ONLINE JOB SEARCH PORTAL

A PROJECT REPORT

for

Project (KCA451)

Session (2023-24)

Submitted by

HARSH CHAUHAN

2200290140064

HARSH AWASTHI

2200290140063

Submitted in partial fulfillment of the

Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of

Ms. Shruti Agrawal

Teaching Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(MAY 20224)

DECLARATION

I hereby declare that the work presented in report entitled “Online Job Search Portal” was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University of Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, that are not my original contribution. I have used quotation marks to identify verbatim sentences and give credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Harsh Chauhan
Roll No.: 2200290140064

Name: Harsh Awasthi
Roll No.: 2200290140063

(Candidate Signature)

CERTIFICATE

Certified that **Harsh Chauhan 2200290140064 , Harsh Awasthi 2200290140063** have carried out the project work having "**Online Job Search Portal**" (**Major Project-KCA451**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Harsh Chauhan 2200290140064
Harsh Awasthi 2200290140063

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Ms. Shruti Agrawal	**Dr. Arun Tripathi**
Teaching Professor	**Head**
Department of Computer Applications	**Department of Computer Applications**
KIET Group of Institutions, Ghaziabad	**KIET Group of Institutions, Ghaziabad**

ABSTRACT

The evolution of the digital age has transformed the job market landscape, prompting the need for efficient and accessible platforms to match job seekers with employers. This abstract presents the design and development of an Online Job Portal Web Application, aimed at streamlining the job search and recruitment process.

The platform prioritizes user experience by offering a seamless interface for both job seekers and employers. Job seekers can create personalized profiles, upload resumes, and browse through a wide array of job listings filtered by various criteria such as industry, location, and experience level. Advanced search algorithms ensure relevant job suggestions tailored to the user's preferences.

For employers, the web application provides robust tools for posting job vacancies, managing applications, and filtering candidates based on desired qualifications. Additionally, features such as applicant tracking systems and automated notifications facilitate efficient communication and streamline the hiring process.

The Online Job Portal Web Application incorporates state-of-the-art security measures to safeguard user data and ensure confidentiality throughout the recruitment process. Furthermore, it is designed to be scalable and adaptable to accommodate future advancements and evolving user needs.

In conclusion, this web application serves as a dynamic platform bridging the gap between job seekers and employers, facilitating efficient recruitment processes and fostering employment opportunities in the digital era.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Shruti mam** for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Harsh Chauhan

Harsh Awasthi

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

With tremendous increase in technology, information technology is a fast-developing field. Technology which is in vogue today might become redundant tomorrow. This ever-changing scenario makes it possible to provide the latest and most modern IT solutions to various business and institutions.

I am doing my project on Online Job Search Portal. There is the need for efficient management of a network-based system for handling customer orders. This project is an endeavor to provide a solution to this. The proposed system enables an administrator to keep track of customer orders and maintaining records of the customers. Thus, the project is a sincere effort in simplifying the task of administrators in an easily usable format. I finalized to make this project and hence planned to develop this system using HTML CSS JavaScript for front end and MongoDB database as the Back End.

Implementing a Online job Search Portal provides numerous benefits for businesses of all sizes. From improved efficiency and client satisfaction to data-driven decision making and cost reduction, an SMS can be a valuable tool for driving success in the competitive salon industry

1.2 Motivation

Building a Online Job Search Portal as a college project offers students a unique opportunity to apply their technical skills in a practical and relevant context, while also fostering collaboration, client interaction, and entrepreneurial thinking. It serves as a valuable learning experience that prepares students for future careers in software development and related fields.

Most of the employees are using smart phones and tablets, so eventually they got the basic idea about how to use web browsers and access internet and work on it. And they can easily adept to this online web base Online Job Search Portal with simple system introduction of how to use it.

The system will provide user friendly simple interface which will help them to make reliable, quick and effective service.

1.3 Problem Statement

While online job search portals have revolutionized the employment landscape, various shortcomings hinder their effectiveness in meeting the diverse needs of job seekers and employers. This problem statement highlights key challenges faced by existing online job search portals and proposes avenues for improvement.

Limitations Of Existing System

Before starting the project we have to study its limitations and objective some of them are:

- Many existing portals lack robust personalization features, resulting in job listings that may not align closely with the preferences and qualifications of job seekers. This lack of personalization can lead to frustration and inefficiency in the job search process.
- The algorithms used by some portals to match job seekers with vacancies may not be sufficiently refined, leading to mismatches and missed opportunities. Poorly matched job recommendations can decrease user satisfaction and hamper the effectiveness of the platform.
- Some online job portals suffer from complex user interfaces and navigation structures, making it challenging for users to find relevant information quickly. A confusing user experience can discourage engagement and drive users away from the platform.
- Accessibility features for users with disabilities are often overlooked in many existing job portals, excluding a significant portion of the workforce from accessing opportunities. Additionally, language barriers and lack of support for diverse cultural backgrounds can further limit inclusivity.
- While job seekers are the primary focus of many portals, the tools and features available for employers may be lacking. This can result in suboptimal recruitment processes, hindered employer branding efforts, and decreased overall satisfaction with the platform.
- With the increasing prevalence of cyber threats, ensuring the security and privacy of user data is paramount. Many existing portals may not employ robust security measures, leaving user information vulnerable to breaches and unauthorized access.

Proposed System

The proposed online job search application aims to streamline the job search process for both job seekers and employers, providing a seamless platform for connecting talent with opportunities.

- With improved computerization being involved in the maintenance of monitoring of the processes from customer registration to activation. Report generation will help make it easy to analyze the performance at the Bank.
- Customer details, error and inconsistencies can be kept at par.
- Validation of data will ensure only accurate, valid and complete data is stored in the database.

1.4 Objective And Scope

The objective of a modern online job search portal is to leverage cutting-edge technology and user-centric design principles to create an intuitive, efficient, and personalized platform for connecting job seekers with relevant employment opportunities. The primary goal is to enhance the job search experience by providing advanced features, personalized recommendations, and streamlined communication channels.

Website

The Site should be displayed with the 'Home', 'About us', 'Services', 'Specials', 'Contact us' and Registration pages.

User Authentication and Account Management:
Allow users (both job seekers and employers) to register, log in, and manage their accounts.
Implement features for password recovery, account verification, and profile customization.

Job Search and Discovery:
Provide robust search functionality to enable users to find relevant job listings.
Offer advanced search filters such as location, industry, job type, salary range, and experience level.

Job Listing and Management:
Allow employers to create, edit, and manage job listings.
Provide tools for employers to customize job descriptions, specify requirements, and set application deadlines.

Application Management:
Enable job seekers to apply for jobs directly through the portal.
Provide features for job seekers to upload resumes, cover letters, and other relevant documents.

User Profile Management:
Allow users to create and manage their profiles.
Provide options for job seekers to showcase their skills, experience, education, and other relevant information.

Communication and Messaging:
Facilitate communication between job seekers and employers through in-app messaging.
Provide features for employers to communicate with applicants, schedule interviews, and provide feedback.

User Type – Job Seeker/User

- Allow job seekers to register on the platform by providing basic information such as name, email address, and password.
- Offer options for social media or single sign-on (SSO) registration to streamline the process.
- Enable job seekers to create detailed profiles that showcase their skills, experience, education, certifications, and employment history.
- Provide fields for additional information such as desired job roles, industries, salary expectations, and location preferences.
- Allow users to upload resumes, cover letters, and other relevant documents to their profiles.
- Provide advanced search filters such as location, industry, job title, keywords, salary range, and experience level.
- Allow users to save search queries and set up job alerts to receive notifications about new job listings matching their criteria.
- Offer options for users to save favorite job listings for later viewing or apply directly through the platform.
- Allow job seekers to update their profiles, add new skills or experiences, and modify their job preferences.

- Provide options for users to make their profiles public or private, depending on their preferences.
- Offer tools for users to monitor profile completeness and receive suggestions for improving their profiles.
- Facilitate communication between job seekers and employers through an in-app messaging system.
- Provide features for job seekers to communicate with employers, ask questions about job listings, and request additional information.
- Implement notifications for new messages, application updates, and other relevant events to keep job seekers informed.
- Offer tools for job seekers to create, edit, and format professional resumes directly within the platform.
- Offer tools for job seekers to assess their skills, strengths, and career interests to better align with job opportunities.

User/Job Provider Companies Module:

- Allow employers or job provider companies to register on the platform by providing company information such as name, industry, size, location, and contact details.
- Offer options for company representatives to create user accounts and manage company profiles.
- Enable companies to create detailed profiles that showcase their mission, values, culture, and employer brand.
- Provide fields for additional information such as company history, team members, benefits, and perks.
- Offer a user-friendly interface for employers to create, edit, and manage job listings.
- Provide fields for job details such as job title, description, requirements, responsibilities, salary range, and location.

- Offer options for employers to customize job listings with additional features such as company logos, images, videos, and employee testimonials.
- Allow employers to specify application instructions, deadlines, and contact information for applicants.
- Implement features for employers to track the performance of their job listings, including views, applications, and conversion rates.
- Provide tools for employers to manage job applications received through the platform.
- Enable employers to review applicant profiles, resumes, cover letters, and other submitted documents.
- Offer features for employers to communicate with applicants, schedule interviews, and provide feedback throughout the hiring process.

User Type – Admin

- Provide an intuitive dashboard for administrators to view key metrics and monitor platform activity.
- Display statistics such as the number of registered users, active job listings, applications received, and user engagement metrics.
- Offer visualizations such as charts, graphs, and tables to present data in a clear and actionable format.
- Allow administrators to manage user accounts, including creating new accounts, approving registrations, and resetting passwords.
- Provide options for user role management, allowing administrators to assign roles and permissions to different user groups such as job seekers, employers, and moderators.
- Implement features for suspending or banning user accounts in case of violations or misconduct.
- Offer tools for administrators to manage job listings, including approving new listings, editing existing listings, and removing expired or inappropriate listings.

- Provide options for content moderation, allowing administrators to review and moderate user-generated content such as comments, reviews, and forum posts.
- Implement features for flagging and reporting inappropriate content and taking appropriate action.
- Allow administrators to generate customized reports on platform activity, user interactions, and other relevant data for decision-making and analysis.
- Offer tools for managing system resources, optimizing performance, and ensuring.
- Implement features for managing third-party integrations, APIs, and external services used by the platform.
- Implement moderation features to ensure that content on the platform complies with community guidelines, terms of service, and legal requirements.
- Provide tools for reviewing and addressing user-reported issues, violations, and disputes in a timely manner.
- Offer options for enforcing platform policies, issuing warnings, suspending accounts, or taking other disciplinary actions as needed.
- Implement robust security measures to protect user data, including encryption of sensitive information, secure authentication mechanisms, and regular security audits.
- Offer support tools for administrators to assist users, address inquiries, and resolve issues effectively.
- Provide options for creating and managing support tickets, tracking their status, and communicating with users to provide updates and solutions.
- Implement a knowledge base or FAQ section to provide self-service resources and guidance for common questions and issues.

1.5 Feasibility study

ECONOMIC FEASIBILITY:

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. This part of feasibility study gives the top management the economic justification for the new system.

TECHNICAL FEASIBILITY:

Technical feasibility centers on the existing manual system of the test management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities.

BEHAVIOURAL FEASIBILITY:

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

RESOURCE REQUIREMENTS

Team: Project manager, front-end and back-end developers, database administrator, UX/UI designer, and QA tester.
Tools: Development frameworks (e.g., React, Node.js), database systems (e.g., MySQL, MongoDB), and project management software (e.g., Jira).

SWOT ANALYSIS:

Strengths: Established brand presence, extensive user base.
Weaknesses: Can be cluttered and overwhelming for users.
Opportunities: Niche markets, enhanced user experience, AI-driven matching.
Threats: High competition, rapidly changing technology.

CHAPTER 2

LITERATURE REVIEW

In [1], Digital Job Searching and Recruitment Platforms: A Semi-systematic Literature Review, August 2023 DOI:10.1007/978-3-031-42134-1_31

The purpose of this paper is to shed light on the new E-recruitment trend that is pervading the lives of job seekers, included students, and job offers. A semi-systematic literature review on digital job searching and recruiting platform in the last five years was conducted with the aim to develop a preliminary conceptual framework.

The purpose of this paper is to shed light on the new E-recruitment trend that is pervading the lives of job seekers, included students, and job offers. A semi-systematic literature review on digital job searching and recruiting platform in the last five years was conducted with the aim to develop a preliminary conceptual framework.

Following a replicable research process, a final sample of 37 publications was located in five subdimensions - Web Application Framework, Use of Artificial Intelligence technologies, Use of Blockchain Technologies, Type of User, User Experience - grouped by two dimensions of analysis: “Technical implementation of the platform”, “Platform usability analysis”.

From our findings it emerges that the first one received strong attention, specifically with regards to subdimensions Web Application Framework and Use of the Artificial Intelligence Technologies; the subdimension Use of the Blockchain Technologies started to attract scholarly attention only from 2020.

In [2] Digital Job Searching and Recruitment Platforms: A Semi-systematic Literature Review Conference paper First Online: 29 August 2023 pp 313–322

The purpose of this paper is to shed light on the new E-recruitment trend that is pervading the lives of job seekers, included students, and job offers. A semi-systematic literature review on digital job searching and recruiting platform in the last five years was conducted with the aim to develop a preliminary conceptual framework. Following a replicable research process, a final sample of 37 publications was located in five subdimensions - Web Application Framework, Use of Artificial Intelligence technologies, Use of Blockchain Technologies, Type of User, User Experience - grouped by two dimensions of analysis: “Technical implementation of the platform”, “Platform usability analysis”. From our findings it emerges that the first one received strong attention, specifically with regards to subdimensions Web Application

Framework and Use of the Artificial Intelligence Technologies; the subdimension Use of the Blockchain Technologies started to attract scholarly attention only from 2020. The second dimension of analysis has received a fair amount of attention over the last five years, but it seems that in 2021 the sub-dimension Type of User is perceived as the most attractive from scholars from different field of studies.

The contribution of this work is twofold. Firstly, it tries to shed lights on the main characteristics of the studies about the job searching and recruiting platforms as derived from the publications included in our review identifying appropriate dimensions and sub-dimensions of analysis that could be useful to analyze these platforms in the future. Secondly, for each sub-dimensions we identified the major challenges that authors have set out to address. This specific aspect will be helpful to identify the future research agenda for the topic investigated.

