JOB PORTAL

A Mini Project Thesis
Submitted in the partial fulfillment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

D. SATHVIKA	**21281A0567**
K. SHIRISHA	**21281A0588**
V. VARUN	**21281A0578**
N. AJAY	**21281A0569**

PROJECT GUIDE
Mr. RAJENDAR KALVALA
(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE

(Approved by AICTE, New Delhi, Affiliated to JNTU, Hyderabad,
Accredited with 'A++' Grade by NAAC, Accredited by NBA)
Singapur, Huzurabad, Karimnagar, Telangana- 505468

2024-2025

JOB PORTAL

A Mini Project Thesis
Submitted in the partial fulfillment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

D. SATHVIKA	**21281A0567**
K. SHIRISHA	**21281A0588**
V. VARUN	**21281A0578**
N. AJAY	**21281A0569**

PROJECT GUIDE
Mr. RAJENDAR KALVALA
(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE

(Approved by AICTE, New Delhi, Affiliated to JNTU, Hyderabad,
Accredited with ‘A++’ Grade by NAAC, Accredited by NBA)
Singapur, Huzurabad, Karimnagar, Telangana- 505468
2024-2025

KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE, SINGAPUR

Sponsored by KAMALA EDUCATION SOCIETY, Approved by AICTE, New Delhi and Affiliated to JNTU-Hyderabad, T.G.

Accredited by NBA (CSE, ECE &EEE) and NAAC with A++

HUZURABAD, KARIMNAGAR, TELANGANA, INDIA- 505468

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that **D. SATHVIKA (21281A0567), K. SHIRISHA (21281A0588), V.VARUN (21281A0578), N.AJAY (21281A0569)** of IV B.Tech (CSE) have satisfactorily completed the dissertation work for Industrial Oriented mini project entitled "JOB PORTAL" towards the partial fulfillment of B. Tech degree in the academic year 2024-2025.

Project Guide **Head of the Department**

Principal **External Examiner**

INDEX

ACKNOWLEDGEMENT

The success of any course depends mostly on the teachers who teach us. Only good teaching can interpret the syllabus and produce desirable changes and competent citizens. This one was a team effort and many people whose names do not appear on the cover deserve credit. First, we thank god almighty for his manifold mercies in carrying out of our project successfully.

We heartfully thank our Management and Director **Dr. K. SHANKER**, and principal **Dr. K. ESHWARAIAH** for providing all the resources in completing our project.

We sincerely thank our Head of the Department of Computer Science & Engineering and Professor **Dr Major. RAVINDRA BABU KALLAM**, for encouraging us in doing real time projects and for his guidance.

We thank our project coordinator **Ms. R. BHARATHI**, Assistant Professors in CSE for her continuous support in completion of this project.

We thank our internal guide **Mr. RAJENDAR KALVALA**, Assistant Professor in department of Computer Science & Engineering, for his guidance and support.

We also thank other teaching and non-teaching staff of Kamala Institute of Technology and Science, for supporting us in every stage of our industrial oriented mini project, entitled **"JOB PORTAL".**

We also thank our parents and friends for their moral support throughout the project work that helped to strengthen our will.

DURISHETTI SATHVIKA	**(21281A0567)**
KUKKALA SHIRISHA	**(21281A0588)**
VARIKELA VARUN	**(21285A0578)**
NARSINGOJU AJAY	**(21281A0569)**

ABSTRACT

The **Job Portal** System is a comprehensive online platform designed to streamline the job search and recruitment process for job seekers. This system provides an intuitive interface that allows users to upload resumes, and apply for jobs with ease. Admin can post job openings, manage applications, and search for qualified candidates efficiently. With a growing number of candidates entering the job market, job seekers require efficient tools to find relevant opportunities. A job portal aggregates listings, enabling users to access a broader range of job openings in one place.

Key features include a robust search engine that filters jobs based on various criteria such as location, industry, and experience level, as well as personalized job recommendations. The system also incorporates tools for communication between administrators and candidates, facilitating timely interactions.

The System is attached with many online recruitment functions and other regular functions, such as User Registration, Management of Job Post, Add Skill, Viewing applied jobs, Resumes management. The Job Portal System aspires to create a dynamic ecosystem where talent meets opportunity, fostering a more efficient, accessible, and effective hiring process for all stakeholders involved. Through continuous innovation and user-centric design, the platform aims to adapt to the evolving needs of the job market, making it an indispensable resource for both job seekers and employers.

Overall, the Job Portal System aims to enhance the job market experience by connecting talent with opportunities, fostering a more efficient and user-friendly recruitment process. As the workforce continues to evolve, such platforms play a crucial role in bridging the gap between talent and opportunity.

LIST OF FIGURES

LIST OF DATABASE FIGURES

LIST OF ABBREVIATIONS

HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
UML	Unified Modelling Language
GUI	Graphical user Interface
JS	Java Script

CHAPTER 1

INTRODUCTION

1.1 About project

Finding jobs that best suits the interests and skill set is quite a challenging task for the job seekers. The difficulties arise from not having proper knowledge on organization's objective their work culture and current job openings. Online Job search portals have certainly made job seeking convenient on both sides. "Job Portal" System is the solution where the job seeker meet aiming at fulfilling their individual requirement. They are the cheapest as well as the fastest source of communication reaching wide range of audience on just a single click irrespective of their geographical distance.

The web application "Job Search Portal" provides an easy and convenient search application for job seekers to find desired jobs. Job seekers can register with the application and update their details and skill set.

There are mainly two modules:

1. Admin 2. Candidate

In this, we have 2 logins such as for Admin, Candidate.

The admin module of the Job Portal System plays crucial role in managing the overall system The admin also has the authority to manage all the job details on the platform. They can add or remove any details as per the system's requirements. The candidate module in the job portal is designed to facilitate job search process for individuals looking for employment opportunities, keep track of application statuses and also provides access for applying jobs.

1.2 Existing System

This application having database which is a repository of an organization's electronically stored data. The databases are designed to facilitate analysis. The classic Functionality of this Application focuses on data storage. However, the means to retrieve and analyze data, to extract, transform and load data, and to manage the data dictionary. To facilitate easy maintenance of records of various Companies, job and job seekers.

Drawbacks

1.Difficult to access of all record.

2.Too difficult to match the suitable candidates to appropriate job.

3.Can't Prevent and reduce human error.

4. Increase manual work.

5. To check for details prospective jobseekers through not easy search provided in the portal.

1.3 Proposed System with Features

The proposed job portal system is designed to streamline the job search and recruitment processes, incorporating advanced features to enhance usability, efficiency, and engagement. The proposed system has designed and developed using structural system analysis design methodology life cycle. The development of the proposed system uses the open-source programming framework to further development and developed functional requirements. This has designed with interactive features to be effective. The efficient outcome will fulfill requirements of the job hunters and job providers with high demand in recruitment process.

- Key features of the proposed system include:

  1. **User Registration and Authentication**

  Job Seekers and Employers will be able to register using email or OAuth providers like Google.

  Flask-Login will be used for user authentication and password hashing.

  2. **Job Search and Filters**

  Job seekers can search for job listings using keywords, location, and other filters like job type (full-time, part-time, freelance), and salary range.

  MongoDB's powerful query system will be used to handle complex search filters and provide relevant job recommendations.

  3. **Resume Parsing**

  Job seekers can upload their resumes, which the system will automatically parse using Python-based libraries such as PyPDF2 or docx to extract key details (contact info, skills, work experience).

CHAPTER 2

LITERATURE SURVEY

Job Procurement: Old and New Ways Job seeking usually involves different ways to look for jobs such as through personal contacts, direct telephone calls to employers, job agency office, scanning online job listings, etc. Before the Internet, became widely uses as a method of seeking jobs, jobseekers spent a lot of time using various methods to look for job openings. Today, jobseekers use online methods which are very convenient and save a lot of time. The following list of methods to be the traditional (old) ways for recruitment:

1) Employment recruitment agencies
2) Job fairs
3) Advertising in the mass media such as newspapers
4) Management Consultants
5) Advertisement in television and radio
6) Existing employee contacts
7) schools, colleges or universities or student services department
8) Workers or professional referrals

These old job seeking methods are too slow, stressful, challenging and also lack quality. In addition, the applicants have to consider the cost and the amount of time to get the information they need, and other preparations they have to make. Finding all available job vacancies is a main step at in the job-seeking process. The Internet is now a powerful tool that jobseekers can use. Today, there are many sites that advertise job positions to be filled by people with certain skills in various fields. The Internet plays an important role in the area of human resource planning and development. Most planning and development organizations are now using computer technology and the Internet for staff recruitment. It should be noted that although the Internet has facilitated the process of job-seeking, it has not replaced the traditional methods, completed.

CHAPTER 3

ANALYSIS

3.1 Hardware Requirements

Processor	:	Pentium Dual Core or above
RAM	:	2 GB or more (preferably 4 GB)
Hard Disk Capacity	:	500gb or more
Monitor	:	Any monitor
Keyboard	:	Standard Keyboard
Mouse	:	Two or Three Button Mouse

3.2 Software Requirements

Operating System	:	Microsoft Windows7 64 bit or above
Scripting Language	:	Python 3.12.6
Front End	:	HTML, CSS ,JavaScript
Framework	:	Flask
Backend	:	MongoDB

3.3 Functional Requirements

Functional requirements for a job portal system detail the specific operations and features the system must support to meet user needs.

1.User Registration:

- Account Creation: Users can register as job seekers or administrators.
- Login and Logout: Secure login and logout functionalities.

2. Job Posting and Management:

- Job Posting: Admin can create, publish, and update job listings with details such as job title, description, requirements, and location.

- Job Listing Management: Admin can deactivate or delete job postings.

3. Job Search and Application:

- Search Functionality: Job seekers can search for jobs using criteria such as keywords, location and salary.
- Application Submission: Job seekers can apply for jobs by submitting resumes, cover letters, and other required documents.

4.Resume Management:

- Resume Upload: Job seekers can upload and update their resume.

3.4 Non-Functional Requirements

Non-functional requirements for a job portal system address aspects such as performance, security, and usability, ensuring the platform operates effectively and reliably.

i. Performance and Scalability: The job portal must handle high traffic volumes efficiently, with quick load times and minimal latency. It should be scalable to accommodate growing numbers of users and job postings without compromising performance.

ii. Security: The system must ensure robust protection of user data through encryption, secure authentication mechanisms, and regular security updates. Compliance with data protection regulations, such as GDPR, is essential to safeguard personal information.

iii. Usability and Accessibility: The portal should offer an intuitive and user-friendly interface to ensure ease of use for all users, including those with disabilities. This includes adherence to accessibility standards and providing a seamless experience across various devices and screen sizes.

iv. Reliability and Availability: High availability is crucial, with the system ensuring minimal downtime through effective redundancy and backup strategies. The portal should offer reliable access and performance under varying loads.

V. Security: User data, including resumes and personal information, must be encrypted both at rest and in transit and the system should implement multi-factor authentication for user accounts to enhance security.

Vi. Modularity: The architecture should support modular components, allowing for easy updates and maintenance without system downtime and Comprehensive technical and user documentation must be provided, covering system functionality.

3.5 Module Description

The modules used in this system are:

1.Admin 2. Candidate

1.Admin:

- The Admin plays a crucial role in managing overall system.
- Manages job details.
- Add or remove job details as per the requirements.

2. Candidate:

- Designed to facilitate the job search process.
- Applying for Jobs.
- keep track of application status.

CHAPTER 4

DESIGN

4.1 Block Diagram:

The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of Implementation. Main operations are to add and manage the job details.

The following figure represents the block diagram.

