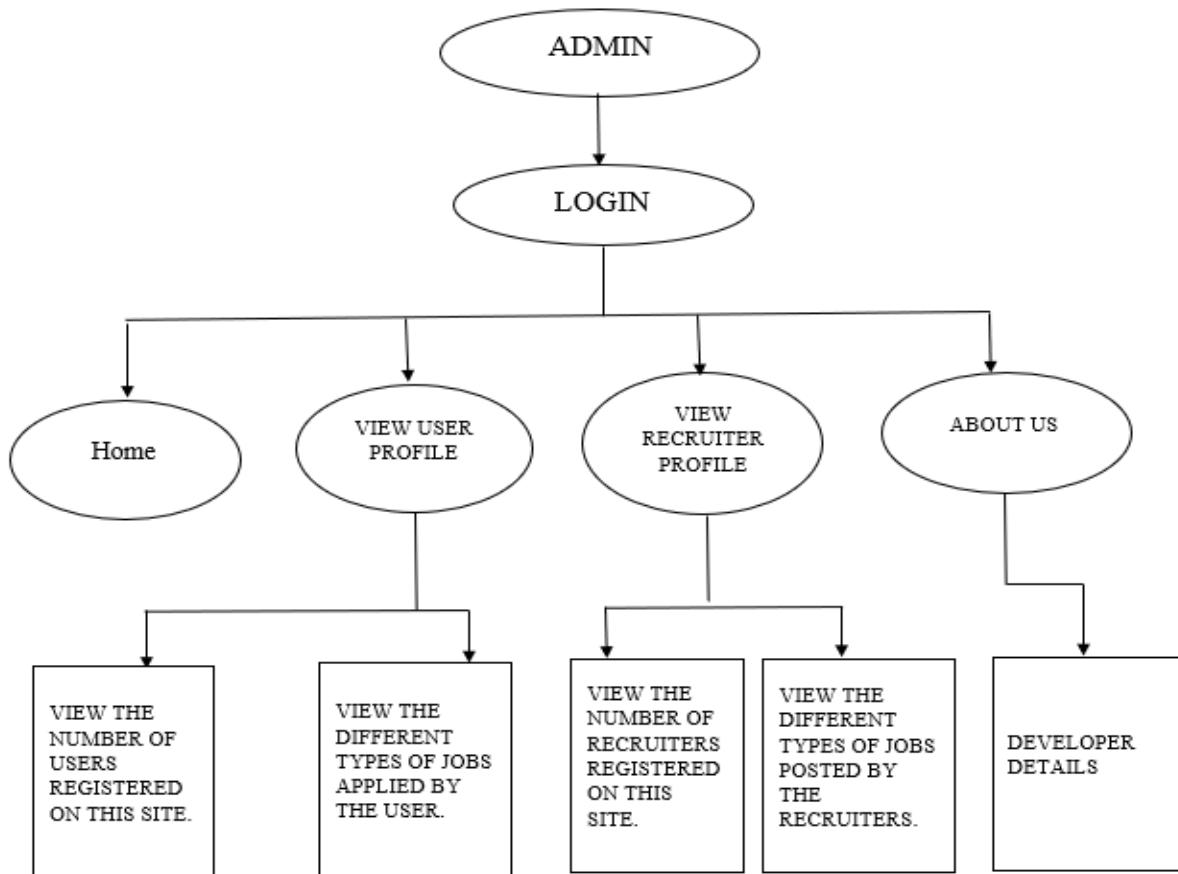


Fig 5.2.1 ADMIN MODULE WORKFLOW

The second module of the job recommendation system is the Admin. The admin module starts with a login page where the email address and password will be given to the admin as this module act as an interface between user module and recruiter module. After login, it will lead to the home page and also contains other options such as View user profile and View recruiter profile. In the user profile, admin can collect data about the number of users registered and jobs applied by the users. In the recruiter profile admin can collect data about the number of recruiters registered and jobs posted by the recruiters.

5.3 RECRUITER MODULE

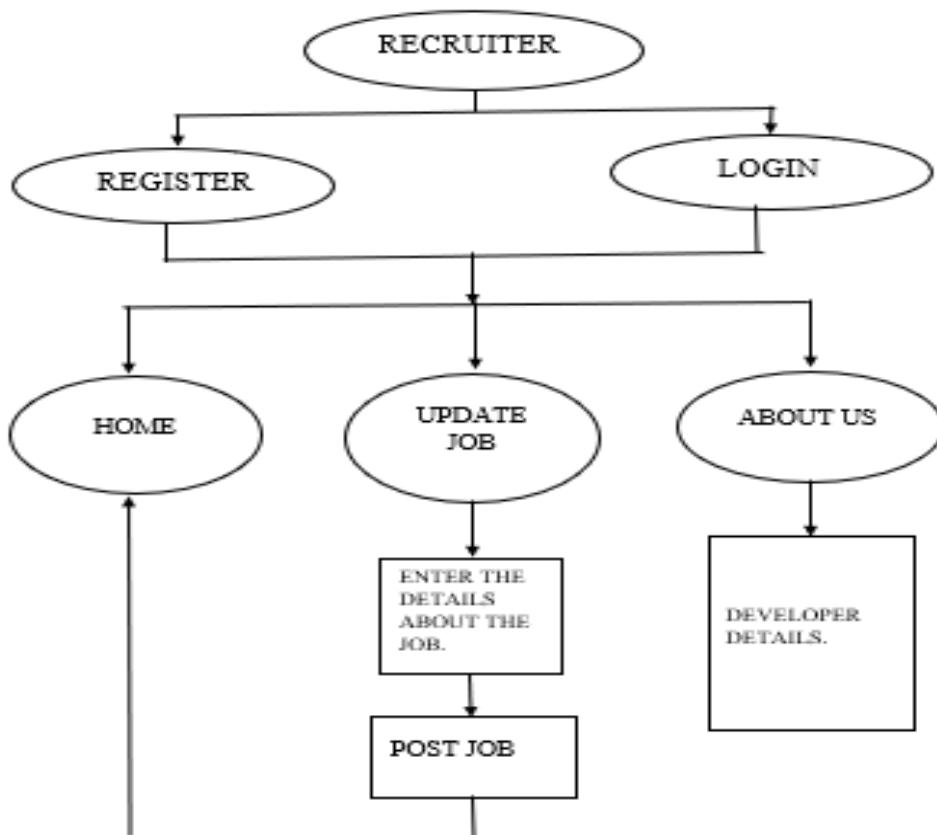


Fig 5.3.1 RECRUITER MODULE WORKFLOW

The final module of job recommendation system is the Recruiter. Here the recruiter has to register first or login if he/she is already a registered member in this site. After Registration/Login, the recruiter will see the home page. The main function of the recruiter in this website is to post the new jobs of his/her company which will be given to the recruiter as “Update Job”.

CHAPTER-6

SOFTWARE DESCRIPTION

6.1 JAVA

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore it is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc. Java Platform is a collection of programs that help programmers to develop and run Java programming applications efficiently. It includes an execution engine, a compiler, and a set of libraries in it. It is a set of computer software and specifications. James Gosling developed the Java platform at Sun Microsystems, and the Oracle Corporation later acquired it.

Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. Oracle offers its own Hotspot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