In [3] Demystifying the User Experience: A Case Study on Online Job Search Engines Chyna Amanova1, Sena Bulak Ozgur, Fortunata Msilu, and Fatih Demir Northern Illinois University, USA

Since the onset of the COVID-19 pandemic in 2020, the demand for job search engines has increased because of the availability of searching for remote job positions. However, there is still a lack of research on how people interact with different job search engines. It is critical to assess the effectiveness and efficiency of these tools for people searching for jobs because search engines help job seekers find positions that best fit their qualifications.

This study assessed a job search engine by utilizing surveys, interviews, and observational methods to provide an improved understanding of individuals' behavior and decision making in their search for career.

The results showed that the job search engine needs improvement in the usability aspect because the website has an unintuitive interface and features, suggesting novice users may face difficulties in using it. The findings contribute to the limited body of literature that examines the user experiences in searching for jobs.

In [4] A Review Study on Online Job Portal
Profile image of International Journal of Scientific Research in Computer Science, Engineering and Information Technology IJSRCSEIT International Journal of Scientific Research in Computer Science, Engineering and Information Technology IJSRCSEIT

The software for the training and placement cell of a college is a need for the students and the institute management for proper placement and training of the students of the institute. It helps the students to provide their profiles to the training and placement cell of the institute, updating their respective profiles with their gradual approach towards the course end. The students also get to know about the companies coming for the on -campus/off-campus/pool/group pool categories of campus interviews. This paper emphasizes the significance of an on-campus online job recruitment system and its function in assisting students in obtaining available employment. It emphasizes the issues with traditional employment practices, particularly for college students.

This paper presents and emphasizes the need for an online employment posting platform for colleges and the efficacy of such a system in connecting students with job possibilities. Historically, Human Resource management has utilized employment websites for candidate sourcing and placement. The current project is an employment website developed for one of the most prestigious engineering schools, which is a variant of such job boards tailored to service the students of the institution. With functions such as job suggestions provided offering learners recommendations based on their skills and candidate filtering to aid employers in application matchmaking, the platform is anticipated to be useful for both students investigating job opportunities.

In [5] JOB PORTAL: FINDING BEST JOB AND BEST CANDIDATE
Profile image of IRJET Journal IRJET Journal 2022, IRJET Volume 4, Issue 3, March – 2019

Online Job Search Portal web applications have become essential tools for streamlining operations, enhancing customer experience, and improving business efficiency in the salon industry. This literature review aims to provide an overview of existing research and development efforts in this field, highlighting key features, challenges, and emerging trends.

While existing research and development efforts have made significant strides in addressing key challenges and incorporating emerging technologies, there remains ample opportunity for further innovation and improvement in this field. Future research directions may include exploring the impact of AI technologies, improving data security and privacy measures, and investigating novel approaches to enhancing user experience in Online Job Search Portals.

CHAPTER 3

SYSTEM DEVELOPMENT AND CODING

3.1 Existing System

The existing systems enables jobseekers to search through print media like poster advertisements, newspapers and visual media like television or company websites for employment opportunities. This is a tedious task as it takes a lot of time and energy to search for the right job position, learn about the position and about the company. Job search for proper match of skill set and salary is challenging.

Job seekers can also find jobs through job fairs where they must first make it possible to attend the fairs which might be sometimes impossible with their schedules and if they visit the fairs they must hand over paper printed resumes. The more the number of candidates the more the number of papers for the company which is a lot of manual effort. Again, jobseekers might get job offers through placement cells in respective colleges but getting hold of the right opportunity at the right time is always challenging. On the other hand, the same goes for employers who are looking for candidates who are best fitted for their job positions.

They must constantly advertise, go to a lot of job fairs which still doesn't guarantee the best way to select from a large pool of candidates. Such conventional and outdated systems are replaced by several well featured national job search portals like Monster, Dice.com, Glassdoor, Indeed etc. All these job search and advertisement portals aims at e-recruitment by providing several simple and useful features to jobseekers and employers making job search and candidate selection a much time saving and easier process.

3.2 Proposed System

With the advancement of technology job seekers are relying greatly on Online Job

Search Portals. Taking motivation from the conventional systems and their drawbacks andinspiration from the existing job search portals, I decided to develop "Dreams Job".

In the proposed system we are trying to develop an online job search web application that reduces challenges for job seekers to find a desired and suitable job according to their qualification. We aim at reducing the challenges by providing advanced search features that gives the candidate ample scope to select jobs that matches their skill set and requirements and gives them back the exact jobs that are available.

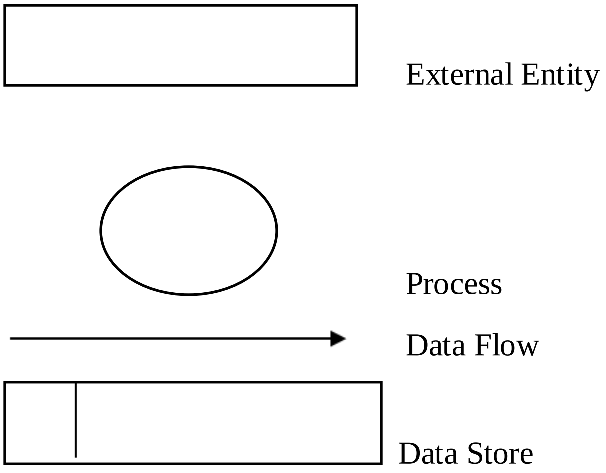
- This in turn is less time taking as the candidate gets all details in one place and do not have to go to company website to learn about the positions. In the proposed system job seekers can upload their resumes in the required file format, see all the available jobs and search for desired jobs and then apply for those jobs. On the other hand, this system enables employers to post their jobs and get a list of all applications which they can screen online and that reduces the huge amount of manual effort and time.
- Online recruitment or e-recruitment is turning out to be both the job seekers and the employers' favorite activity as offer and demand are well met at one place and both must spend less time to get hold of the right roles or candidates. The company can post jobs, see applications and check resumes in the proposed system.

3.3 Data Flow Diagram

The entire system is projected with a physical diagram which specifics the actual storage parameters that are physically necessary for any database to be stored on to the disk. The overall systems existential idea is derived from this diagram.

The content level DFD is provided to have an idea of the functional inputs and outputs that are achieved through the system. The system depicts the input and output standards at the high level of the systems existence.

A DFD does not show a sequence of steps. A DFD only shows what the different process in a system is and what data flows between them. The following are some DFD symbols used in the project



3.3.1 Level 0 Data Flow Diagram

Level 0 Data Flow Diagram will explain the basic flow of data in a system which shows how the new or old user will interact with the system.

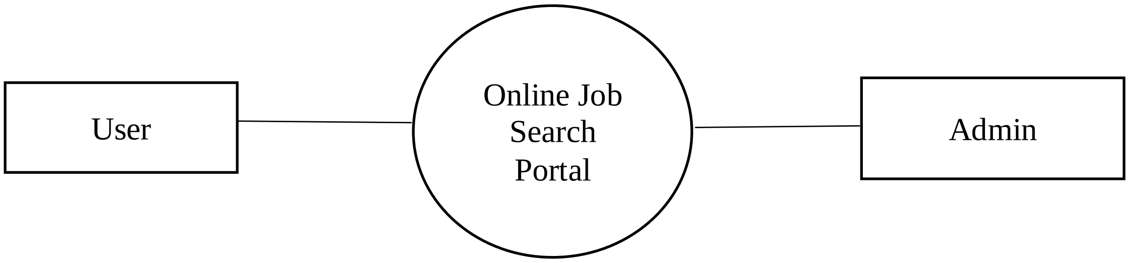


Fig. 3.1 Level 0 DFD of Online Job Search Portal

Fig. 3.1 elaborates the interaction between user and the system. If the user is new then user will first register to the system by providing name, username, email, password, phone. Once successfully registered a message will be display to the user of successfully registered. If the user is old, then they can directly login to the system. Once successfully logged into the system, it will provide a message to the user. This Level 0 DFD provides a simplified overview of the main processes and interactions in the Online Job Search Portal. It serves as a starting point for further elaboration and decomposition into lower-level DFDs, which would provide more detail about each process and its interactions.

3.3.2 Level 1 Data Flow Diagram

Level 1 Data Flow Diagram will explain the basic flow of data in a system which shows how the new or old user will interact with the system with different processes.

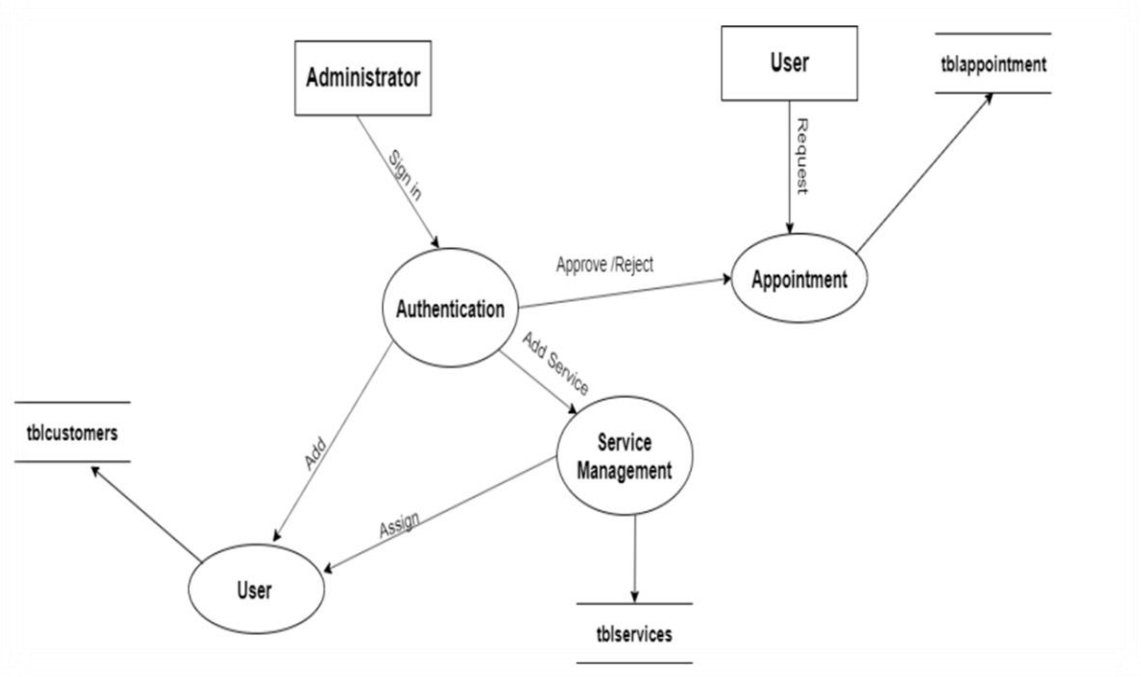


Fig. 3.2 Level 1 DFD of Online Job Search Portal

Fig. 3.2 explains the entire flow of user and system with all processes involved in the system. If the user is new to the system, then register to the system by providing the details to it. And all the details of the user will be stored in the database. If the user is old, then user will log into the system by email and password which will be validated from the database. Then the user will provide the customization of domain and type of the service. After the customization system will make the appointment of the user then feedback is generated and given to the user.

3.4 Data Dictionary

Legal character: [a to z | A to Z]
Digits: [0-9]
Special character: [@, $, #, +, -, /]

Table 1: Data Dictionary

1	Name	Legal Characters
2	Domain	Legal Characters
3	Email	Legal Characters+ Digits+Special Character
4	stylist_id	Digits
5	Schedule	String
6	Phone No.	Digits
7	category	Legal Characters
8	price	Decimal
9	duration	Digits

3.5 Sequence Diagram

Sequence Diagram is used to show the process of the system based on the different timeline.

3.5.1 Sequence Diagram of Registration Process

In this Diagram of Registration Process, it has 4 objects one actor, one boundary object, one control object, one store object.

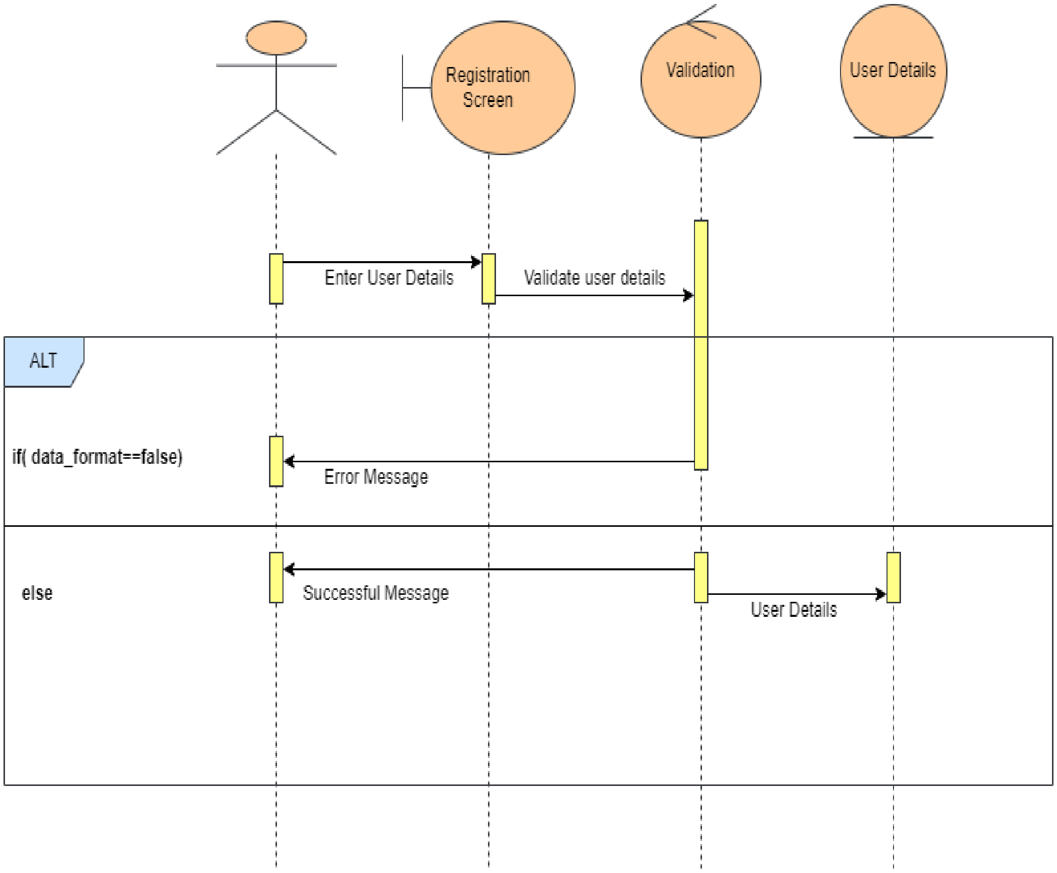


Fig. 3.3 Registration sequence diagram

Fig 3.3 explains about the process of registration where user send the details to the screen then validate those details. If details are not in correct format, then an error message is displayed. If details are in correct format, then successful message is displayed. Then details are stores in user database.