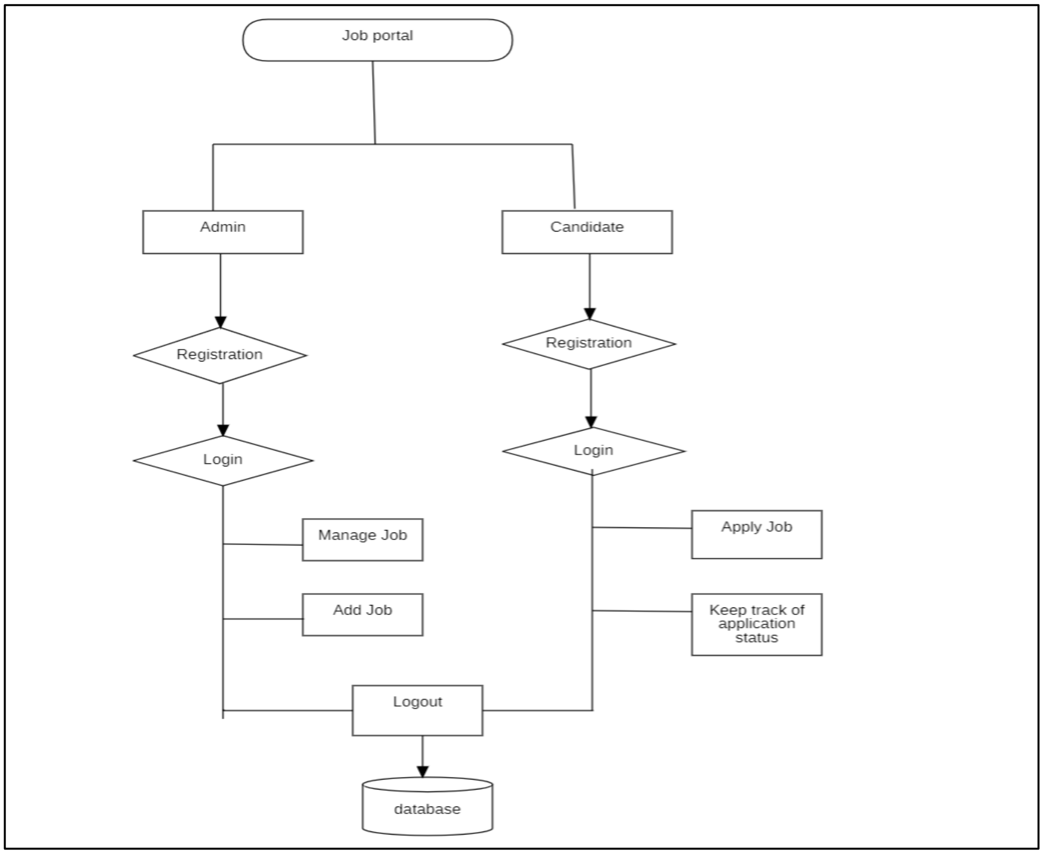


Figure 4.1 Block Diagram

4.2 Data flow diagrams:

- A data flow diagram is a graphical view of how data is processed in a system in terms of input and output.
- The Data flow diagram (DFD) contains some symbol for drawing the data flow diagram.

Data flow diagram is graphical tool used to describe and analyze movement of data through the system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, may be described logically and independently of physical components associated with the system. This is known as logical data flow diagrams.

In physical data flow diagrams each component in a DFD is labelled with a descriptive name. Process is further identified with a number that will be used for identification purpose. Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called as "context diagram".

4.2.1 DFD Symbols:

In the DFD, there are four symbols:

1. Square defines a source (orginator) or destination of data

2.An arrow identifies data flow. is the pipeline through which the information flows. Data move in a specific direction from an origin to a destination.

3.A circle or a bubble represents a process that transforms incoming data flow in to outgoing data flow.

4.An Open Rectangle is a data store, data at rest or a temporary repository of data.

Data Flow Diagram symbol:

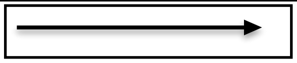
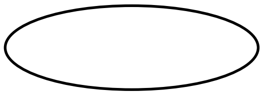
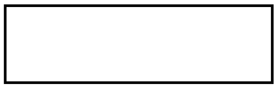
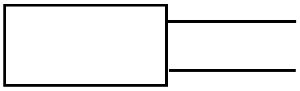
Symbol	Description
	**Data Flow:** Data flow are pipelines through the packets of information flow.
	**Process:** A Process or task performed by the system.
	**Entity:** Entity is object of the system. A source or destination data of a system.
	**Data Store:** A place where data to be stored.

Table 4.2.1 Data flow Diagram symbols

Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

- Process should be named and numbered for an easy interface. Each name should be representative
- of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flow from

  source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source.
- An alternative way is to repeat the source symbol as a destination. Since it is used more than
- once in the DFD it is marked with a short diagonal.
- When a process is exploded into lower-level details, they are numbered.

- The names of data stores and destinations are written in capital letters. Process and data flow names have the first letter of each work capitalized.

4.2.2 Context Diagram

It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

A DFD is also known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data base System.

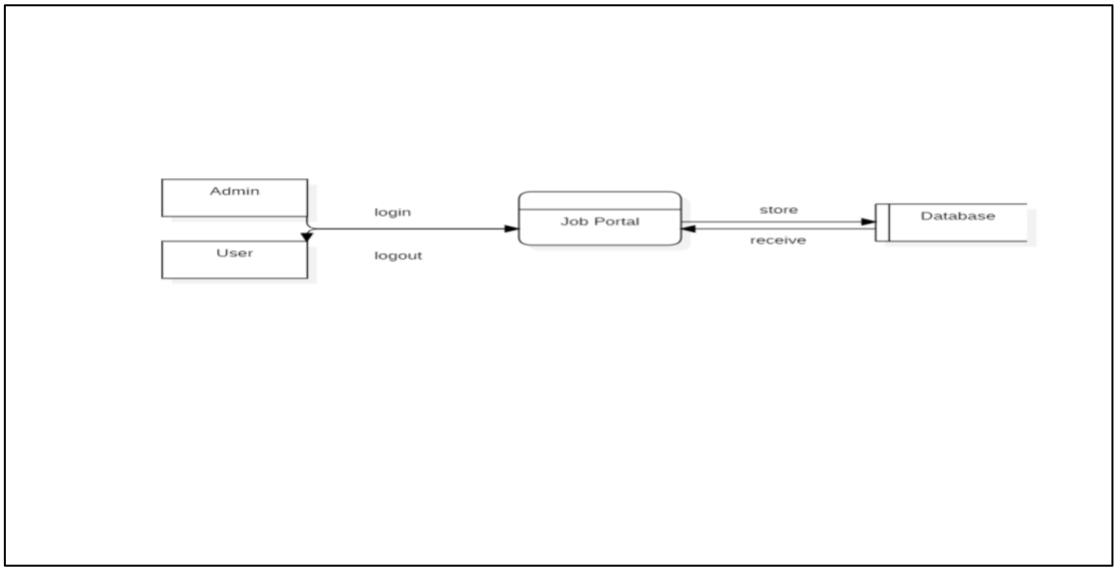


Figure 4.2.2.1 Zero level DFD

Admin DFD

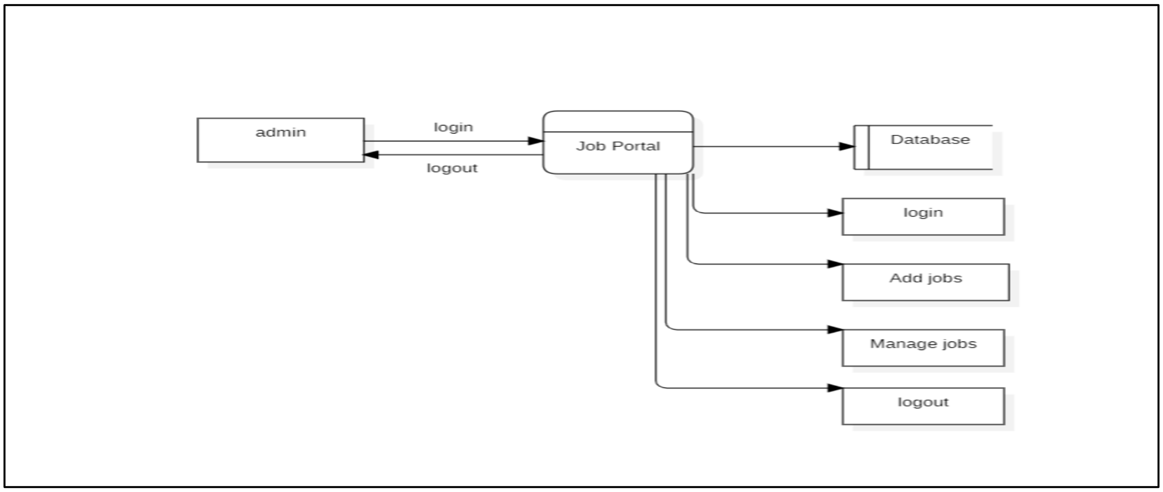


Figure 4.2.2.2 Level-1 DFD (Admin)

User DFD

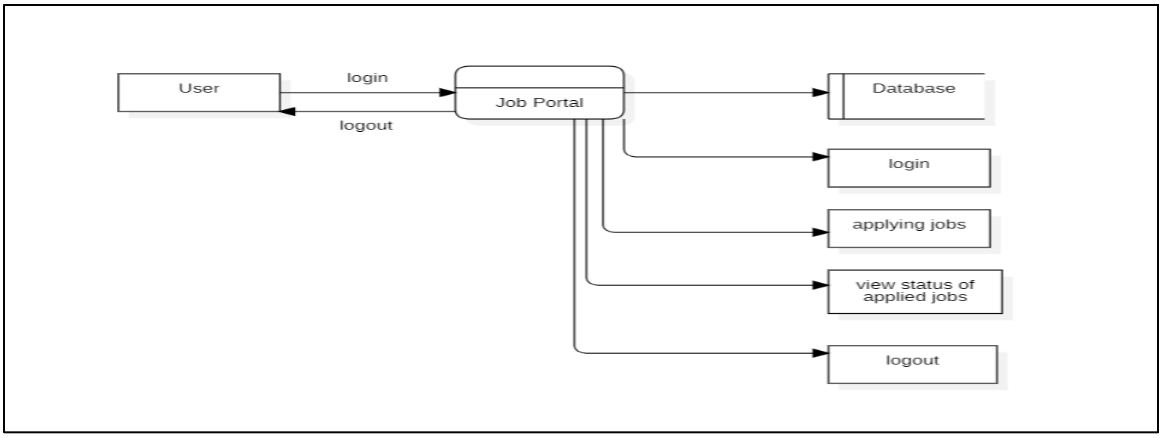


Figure 4.2.2.3 Level-1 DFD (User)

4.3 UML Diagrams

The Unified Modelling Language (UML) is a Standard language for specifying, Visualizing, constructing and documenting the software system and its components. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. Structural models represent the framework for the system and this framework is the place where all other components exist. So, the class diagram, component diagram and deployment diagrams are the part of structural modelling. They all represent the elements and the mechanism to assemble them. But the structural model never describes the dynamic behaviour of the system. Behavioural model describes the interaction in the system. It represents the interaction among the structural diagrams. Behavioural modelling shows the dynamic nature of the system. Architectural model represents the overall framework of the system. It contains both structural and behavioural elements of the system. Architectural model can be defined as the blueprint of the entire system. Package diagram comes under architectural modelling.

There are various kinds of methods in software design. They are as follows:

1. Use Case Diagram
2. Class Diagram
3. Sequence Diagram
4. Activity Diagram
5. Collaboration Diagram

4.3.1 Use Case Diagram

Use case diagrams are one of the five diagrams in the UML for modelling the dynamic Aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modelling the dynamic aspects of systems). Use case diagrams are central to modelling the behaviour of the system, a sub-system, or a class.

Each one shows a set of use cases and actors and relations.

The key points are:

- The main purpose is to show the interaction between the use cases and the actor.
- To represent the system requirement from the user's perspective.
- Use cases are the functions that are to be performed in the module.

In Figure 4.3.1.1, the admin use case diagram portrays the admin as the central actor. Admins can securely log into the system using their credentials, granting them multifaceted capabilities. They are empowered to add new jobs, and also manages the jobs by performing operations such as delete and update.

Additionally, admins play a crucial role in system enhancement by reviewing and considering user requirements. Overall, it encapsulates the diverse responsibilities and control wielded by admins to ensure the system's robustness and responsiveness to user needs.