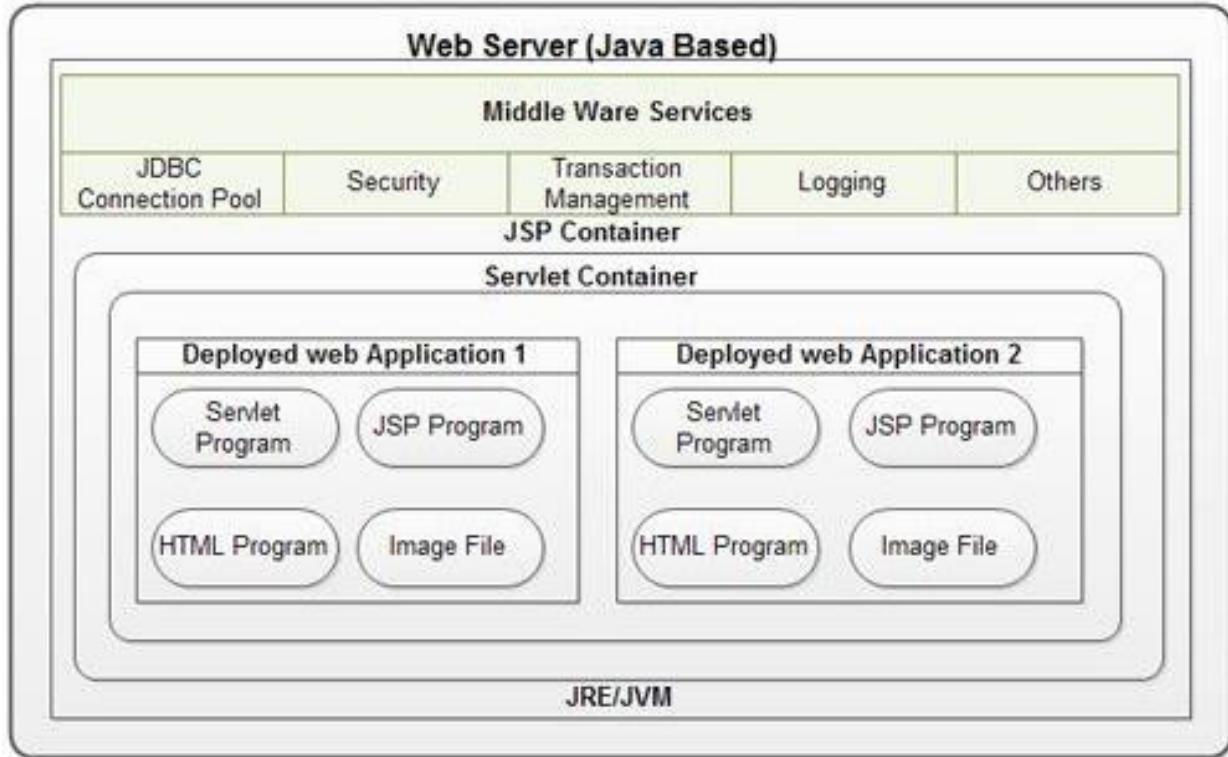


Fig 6.1.1 FRAMEWORK OF JAVA

6.1.1 USES OF JAVA

- Easy to learn – especially for programmers familiar with JSP
- Building and scaling cloud applications.
- Developing chatbots and other marketing tools.
- Powering enterprise-level web applications
- This language is highly stable and supports various interfaces.

6.2 JSP

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime, therefore JSP is a Servlet; each JSP servlet is cached and re-used until the original JSP is modified. Jakarta Server

Pages can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture. JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML.

The resulting page is compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other .jar or Java program, code must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment. JSPs are usually used to deliver HTML and XML documents, but through the use of Output Stream, they can deliver other types of data as well. The Web container creates JSP implicit objects like request, response, session, application, config, page, page Context, out and exception. JSP Engine creates these objects during translation phase.

Java code is not required to be complete or self-contained within a single scriptlet block. It can straddle markup content, provided that the page as a whole is syntactically correct. For example, any Java if/for/while blocks opened in one scriptlet must be correctly closed in a later scriptlet for the page to successfully compile. This allows code to be intermingled and can result in poor programming practices.

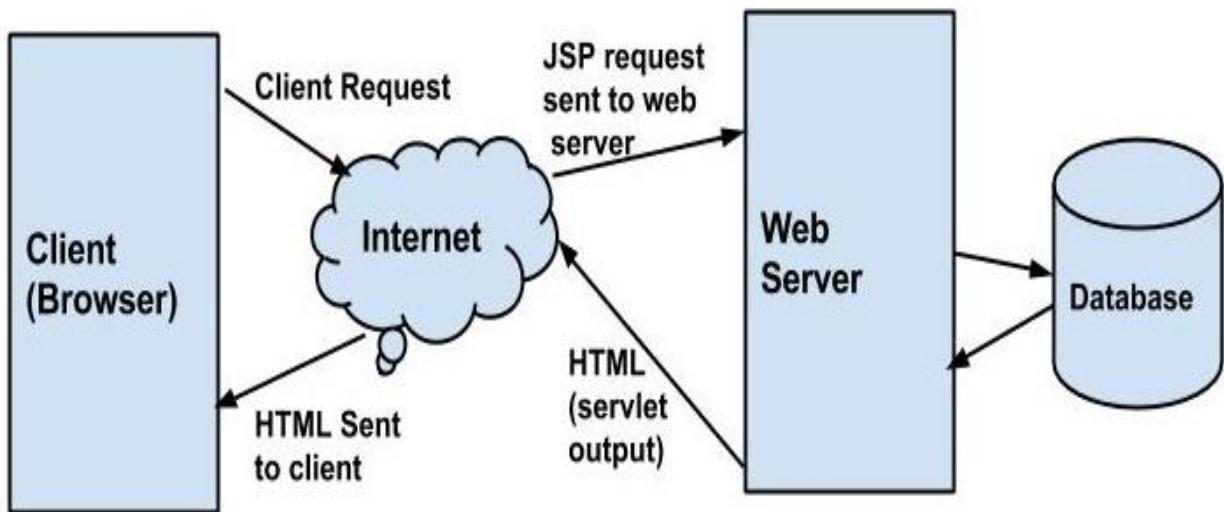


Fig 6.2.1 FRAMEWORK OF JSP

6.2.1 USES OF JSP

- They are easy to maintain.
- No recompilation or redeployment is required.
- JSP has access to entire API of JAVA.
- JSP are extended version of Servlet.

6.3 HTML

HTML is a markup language that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on.

The main parts of our element are as follows:

1. **The opening tag:** This consists of the name of the element (in this case, p), wrapped in opening and closing angle brackets. This states where the begins or starts to take effect — in this case where the paragraph begins.
2. **The closing tag:** This is the same as the opening tag, except that it includes a forward slash before the element name. This states where the element ends in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.
3. **The content:** This is the content of the element, which in this case, is just text.
4. **The element:** The opening tag, the closing tag, and the content together comprise the element.

6.3.1 USES OF HTML

- Structuring web pages. With tags and elements, we can define the headings, paragraphs, and other contents of a web page
- Navigating the internet
- Embedding images and videos
- Improving client-side data storage and offline capabilities

6.4 CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is used to style and layout web pages - for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS

mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML. Styling text is one of the most common things you'll do with CSS. Here we look at text styling fundamentals, including setting font, boldness, italics, line and letter spacing, drop shadows, and other text features. We round off the module by looking at applying custom fonts to your page, and styling lists and links.

6.4.1 USES OF CSS

- Easy to apply to well-structured html
- Smaller file sizes
- Separates style and context
- Good for maintenance

6.5 SQL

Structured Query Language, abbreviated as SQL. SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e., data incorporating relations among entities and variables. SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify how to reach a record, e.g., with or without an index. Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language

and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update, and delete), data definition (schema creation and modification), and data access control. Although SQL is essentially a declarative language (4GL), it also includes procedural elements. SQL was one of the first commercial languages to use Edgar Codd's relational model. The model was described in his influential paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organization for Standardization (ISO). Since then, the standard has been revised to include a larger set of features. Despite the existence of standards, most SQL code requires at least some changes before being ported to different database systems.

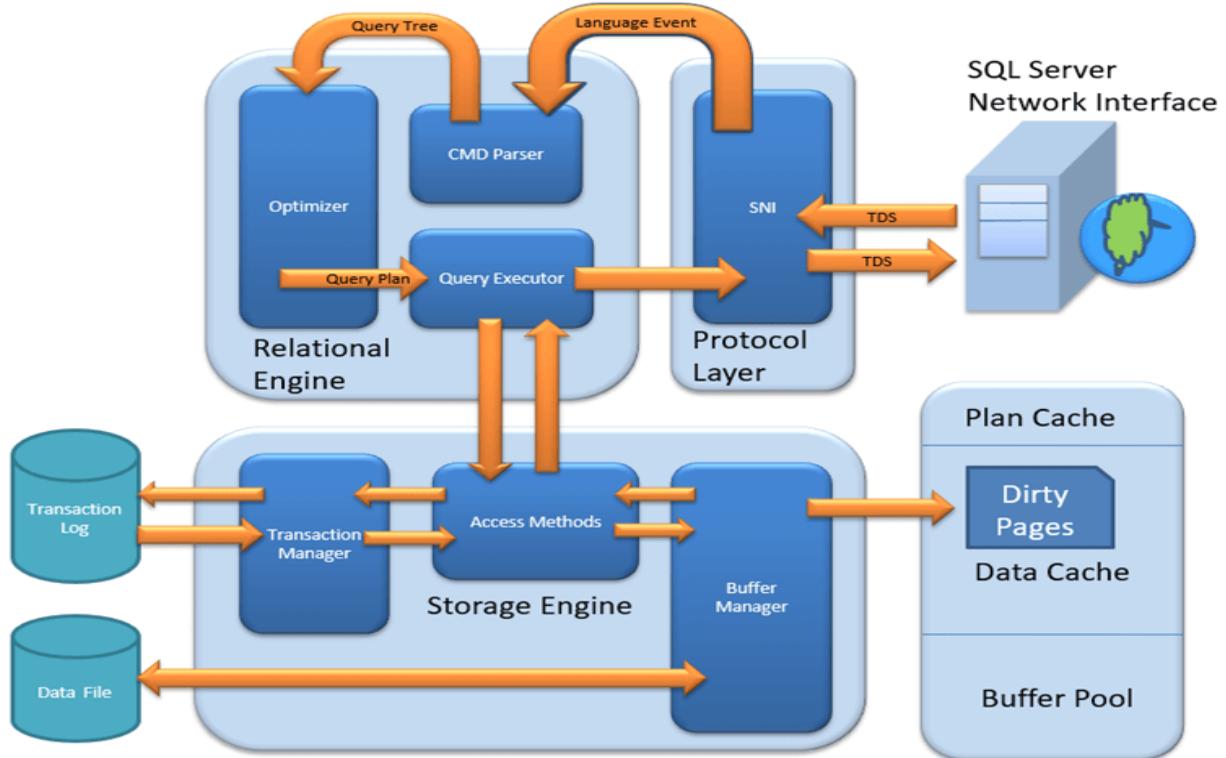


Fig 6.5.1 FRAMEWORK OF SQL

6.5.1 USES OF SQL

- Alter data within a table. If you access a database that has organized data within a table, you can use SQL to manipulate this data.
- Retrieve data from the database
- Change the data structure.
- Define the database's schema.
- Grant access to data.