3.5.2 Sequence Diagram of Add Bookings

Following sequence diagram Figure 3.7 illustrate the add booking for customer from booking dashboard.

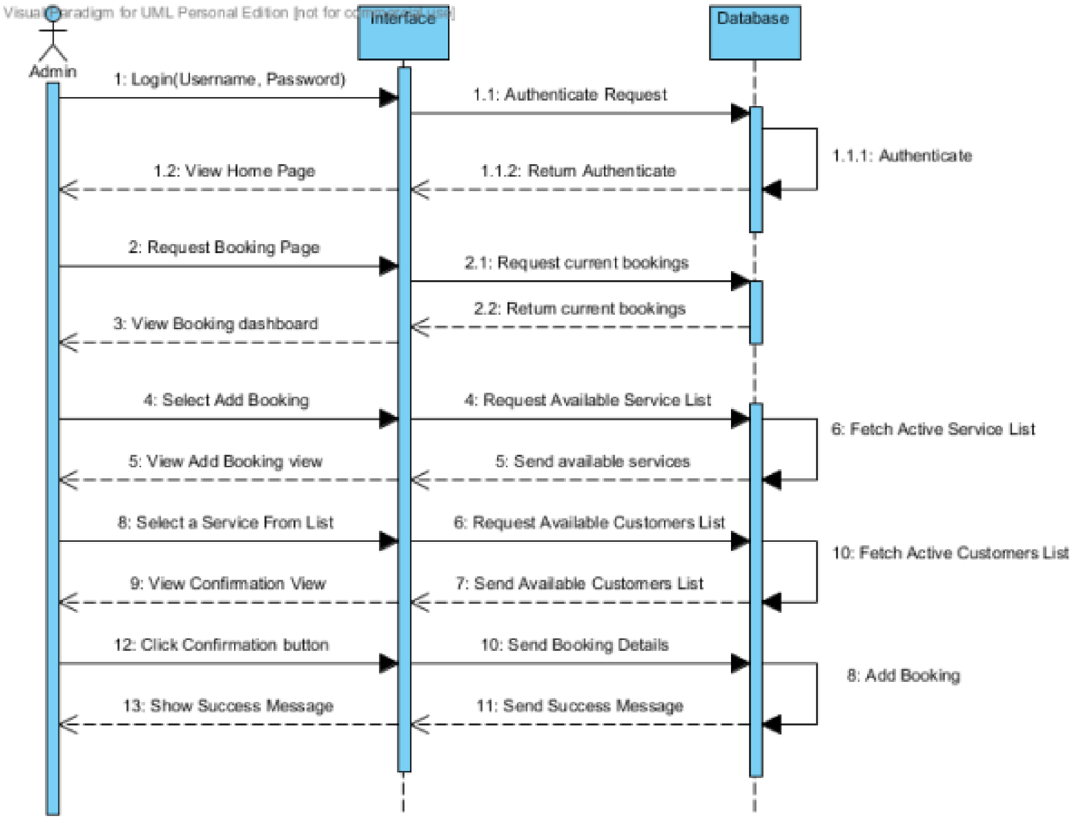

Visual Paradigm for UML Personal Edition [not for commercial use]
Admin
Interface
Database
1: Login(Username, Password)
1.1: Authenticate Request
1.1.1: Authenticate
1.2: View Home Page
1.1.2: Return Authenticate
2: Request Booking Page
2.1: Request current bookings
2.2: Return current bookings
3: View Booking dashboard
4: Select Add Booking
4: Request Available Service List
6: Fetch Active Service List
5: View Add Booking view
5: Send available services
8: Select a Service From List
6: Request Available Customers List
10: Fetch Active Customers List
9: View Confirmation View
7: Send Available Customers List
12: Click Confirmation button
10: Send Booking Details
8: Add Booking
13: Show Success Message
11: Send Success Message

Fig. 3.4 Sequence Diagram Of Add Bookings

3.5.3 Sequence Diagram of Login Process

In this Diagram of Login Process, it has 4 objects one actor, one boundary object, one control object, one store object.

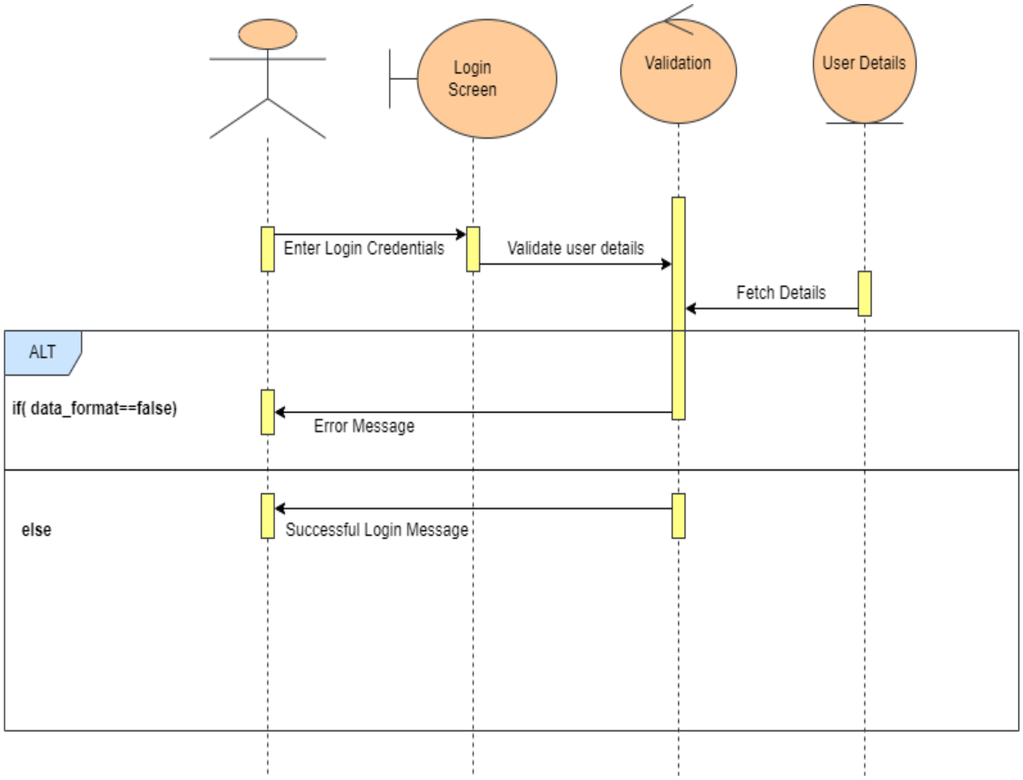


Fig. 3.5 Login sequence diagram

Fig 3.3 explains about the process of login where user send the details to the screen then validate those details. If details are not correct from fetched data from database, then an error message is displayed. If details are correct from fetched data from database, then successful message is displayed.

3.5.4 Sequence Diagram of Add Services

Following sequence diagram Figure 3.8 illustrate add services in to system by system administrator.

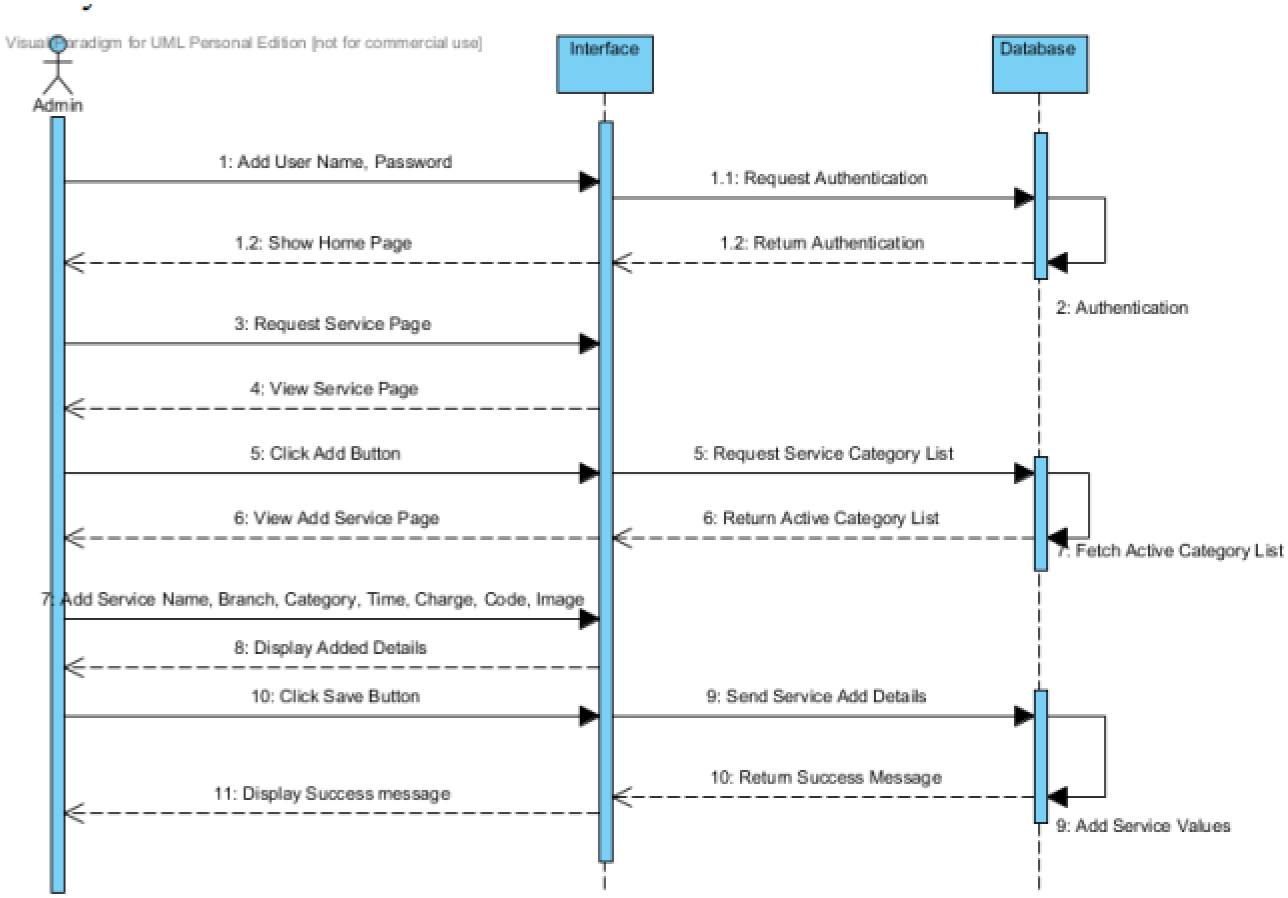

Visual Paradigm for UML Personal Edition [not for commercial use]
Admin
Interface
Database
1: Add User Name, Password
1.1: Request Authentication
1.2: Show Home Page
1.2: Return Authentication
2: Authentication
3: Request Service Page
4: View Service Page
5: Click Add Button
5: Request Service Category List
6: View Add Service Page
6: Return Active Category List
7: Fetch Active Category List
7: Add Service Name, Branch, Category, Time, Charge, Code, Image
8: Display Added Details
10: Click Save Button
9: Send Service Add Details
10: Return Success Message
11: Display Success message
9: Add Service Values

Fig 3.6 Sequence Diagram Of Add Services

3.6 Unified Modeling Language Diagrams (UML):

• The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

• A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View

i. This view represents the system from the user's perspective.

ii. The analysis representation describes a usage scenario from the end-user's perspective.

Structural model view

In this model the data and functionality are arrived from inside the system.

This model view models the static structures.

Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View

In these the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are

UML Analysis modelling, which focuses on the user model and structural model views of the system?

UML design modelling, which focuses on the behavioral modelling, implementation modelling and environmental model views.

3.6.1 Use Case Description of Admin

Primary Actor: Portal Admin

Goal in Context: The admin of the online job search portal utilizes the admin module to manage various aspects of the platform efficiently, including user management, job postings, application tracking, and generating analytical reports.

Scope: Online Job Search Portal

Level: User Goal

Preconditions:

- The owner must have access to the Online Job Search Portal.
- The owner must be authenticated and authorized to access the owner module.
- Success Scenario:
- The owner logs into the Online Job Search Portal.
- The owner accesses the owner module from the dashboard.
- The owner views a summary dashboard displaying key metrics such as revenue, appointments booked, inventory status, and staff performance.