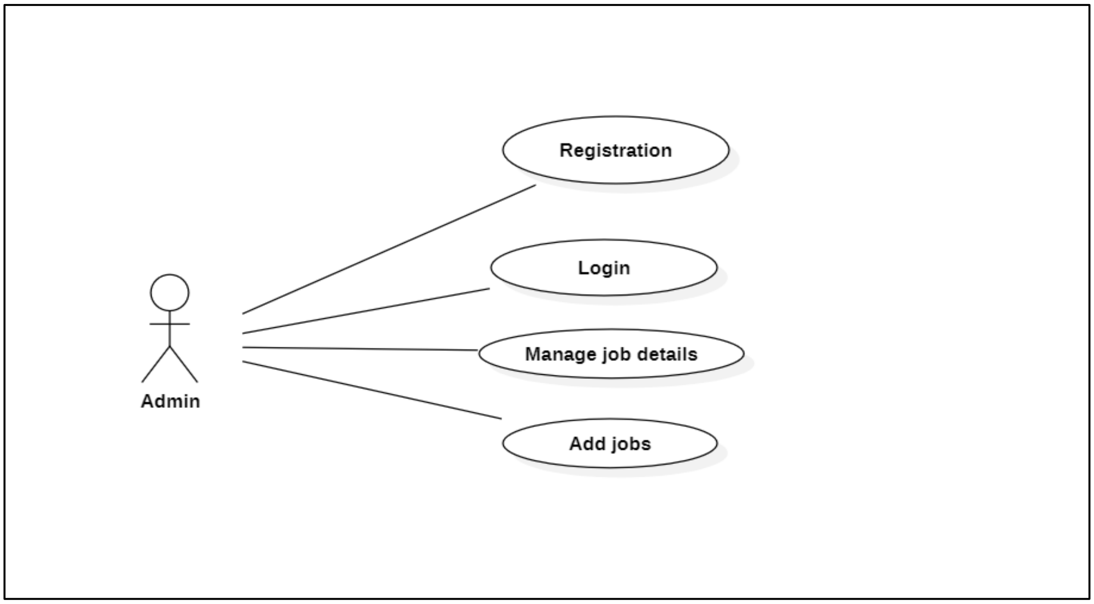


Figure 4.3.1.1 Admin Use case Diagram

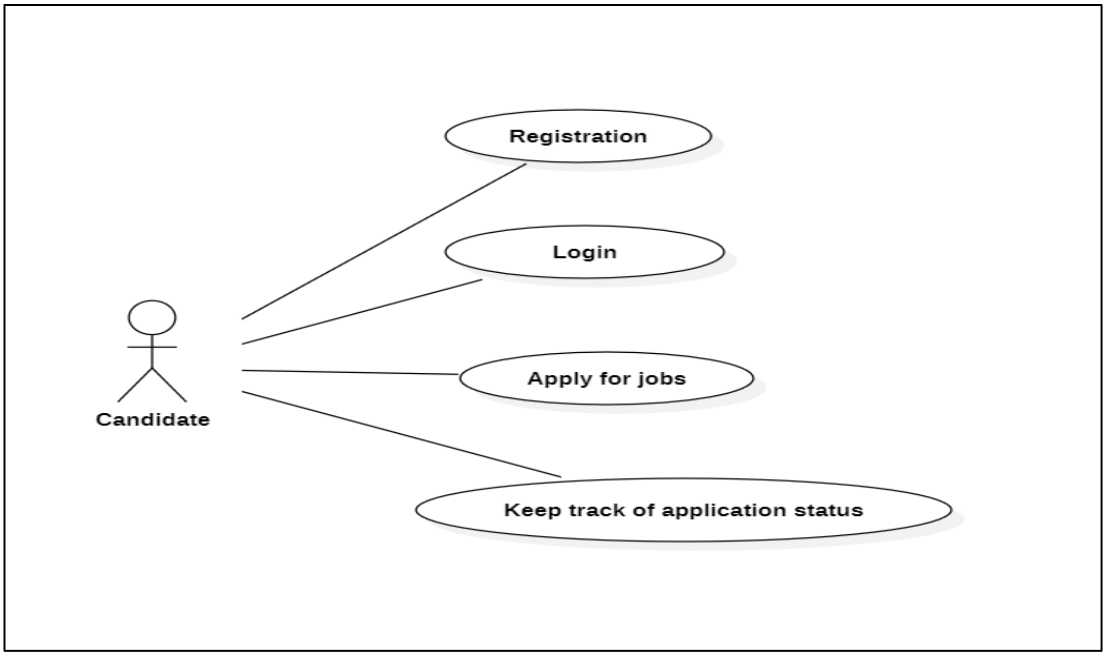


Figure 4.3.1.2 Candidate Use case Diagram

The above Figure 4.3.1.2 Shows the can Candidate Use case Diagram. The actor in the following diagram is Candidate. Candidates can view information about the website to gain insights. They have the option to sign up and create accounts, followed by secure login access. Once logged in, Candidates can explore the homepage and view dashboards. Additionally, users can apply the jobs and view the status of the applied jobs. Finally, users can logout when they've completed their interactions with the website.

4.3.2 Class Diagram

A "Class Diagram" shows a set of classes, interfaces and collaborations and their Relationships. These diagrams are the most common diagrams in modelling object-oriented systems. The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the Constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams which can be mapped directly with object-oriented languages.

The below Figure 4.3.2 shows the methods that are used in this class diagram and the Description of each class is listed below.

1.**Admin:** This class is used for adding and updating or deleting the jobs.

2.**Job:** This class represents the position, company, description, location, vacancies so on
Where admin requires these details to post the jobs.

3.**Candidate:** This class inherits from User class and used for applying the jobs and can view the applied status of jobs.

4.**Application:** This class represents the job position, company, status.

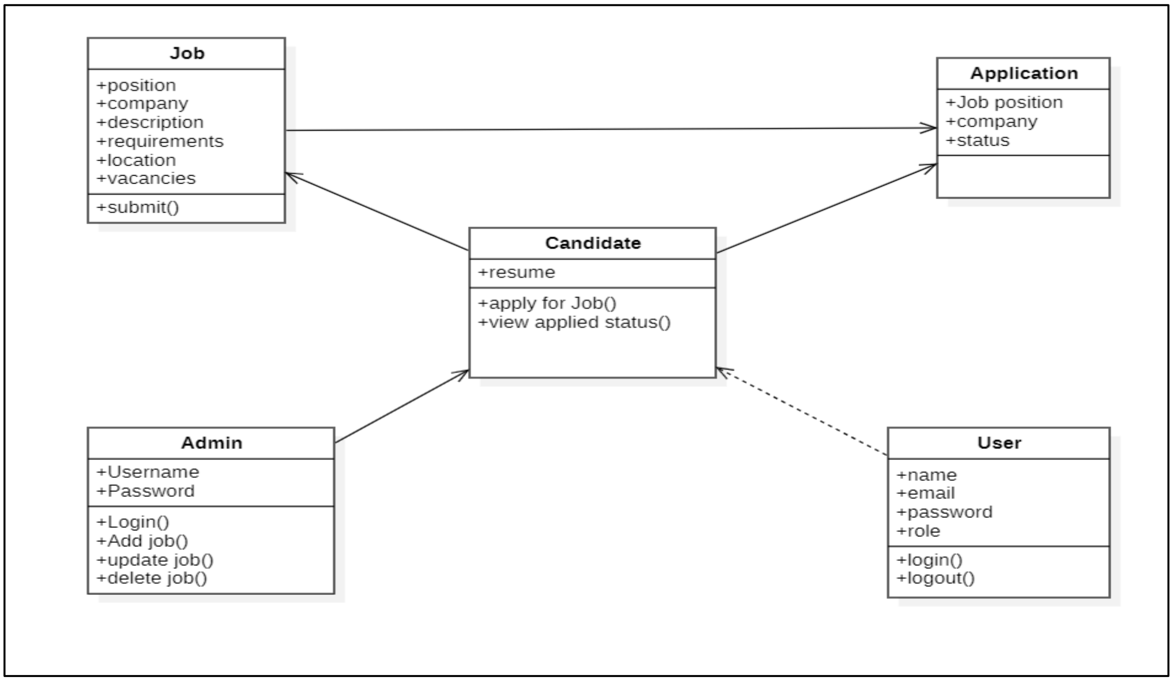


Figure 4.3.2 Class Diagram

4.3.3 Sequence Diagram

Sequence diagram is an interaction diagram which focuses on the time ordering of messages. It shows a set of objects and messages exchanged between these objects. This diagram illustrates the dynamic view of a system.

The key points are:

1. The main purpose is to represent the logical flow of data with respect to a process.
2. A sequence diagram displays the objects and not the classes.

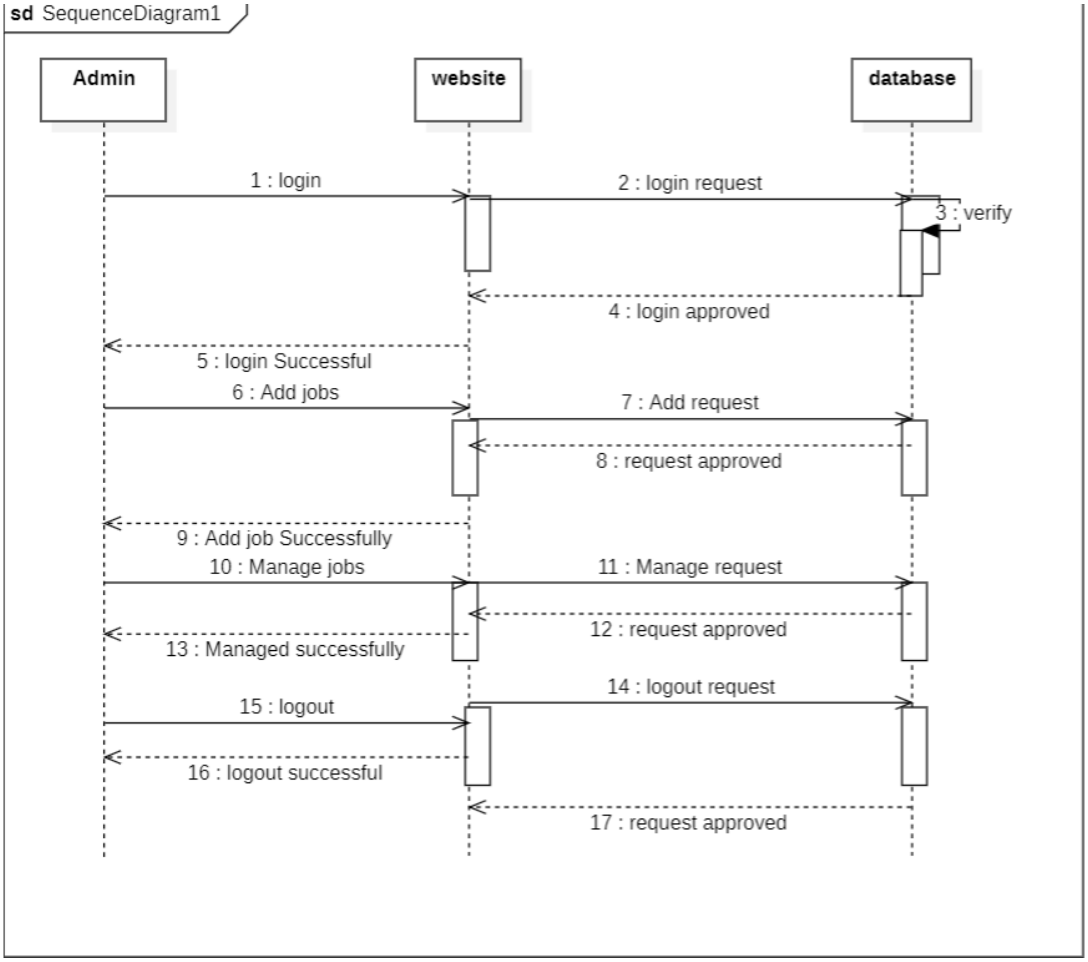


Figure 4.3.3.1 Admin Sequence Diagram

The above Figure 4.3.3.1 represents admin sequence diagram. This diagram shows the steps involved in the admin login process. The admin first logs in to the system, and the database verifies the credentials. Upon successful verification, the admin can add and manage the jobs properly.

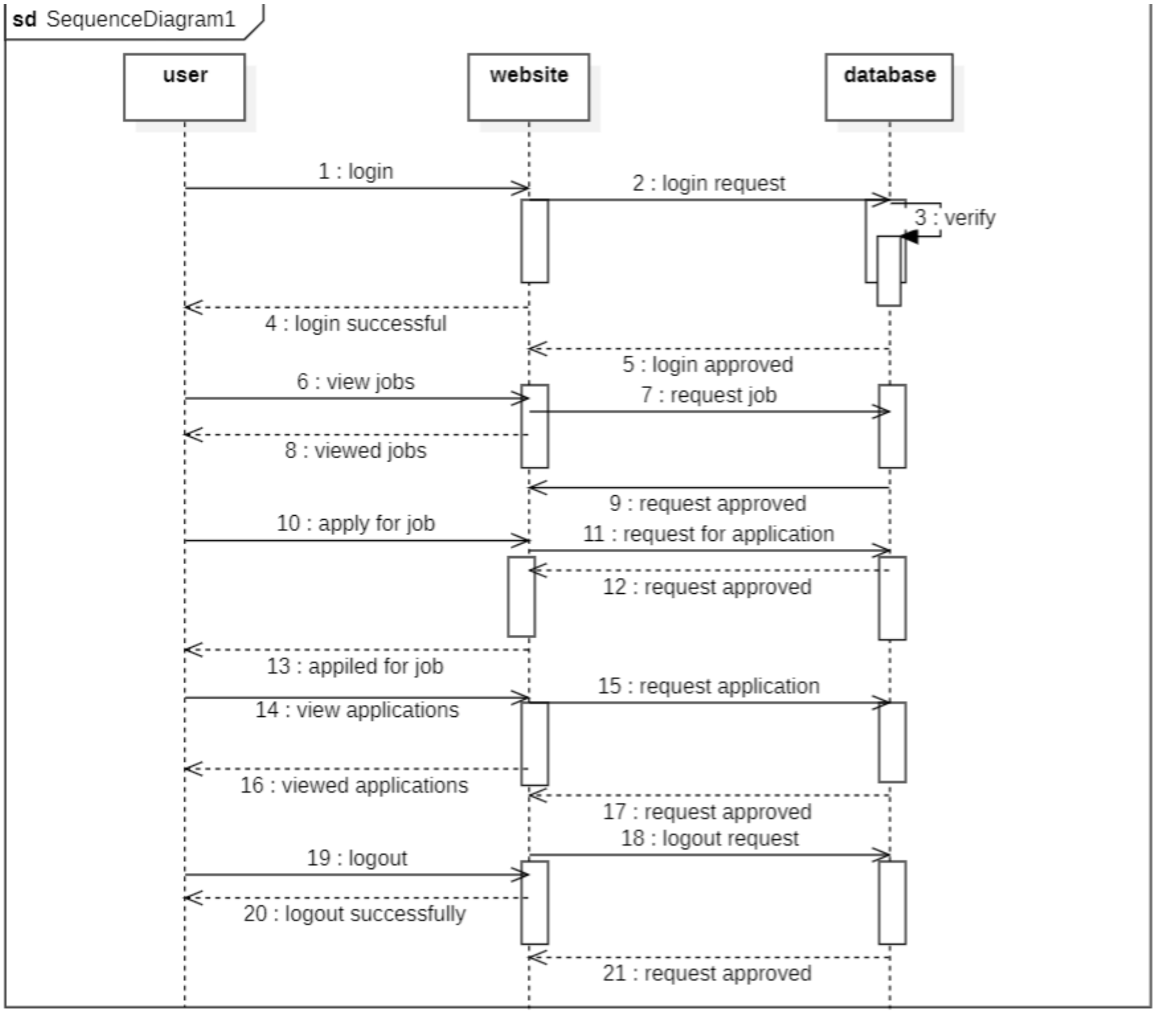


Figure 4.3.3.2 User Sequence Diagram

The above Figure 4.3.3.2 represents user sequence diagram. This sequence diagram shows the steps involved in the user login process. Users first create an account on the website, and the database receives their signup requests. After successfully completing their signup, users can log in with their credentials. The database also receives their login requests. Upon successful login, users can apply the jobs and can keep track of their application status.