6.6 NET BEANS

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems. The IDE provides comprehensive support for JDK 7 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the Glassfish Server, the reference implementation of Java EE.

Apache NetBeans is top level Apache Project dedicated to providing rock solid software development products (the Apache NetBeans IDE and the Apache NetBeans Platform) that address the needs of developers, users and the businesses who rely on NetBeans as a basis for their products; particularly, to enable them to develop these products quickly, efficiently and easily by leveraging the strengths of the Java platform and other relevant industry standards. The two base products,

the Apache NetBeans IDE and Apache NetBeans Platform, are free for commercial and non-commercial use, under the Apache license. The Apache NetBeans project is also a vibrant community in which people from across the globe can ask questions, give advice, contribute and ultimately share in the success of our products. The platform does not add a lot of overhead to your application — but it can save a huge amount of time and work. The Apache NetBeans Platform provides a reliable and flexible application architecture.

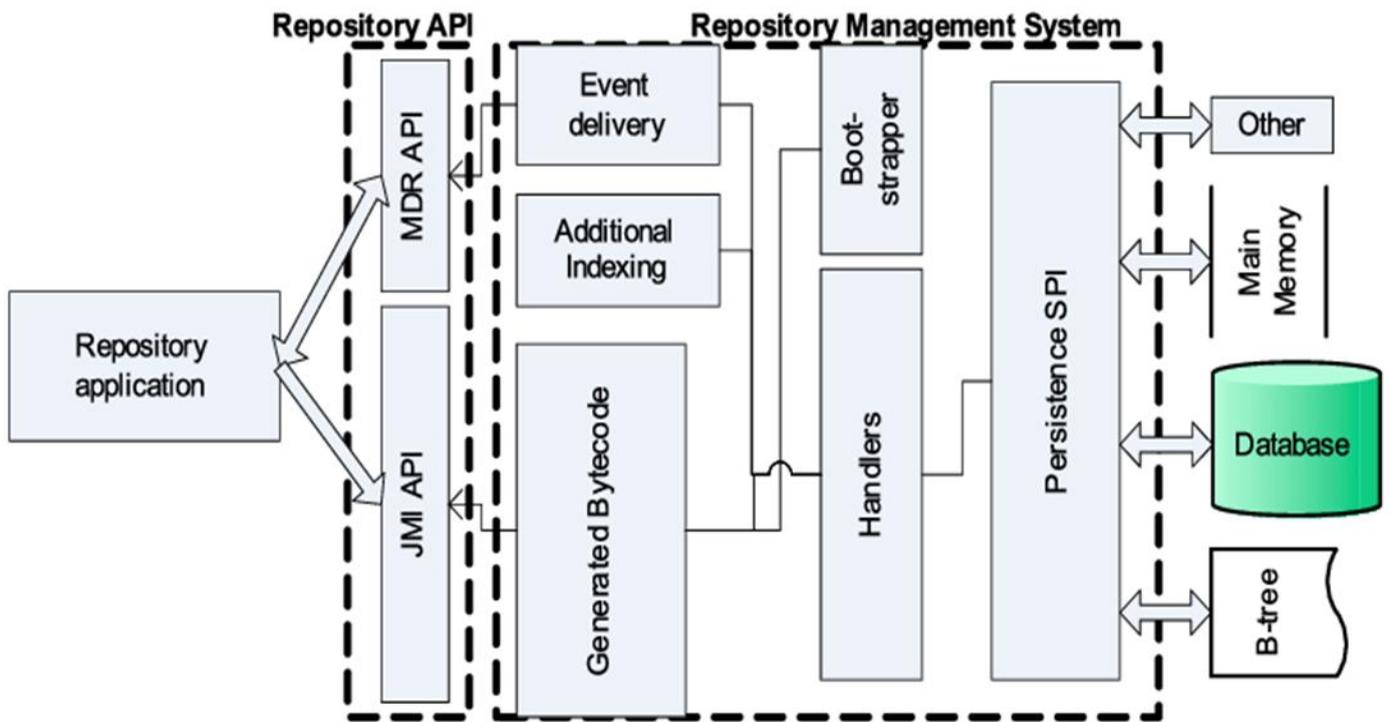


Fig 6.6.1 ARCHITECTURE OF NETBEAN

6.6.1 USES OF NETBEANS

- Debug, builds, exports the application code.
- HTML5, PHP, C/C++, JavaScript, JSP, XML, Groovy and Javadoc.
- Cross-platform support for Linux, Windows, and Mac.
- MySQL, Java DB, Oracle, Postgre SQL, Elastic search, Mongo DB database connection wizard

CHAPTER-7

DIAGRAMS

7. DIAGRAMS

7.1 USECASE DIAGRAM

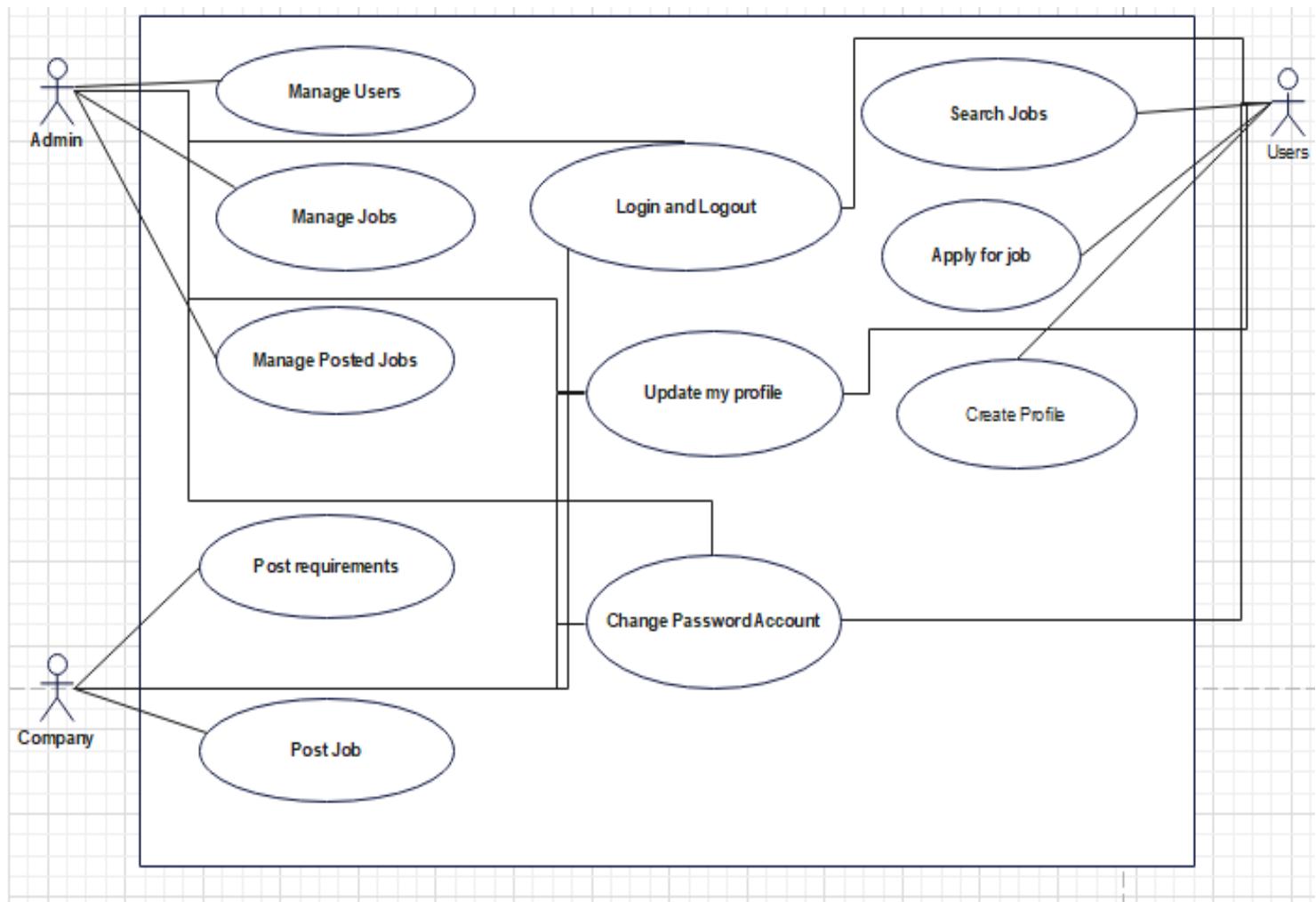


Fig 7.1.1 USECASE DIAGRAM

7.1.1 Description

Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions are interacted. Here the user is the actors and sensing the devices ordering them in the attribute and controlling them are the functions