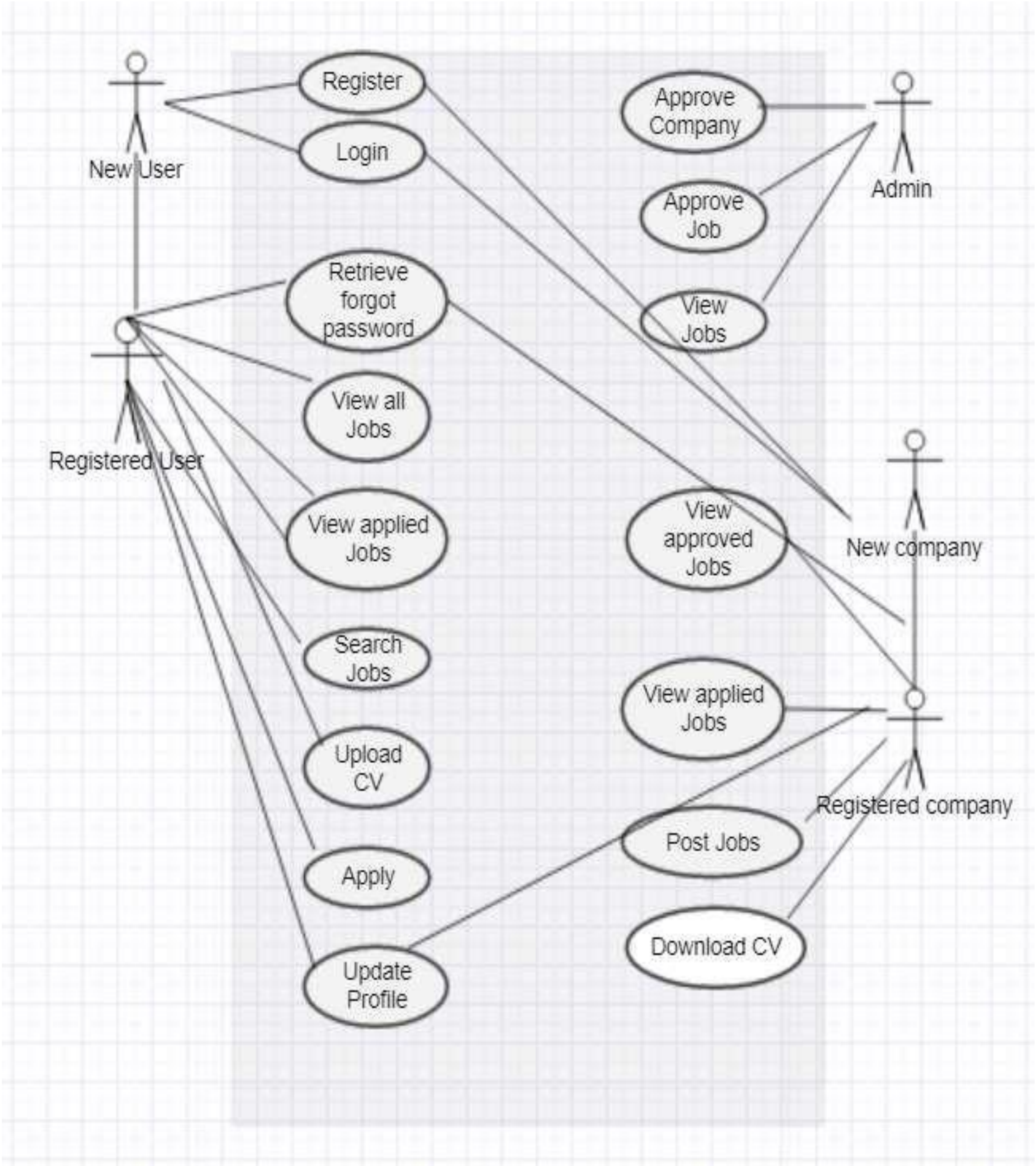


Fig 3.7 illustrates the identified use cases for online job search portal in the new system. It's simply describing the functionalities for owner in the system. In addition to the other users of system owner has additional use cases for system reports. Using system reports owner can get better decisions on the salon management.

3.6.2 Use Case Description of User/Jobseeker

Primary Actor: Jobseeker

Goal in Context: The jobseeker on the online job search portal utilizes the jobseeker module to manage various aspects of their job search efficiently, including profile management, job application submissions, application tracking, and receiving notifications for job alerts and interview requests.

Scope: Online Job Search Portal

Level: User Goal

Preconditions:

- The jobseeker must have access to the Online Job Search Portal.
- The jobseeker must be authenticated and authorized to access the jobseeker module.
- The jobseeker logs into the Online Job Search Portal.
- The jobseeker accesses the jobseeker module from the dashboard.
- The jobseeker views a summary dashboard displaying profile status, application activity, and recent job postings.
- The jobseeker navigates to the job search section to find, apply, and track job applications.

Technology and Data Variations List:

- The system should support user management functionalities, including adding, editing, and removing user accounts.
- Data related to user accounts, system configurations, and access control should be stored securely in a centralized database

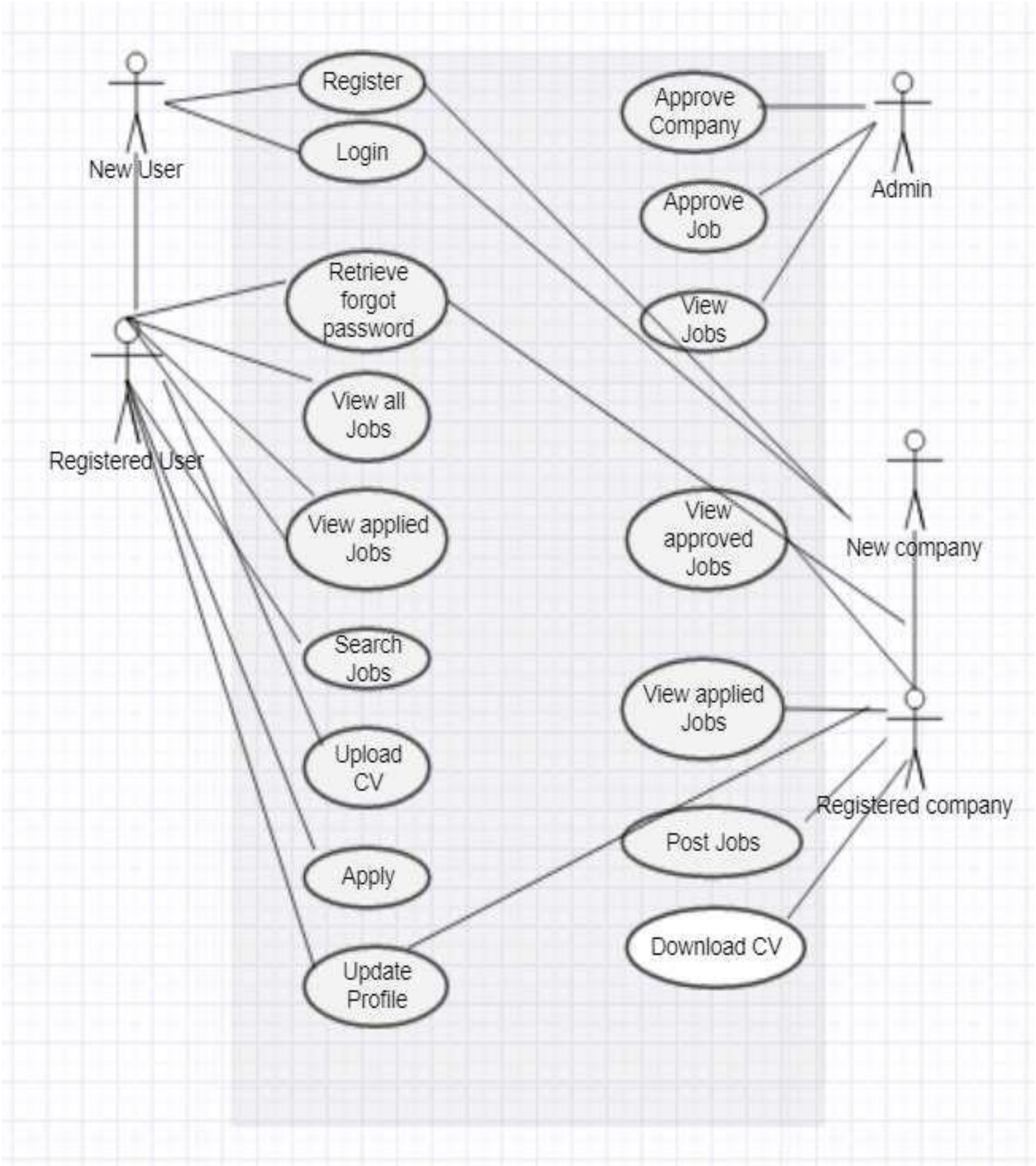


Fig 3.8 is for identified use cases for salon admin in online Job Portal. Its express the deferent use cases that individual branc on the system.

3.6.3 Use Case Description of Companies

Following Figure 3.3 is for identified use cases for companies in online job search portal. Its express the deferent use cases that company work on the system.

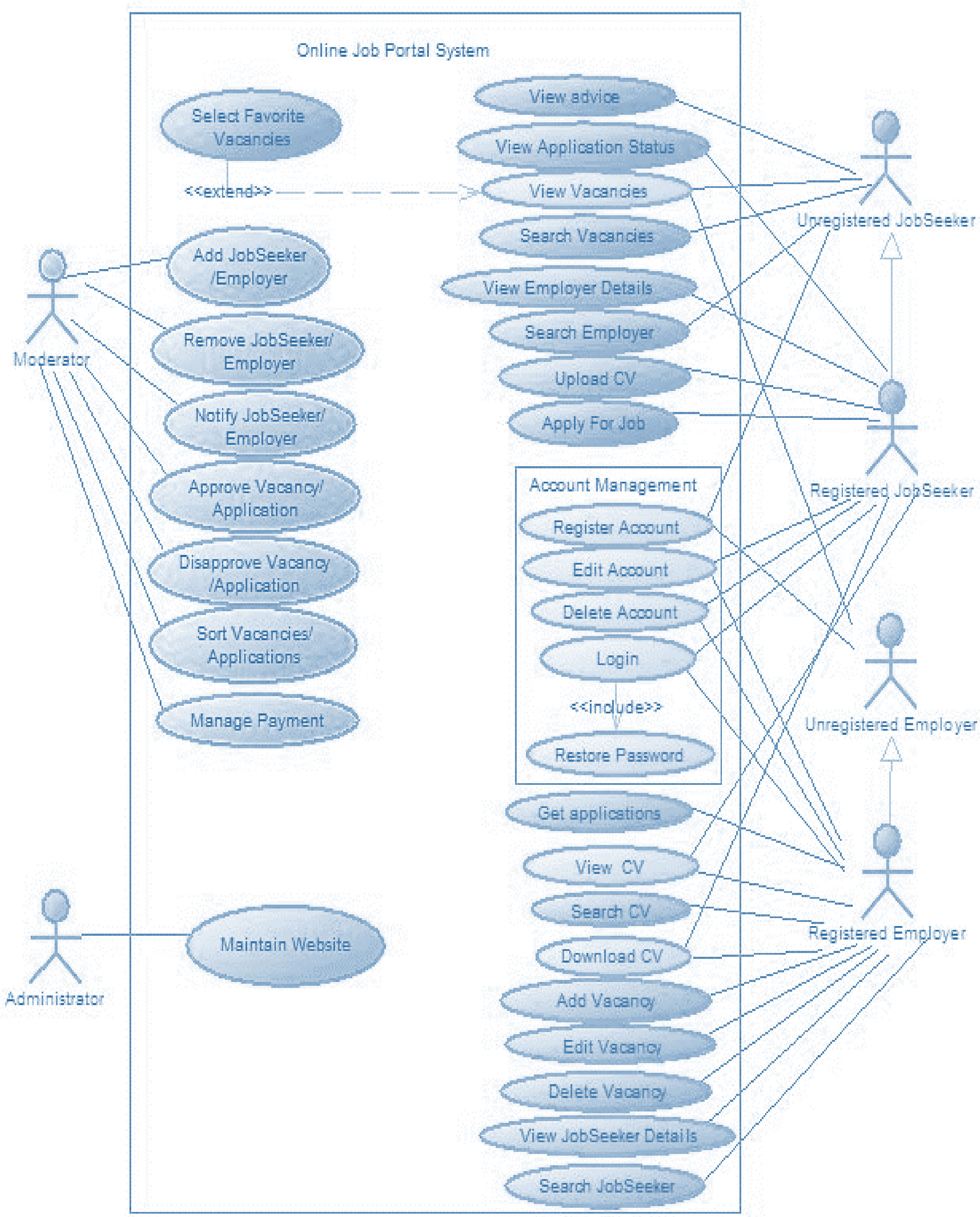
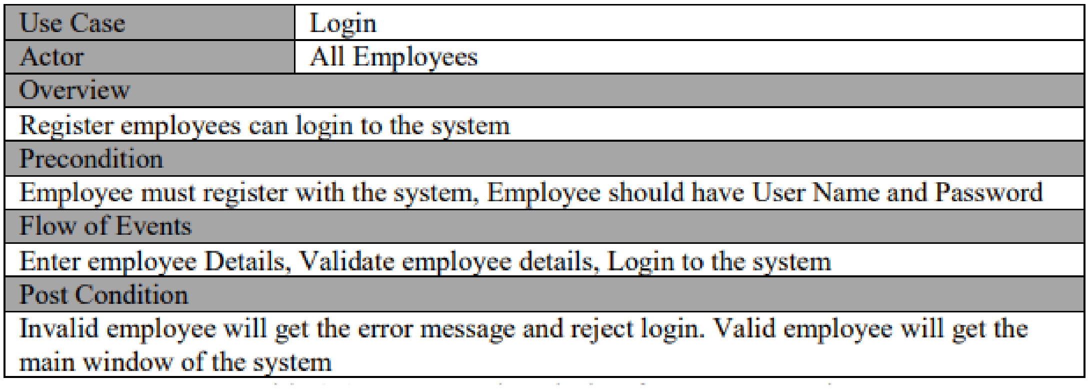


Fig 3.9 is for identified use cases for companies registered in online job search portal. Its express the deferent use cases that companies available for hiring on the system.

3.6.4 Use Case Description of Login

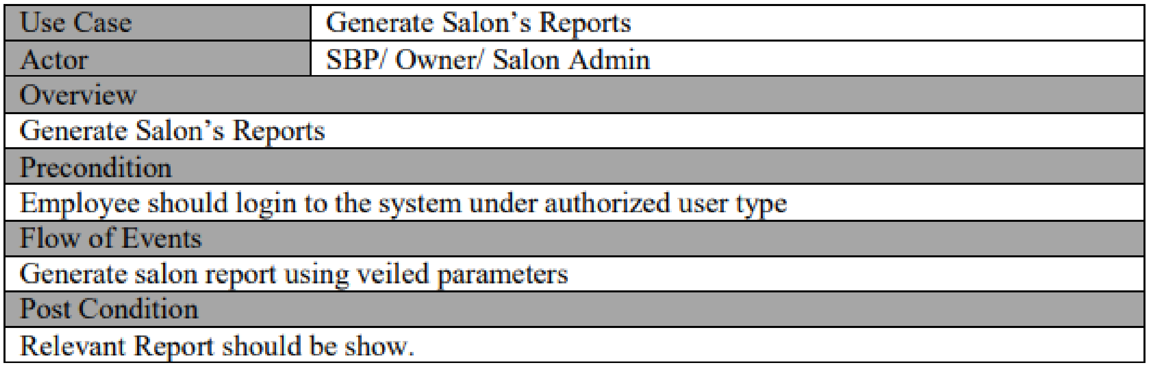
Bellow table 3.1 is Use case description for the login use case.