4.3.4 Activity Diagram

Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as a rounded box containing the name of operation. This activity diagram describes the behavior of the system.

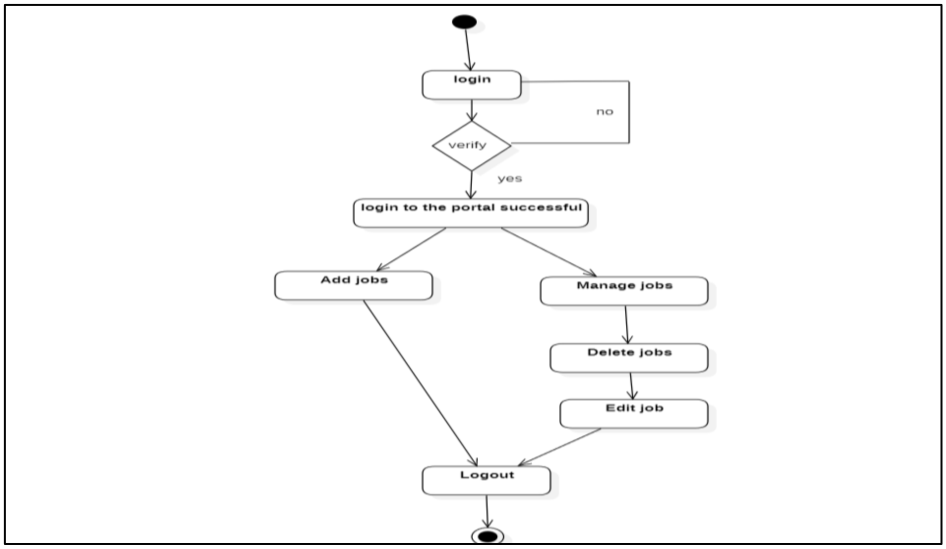


Figure 4.3.4.1 Admin Activity Diagram

The above Figure 4.3.4.1 represents the admin activity diagram. The admin first logs in to the website. Then they have authentication to access the website, where they can perform operations, such as adding and managing the jobs. After performing the operations, then admin exit from the website.

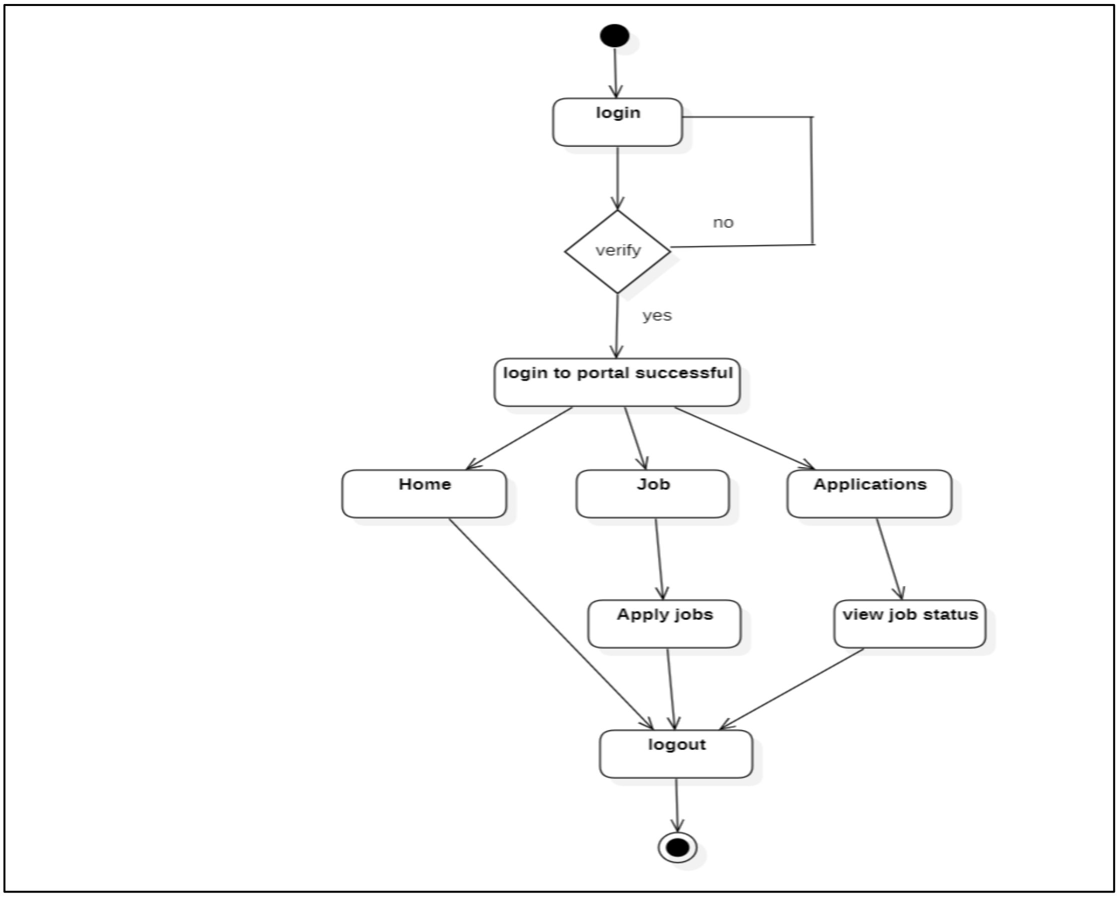


Figure 4.3.4.2 User Activity Diagram

Figure 4.3.4.2 illustrates the User Activity Diagram, depicting the sequential actions users take on the website. Users initiate the process by signing up, sending signup requests to the database. Upon successful signup, they can log in using credentials. Users can further contribute to system improvement by providing suggestions to the admin based on their experiences.

4.3.5 Collaboration Diagram

A collaboration diagram is an introduction diagram that emphasizes the structural organization of the objects that send and receive messages. Graphically a collaboration diagram is a collection of vertices and arcs.

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. It is also called a communication diagram. Collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other. Collaboration diagrams convey the same information as sequence diagrams, but they focus on object roles instead of the times that messages are sent. In the sequence diagram, object roles are the vertices and messages are the connecting links. The object-role rectangles are labelled with either class or object names (or both). Class names are preceded by colons (:). Each message in the collaboration diagram has a sequence number. The top-level message is numbered 1. Messages at same level have the same decimal prefix but suffixes of 1, 2, etc.

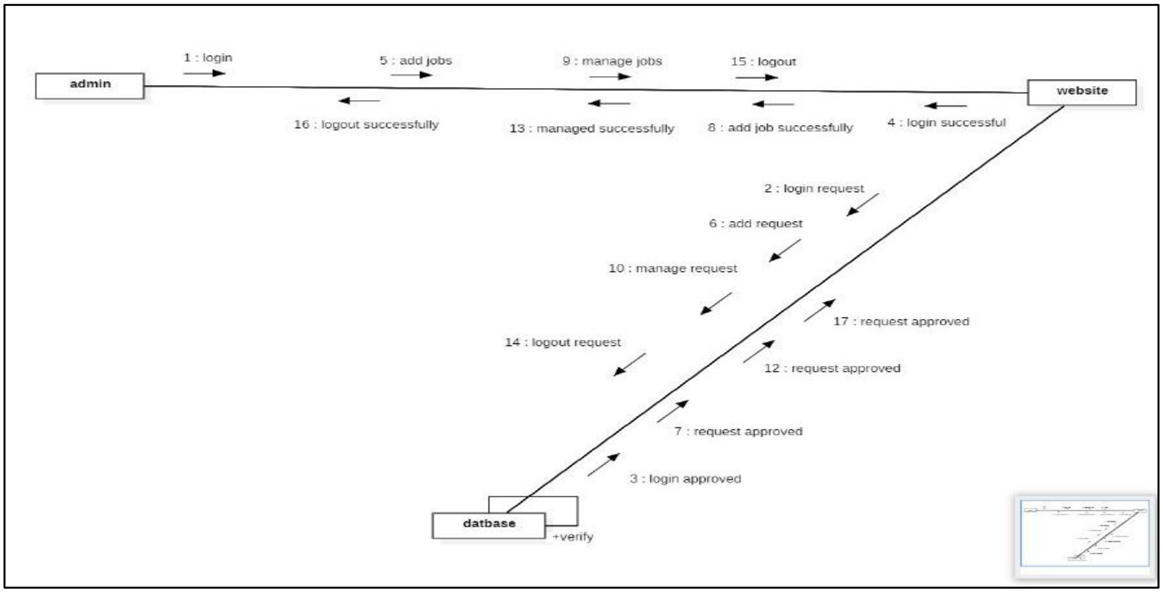


Figure 4.3.5.1 Admin Collaboration diagram

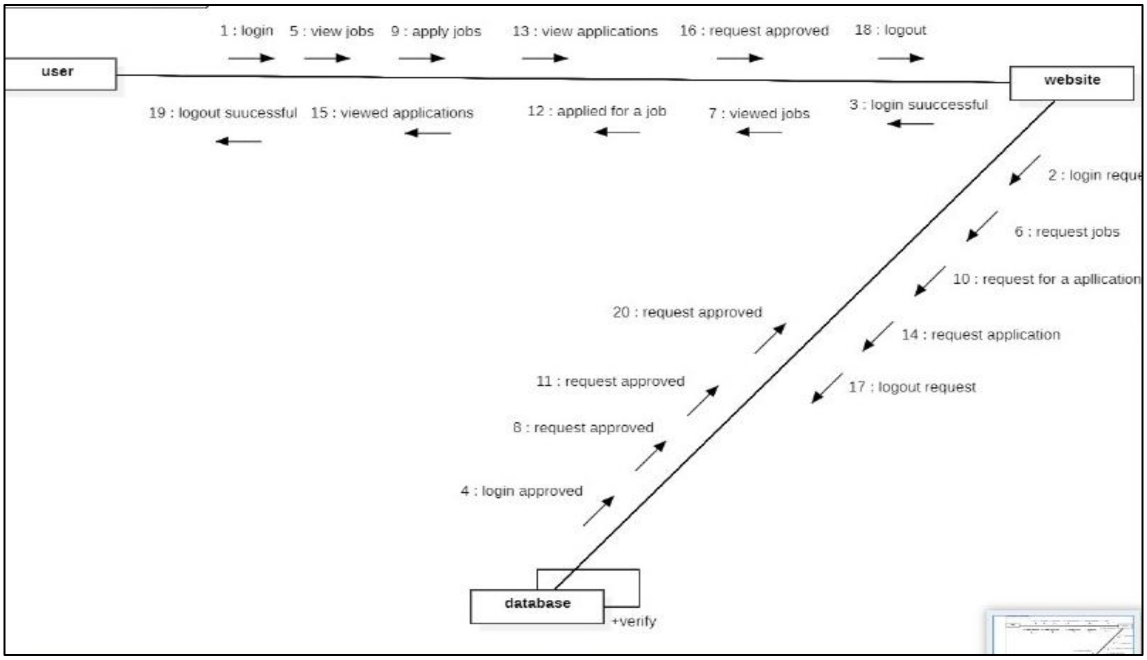


Figure 4.3.5.2 User Collaboration Diagram

The figures 4.3.5.1 and 4.3.5.2 show the collaboration diagrams of admin and user. In the case of admin, the admin, website, and database are the objects. The admin logs in to the application, and the login information is stored in the database. The admin’s actions are stored in the database so that the information can be retrieved later. In the case of user, the user, website, and database are the objects. Users first sign up for the application, and their registration information is stored in the database. After logging in, users have access to the website