7.2 E-R DIAGRAM

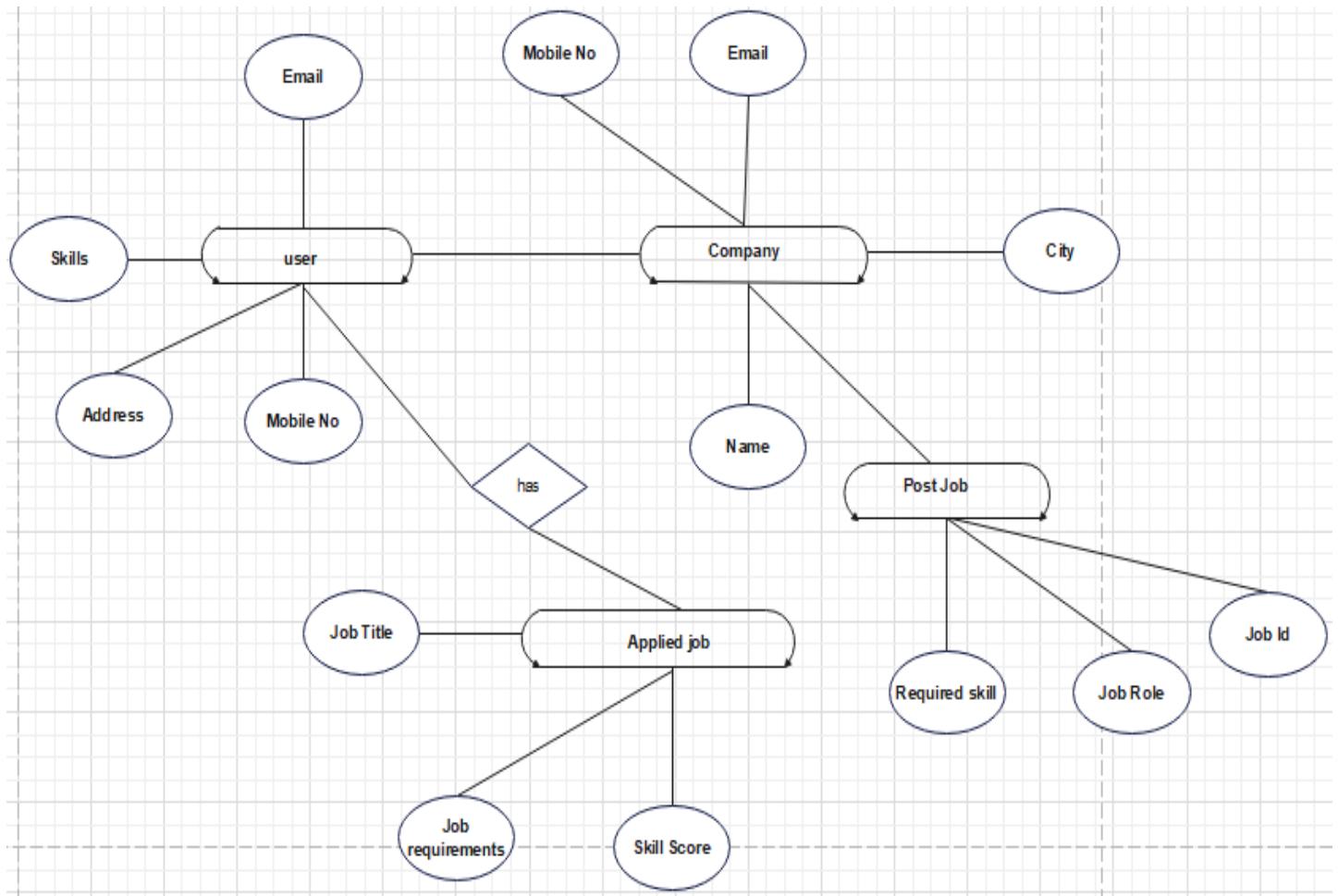


Fig 7.2.1 E-R DIAGRAM

7.2.1 Description

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. It focuses on element without entities rather than relation.

7.3 SEQUENCE DIAGRAM

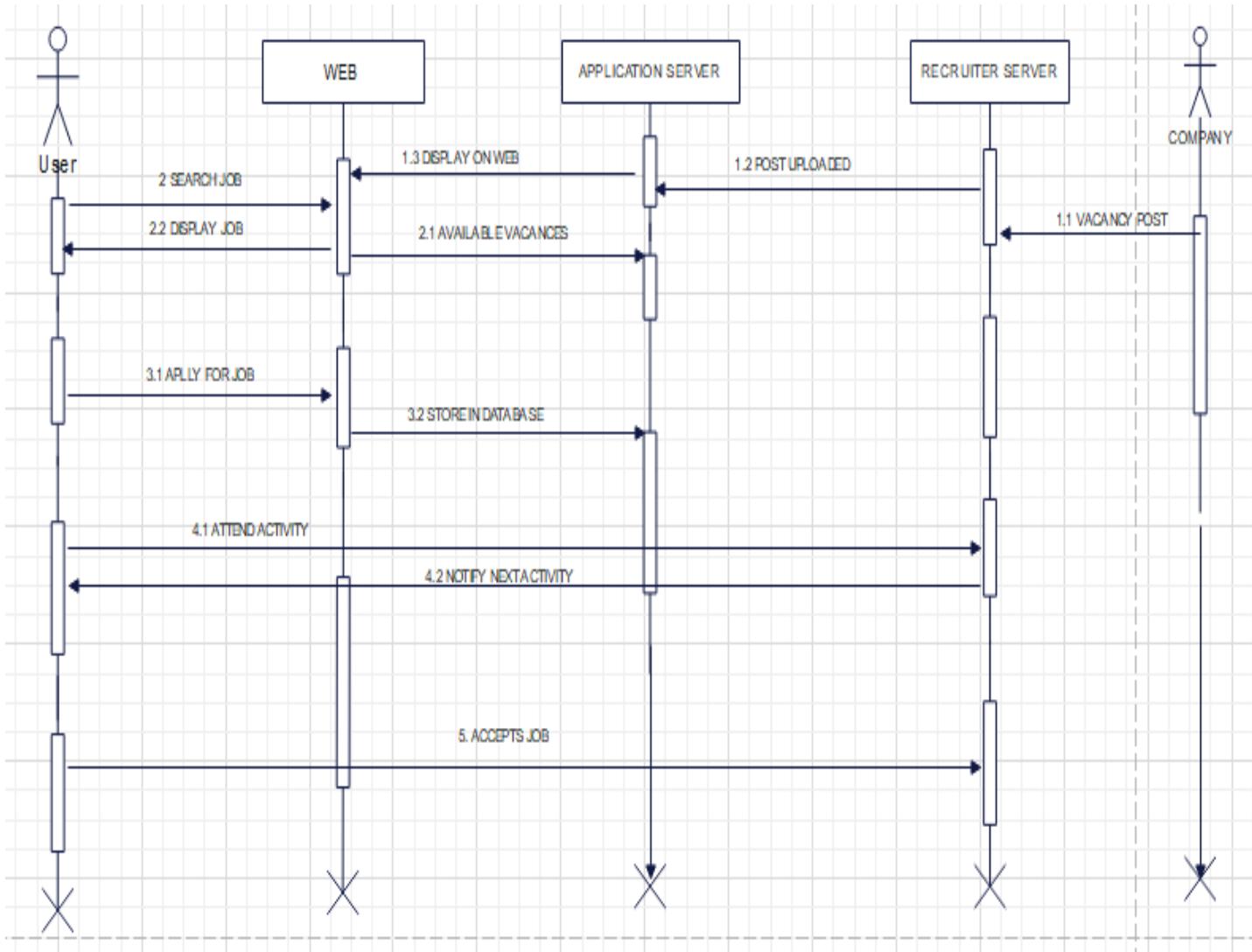


Fig 7.3.1 SEQUENCE DIAGRAM

7.3.1 Description

Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are show as arrows. Here user embedding data and room are objects they interact each other. The arrow shows interaction like send cover video and data, reserving room etc.