Use Case	Login
Actor	All Employees
Overview	
Register employees can login to the system	
Precondition	
Employee must register with the system, Employee should have User Name and Password	
Flow of Events	
Enter employee Details, Validate employee details, Login to the system	
Post Condition	
Invalid employee will get the error message and reject login. Valid employee will get the main window of the system	

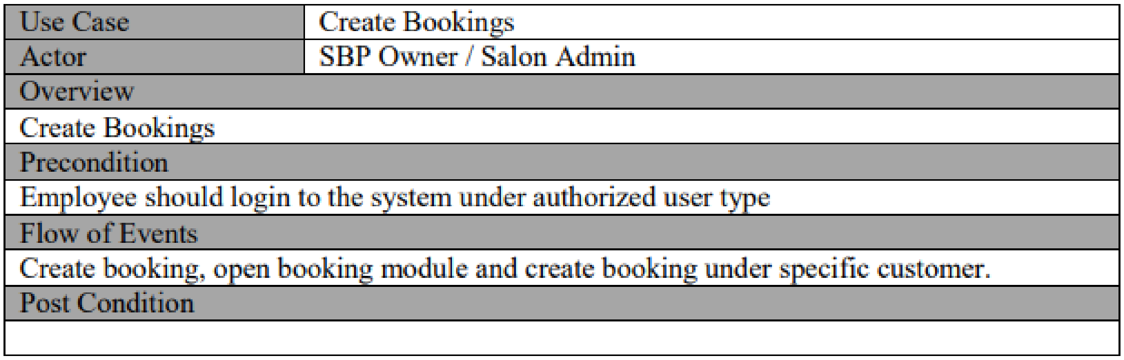
3.6.5 Use Case Description of Generate Job Reports

Table 3.2 is the use case description for the Generate Salon's Reports use case.



Use Case	Generate Salon's Reports
Actor	SBP/ Owner/ Salon Admin
Overview	
Generate Salon's Reports	
Precondition	
Employee should login to the system under authorized user type	
Flow of Events	
Generate salon report using veiled parameters	
Post Condition	
Relevant Report should be show.	

3.6.6 Use Case Description of Companies Registration Use Case



Use Case	Create Bookings
Actor	SBP Owner / Salon Admin
Overview	
Create Bookings	
Precondition	
Employee should login to the system under authorized user type	
Flow of Events	
Create booking, open booking module and create booking under specific customer.	
Post Condition	

3.7 ENTITY RELTIONSHIP DIAGRAM (ERD)

- This document is an entity-relationship diagram, or “ERD,” for a system to manage Inventory Management System.

- An ERD is a model that identifies the concepts or entities that exist in a system and the relationships between those entities. Manage Pages Manage Appointment (Accept / reject) Update Own Profile Change Password Add New Customer Generate Invoices Generate Reports

- An ERD is often used as a way to visualize a relational database: each entity represents a database table, and the relationship lines represent the keys in one table that point to specific records in related tables.

- ERD may also be more abstract, not necessarily capturing every table needed within a database, but serving to diagram the major concepts and relationships.

- It may assist the database design process for an e-resource management system, but does not identify every table that would be necessary for an electronic resource management database.

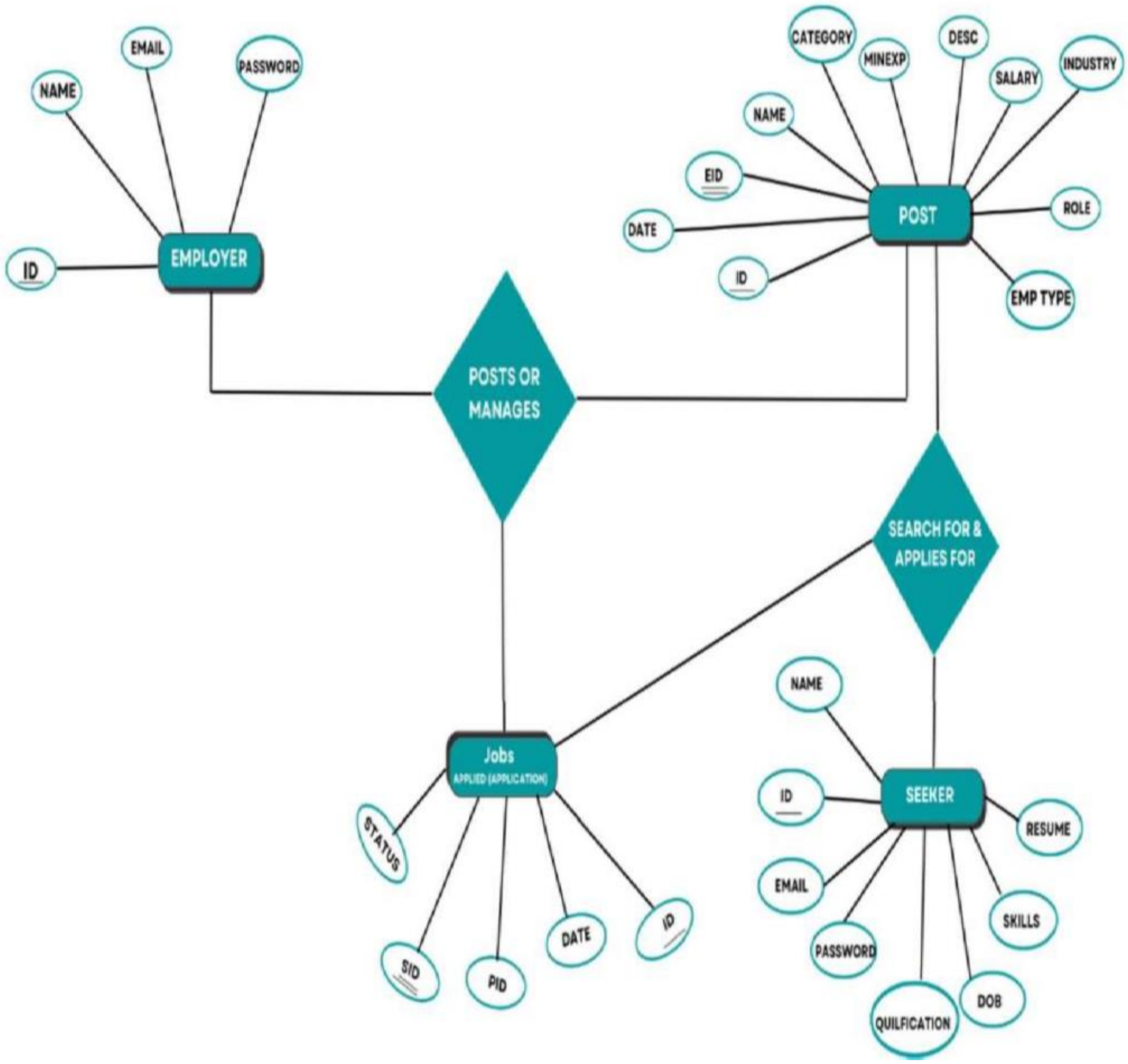


Fig 3.10 is for representation of ER (Entity-Relationship) diagram in online job search portal. Its express the deferent use cases that companies available for hiring on the system.

3.8 CODING

Creating a MERN stack (MongoDB, Express, React, Node.js) online job portal involves several steps.

Index.js

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" href="img/favicon.svg" type="image/x-icon">
<title> Jobvio.in | Home</title>
<link href="css/simpleGridTemplate.css" rel="stylesheet" type="text/css">
<link href="css/bootstrap.css" rel="stylesheet" type="text/css">
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<linkrel="stylesheet"href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
integrity="sha384-
mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"
crossorigin="anonymous">
<linkrel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<link href="css/animate.min.css" rel="stylesheet" type="text/css">
<!--FONTS-->
<link href="https://fonts.googleapis.com/css2?family=Sora:wght@200&display=swap"
rel="stylesheet">
<style>
.tiltContain{margin-top:0%;}
.btnTilt{height: 75px;background:rgba(225,225,225,0.2) ; color:white; font-family:
Sora;}
.textDarkShadow{text-shadow: 0px 0px 3px #000,3px 3px 5px #003333;}
/*-------------------------------------------------------*/
.btn{cursor: pointer; transition: 0.8s;}.btn:hover{ transform: scale(1.1);}
/*---------------------------------------------------------------*/
.dm{padding-top: 100px;}
/*-----------------------------------------------------------------------*/
.mbbtn{width: 120px; height: 40px; background-color: #e9c46a; color: black;
transition:0.4s;}
```

```
.mbbtn:hover{transform:scale(1.08); background-color: #e9c46a; color: black;}
/*------------------------------------------------------------------------------------*/
.floating { animation-name: floating; animation-duration: 3s; animation-iteration-count:
infinite; animation-timing-function: ease-in-out;
margin-left: 30px; margin-top: 5px; }
@keyframes floating { 0% { transform: translate(0, 0px); }
50% { transform: translate(0, 15px); }
100%  { transform: translate(0, -0px); }   }
/*--------------------------------------------------------------------------------------------------------*/
.crd { height:320px; width:460px; border-radius:20px; cursor: pointer; transition:0.8s;}
.crd:hover { transform: scale(1.05);}
/*------------------------------------------------------------------------------------------------------------------------------*/
</style>
<body onload="logoBeat()" style="font-family: 'Sora', sans-serif;">
<?php
include 'navBar.php';
include 'signinEmployerModals.php';
?>
<!-- Main Container -->
<div class="container-fluid" style=" background-color: #18303B; background-position:
center; background-size: cover; background-repeat: no-repeat; background-attachment:
fixed;">
<div class="hero" style=" color:whitesmoke; height: 1700px;">
<div>
<!--images in home page-->
<div  class="floating-area">
<img class="floating" src="img/homepage SVG.svg" width="42%" style="padding-
top:90px; padding-left: 50px; padding-bottom:100px; float: left;"/>
</div>
<!-- Texts in home page with heading and sub texts-->
<div class="animated slideInRight">
<h1 style="padding-top:250px; padding-right:290px; font-size: 30px; float: right;">
<strong style="font-family: 'Schoolbell', cursive; font-family:  'Vollkorn',  serif;">Find
the career you deserve!</strong>
</h1>
<h3 style="padding-top:10px; padding-right:285px; font-size: 20px; color:white; float:
right;">
Find out what you like doing best and <br>get someone to pay you for doing it.<br>
```

```
<br> Find the job meant for you. <b style="color: #e9c46a">#jobvio</b>
</h3>
<div class="butn" style="padding-top:15px; padding-right:530px;float:right;">
<button type="button" class="btn" style="width:120px; height:40px; background-
color:#e9c46a; border-radius: 0 !important;">
<a href = "mailto: rakshith.19cs121@sode-edu.in" style="color:black;"> Say Hi! </a>
</button>
</div>
</div>
<div style="width: 100%" class="row" >
<div class="col-md-9" >
<div style=" margin-top: 30px; padding-left: 50px;">
<h1 id="jbs">Find jobs</h1>
<form class="example" action="index.php">
<input style="color:#000; height:45px; width:800px; border-radius:30px 0px 0px
30px;" type="text" placeholder="     Search For Jobs.." name="q">
<button type="submit" style="height:45px; width:160px; border-radius:0px 30px 30px
0px; background-color: #257059; "><i class="fa fa-search bb"></i></button>
</form>
</div>
\<div class="container" style="padding-left:50px; padding-top:50px; padding-
bottom:50px;">
<!---------------------SUM OF POSTS & ACTIVE USERS USING VIEW TABLES-----
-------------------------->
<!-- sum of posts -->
<?php
include 'connect.php';
$sql = "select * from `totalposts`";
$totalresult = $conn->query($sql);
if ($totalresult->num_rows > 0) {
while($row = $totalresult->fetch_assoc())
{
$numberofposts = $row['AllPosts'];
?>
<h3 style="font-family: 'Schoolbell', cursive; font-family: 'Vollkorn', serif; color:
#f2cc8f; padding-left:4px;"> Total Job Posts Available: <?php echo $numberofposts; ?>
</h3>
<?php }} ?>
<!-- active users -->
<?php
include 'connect.php';
$sql = "select * from `totalactiveusers`";
$userresult = $conn->query($sql);
if ($userresult->num_rows > 0) {
```

```
while($row = $userresult->fetch_assoc())
{
$numberofusers = $row['TotalActiveUsers'];
?>
<h3 style="font-family: 'Schoolbell', cursive; font-family: 'Vollkorn', serif; color:
#f2cc8f; padding-left:4px;"> Active Users: <?php echo $numberofusers; ?> </h3>
<?php }} ?>
<!-------------------------------------------------------------------------------------------------------->
<div class="row">
<?php
$name=$category=$minexp=$salary=$industry=$desc=$role=$eType=$status="";
include 'connect.php';
$sql = "select *,(select name from employer where id=post.eid)as ename from post
order by date";
if(isset($_GET['q']))
{
$sql = "select *,(select name from employer where id=post.eid)as ename from post
where name LIKE '%".$_GET['q']."%' order by date";
}
if(isset($_GET['industry']))
{
$sql = "select *,(select name from employer where id=post.eid)as ename from post
where industry='".$_GET['industry']."' order by date";
}
if(isset($_GET['category']))
{
$sql = "select *,(select name from employer where id=post.eid)as ename from post
where category='".$_GET['category']."' order by date";
}
$result = $conn->query($sql);
if($result->num_rows>0)
{
while(   $row=$result->fetch_assoc())
{
$pid= $row['id'];
$jobtitle= $row['name'];
$category=$row['category'];
$minexp=$row['minexp'];
$salary=$row['salary'];
$industry=$row['industry'];
$desc=$row['desc'];
$role=$row['role'];
$ename =$row['ename'];
$status=$row['status'];
?>
<div class="col-md-4 crd" style="margin: 20px; background-color:#257059; padding:
20px;">
<h3 style="color: #e9c46a"> <b> <?php echo $jobtitle;?> <b> </h3>
<!------------------------------------------------------------------------------>
<h5 style="color:#99D98C">By <?php echo $ename;?> </h5> <br>
<!------------------------------------------------------------------------------>
<h5> <b style="color:#F8D4A7">Job Description:</b> <br> </h5>
<h5><?php echo $desc;?></h5>
<!------------------------------------------------------------------------------>
<h5><b style="color:#F8D4A7">Experience Required:</b>
<?php echo $minexp;?> years </h5>
<!------------------------------------------------------------------------------>
<h5><b style="color:#F8D4A7">Salary:</b>
<?php echo $salary;?> </h5> <br>
<h5><b style="color:#F8D4A7">Last date to apply:</b>
<?php echo $role;?> </h5> <br>
<!------------------------------------------------------------------------------>
<a href="applyJob.php?id=<?php echo $pid;?>" class="pull-right" style="font-family:
'Sora', sans-serif; color:#e9c46a;" ><h3><strong>Apply</strong></h3></a>
</div>
<!–    ------------------------------------------------------------------------------------------->
<!--    Error message -->
<?php }}else{
echo '<div class="" style="justify-content:center;">';
echo ' <img src="img/err.svg" width=600px /> ';
echo '</div>';
} ?>
</div>
</div>
</div>
<div style=" height: 100vh; " class="col-md-3">
<br><br>
<div class="animated slideInRight dm">
<img class="floating" src="img/rytsrc.svg" style="width:300px; height: 300px;"/>
</div>
<br><br>
<div style="padding-top:10px; padding-right:30px;">
<h3>Jobs By Category</h3>
<form>
<div>
<select class="form-control" name='category' style="border-radius:0px;">
<?php include "categoryOptions.php";?>
</select><br>
```