4.4 Data Dictionary

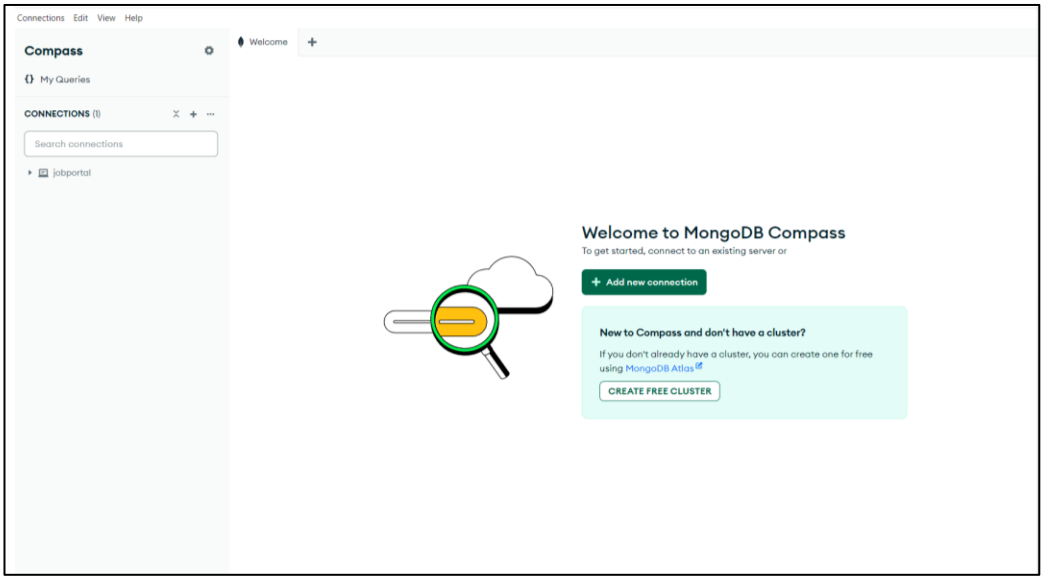
A database dictionary, or Metadata Repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format”. The term can have one of several closely related meanings pertaining to databases and database management systems (DBMS).

- A document describing a database or collection of databases.
- An integral component of a DBMS that is required to determine its structure.
- A piece of middleware that extends or supplants the native data dictionary of a DBMS.

The terms data dictionary and data repository indicate a more general software utility than a catalogue. A catalogue is closely coupled with the DBMS software. It provides the information stored in it to the user and the DBA, but it is mainly accessed by the various software modules of the DBMS itself, such as DDL and DML compilers, the query optimizer, the transaction processor, report generators, and the constraint enforcer. On the other hand, a data dictionary is a data structure that stores metadata, i.e., (structured) data about information. The software package for a stand-alone data dictionary or data repository may interact with the software modules of the DBMS, but it is mainly used by the designers, users and administrators of a computer system for information resource management. These systems maintain information on system hardware and software configuration, documentation, application and users as well as other information relevant to system administration. If a data dictionary system is used only by the designers, users, Otherwise, it is called an active data dictionary or data dictionary. When a passive data dictionary is updated, it is done so manually and independently from any changes to a DBMS (database) structure. With an active data dictionary, the dictionary is updated first and changes occur in the DBMS automatically as a result.

4.4.1 Data figures

Database (MONGODB):



Welcome to MongoDB Compass

To get started, connect to an existing server or

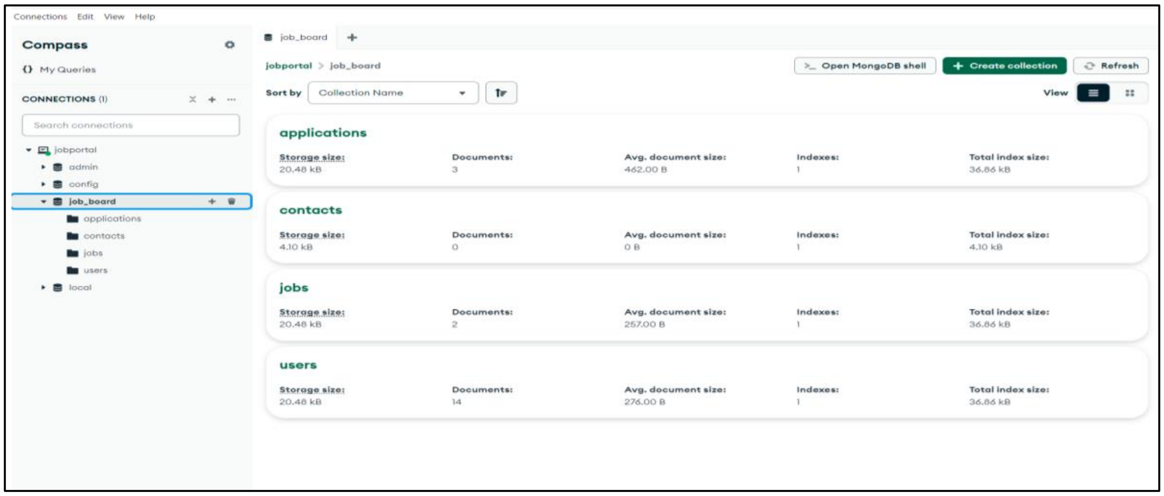
New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using MongoDB Atlas

Fig 4.5.1.1 Screenshot of MONGODB Interface

Database Details:

- **Name:** DB Details.
- **Description:** This figure shows the details about the database.

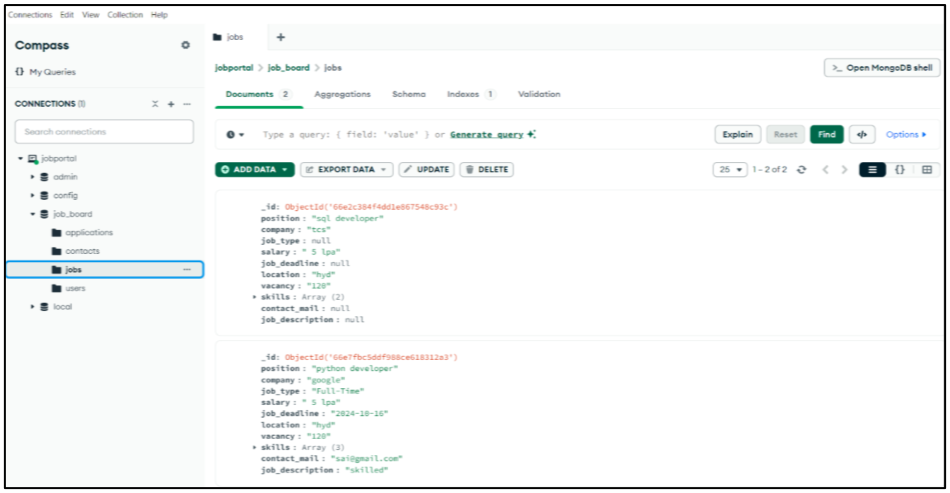


	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
applications	20.48 kB	3	462.00 B	1	36.86 kB
contacts	4.10 kB	0	0 B	1	4.10 kB
jobs	20.48 kB	2	257.00 B	1	36.86 kB
users	20.48 kB	14	276.00 B	1	36.86 kB

Figure 4.5.1.2 DB Details

Job Details:

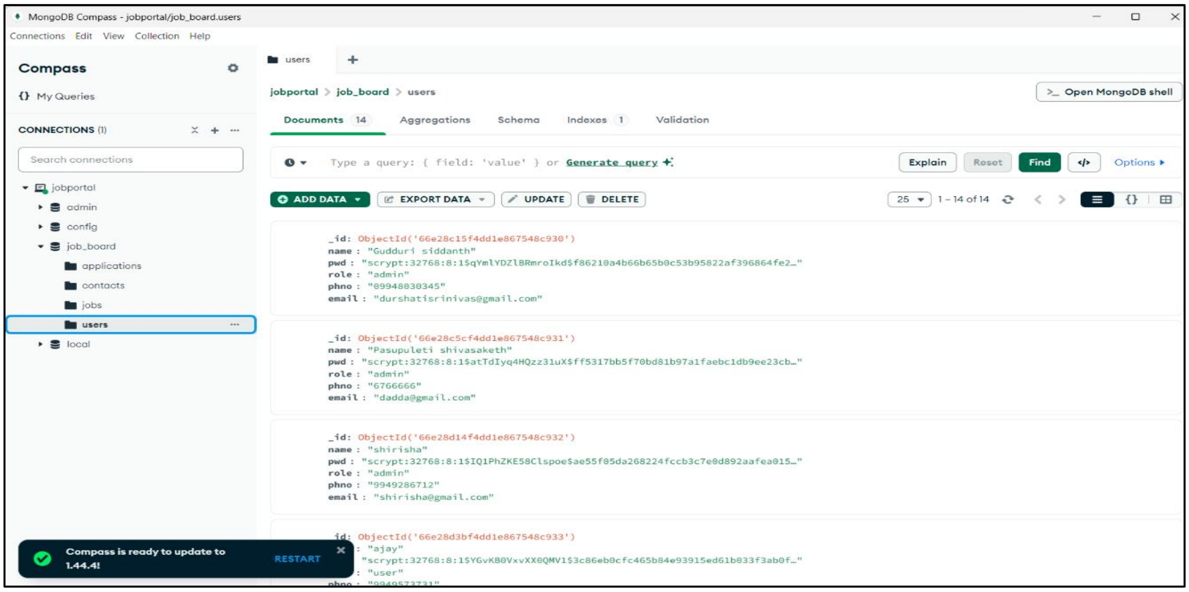
- **Name:** Job Details
- **Description:** This figure shows the information about Jobs. Admin can also view the details and can update it.



```
_id: ObjectId('66e2c384f4dd1e867548c93c')
position : "sql developer"
company : "tcs"
job_type : null
salary : " 5 lpa"
job_deadline : null
location : "hyd"
vacancy : "120"
skills : Array (2)
contact_mail : null
job_description : null
```

```
_id: ObjectId('66e7fbc5ddf988ce618312a3')
position : "python developer"
company : "google"
job_type : "Full-Time"
salary : " 5 lpa"
job_deadline : "2024-10-16"
location : "hyd"
vacancy : "120"
skills : Array (3)
contact_mail : "sai@gmail.com"
job_description : "skilled"
```

Figure 4.5.1.3 Job Details DB



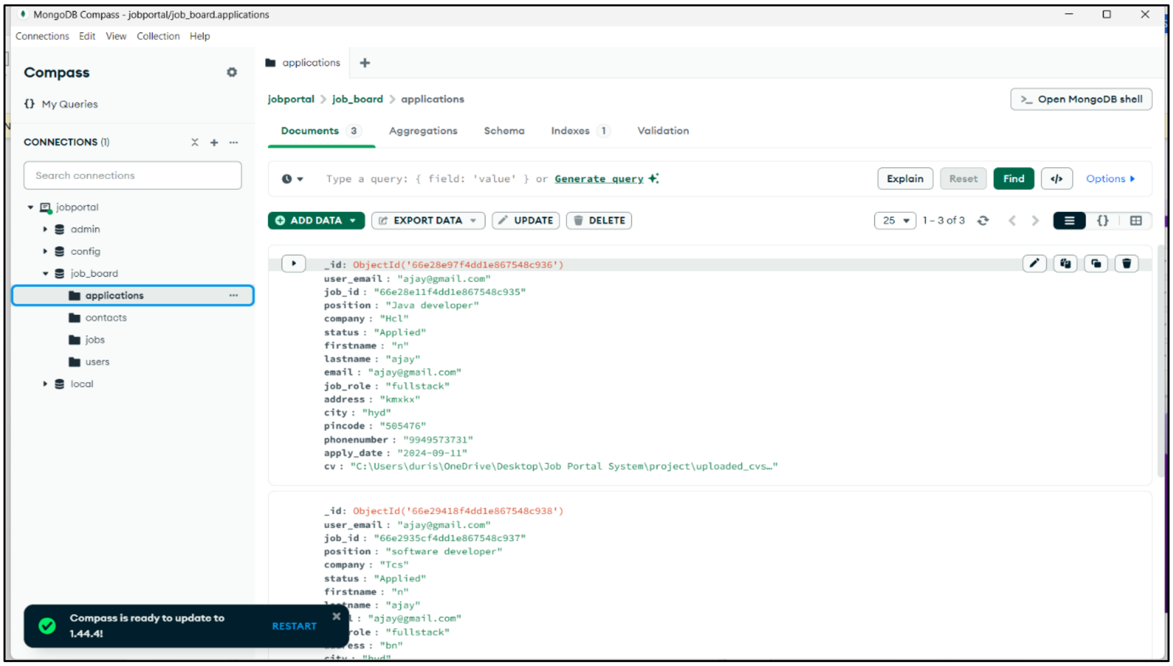
```
_id: ObjectId('66e28c15f4dd1e867548c930')
name : "Gudduri siddanth"
pwd : "scrypt:32768:8:1$qYmlYDZlBRmroIkd$f86210a4b66b65b0c53b95822af396864fe2…"
role : "admin"
phno : "09948030345"
email : "durshatisrinivas@gmail.com"

_id: ObjectId('66e28c5cf4dd1e867548c931')
name : "Pasupuleti shivasaketh"
pwd : "scrypt:32768:8:1$atTdIyq4HQzz31uX$ff5317bb5f70bd81b97a1faebc1db9ee23cb…"
role : "admin"
phno : "6766666"
email : "dadda@gmail.com"

_id: ObjectId('66e28d14f4dd1e867548c932')
name : "shirisha"
pwd : "scrypt:32768:8:1$IQ1PhZKE58Clspoe$ae55f05da268224fccb3c7e0d892aafea015…"
role : "admin"
phno : "9949286712"
email : "shirisha@gmail.com"

 id: ObjectId('66e28d3bf4dd1e867548c933')
     : "ajay"
      "scrypt:32768:8:1$YGvKB0VxvXX0QMV1$3c86eb0cfc465b84e93915ed61b033f3ab0f…"
     : "user"
phno : "9949573731"
```