7.4 ACTIVITY DIAGRAM

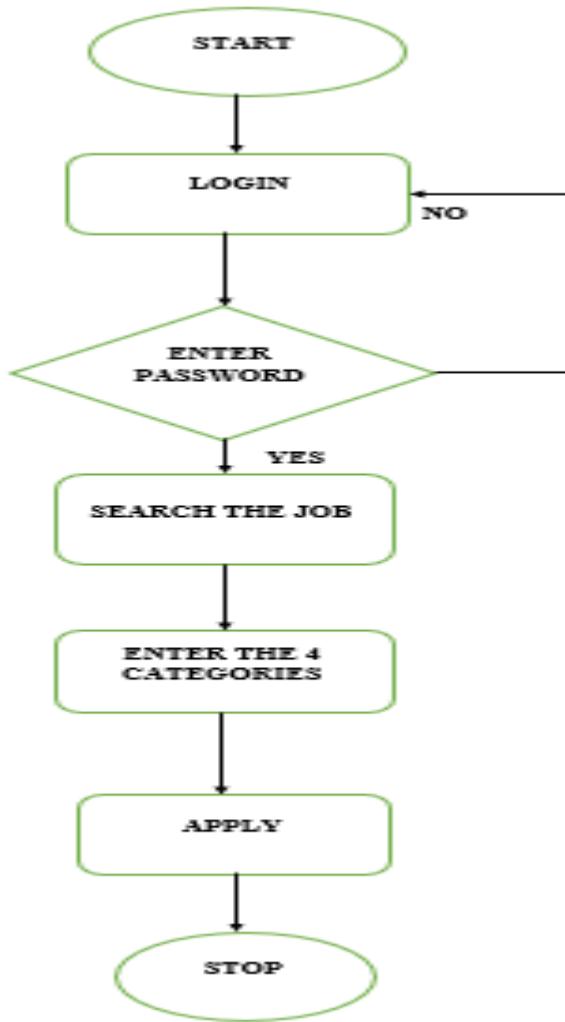


Fig 7.4.1

7.4.1 Description

Activity diagrams represent workflows in a graphical way. They can be used to describe business workflow or the operational workflow of any component in a system.

7.5 DATABASE DIAGRAM

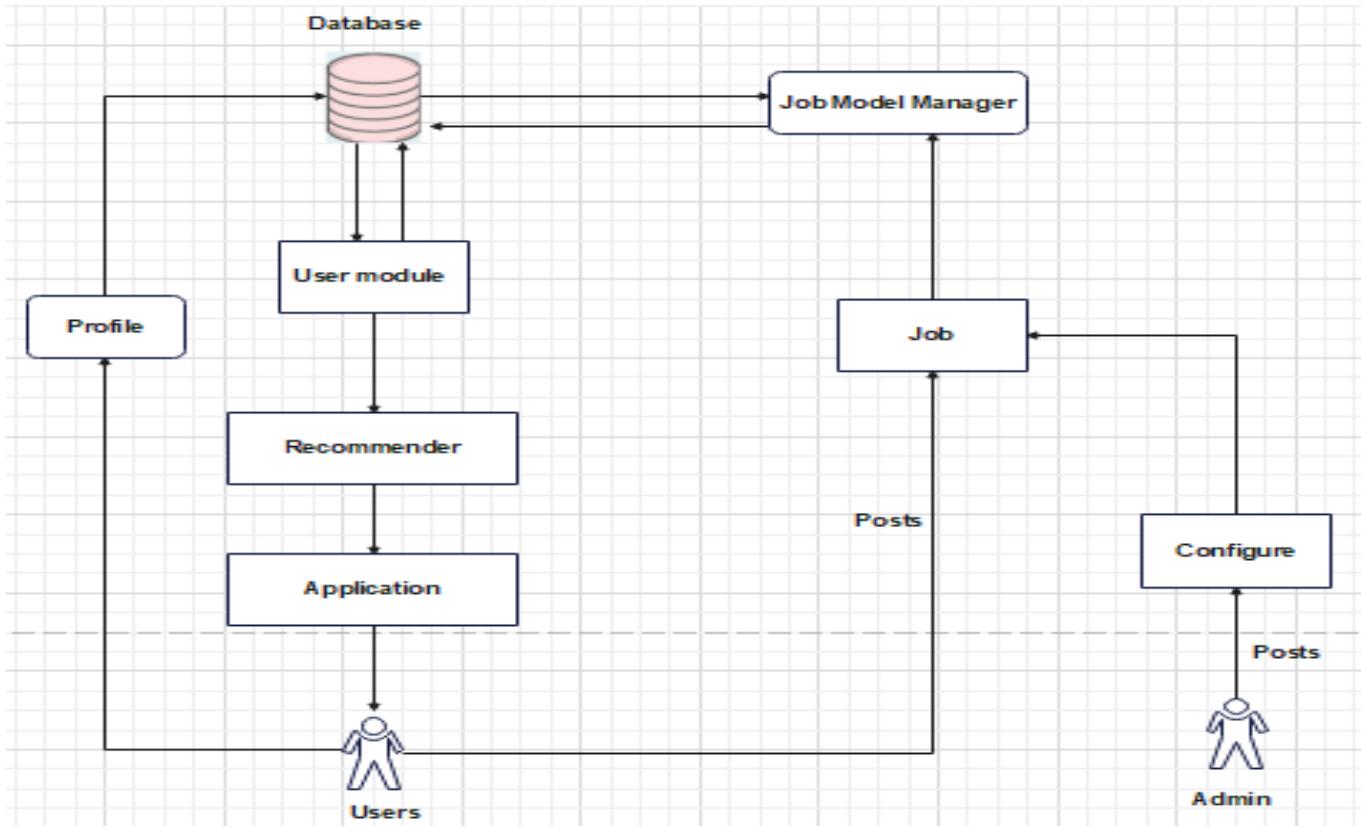


Fig 7.5.1 DATABASE DIAGRAM

7.5.1 Description

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly

7.6 INPUT AND OUTPUT DIAGRAM

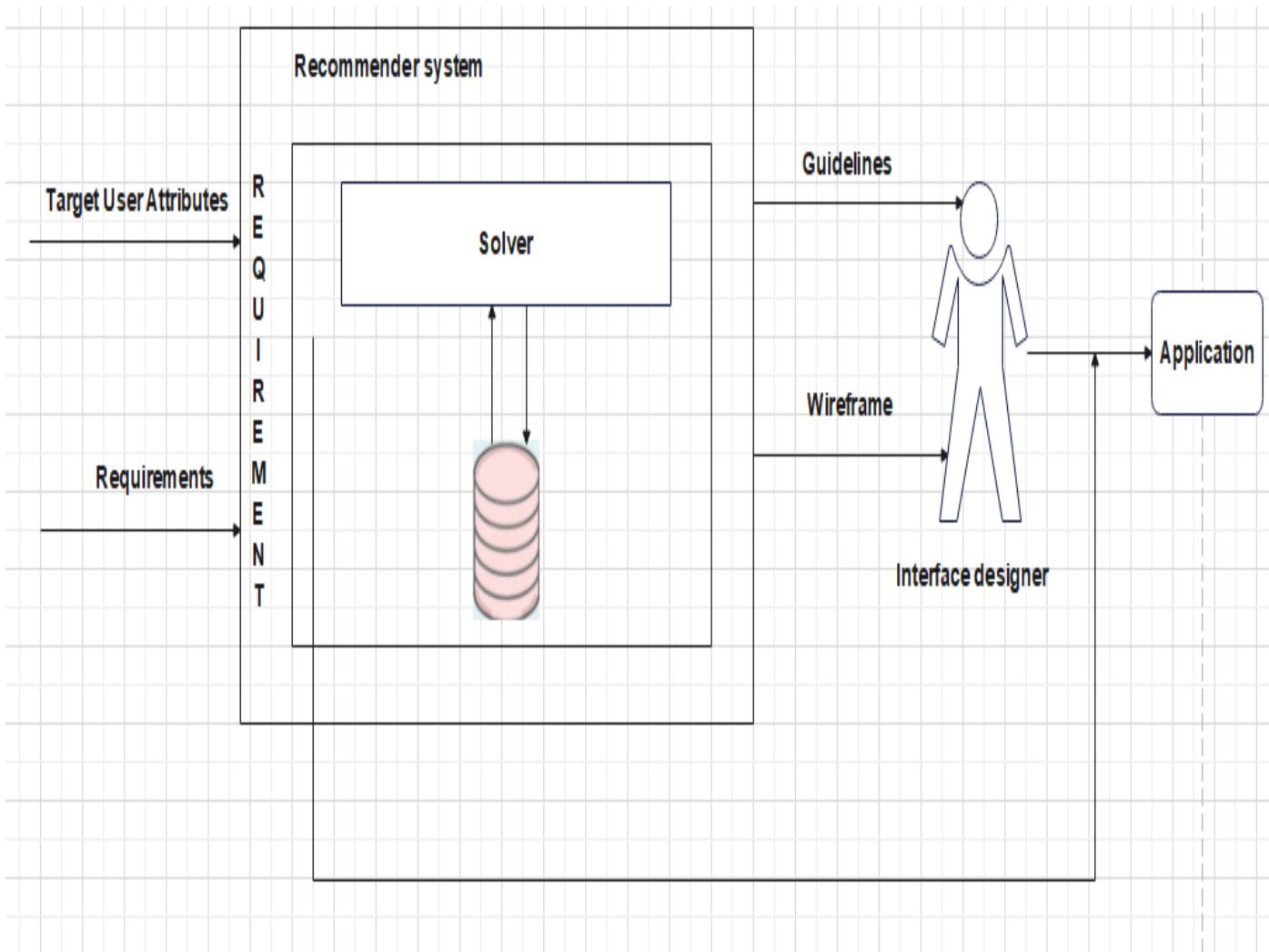


Fig 7.6.1 INPUT AND OUTPUT DIAGRAM

7.6.1 Description

An Input-Output Diagram is a simple high-level representation of a system that shows: The major inputs to a system and their suppliers. The major outputs from a system and their customers. The major components of the system necessary for it to achieve its purpose through transforming inputs to outputs.