```
<input class=" btn-success pull-right mbbtn" type="submit" value="Search"
style="border-radius:0px;"/>
</div>
</form>
</div><br><br>
<br><br>
<div style="padding-top:10px; padding-right:30px;">
<h3>Jobs By Industry</h3>
<form>
<select class="form-control" name='industry' style="border-radius:0px;">
<?php include "industryOptions.php";?>
</select><br>
<input class=" btn-success pull-right mbbtn" type="submit" value="Search"
style="border-radius:0px;"/>
</form>
</div>
</div>
</div>
</div>
</div>
<!--first row -->
<script src="js/tilt.jquery.min.js"></script>
<script src="js/signinModal.js"></script>
<?php include 'footer.php';?>
<button style="display:none;" type="button" class="btn btn-info btn-lg" data-
toggle="modal" data-target="#msgModal" id="msgModalBtn">Open Modal</button>
<!-- Modal -->
<div id="msgModal" class="modal fade" role="dialog">
<div class="modal-dialog">
<!-- Modal content-->
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal">&times;</button>
<?php if(isset($_GET['msg'])){
$msg= $_GET['msg'];
if($msg=='success')
{
echo  "<h4 class='modal-title'>Job Applied Successfully!</h4>";
}
else if($msg=='error')
{
echo "<h4 class='modal-title'>Some Error occured Pls try again later!</h4>";
}
else if($msg=='dup')
{
echo "<h4 class='modal-title'>You have already applied for this job.\n "
"Check your application status in 'Jobs Applied' section</h4>";
}
}?>
</div>
</div>
</div>
</div>
<?php
if(isset($_GET['msg']))
{
if($_GET['msg']=='login')
{
?>
<script>
$('#loginAnchor').trigger( "click" );
</script>
<?php }else{
?>
<script>
$('#msgModalBtn').trigger( "click" );
</script>
<?php
}
}?>
</body>
</html>
```

Appliedjobs.js

```
<?php include 'authorizeSeeker.php';?>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" href="img/favicon.svg" type="image/x-icon">
<title> Jobs Applied | Job Seeker</title>
<link href="css/simpleGridTemplate.css" rel="stylesheet" type="text/css">
<link href="css/bootstrap.css" rel="stylesheet" type="text/css">
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<link type="text/css" rel="css/bootstrap.min.css">
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
```

```
<script type="text/javascript" src="js/jquery-3.3.1.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.min.js"></script>
<script type="text/javascript" src="js/dataTables.bootstrap.min.js"></script>
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<link href="css/animate.min.css" rel="stylesheet" type="text/css">
<link href="https://cdn.datatables.net/1.10.19/css/dataTables.bootstrap.min.css"
rel="stylesheet" type="text/css">
<link href="css/kodchasan.css" rel="stylesheet">
<!--FONTS-->
```

Postjob.js

```
<?php
// Create connection
include 'authorizeEmployer.php';
$id=0;
$name=$category=$minexp=$salary=$industry=$desc=$role=$eType=$status=$msg=""
;
if($_SERVER['REQUEST_METHOD']=='POST'||
$_SERVER['REQUEST_METHOD'] == 'GET') {

include 'connect.php';
if(isset($_GET['update'])&& isset($_GET['id'])){
$id = $_GET['id'];
$sql="select * from post where eid=$eid and id=$id";
$result = $conn->query($sql);
if(   $row=$result->fetch_assoc()){
$name= $row['name'];
$category=$row['category'];
$minexp=$row['minexp'];
$salary=$row['salary'];
$industry=$row['industry'];
$desc=$row['desc'];
$role=$row['role'];
$eType=$row['eType'];
$status=$row['status'];
}
}
if(isset($_POST['submitPost'])){
$id= $_POST['id'];
$name= $_POST['name'];
$category=$_POST['category'];
$minexp=$_POST['minexp'];
$salary=$_POST['salary'];
$industry=$_POST['industry'];
$desc=$_POST['desc'];
$role=$_POST['role'];
$eType=$_POST['eType'];
$status=$_POST['status'];
if($id>0){
$sql = "Update `post` set `date`=CURRENT_DATE(),"
. "`name`='$name', "
. "`category`='$category', "
. "`minexp`='$minexp', "
. "`desc`='$desc', "
. "`salary`='$salary', "
. "`industry`='$industry', "
. "`role`='$role', "
. "`employmentType`='$eType', "
. "`status`= '$status' "
. "where id=$id and eid=$eid;";
}else{
$sql = "INSERT INTO `post` (`id`, `date`, `eid`, `name`, `category`, `minexp`, `desc`,
`salary`, `industry`, `role`, `employmentType`, `status`) "
. "VALUES (NULL, CURRENT_DATE(), '$eid', '$name', '$category', '$minexp',

'$desc', '$salary', '$industry', '$role', '$eType', '$status');";
}
if ($conn->query($sql) === TRUE) {
if($_GET['update']){
header('location: employerAccount.php');
}else{
$msg="New Post has been created successfully";
}
} else {
$msg="Error: " . $sql . "<br>" . $conn->error;
}
}
}
?
```

```
 <link href="https://fonts.googleapis.com/css2?family=Sora:wght@200&display=swap"
 rel="stylesheet">
<style>
.tiltContain{margin-top:0%;}
.btnTilt{height: 75px;background:rgba(225,225,225,0.2) ; color:white; font-family:
Comfortaa;}
.textDarkShadow{
text-shadow: 0px 0px 3px #000,3px 3px 5px #003333;
}
```

Postcreated.html

```
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" href="img/favicon.svg" type="image/x-icon">
<title> Create A Post | Employer</title>
<link href="css/simpleGridTemplate.css" rel="stylesheet" type="text/css">
<link href="css/bootstrap.css" rel="stylesheet" type="text/css">
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<linkrel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script  src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<link href="css/animate.min.css" rel="stylesheet" type="text/css">
<!--FONTS-->
<link href="https://fonts.googleapis.com/css2?family=Sora:wght@200&display=swap"
rel="stylesheet">
<style>
</style>
<?php
include 'navBar.php';
?>
```

```
<div style="background-color: #FF5757; width: 100%; height: 650px; padding-top:
200px; text-align: center;">
<h1 style="font-size: 80px;">Your Post Published!!</h1>
<img src="img/5.gif" width="300" height="300"/>
</div>
<?php
include 'footer.php';
?>
```

EmployeerAccount.js

```
<?php include 'authorizeEmployer.php';?>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" href="img/favicon.svg" type="image/x-icon">
<title> Account | Employer</title>
<link href="css/simpleGridTemplate.css" rel="stylesheet" type="text/css">
<link href="css/bootstrap.css" rel="stylesheet" type="text/css">
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<linkrel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<script src="https://code.jquery.com/jquery-3.3.1.js"></script>
<script src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/1.10.19/js/dataTables.bootstrap.min.js"></script>
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<link href="css/animate.min.css" rel="stylesheet" type="text/css">
<linkhref="https://cdn.datatables.net/1.10.19/css/dataTables.bootstrap.min.css"
rel="stylesheet" type="text/css">
<!--FONTS-->
<link href="https://fonts.googleapis.com/css2?family=Sora:wght@200&display=swap"
rel="stylesheet">
<style>
.tiltContain{margin-top:0%;}
.btnTilt{height: 75px;background:rgba(225,225,225,0.2) ; color:white; font-family:
Comfortaa;}
.textDarkShadow{
text-shadow: 0px 0px 3px #000,3px 3px 5px #003333;
}
.pc {
```

```
animation-name: pc;
animation-duration: 3s;
animation-iteration-count: infinite;
animation-timing-function: ease-in-out;
margin-left: 30px;
margin-top: 5px;
```

SeekerAccount.js

```
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" href="img/favicon.svg" type="image/x-icon">
<title> Account | Job Seeker</title>
<link href="css/simpleGridTemplate.css" rel="stylesheet" type="text/css">
<link href="css/bootstrap.css" rel="stylesheet" type="text/css">

<link href="css/Animate.css" rel="stylesheet" type="text/css">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link href="css/Animate.css" rel="stylesheet" type="text/css">
<link href="css/animate.min.css" rel="stylesheet" type="text/css">
<!--FONTS-->
<link href="https://fonts.googleapis.com/css2?family=Sora:wght@200&display=swap"
rel="stylesheet">
<style>
.tiltContain{margin-top:0%;}
.btnTilt{height: 75px;background:rgba(225,225,225,0.2) ; color:white; font-family:
Sora;}
.textDarkShadow{
text-shadow: 0px 0px 3px #000,3px 3px 5px #003333;
}
.pc {
animation-name: pc;
animation-duration: 3s;
animation-iteration-count: infinite;
animation-timing-function: ease-in-out;
margin-left: 30px;
margin-top: 5px;
}
@keyframes pc {
```