Figure 4.5.1.4 User Details DB



```
_id: ObjectId('66e28e97f4dd1e867548c936')
user_email : "ajay@gmail.com"
job_id : "66e28e11f4dd1e867548c935"
position : "Java developer"
company : "Hcl"
status : "Applied"
firstname : "n"
lastname : "ajay"
email : "ajay@gmail.com"
job_role : "fullstack"
address : "kmxkx"
city : "hyd"
pincode : "505476"
phonenumber : "9949573731"
apply_date : "2024-09-11"
cv : "C:\Users\duris\OneDrive\Desktop\Job Portal System\project\uploaded_cvs…"

_id: ObjectId('66e29418f4dd1e867548c938')
user_email : "ajay@gmail.com"
job_id : "66e2935cf4dd1e867548c937"
position : "software developer"
company : "Tcs"
status : "Applied"
firstname : "n"
  tname : "ajay"
    l : "ajay@gmail.com"
   role : "fullstack"
    ess : "bn"
city : "hyd"
```

Figure 4.5.1.5 Application Details DB

CHAPTER 5
IMPLEMENTATION

5.1 Technology Used

- **Backend**

PYTHON

Python is a popular high-level programming language that was first released in 1991 by Guido Rosum. It is a versatile language that can be used for a variety of purposes, including web van development, data analysis, scientific computing, artificial intelligence, and more. Python is known for its clean syntax, readability, and ease of use, which makes it a popular choice among beginners and experienced developers alike. It also has a large and active community of developers who contribute to its development and maintenance.

Python supports multiple programming paradigms, including object-oriented, functional, and procedural programming. It also has a vast standard library that provides a wide range of modules for various purposes, such as web development, data analysis, and networking. One of the main reasons why Python is so popular is its flexibility and yersatility. It is widely used in many industries and has become the language of choice for many data scientists and machine learning engineers. It is also widely used in education and it is a great language for starting learning programming.

Features of python:

- Object-oriented
- Large standard library
- High-level Language
- Third-party libraries

Object-oriented:

Python supports object-oriented programming, which makes it easy to create reusable and modular code. It allows developers to define classes and functions, and organize code into logical units.

Large standard libraries:

Python comes with a comprehensive standard library that provides developers with a wide range of modules and functions to accomplish common programming tasks.

High-level Language:

Python is considered a high-level language because it provides a level of abstraction that simplifies programming and enhances developer productivity. Its syntax is designed to be readable and intuitive, allowing developers to write code more easily and maintain it more effectively. It employs dynamic typing, meaning variable types are determined at runtime rather than at compile time, which reduces the complexity of code. The language also includes a rich standard library with a wide range of modules and functions. Additionally, with built-in high-level data structures like lists and dictionaries.

Third-party libraries:

Python has a rich ecosystem of third-party libraries that greatly extend its functionality beyond the built-in features. These libraries, available through repositories like the Python Package Index (PyPI), can be easily installed using tools like pip. They cover a wide range of applications, from web development with frameworks like Django and Flask to data analysis with Pandas and NumPy.

VISUAL STUDIO

Visual Studio is a comprehensive integrated development environment (IDE) created by Microsoft. It is widely used for developing a variety of applications, from desktop and web applications to mobile apps and cloud-based solutions.

Key Features of Visual Studio

1. **Rich IDE Experience:**
   - Code Editor: Visual Studio offers a powerful code editor with features such as IntelliSense (code completion), syntax highlighting, code refactoring, and live code analysis.
   - Debugging Tools: The IDE provides advanced debugging capabilities, including breakpoints, watch windows, call stacks, and live debugging. You can debug both managed and native code.
2. **Project and Solution Management:**
   - Solution Explorer: This tool helps manage project files and dependencies, allowing for easy navigation and organization of your codebase.
   - Project Templates: Visual Studio includes numerous project templates for different types of applications, such as ASP.NET web apps, desktop apps, and mobile apps.
3. **Version Control Integration:**
   - Git Integration: Visual Studio supports Git, enabling version control directly within the IDE. It provides tools for managing repositories, committing changes, and resolving conflicts.
   - Team Foundation Version Control (TFVC): For teams using Microsoft's version control system, Visual Studio integrates with TFVC for source control management.
4. **Extensibility:**
   - Extensions and Add-ons: Visual Studio has a rich ecosystem of extensions available through the Visual Studio Marketplace. These extensions can add new features, integrate with other tools, and customize the IDE to fit specific needs.
   - Customizable UI: You can customize the IDE’s appearance and behaviour, including tool windows, keyboard shortcuts, and themes.

5. **Language Support:**
   - Multilingual Support: Visual Studio supports a wide range of programming languages, including C#, C++, Visual Basic, F#, Python, JavaScript, and more.

FLASK

Flask is a lightweight and flexible web framework for Python, designed to facilitate the creation of web applications with minimal overhead. It follows the WSGI (Web Server Gateway Interface) standard, allowing developers to build web apps with simplicity and ease. Flask is often praised for its simplicity and modularity, giving developers the freedom to choose their tools and libraries. Unlike more opinionated frameworks, Flask provides only the essentials to get a web server up and running, while leaving the rest to the developer's discretion. This makes it a popular choice for small to medium-sized projects and for those who prefer a more hands-on approach to web development.

Flask is widely used in web development for creating dynamic web applications with Python due to its simplicity and flexibility. It allows developers to quickly set up a web server and define routes to handle HTTP requests, making it ideal for building everything from small prototypes to complex applications. Flask's minimalistic design means it doesn't impose a specific project structure, enabling developers to customize their setup according to their needs. Its extensible nature supports a wide range of plugins and libraries, making it easy to integrate features such as authentication, database interactions, and form validation. Flask a popular choice for developers looking to build robust and scalable web application.

MONGODB

MongoDB: A NoSQL Database

MongoDB is a popular open-source, document-oriented NoSQL database. Unlike traditional relational databases that store data in tables with rows and columns, MongoDB stores data in flexible JSON-like documents. This makes it well-suited for handling complex, semi-structured data and rapidly changing data structures.

Key Features of MongoDB:

Document-Oriented: Stores data in flexible JSON-like documents, making it easy to model complex data structures.

Schema-Less: While it supports optional schemas, MongoDB allows for dynamic data structures, making it adaptable to changing requirements.

Indexing: Provides efficient indexing mechanisms for querying and retrieving data.

Replication: Ensures data durability and availability through replication across multiple nodes.

Sharding: Scales horizontally by distributing data across multiple servers, handling large datasets.

Aggregation Framework: Offers powerful aggregation pipelines for complex data analysis and reporting.

Full-Text Search: Enables full-text search capabilities for text-based data.

Use Cases for MongoDB:

E-commerce: Storing product information, customer data, and order details.

Real-time Analytics: Processing and analyzing large volumes of data in real-time.

IoT (Internet of Things): Storing sensor data and time-series data.

Gaming: Managing user data, game state, and leaderboards.

Social Media: Storing user profiles, posts, and interactions.

Advantages of MongoDB:

Flexibility: Easily adapts to changing data structures.

Scalability: Handles large datasets and high traffic.

Performance: Provides efficient query performance.

ACID Transactions: Supports multi document transactions for data integrity.

Integration with Big Data: Compatible with frameworks like Hadoop and Spark.

5.2 Frontend Design

- **Front end**

HTML

HTML is a hypertext mark-up language that is in reality a backbone of any website. Every website can't be structured without the knowledge of HTML. If we make our web page only with the help of HTML, then we can't add many of the effective features in a web page, for making a webpage more effective we use various platforms such as CSS. So here we are using this language to make our web pages more effective as well as efficient. And to make our web pages dynamic we are using JavaScript.

CSS

CSS Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that 35 previously could only be defined in a page's HTML. The basic purpose of CSS is to separate the content of a web document (written in any mark-up language) from its presentation (that is written using Cascading Style Sheets CSS gives the option of selecting various style schemes and rules according to the requirements and it also allows the same HTML document to be presented in more than one varying style.

JavaScript

JavaScript is considered to be one of the most famous scripting languages of all time. JavaScript, by definition, is a Scripting Language of the World Wide Web. The main usage of JavaScript is to add various Web functionalities, Web form validations, browser detections, creation of cookies, and so on. JavaScript is one of the most popular scripting languages and that is why it is supported by almost all web browsers available today like Firefox.

We used the browser Opera or Internet Explorer. JavaScript is considered to be one of the most powerful scripting languages in use today. It is often used for the development of client-side web development. JavaScript is used to make web pages more interactive and dynamic.
JavaScript is a lightweight programming language, and it is embedded directly into the HTML code. JavaScript, as the name suggests, was influenced by many languages, especially Java.

The advantages of JavaScript: -

An interpreted language.
Embedded within HTML.
The minimal syntax is easy to learn
Performance sign for simple, small programs.
Client-Side Scripting.
Cross-Platform Compatibility.
Lightweight and Efficient.

CHAPTER 6
TESTING

Introduction:

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding. System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for member acceptance testing. A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected. Software testing is the process of testing the functionality and correctness of software by running it. Process of executing a program with the intent of finding an error.

Verification:

Verification is the process to make sure that product satisfies the conditions imposed at the tester of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation:

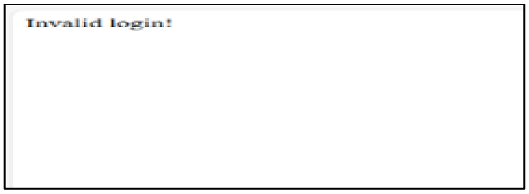
Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

6.1 Testing:

Testing Member Login Page:

1. Figure 6.1 represents “Member Login Failed” when he/she enters the wrong password. The expected output is to show “Invalid login details”.

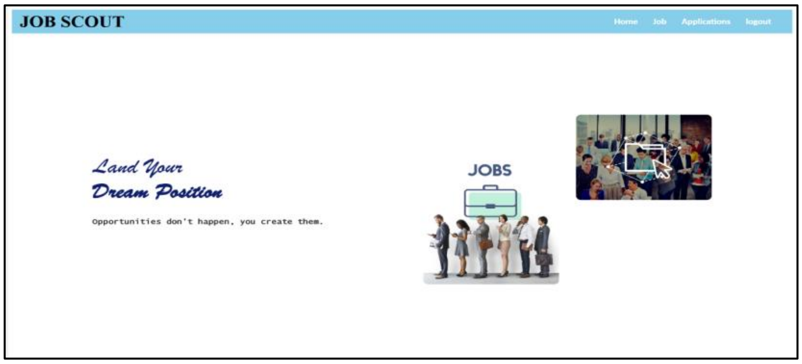
2. Figure 6.2 shows “Successful Login of member”.



Invalid login!

6.1 Screenshot of Member login failed

Figure 6.1 shows attempting login by leaving username, password fields empty. The expected output is “Invalid login!”. Similarly, shows trying to login with username and leaving the password empty. The expected output is to show invalid login.



Land Your
Dream Position

Opportunities don't happen, you create them.

6.2 Screenshot of Member login successful

Figure 6.2 shows trying to login with username and password. The expected output is showing if username and password are valid.

Test No	Data Input	Expected Output	Actual Output	Pass/Fail
1	All empty fields	Please fill out this field	Please fill out this field	Pass
2	Username and leaving password Empty	Please fill out this field	Please fill out this field	Pass
3	Correct username and password	Successful login	Successful login	Pass
4	Incorrect password	Invalid login details	Invalid login details	Pass

6.2 Test Strategy:

Field-testing is performed manually, and functional testing is written in detail.

Test Objectives

1.All field entries must work properly.

2.Pages must be activated from the identified link.

3.The entry screen, messages and responses must not be delayed.

4.Features to be tested.

5.Verify that entries are in correct format.

6.No duplicate entries should be allowed.