CHAPTER-8

SYSTEM TESTING

8. SYSTEM TESTING

8.1 White Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

White-box test design techniques include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing
- Statement coverage
- Decision coverage

White-box testing is a method of testing the application at the level of the source code. The test cases are derived through the use of the design techniques mentioned above: control flow testing, data flow testing, branch testing, path testing, statement coverage and decision coverage as well as modified condition/decision coverage. White-box testing is the use of these techniques as guidelines to create an error free environment by examining any fragile code. These White-box testing techniques are the building blocks of white-box testing, whose essence is the careful testing of the application at the source code level to prevent any hidden errors later on. These different techniques exercise every visible path of the source code to minimize errors and create an error-free environment. The whole point of white-box testing is the ability to know which line of the code is being executed and being able to identify what the correct output should be.

Levels

1. Unit testing. White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.
2. Integration testing. White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour.

1. Open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.
2. Regression testing. White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

8.2 Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

8.3 Test procedures

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of *what* the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of *how* the software produces the output in the first place.

8.4 Test cases

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-function tests may also be used. The test designer selects both valid and invalid

inputs and determines the correct output without any knowledge of the test object's internal structure.

8.5 Test design techniques

- Decision table testing
- All-pairs testing
- State transition tables
- Equivalence partitioning
- Boundary value analysis

8.6 Performance testing

In software engineering, performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage. Performance testing is a subset of performance engineering, an emerging computer science practice which strives to build performance into the implementation, design and architecture of a system.

8.7 Unit testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with a more

associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation.

Testing will not catch every error in the program, since it cannot evaluate every execution path in any but the most trivial programs. The same is true for unit testing. Additionally, unit testing by definition only tests the functionality of the units themselves. Therefore, it will not catch integration errors or broader system-level errors (such as functions performed across multiple units, or non-functional test areas such as performance). Unit testing should be done in conjunction with other software testing activities, as they can only show the presence or absence of particular errors; they cannot prove a complete absence of errors. In order to guarantee correct behaviour for every execution path and every possible input, and ensure the absence of errors, other techniques are required, namely the application of formal methods to proving that a software component has no unexpected behaviour. Unit testing embedded system software presents a unique challenge: Since the software is being developed on a different platform than the one it will eventually run on, you cannot readily run a test program in the actual deployment environment, as is possible with desktop programs.

8.8 Software testing

Software testing is a combinatorial problem. For example, every Boolean decision statement requires at least two tests: one with an outcome of "true" and one with an outcome of "false". As a result, for every line of code written, programmers often need 3 to 5 lines of test code. This obviously takes time and its investment may not be worth the effort. There are also many problems that cannot easily be tested at all – for example those that are nondeterministic or involve multiple threads. In addition, code for a unit test is likely to be at least as buggy as the code it is testing. Fred Brooks in *The Mythical Man-Month* quotes: never take two chronometers to sea. Always take one or three. Meaning, if two chronometers contradict, how do you know which one is correct. Another challenge related to writing the unit tests is the difficulty of setting up realistic and useful tests. It is necessary to create relevant initial conditions so the part of the application being tested behaves like part of the complete system. If these initial conditions are not set correctly, the test will not be exercising the code in a realistic context, which diminishes the value and accuracy of unit test results. To obtain the intended benefits from unit testing, rigorous discipline is needed throughout the software development process. It is essential to keep careful records not only of the tests that have been performed, but also of all changes that have been made to the source code of this or any other unit in the software. Use of a version control system is essential. If a later version of the unit fails a particular test that it had previously passed, the version-control software can provide a list of the source code changes (if any) that have been applied to the unit since that time. It is also essential to implement a sustainable process for ensuring that test case failures are reviewed daily and addressed immediately if such a process is not implemented and ingrained into the team's workflow, the application will evolve out of sync.

with the unit test suite, increasing false positives and reducing the effectiveness of the test suite.

8.9 Functional testing

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does.

Functional testing typically involves five steps. The identification of functions that the software is expected to perform

1. The creation of input data based on the function's specifications
2. The determination of output based on the function's specifications
3. The execution of the test case
4. The comparison of actual and expected outputs

8.10 Testing types

8.10.1 Load testing

Load testing is the simplest form of performance testing. A load test is usually conducted to understand the behaviour of the system under a specific expected load. This load can be the expected concurrent number of users on the application performing a specific number of transactions within the set duration. This test will give out the response times of all the important business

critical transactions. If the database, application server, etc. are also monitored, then this simple test can itself point towards bottlenecks in the application software.

8.10.2 Stress testing

Stress testing is normally used to understand the upper limits of capacity within the system. This kind of test is done to determine the system's robustness in terms of extreme load and helps application administrators to determine if the system will perform sufficiently if the current load goes well above the expected maximum.

8.10.3 Soak testing

Soak testing, also known as endurance testing, is usually done to determine if the system can sustain the continuous expected load. During soak tests, memory utilization is monitored to detect potential leaks. Also important, but often overlooked is performance degradation. That is, to ensure that the throughput and/or response times after some long periods of sustained activity are as good as or better than at the beginning of the test. It essentially involves applying a significant load to a system for an extended, significant period of time. The goal is to discover how the system behaves under sustained use.

8.10.4 Spike testing

Spike testing is done by suddenly increasing the number of or load generated by, users by a very large amount and observing the behaviour of the system. The

goal is to determine whether performance will suffer, the system will fail, or it will be able to handle dramatic changes in load.

8.10.5 Configuration testing

Rather than testing for performance from the perspective of load, tests are created to determine the effects of configuration changes to the system's components on the system's performance and behaviour. A common example would be experimenting with different methods of load-balancing.

8.11 Isolation testing

Isolation testing is not unique to performance testing but involves repeating a test execution that resulted in a system problem. Often used to isolate and confirm the fault domain.

8.12 Integration testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

8.13 System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer.

8.14 Structure Testing

It is concerned with exercising the internal logic of a program and traversing particular execution paths.

8.15 Output Testing

- Output of test cases compared with the expected results created during design of test cases.

- Asking the user about the format required by them tests the output generated or displayed by the system under consideration.
- Here, the output format is considered into two was, one is on screen and another one is printed format.
- The output on the screen is found to be correct as the format was designed in the system design phase according to user needs.

8.16 User acceptance Testing

- Final Stage, before handing over to the customer which is usually carried out by the customer where the test cases are executed with actual data.
- The system under consideration is tested for user acceptance and constantly keeping touch with the prospective system user at the time of developing and making changes whenever required.
- It involves planning and execution of various types of test in order to demonstrate that the implemented software system satisfies the requirements stated in the requirement document.