```
0% { transform: translate(0,  0px); }
50% { transform: translate(0, 15px); }
100% { transform: translate(0, -0px); }
}
</style>
<body onload="logoBeat()" style="font-family: 'Sora', sans-serif;">
<?php
include 'navBar.php';
include 'signinEmployerModals.php';
?>
<!-- Main Container -->
<div class="container-fluid"  style="background-color:#FFDAB9;">
<?php
include 'connect.php';
$sid = $_SESSION["sid"];
$sqlE = "select * from seeker where id = '$sid' ;";
$resultE = $conn->query($sqlE);
```

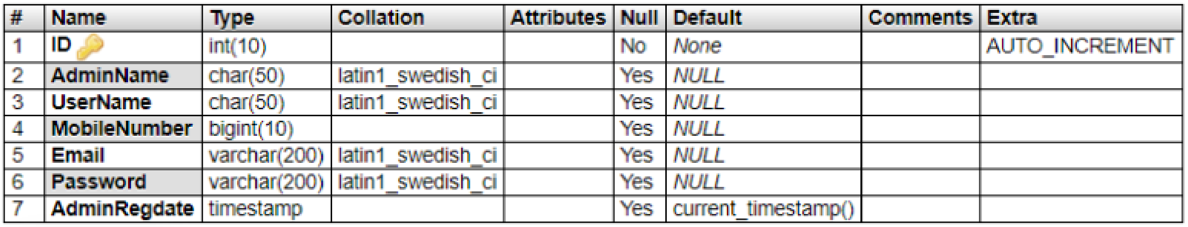
CHAPTER 4

PROPOSED WORK

4.1 Database Design

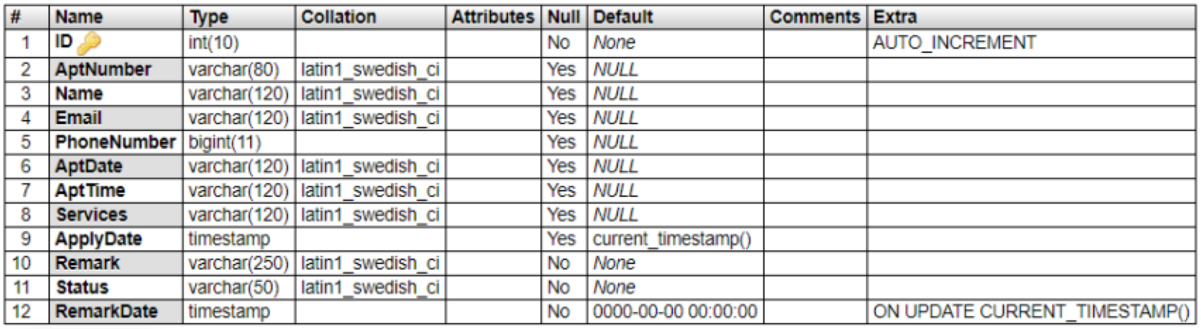
A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

Online Job Search Portal (MSMS) contains 7 MySQL tables :
tbladmin table Structure : This table store the admin personal and login details



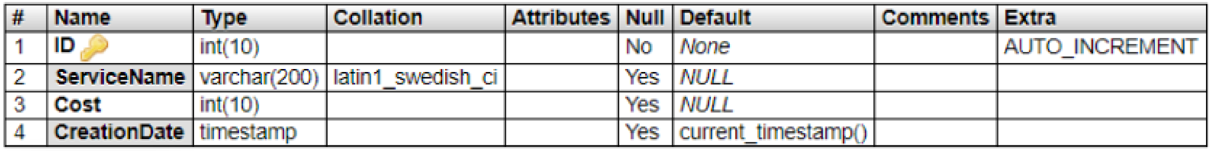
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	AdminName	char(50)	latin1_swedish_ci		Yes	*NULL*		
3	UserName	char(50)	latin1_swedish_ci		Yes	*NULL*		
4	MobileNumber	bigint(10)			Yes	*NULL*		
5	Email	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
6	Password	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
7	AdminRegdate	timestamp			Yes	current_timestamp()		

tblappointment table Structure : This table store the user appointment details.



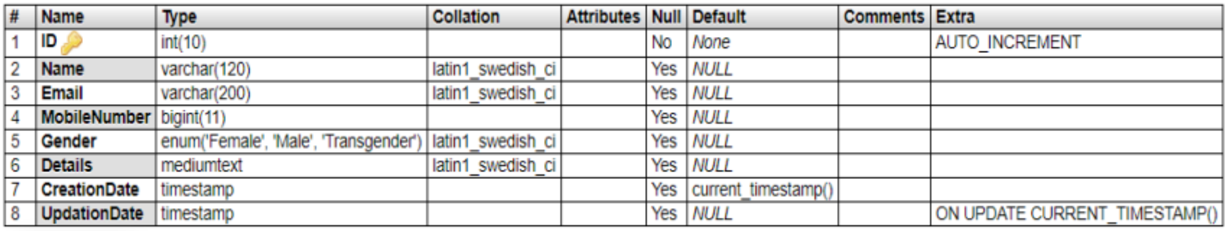
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	AptNumber	varchar(80)	latin1_swedish_ci		Yes	*NULL*		
3	Name	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
4	Email	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
5	PhoneNumber	bigint(11)			Yes	*NULL*		
6	AptDate	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
7	AptTime	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
8	Services	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
9	ApplyDate	timestamp			Yes	current_timestamp()		
10	Remark	varchar(250)	latin1_swedish_ci		No	*None*		
11	Status	varchar(50)	latin1_swedish_ci		No	*None*		
12	RemarkDate	timestamp			No	0000-00-00 00:00:00		ON UPDATE CURRENT_TIMESTAMP()

tblservices table Structure : This table store the services details.



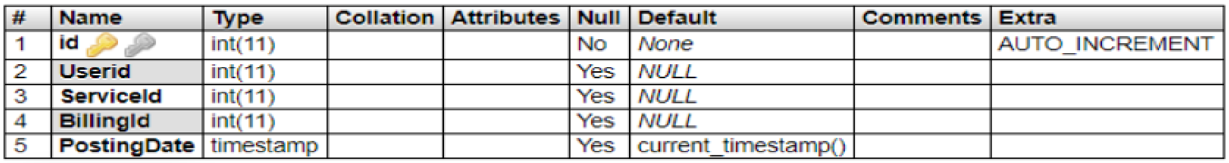
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID 🔑	int(10)			No	*None*		AUTO_INCREMENT
2	ServiceName	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	Cost	int(10)			Yes	*NULL*		
4	CreationDate	timestamp			Yes	current_timestamp()		

tblcustomers table Structure : This table store the customer details.



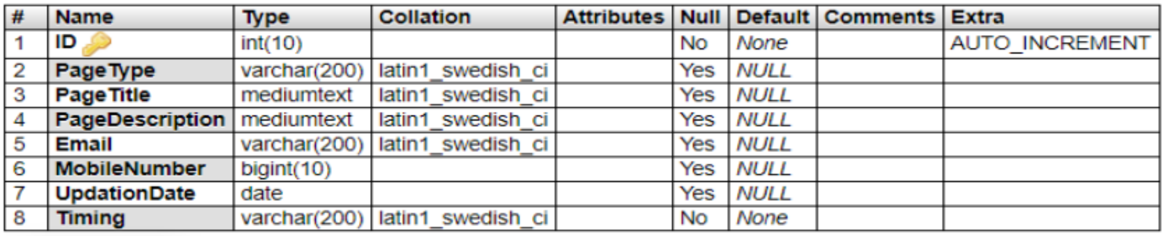
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID 🔑	int(10)			No	*None*		AUTO_INCREMENT
2	Name	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
3	Email	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
4	MobileNumber	bigint(11)			Yes	*NULL*		
5	Gender	enum('Female', 'Male', 'Transgender')	latin1_swedish_ci		Yes	*NULL*		
6	Details	mediumtext	latin1_swedish_ci		Yes	*NULL*		
7	CreationDate	timestamp			Yes	current_timestamp()		
8	UpdationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

tblinvoice table Structure : This table store the customer invoice details.



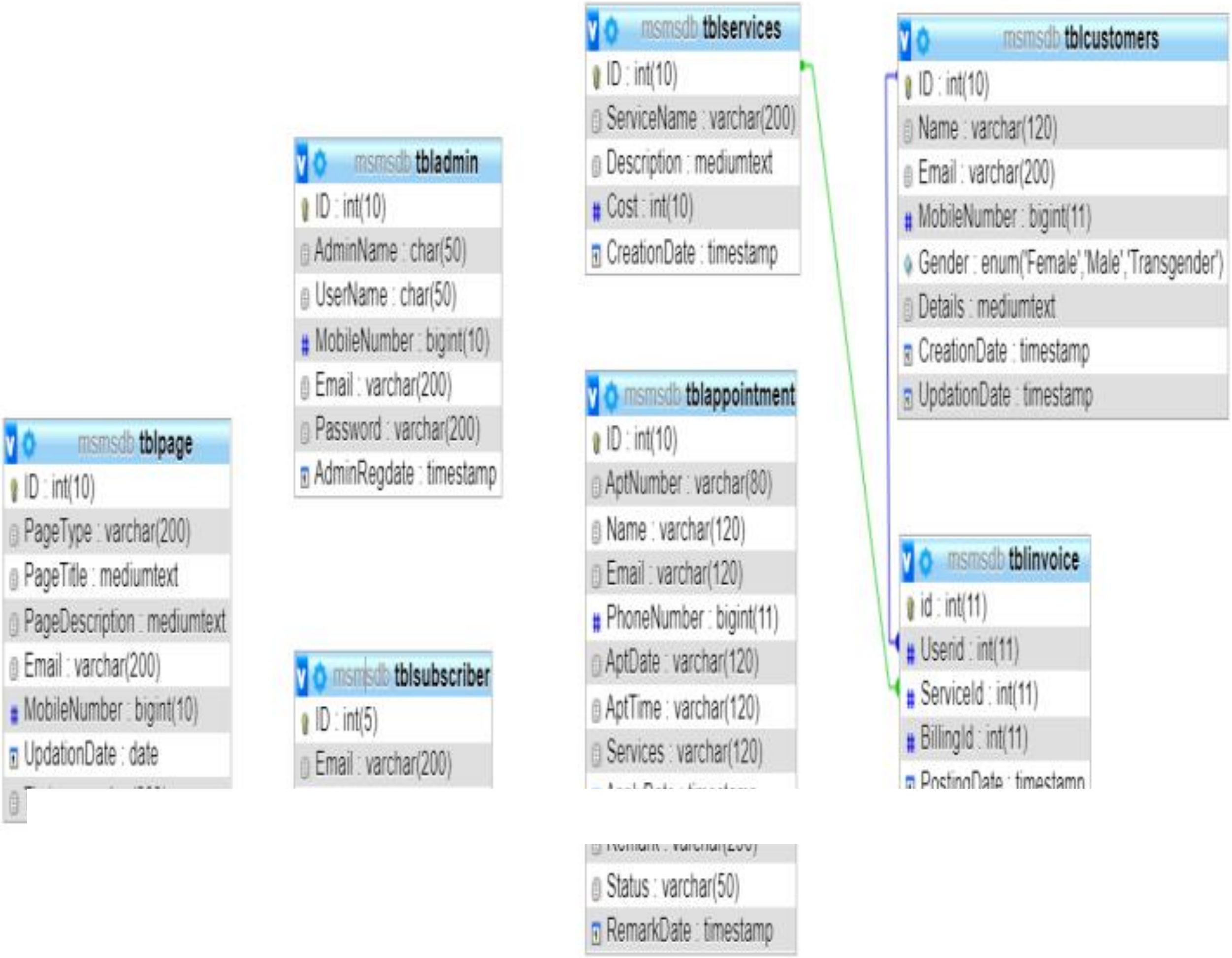
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	Userid	int(11)			Yes	*NULL*		
3	ServiceId	int(11)			Yes	*NULL*		
4	BillingId	int(11)			Yes	*NULL*		
5	PostingDate	timestamp			Yes	current_timestamp()		

tblpage table Structure : This table store the pages information.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID 🔑	int(10)			No	*None*		AUTO_INCREMENT
2	PageType	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	PageTitle	mediumtext	latin1_swedish_ci		Yes	*NULL*		
4	PageDescription	mediumtext	latin1_swedish_ci		Yes	*NULL*		
5	Email	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
6	MobileNumber	bigint(10)			Yes	*NULL*		
7	UpdationDate	date			Yes	*NULL*		
8	Timing	varchar(200)	latin1_swedish_ci		No	*None*		

tblsubscribe table Structure: This table store emails of subscribers



msmsdb tblpage
ID : int(10)
PageType : varchar(200)
PageTitle : mediumtext
PageDescription : mediumtext
Email : varchar(200)
MobileNumber : bigint(10)
UpdationDate : date

msmsdb tbladmin
ID : int(10)
AdminName : char(50)
UserName : char(50)
MobileNumber : bigint(10)
Email : varchar(200)
Password : varchar(200)
AdminRegdate : timestamp

msmsdb tblsubscriber
ID : int(5)
Email : varchar(200)

msmsdb tblservices
ID : int(10)
ServiceName : varchar(200)
Description : mediumtext
Cost : int(10)
CreationDate : timestamp

msmsdb tblappointment
ID : int(10)
AptNumber : varchar(80)
Name : varchar(120)
Email : varchar(120)
PhoneNumber : bigint(11)
AptDate : varchar(120)
AptTime : varchar(120)
Services : varchar(120)
Status : varchar(50)
RemarkDate : timestamp

msmsdb tblcustomers
ID : int(10)
Name : varchar(120)
Email : varchar(200)
MobileNumber : bigint(11)
Gender : enum('Female','Male','Transgender')
Details : mediumtext
CreationDate : timestamp
UpdationDate : timestamp

msmsdb tblinvoice
id : int(11)
Userid : int(11)
ServiceId : int(11)
BillingId : int(11)
PostingDate : timestamp

4.2 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control .

Figure 4.1 shows the "Employer Activity " in the system. It describes the activities between user and the system during the insert salon service to the system. In this activity diagram describes the how SBP staff add new service to the system.

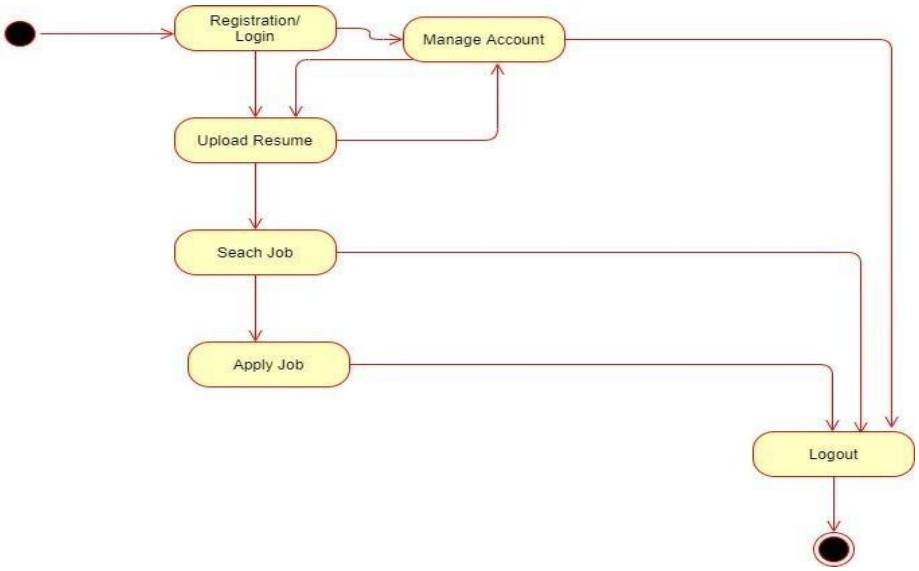
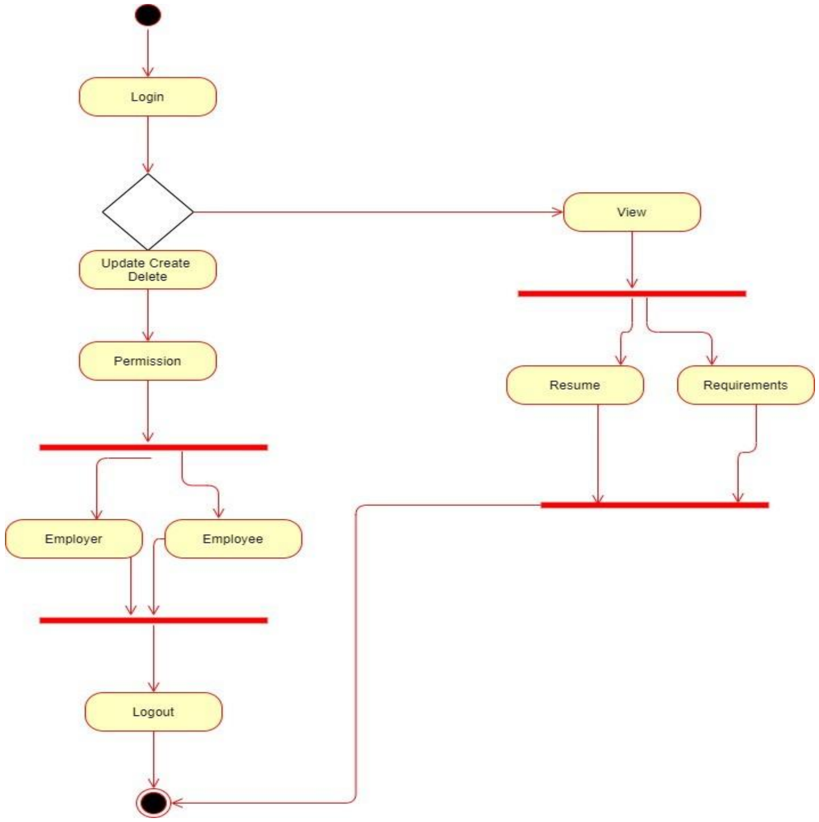


Figure 4.2. Employee Activity DiagramThis type of diagram helps in visualizing the sequence of actions,



decision points, and interactions between different tasks in a process. Here’s an explanation of how to create and understand an Employee Activity Diagram

.

4.2 Technology Description

- **Selection of Operating System:** Our website is platform independent, so it does not depend on the operating system.
- **Selection of Software:** Visual Studio is used to create our software.

4.3 Approach Used

Developing a Online Job Search Portal web application involves several key phases. First, gather requirements by meeting with salon owners and staff to understand their needs. Next, design the system architecture and user interface, considering factors like scalability and usability. Then, proceed with development, coding both the frontend and backend components using appropriate technologies and frameworks. Integrate third-party services as needed, such as payment gateways or SMS APIs.

Thoroughly test the application for functionality, usability, performance, and security. Once testing is complete, deploy the application to a production environment and provide training and support to salon staff. Maintain the application by applying updates and addressing user feedback to ensure it continues to meet the salon's needs over time. Throughout the process, communication with stakeholders is essential for success

4.4 Dependencies Required

For a web-based Online Job Search Portal, you'll need a combination of frontend and backend technologies, as well as dependencies for specific functionalities. Here's an overview:

Frontend Dependencies:

- **HTML/CSS/JavaScript:** Fundamental for building the structure, styling, and interactivity of the web application.
- **Frontend Framework/Library:** Consider using a framework like React.js, Angular, or Vue.js to streamline development and manage components efficiently.
- **UI Framework/Library:** You might want to use UI frameworks like Bootstrap, Material-UI, or Semantic UI to ensure consistent and responsive design across different devices.
- **AJAX/HTTP Library:** Libraries like Axios or Fetch API for making asynchronous HTTP requests to the backend server. and responsive Single Page Applications (SPAs) by allowing the mapping of components to different routes.

Backend Dependencies:

- **Server-Side Language**: Choose a language to handle server-side logic. Popular options include Node.js (JavaScript), Python (Django or Flask), Ruby (Ruby on Rails), or PHP (Laravel or Symfony).
- **Web Server:** Apache or Nginx to serve the web application.
- **Database:** Select a database system to store salon data. Options include relational databases like MySQL, PostgreSQL, or SQLite, or NoSQL databases like MongoDB.
- **Web-vitals:** Version 2.1.4 is used in this project. Web Vitals is a set of metrics that help measure the performance and user experience of a web application. It includes tools and libraries for capturing and reporting essential performance metrics.

4.5 Algorithms and Flowcharts

The description of major functionalities of the Mock Interview System are given below:

- **Sign up():**
  This function will make the user to sign up to the system.

Algorithm