6.3 Black Box Testing:

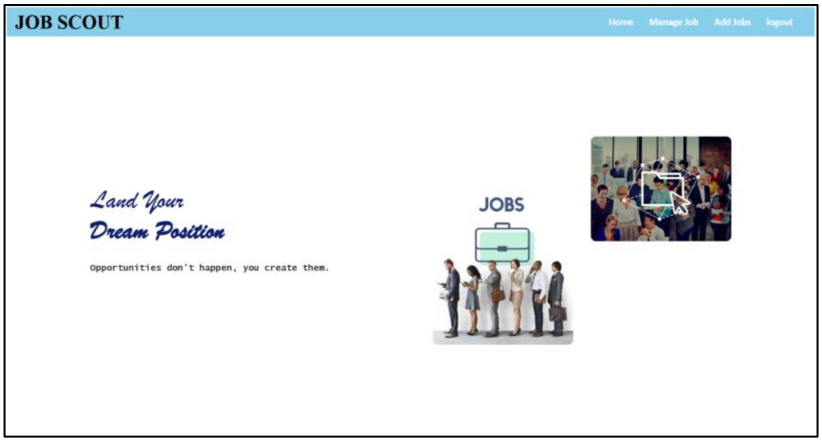
The base of the black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system. Now a days, it is becoming to route the testing work to a third party as the developer of the system knows too much of the internal logic and coding of the system which makes it unfit to rest application by the developer.

Advantages of Black Box Testing

- Used to verify contradictions in actual system and specifications.
- Test cases can be designed as soon as the functional specifications are complete.
- Well suited and No need of technical knowledge.
- Efficient for large code segments.

Disadvantages of Black Box Testing

- The test input needs to be from a large sample space.
- Chances of having unidentified paths during this testing.
- Test cases are challenging to design without having clear functional specifications
- It is a high probability of repeating tests already performed by the programmer.
- There are chances of having unidentified paths during the testing process.



Land Your
Dream Position

Opportunities don't happen, you create them.

6.3 Screenshot of Admin Login Successful

6.4 White Box Testing:

White box testing requires access to source code. Though white box testing can be performed any time in the life cycle after the code is developed, it is a good practice to perform white box testing during unit testing phase.

In designing of database, the flow of specific inputs through the code, expected output and the functionality of conditional loops are tested.

1.Guarantee that all independent paths have been executed.

2.Executed all logical decisions on their true and false sides.

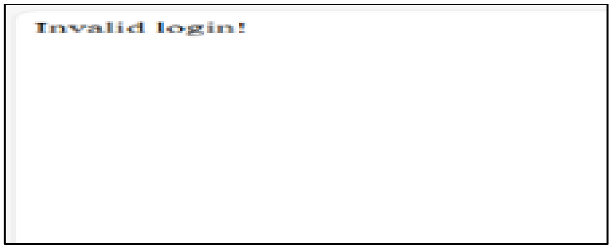
3.Executed all loops at their boundaries and within their operational bounds.

4.Executed internal data structures to ensure their validity.

Unlike black box testing that only checks if a system works from outside, white box testing dives into the internal mechanics of the application. In the figure 6.4,we performed white box testing by purposely feeding the login process wrong information, particularly examining how the system manages admin authentication. This method grants us the ability to closely inspect the application's inner workings, revealing potential weakness. By deliberately probing the system's logic with incorrect data, we aim to identify and address vulnerabilities, ensuring a robust and secure application.

Advantages of White Box Testing:

- Testing can commence even before the GUI is ready.
- Code optimization by finding hidden errors.
- White box test cases can be easily automated.
- It can provide stability and usability of the test cases.
- Results in cleaner, more maintainable, and higher-quality code.
- Tests all paths, branches, and conditions in code.
- Uncovers hidden security vulnerabilities within the code.

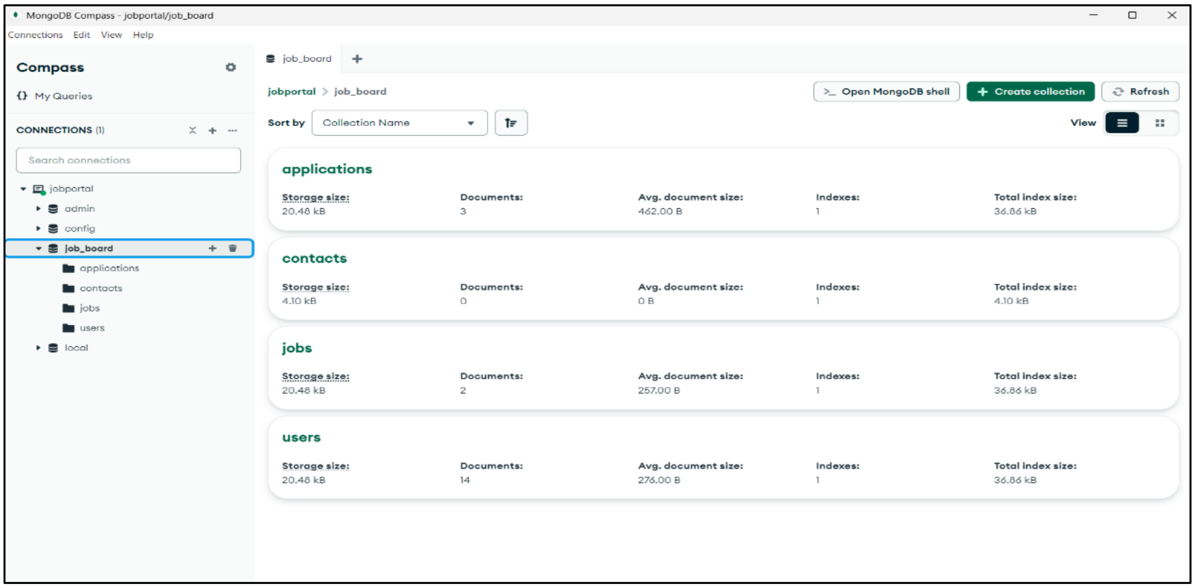


Invalid login!

Figure 6.4 The admin has failed because of invalid login credentials

6.5 Database Testing

This test involves testing (Figure 6.5) whether the values entered through the form gets stored and saved in the database correctly or not.

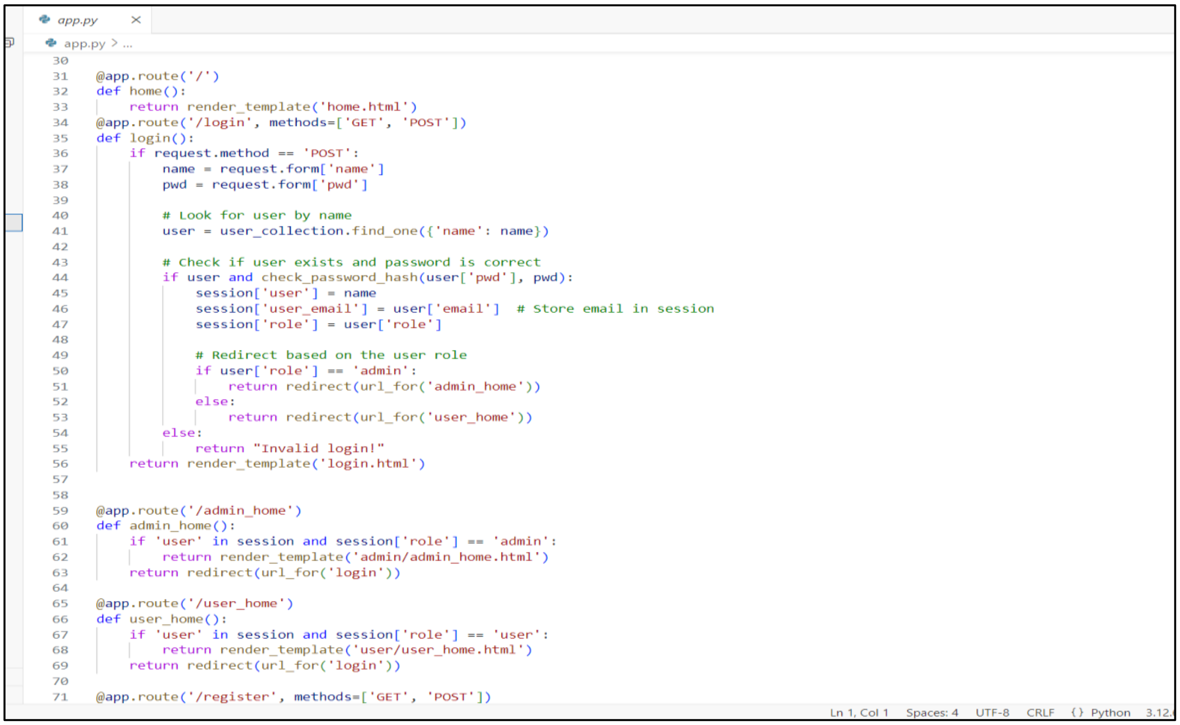


	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
applications	20.48 kB	3	462.00 B	1	36.86 kB
contacts	4.10 kB	0	0 B	1	4.10 kB
jobs	20.48 kB	2	257.00 B	1	36.86 kB
users	20.48 kB	14	276.00 B	1	36.86 kB

Figure 6.5 Database Testing

6.6 Performance Testing:

The performance of the application can be tested through this test. Since the number of codes is less, performance is good. The below figure 6.6 shows the performance testing of the code.



```python

@app.route('/')
def home():
    return render_template('home.html')
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        name = request.form['name']
        pwd = request.form['pwd']

        # Look for user by name
        user = user_collection.find_one({'name': name})

        # Check if user exists and password is correct
        if user and check_password_hash(user['pwd'], pwd):
            session['user'] = name
            session['user_email'] = user['email']  # Store email in session
            session['role'] = user['role']

            # Redirect based on the user role
            if user['role'] == 'admin':
                return redirect(url_for('admin_home'))
            else:
                return redirect(url_for('user_home'))
        else:
            return "Invalid login!"
    return render_template('login.html')


@app.route('/admin_home')
def admin_home():
    if 'user' in session and session['role'] == 'admin':
        return render_template('admin/admin_home.html')
    return redirect(url_for('login'))

@app.route('/user_home')
def user_home():
    if 'user' in session and session['role'] == 'user':
        return render_template('user/user_home.html')
    return redirect(url_for('login'))

@app.route('/register', methods=['GET', 'POST'])
```

Figure 6.6 Performance Testing

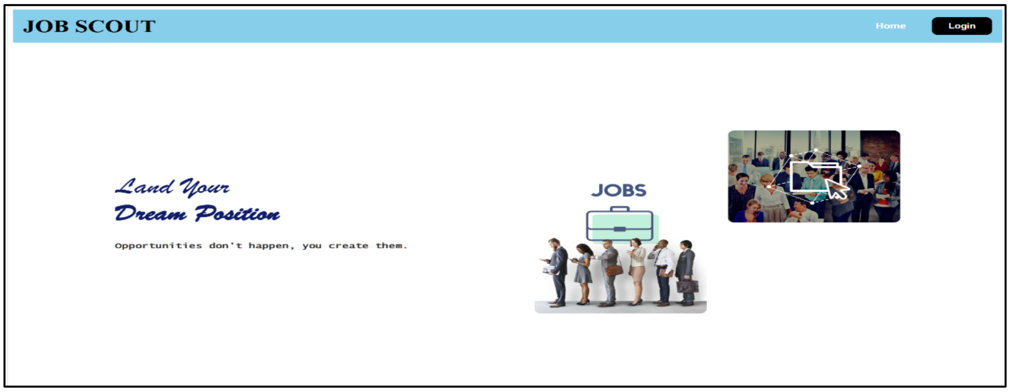
6.7 Test Results

Test cases are done on every page on this website. All the test cases are passed successfully. No defects were encountered.

CHAPTER 7

RESULTS

7.1 Home Page:



Land Your
Dream Position

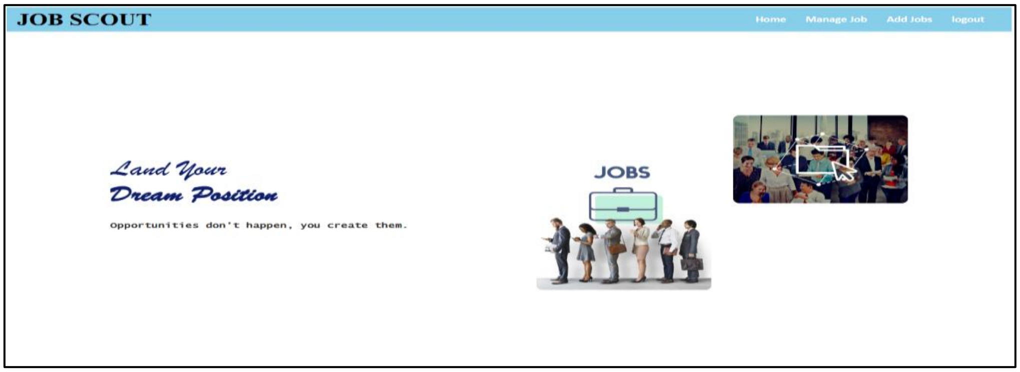
opportunities don't happen, you create them.

7.1 Screenshot of Home page

The figure 7.1 is the home page from where a candidate can Login as user or admin to the site if he/she has an account, otherwise they need to register first.