CHAPTER-9

SAMPLE CODING

9. CODING

9.1 JSP CODE

```
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%
String fs=request.getParameter("FirstName");
String c=request.getParameter("LastName");
String l=request.getParameter("EmailId");
String jt=request.getParameter("CompanyName");
String a=request.getParameter("ComapnyUrl");
String g=request.getParameter("title");
String p=request.getParameter("Location");
String add=request.getParameter("jd");
String email=request.getParameter("vacancies");
<!DOCTYPE html>
<html>
<style>
input[type=submit] {
background-color: #0041C2;
width:10%;
border-style: outset;
border-color:silver;
color: #fff;
padding: 7px 7px;
```

```

text-decoration: inherit;
margin: 2px 2px;
cursor: pointer;
while(resultSet.next()){
%>
<tr>
<td><%=resultSet.getString("FirstName") %></td>
<td><%=resultSet.getString("Lastname") %></td>
<td><%=resultSet.getString("title") %></td>
<td><%=resultSet.getString("Location") %></td>
<td><%=resultSet.getString("jd") %></td>
</tr>
<%
}
connection.close();
} catch (Exception e) {
e.printStackTrace();
}
%>
</table>
<br>
<div align='center'>
<form action="recprofile.html" method="get">
<input type="submit" value="Go Back">
</form></div>
</body>
</html>

```

9.2 JAVA CODE

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/***
 *
 * @author sanje
 */
@WebServlet(urlPatterns = {" /Fodra"})
public class Fodra extends HttpServlet {
/***
 * Processes requests for both HTTP
 * <code>GET</code> and
 * <code>POST</code> methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
```

```
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        /* TODO output your page here. You may use following sample code. */
        response.sendRedirect("welcome.html");
    } finally {
        @Override
        protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }
    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}
```

9.3 HTML CODE

```
<!DOCTYPE html>

<!-- Template by html.am -->

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>JOB RECOMMENDATION WEBSITE</title>
    <style type="text/css">
      table, th, td {
        border:dotted;
      }
      body {
        margin: 0;
        padding: 0;
        overflow: hidden;
        height: 100%;
        max-height: 100%;
        font-family:Sans-serif;
        line-height: 1.5em;
      }
      #header {
        position: absolute;
        top: 0;
        left: 0;
        width: 100%;
        overflow: hidden; /* Disables scrollbars on the navigation frame. To enable scrollbars, change "hidden" to "scroll" */
      }
    </style>
  </head>
  <body>
    <div id="header">
      <h1>Job Recommendation Website</h1>
      <p>A template for a job recommendation website using HTML, CSS, and JavaScript. The website features a header with a logo and navigation menu, a hero section with a large image and text, a grid of job cards, and a footer with social media links and copyright information. The code is modular and follows best practices for accessibility and performance.</p>
    </div>
    <div id="content">
      <h2>Job Recommendations</h2>
      <div>
        <h3>Recent Job Posts</h3>
        <ul>
          <li>Software Engineer at Google</li>
          <li>Machine Learning Researcher at Facebook</li>
          <li>Product Manager at Amazon</li>
          <li>Data Scientist at Netflix</li>
        </ul>
      </div>
      <h2>About Us</h2>
      <div>
        <h3>Our Mission</h3>
        <p>Our mission is to connect professionals with opportunities that align with their skills and interests. We believe in the power of technology to facilitate meaningful connections and drive career growth.</p>
      </div>
      <h2>Contact Us</h2>
      <div>
        <h3>Address</h3>
        <p>123 Main Street, Suite 100<br/>Anytown, USA 12345</p>
        <h3>Email</h3>
        <p>info@jobrecommendation.com</p>
        <h3>Phone</h3>
        <p>(555) 123-4567</p>
      </div>
    </div>
  </body>
</html>
```

```

background: #BCCE98;
}

main {
position: fixed;
top: 100px; /* Set this to the height of the header */
left: 0;
right: 0;
bottom: 0;
overflow: auto;
background: #fff;
}

innertube {
margin: 15px; /* Provides padding for the content */
}

bordercolor="Skyblue" >

<tr>

<th class="style4"><span class="style9"><a href="User.html"> <font color="Black" size="+1" face="Times New Roman">USER</a></font></span></th>

<th class="style4"><span class="style9"><a href="Webindex.jsp"><font color="Black" size="+1" face="Times New Roman">ADMIN</a></font></span></th>

<th class="style4"><span class="style9"><a href="ad.html"><font color="Black" size="+1" face="Times New Roman">RECRUITER</a></font></span></th>