7.2 Admin Pages

Admin Dashboard Page:



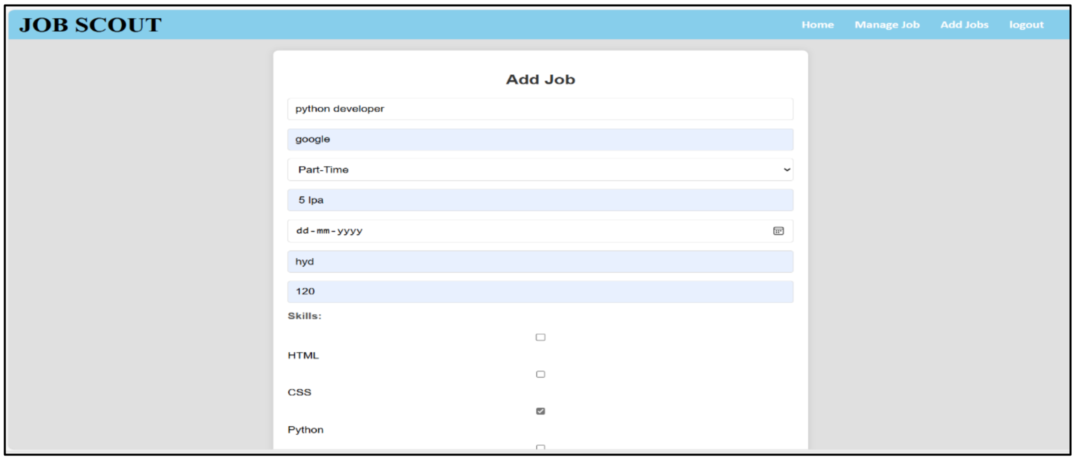
Land Your
Dream Position

opportunities don't happen, you create them.

7.2 Screenshot of Admin Dashboard Page

The figure 7.2 is the Admin Dashboard Page from where he/she Can add and manage the Jobs.

Admin adding Jobs:



Add Job

python developer

google

Part-Time

5 lpa

dd-mm-yyyy

hyd

120

Skills:

□

HTML

□

CSS

☑

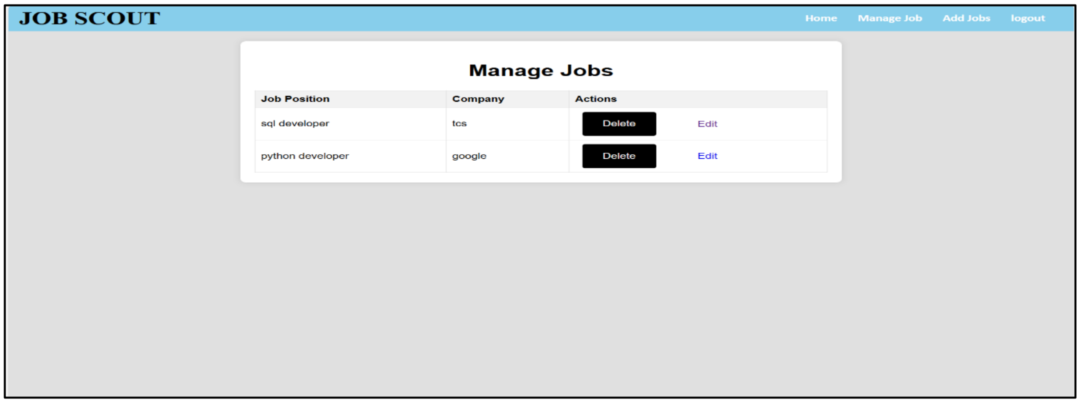
Python

□

7.3 Screenshot of Admin adding jobs

The figure 7.3 is about Admin adding jobs where he can fill the form by mentioning the details such as position, company, part-time, location, vacancies, skills, contact email and job description.

Admin managing Jobs:



Manage Jobs

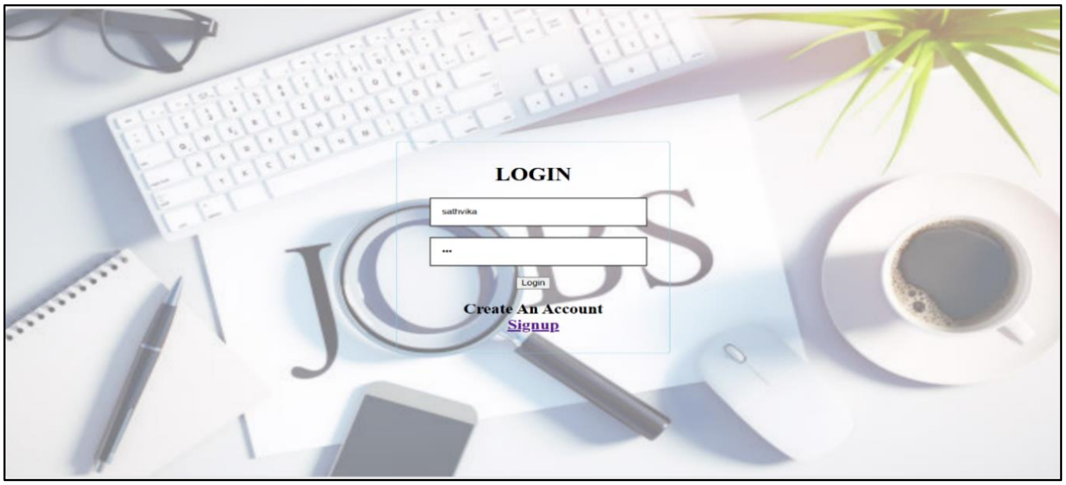
Job Position	Company	Actions	
sql developer	tcs	Delete	Edit
python developer	google	Delete	Edit

7.4 Screenshot for Admin managing job

The figure 7.4 is the admin managing jobs. Here the admin Manages the jobs by performing two operations such as Edit and Delete.

7.3 Member Pages

Member login Page:



LOGIN

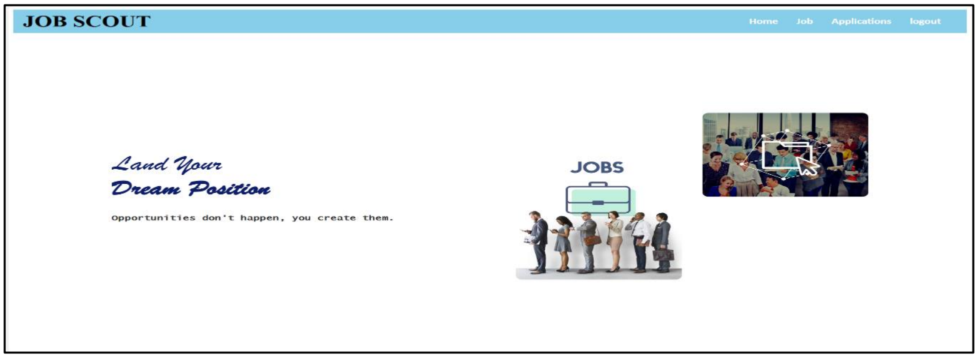
sathvika

•••

7.5 Screenshot of member login form

The above figure 7.5 is the member login form. Here, the member is supposed to provide his/her email and password. So that he/she can access the further services.

Member Dashboard:



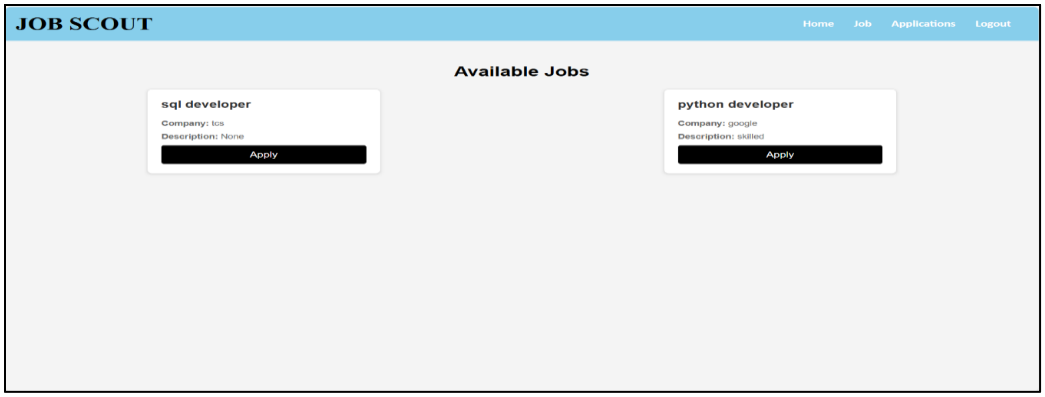
Land Your
Dream Position

Opportunities don't happen, you create them.

7.6 Screenshot of Member Dashboard

The above figure 7.6 shows the member dashboard after he/she logins into the site. The page opens only when the member is verified. The page provide services to access his Job page and to keep track of application statues of the jobs.

Member Job Page:



Available Jobs

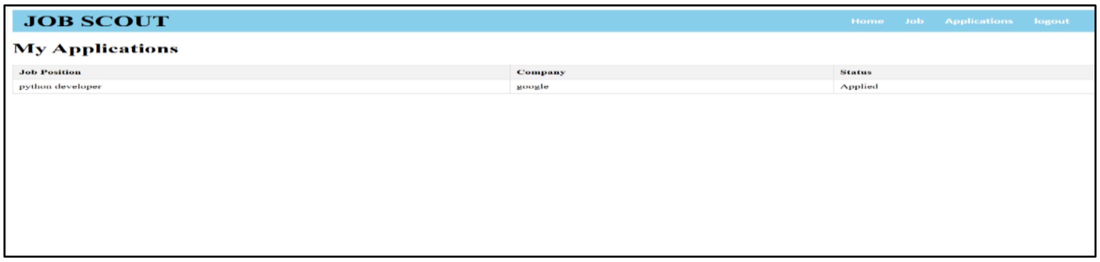
sql developer
Company: tcs
Description: None

python developer
Company: google
Description: skilled

7.7 Screenshot of Member Job Page

The above figure 7.7 is the job page of member where he/she can apply for the job by entering the details such as name, email, job role, Address, city, phone number and can upload their resume/cv.

Member Application Page:



My Applications

Job Position	Company	Status
python developer	google	Applied

7.8 Screenshot for Member Application Page

The above figure 7.8 is the application page of member, where we can see the applied job statuses.

\

CHAPTER 8
CONCLUSION

It can be concluded that this project of Online Job Portal was a real learning experience. It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge. The principles of software production were well implemented throughout the system. The project has been made as per as the given specifications. A Job Portal provides an efficient search for online information on job vacancies for Job Seekers. The main goal of this portals to attempt to produce the right graduates based on the industry needs. This will provide better opportunities and guidance in future in developing projects independently. However, it is important that be aware that the Job Portals can never fulfill all the problems of jobless graduates.

CHAPTER 9

FUTURE SCOPE AND ENHANCEMENTS

This project fulfills the primary requirements of the job seekers. It can be extended in several ways-We can provide recommendations and email updates for new job postings based on the job seeker’s search history. Since, the job seekers might be interested in building a strong Resume, we can provide tips and information for the same. We can also provide templates for building the Resumes which might interest most applicants. The mobile application is developed fulfilling the functionalities of job seeker, it can be extended to support functionalities of Employer as well.

- Online payment for premier job posting: Online payment method for every post or annual subscription charges for regular based job posting agents.
- Banner management: Organization can use their banner at the homepage to display for visitors in order to give priority for their job post.
- Advanced job filtering and search options.
- Email notifications for both users and admins.
- Role-based access control for better security.
- Integration of a resume parsing feature to improve the application process for users.

BIBLIOGRAPHY

[1] Robbins, J. N. (2018). Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. 5th Edition. O'Reilly Media. ISBN-10: 1491960205, ISBN-13: 978-1491960202.

[2] Jacobson, I., Rumbaugh, J., & Booch, G. (2005). Unified Modelling Language. 2nd Edition. Addison-Wesley. ISBN-10: 032171895X, ISBN-13: 978-0321267979.

[3] Pusuluri, N. (2008). Software Testing Concepts and Tools. 2nd Edition. Dreamtech Press. ISBN-10: 8177227122, ISBN-13: 978-8177227123.

[4] Teorey, T. J., Lightstone, S. S., & Nadeau, T. (2010). Database Modelling and Design. 5th Edition. Morgan Kaufmann. ISBN-10: 0123820200, ISBN-13: 978-0123820204.

[5] Earp, R. (2007). Database Design using Entity-Relationship Diagrams. 2nd Edition. Auerbach Publications. ISBN-10: 1439861765, ISBN-13: 978-1439861769.

[6] Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design. 3rd Edition. Prentice Hall. ISBN-10: 0131489062, ISBN-13: 978-0131489066.

[7] Bank, D. (2019). MongoDB: The Definitive Guide. 3rd Edition. O'Reilly Media. ISBN-10: 1491954469, ISBN-13: 978-1491954461.

[8] Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. 2nd Edition. O'Reilly Media. ISBN-10: 1491991739, ISBN-13: 978-1491991732.