</body>

</html>

```

CHAPTER-10

SCREENSHOTS

10. SCREENSHOTS

10.1 WELCOME PAGE



Fig 10.1.1 WELCOME PAGE

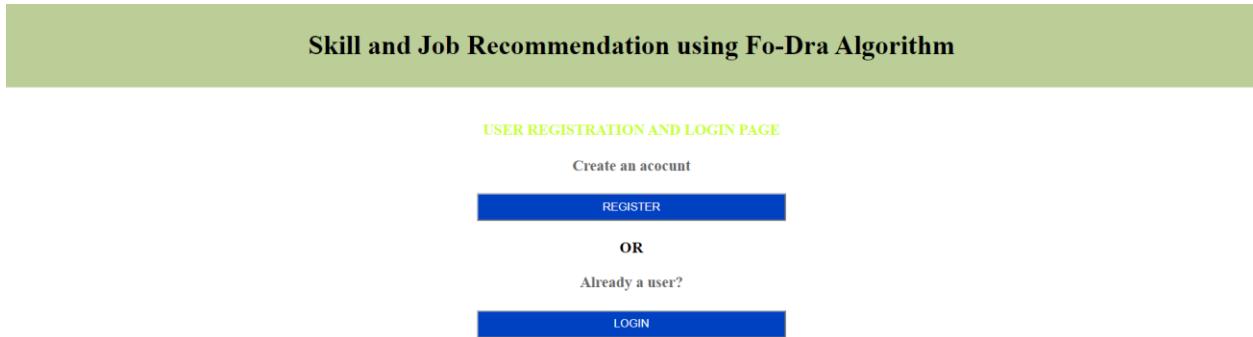


Fig 10.1.2 USER PAGE

10.2 USER MODULE

Skill and Job Recommendation using Fo-Dra Algorithm

USER REGISTRATION PANEL

First Name	<input type="text" value="enter first name..."/>
Last Name	<input type="text" value="Enter last name..."/>
Password	<input type="password" value="enter password..."/>
Confirm Password	<input type="password" value="confirm password..."/>
Phone Number	<input type="text" value="enter phone number..."/>
Email	<input type="text" value="enter email address..."/>
Address	<input type="text" value="enter address..."/>
Gender	<input type="radio"/> Male <input type="radio"/> Female

REGISTER

Fig 10.2.1 USER REGISTER PAGE

Skill and Job Recommendation using Fo-Dra Algorithm

USER LOGIN PANEL

Email id	<input type="text" value="enter email address..."/>
Password	<input type="password" value="enter password..."/>

LOGIN

Fig 10.2.2 USER LOGIN PAGE

A JOB SEARCHING
WEBSITE

Skill and Job Recommendation Using Fo-dra Algorithm

HOME

TOP JOBS

SEARCH

FIND RECENT JOBS

ABOUT US

Welcome to Job Recommendation System

Our Recommendation system provides the user to search the job which will be suitable to their qualification, eligibility and skills which they acquired through academic and also through other sources. In this the Fodra Algorithm is implemented which is used to recommend the suitable job to the users based on the filtration process

Purpose of Job Recommendation System

In this current situation finding a suitable job for a person is very tough and also many of the people struggle to get a job for their qualification. This is mainly due to high population and less job all around the world. In recent days popularity of online recruiting platforms in modern industry many of the users choose these platforms as means to search their job. In the last year, Job recommender systems have become popular since they successfully reduce information overload by generating personalized job suggestions. So the main objective of our project is to recommend jobs to the users and also to recommend the job which suits the qualifications of the users.

What we are doing

We created a website or a platform for the people who are searching for a job. This also help everyone to find their suitable job based on their qualifications

Apply Process

1. Search a job
2. Apply for job
3. Get your Job

Fig 10.2.3 HOME PAGE

A JOB SEARCHING
WEBSITE

Skill and Job Recommendation Using Fo-dra Algorithm

HOME

TOP JOBS

SEARCH

FIND RECENT JOBS

ABOUT US

PHP DEVELOPER		
TCS Solutions		
Location:Banglore		
Experience:0 years or fresher		
View Job		

TECHNICAL SUPPORT ENGINEER		
EClinicalWorks		
Location:Ahmedabad		
Experience:0 years or fresher		
View Job		

SOFTWARE ENGINEER		
Nitor Infotech		
Location:Pune		
Experience:0 years or fresher		
View Job		

JUNIOR ENGINEER		
Crestaerospace Technologies		
Location:Banglore		
Experience:0 years or fresher		
View Job		

Fig 10.2.4 TOP JOBS

44

A JOB SEARCHING
WEBSITE

Skill and Job Recommendation Using Fo-dra Algorithm

Search your dream job now!!!

HOME			
TOP JOBS	Software Developer	B.E/B.TECH	Chennai
SEARCH	20.000-30.000		
FIND RECENT JOBS			
ABOUT US	<input type="button" value="Search"/>		

Fig 10.2.5 SEARCH JOB PAGE

A JOB SEARCHING
WEBSITE

Skill and Job Recommendation Using Fo-dra Algorithm

Results of the category searched by the user

Job Title:
Software Developer

Course:
B.E/B.TECH

Location:
Chennai

Salary:
20,000-30,000

Job Description:

We are currently in search of an experienced Software Developer to create and maintain software, in-house hardware and network systems. The successful candidate will also help to develop critical products for our clients, and offer excellent technical support. This is a fantastic opportunity to engage in a positive and creative work environment that offers excellent benefits and flexible schedules.

Fig 10.2.6 RESULT PAGE

10.3 ADMIN MODULE

Skill and Job Recommendation using Fo-Dra Algorithm

ADMIN LOGIN PANEL

Email Address

Password

Fig 10.3.1 ADMIN LOGIN PAGE

A JOB SEARCHING
WEBSITE

HOME

VISIT USER PROFILE

VISIT COMPANY PROFILE

ABOUT US

Skill and Job Recommendation Using Fo-dra Algorithm

Welcome Admin!!!

Visit the number of users registered on this site

Visit the different types of jobs registered on this site by the users

Fig 10.3.2 USER PROFILE VIEWING PAGE

Skill and Job Recommendation Using Fo-dra Algorithm

HOME
VISIT USER PROFILE
VISIT COMPANY PROFILE
ABOUT US

Here are the details about the different types of jobs applied by the user using Fo-Dra Algorithm

Job Title	Course	Location	Salary
Software Developer	B.E/B.TECH	Chennai	20,000-30,000
Software Developer	ME/M.TECH	Nagpur	30,000-40,000
Software Developer	MCA	Hyderabad	30,000-40,000
Software Developer	MBA/PGDM	Mumbai	20,000-30,000
Marketing	B.Com	Hyderabad	10,000-20,000
Marketing	M.Com	Lucknow	20,000-30,000
Marketing	MBA/PGDM	Kolkata	10,000-20,000
Software Testing	B.E/B.TECH	Mumbai	10,000-20,000
Software Testing	ME/M.TECH	Delhi	20,000-30,000
Software Testing	MCA	Banglore	20,000-30,000
Web designing	B.E/B.TECH	Coimbatore	10,000-20,000
Web designing	ME/M.TECH	Chennai	20,000-30,000
Web designing	MCA	Hyderabad	20,000-30,000
IT Hardware Engineer	B.E/B.TECH	Chennai	10,000-20,000
IT Hardware Engineer	ME/M.TECH	Kolkata	10,000-20,000
Finops Developer	MBA/PGDM	Banglore	20,000-30,000

Fig 10.3.3 JOB VIEWING PAGE

Skill and Job Recommendation Using Fo-dra Algorithm

HOME
VISIT USER PROFILE
VISIT COMPANY PROFILE
ABOUT US

Here are the details about the jobs posted by the recruiters on this site using Fo-Dra Algorithm

First Name	Last Name	Email	Company Name	Company Url(optional)	Job Title	Location	Job Description	Vacancies
Virat	Kohli	tcsptvtltd23@gmail.com	TCS limited	TCS limited	Software Tester	Chennai	We are looking for a results-driven software tester to be responsible for designing and running tests on software usability. The software tester will conduct tests, analyze the results, and report observations to the design team. You may also interact with clients in order to understand the requirements of the product.	10
MS	Dhoni	msd7777@gmail.com	Panimalar Solutions	Panimalar Solutions	Software Developer	Banglore	fbfebgbggbg	3

Go Back

Fig 10.3.4 JOB DETAIL PAGE

10.4 RECRUITER MODULE

Skill and Job Recommendation using Fo-Dra Algorithm

RECRUITER/EMPLOYER REGISTRATION AND LOGIN PAGE

[Create an account](#)

[REGISTER](#)

OR

[Already a user?](#)

[LOGIN](#)

Fig 10.4.1 RECRUITER PAGE

Skill and Job Recommendation using Fo-Dra Algorithm

**RECRUITER/EMPLOYER
REGISTRATION PANEL**

First Name	<input type="text" value="enter first name..."/>
Last Name	<input type="text" value="Enter last name..."/>
Password	<input type="text" value="enter password..."/>
Confirm Password	<input type="text" value="confirm password..."/>
Company Name	<input type="text" value="enter company name..."/>
Select Industry	<input type="text" value="--Select--"/>
Type of Company	<input type="radio"/> Private <input type="radio"/> Public
Designation	<input type="text" value="enter designation..."/>
PinCode	<input type="text" value="enter incode..."/> <small>Format: 123456</small>
Company Address	<input type="text" value="enter address..."/>
Company Email	<input type="text" value="enter email address..."/>
Gender	<input type="radio"/> Male <input type="radio"/> Female <small>Minimum</small>

Fig 10.4.2 RECRUITER REGISTRATION



Fig 10.4.3 RECRUITER LOGIN

The image shows a form titled "UPDATE YOUR COMPANY DETAILS". The left sidebar lists "A JOB SEARCHING WEBSITE" with links for "HOME", "UPDATE COMPANY DETAILS", and "ABOUT US". The main form contains fields for "FIRST NAME", "LAST NAME", "EMAIL ID", "COMPANY NAME", "COMPANY URL (optional)", "JOB TITLE", "LOCATION", "JOB DESCRIPTION" (with a note about min 50 words), and "JOB VACANCIES". Each field has a corresponding input box and a character limit note. At the bottom are "Update" and "Reset" buttons.

FIRST NAME	<input type="text"/> [max 30 characters a-z and A-Z]
LAST NAME	<input type="text"/> [max 30 characters a-z and A-Z]
EMAIL ID	<input type="text"/>
COMPANY NAME	<input type="text"/> [max 30 characters a-z and A-Z]
COMPANY URL (optional)	<input type="text"/>
JOB TITLE	<input type="text"/> [max 30 characters a-z and A-Z]
LOCATION	<input type="text"/> [max 20 characters a-z and A-Z]
JOB DESCRIPTION	<input type="text"/> [min 50 words a-z and A-Z]
JOB VACANCIES	<input type="text"/>

Fig 10.4.4 JOB POSTING PAGE

CHAPTER-11

CONCLUSION

11.1 CONCLUSION

In our project, we implemented FoDRA Algorithm to enable job seekers to search for the job using four dimensions and get accurate jobs they are looking for. FoDRA is an efficient algorithm to develop a job recommendation system. Whenever there are lot of criteria's to be considered, FoDRA is an algorithm which implies to recommend suitable jobs for the job seekers. In this project, we managed to develop a prototype which recommends the perfect job for the job seekers based on their Job title, Qualification, Location and salary which will be an essential feature for them to get the job. To further optimize the recommendation system, and integrate the system for better performance we keep in check the sparsity of user profile and use some methods for filling user's preference and how it can be utilized.

11.2 FUTURE ADVANCEMENTS

In our project, we used four essential features which were combined to recommend suitable jobs to the job seekers. But when the job seeker expect more features apart from those four features such as Working Environment, Food facility, Transport Facility etc.. it will be difficult to incorporate some more features according to the requirement of the job seeker. This Job Recommendation system accommodates those features only on the job seeker side. Another future advancement to this project will be made when these features will be integrated into the recruiter side also which will be an update to the system.

REFERENCES

REFERENCES

1. Job Seeking: The Process and Experience of Looking for a Job Connie R. Wanberg, Abdifatah A. Ali, and Borbala Csillag Department of Work and Organizations, Carlson School of Management, University of Minnesota, Minneapolis, Minnesota55455.
2. The quest for sustainable employment: Challenges faced by young people during the job-search process. Received 24 May 2017. Accepted24 Jan 2019.
3. Job search strategies of individuals at the risk of the poverty, published January11,2019.
4. Job Search Methods and Outcome ,published July 2002,source Repec.
5. Online Job and Candidate Recommendation System, published March 2019.
6. A Study of Issues in Job Portals: Research Analysis, Nitesh Kumar School of Computer Science and Engineering, Lovely Professional University Punjab, India.
7. Enhanced Job Recommendation System ,Department of Computer Engineering, Sinhgad Academy of Engineering, Pune, India.
8. Job Recommendation System Using Profile Matching And Web-Crawling. Computer Science & Engineering, K.K.Wagh College of Engineering, Nashik ,India.
9. SQL server Architecture published by Richard Peterson on January 21, 2023 source - guru99

- 10 Efficient and Scalable Job Recommender System Using Collaborative Filtering, published may 2020.
- 11 Job Recommendation System Using Content and Collaborative-Based Filtering, published August 2021.
- 12 Online Job and Candidate recommendation published on march 2019 source research gate.
- 13 Job Recommender system review by Corne de Ruijter and Sandjai Bhulai of Amsterdam university published on November 2021 Source Research gate.
- 14 Definition of java and its uses by James Hartman published on March 4th 2023 source Guru99
- 15 Definition and uses of JSP, SQL and NetBeans from java point, oracle and Geeks for Geeks.
- 16 Definition uses and the structure of Html and CSS from the source of MDN web docs.
- 17 Architectural diagram for JSP published by Krishna Srinivasan on January 15, 2014 source Java beat.
- 18 Architecture of java-based Server published by Dinesh Thakur from the source A Computer Notes.
- 19 Architecture of OMG MOF-based repository by Ilia petrov and Alejandro Buchmann of Reutlingen and Vedia Industrial Mexico published on November 2008.