```
1 save the requested name, email and password to a constant variable
2 write the regular expression to validate the password
3 try to find the entered email in the database
4 if user is not present in the database then go to line 5 else go to line 8.
5 if password entered is validated properly and is right then go to line 6 else go  line 7.
6 save the details of the user to the database and print “successfully registered message”.
7error message is given to the user.
8 message printed “User already exists.
```

Flowchart

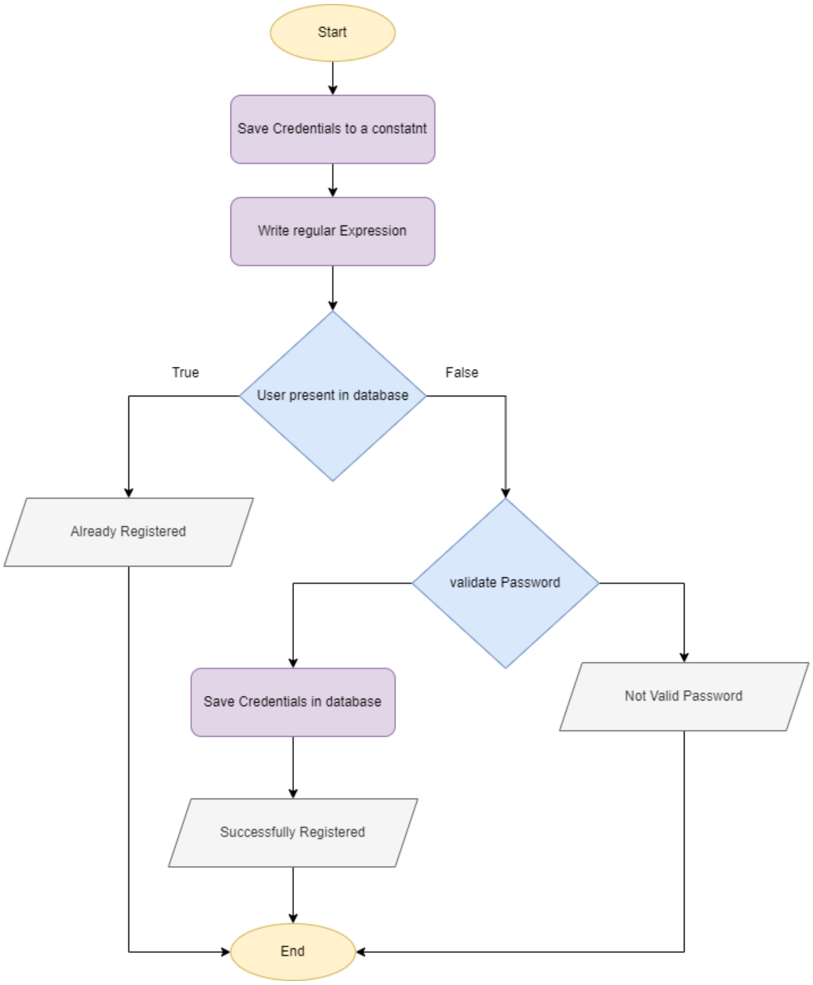


Fig. 4.4 Flowchart of signup process

- **LogIn():**
  This function will make the user to logged in to the system.

Algorithm

1 save the requested name, email and password to a constant variable
2 try to find the entered email in the database
4 if user found in database then go to line 5 else go to line 6.
5 open dashboard of the user.
6 error message is given to the user.

Flowchart

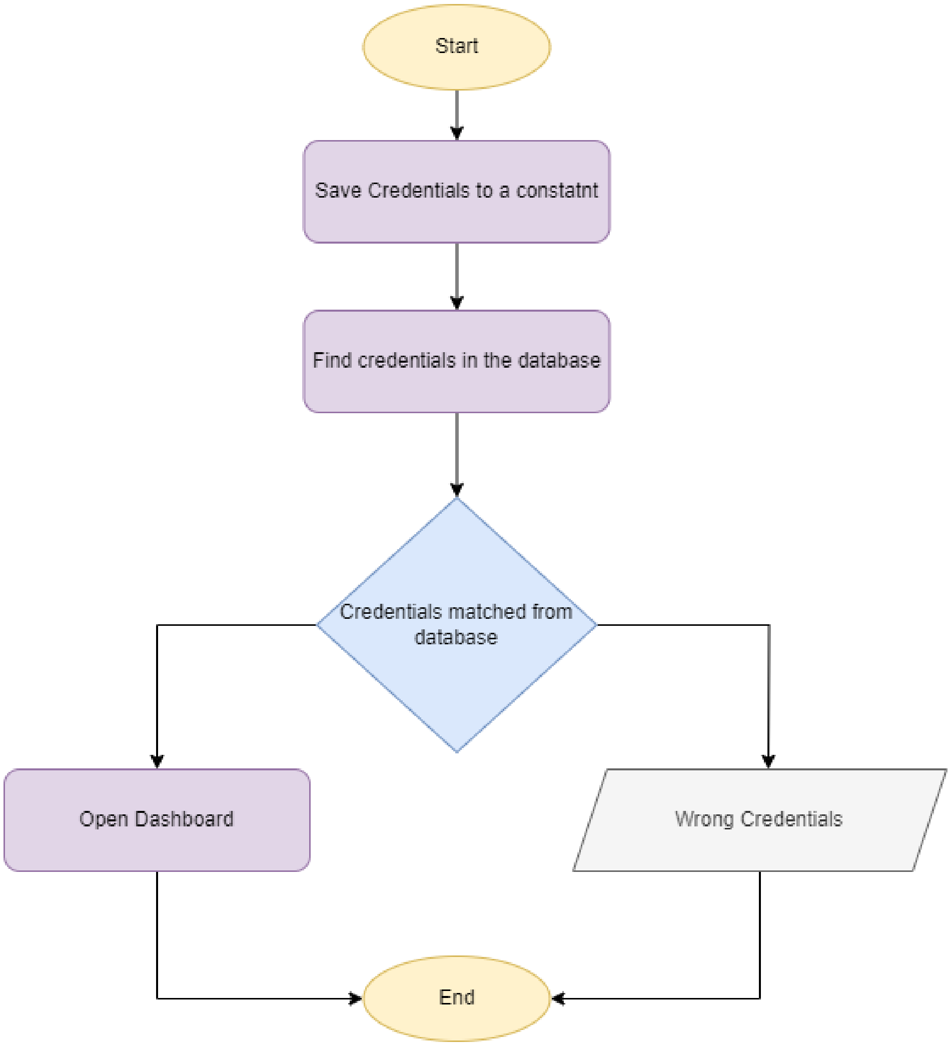

Start
Save Credentials to a constatnt
Find credentials in the database
Credentials matched from database
Open Dashboard
Wrong Credentials
End

Fig 4.6 Flowchart of LogIn

- **Database Connectivity ():**
  This function will connect the server to the database.

Algorithm

1 call require("mongoose") method and store the return value in a constant as mongoose.
2 call mongoose.Schema({token:String}) and store the return value in constant blackschema
3 call mongoose.model("black",blackschema) and store the return value in constant blackmodel
4 module.exports={blackmodel}

Flowchart

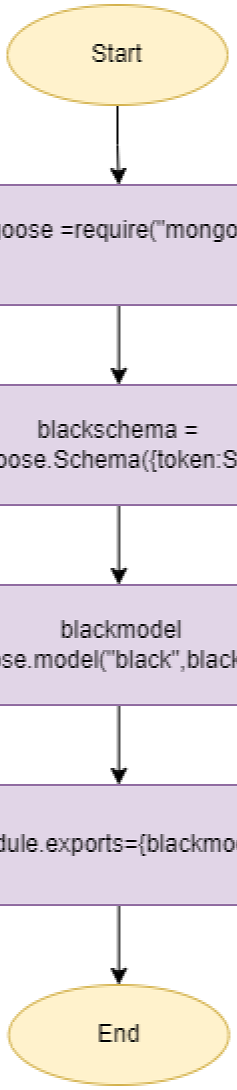

Start
mongoose =require("mongoose")
blackschema =
mongoose.Schema({token:String})
blackmodel
=mongoose.model("black",blackschema)
module.exports={blackmodel}
End

Fig 4.4 Flowchart of Database Connectivity

Flowchart

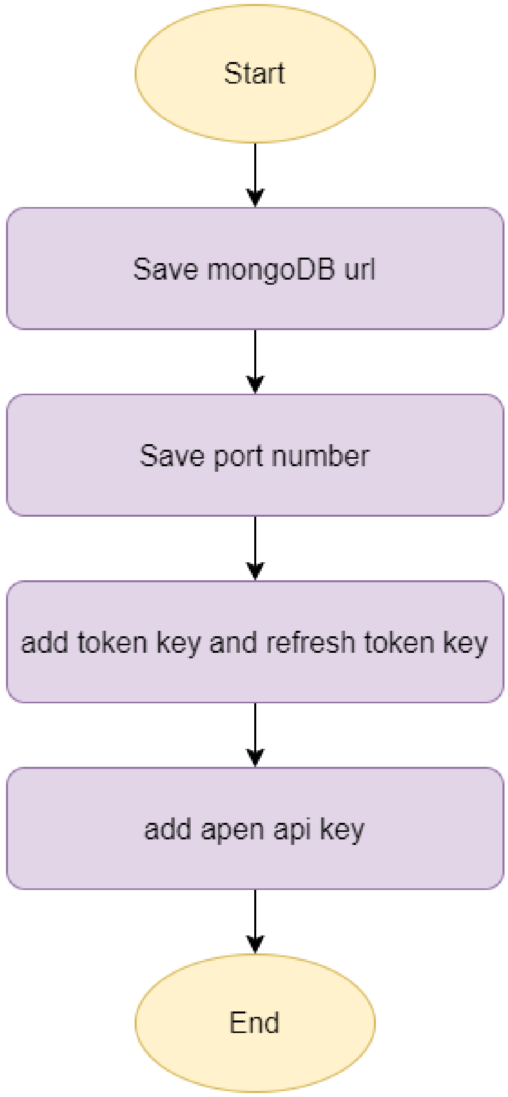


Fig 4.5 Flowchart of API Connectivity

- **Feedback Generation ():**
  This function will generate the feedback.

Algorithm

1 save the history by using HistoryModel.findById method
2 if history is not present then go to 3 else go to 4
3 return error status that history is not present go to end.
4 get the last conversation history
5 convert the last conversation to json.

Flowchart

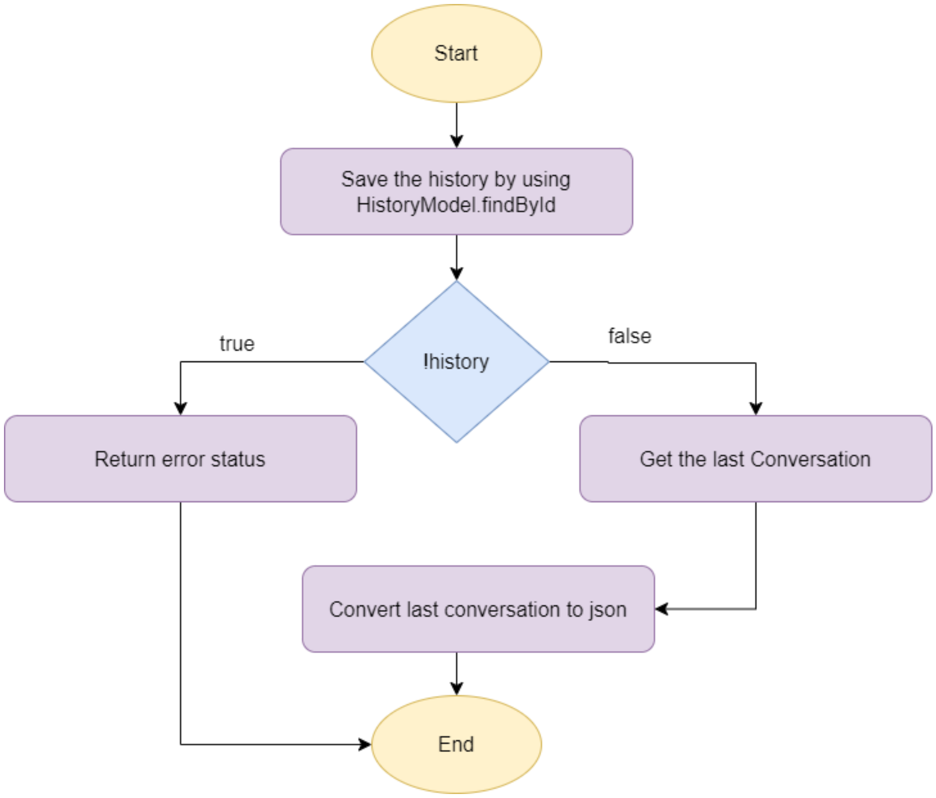

Start
Save the history by using HistoryModel.findById
!history
true
false
Return error status
Get the last Conversation
Convert last conversation to json
End

Fig. 4.6 Flowchart of Feedback Generation

CHAPTER 5

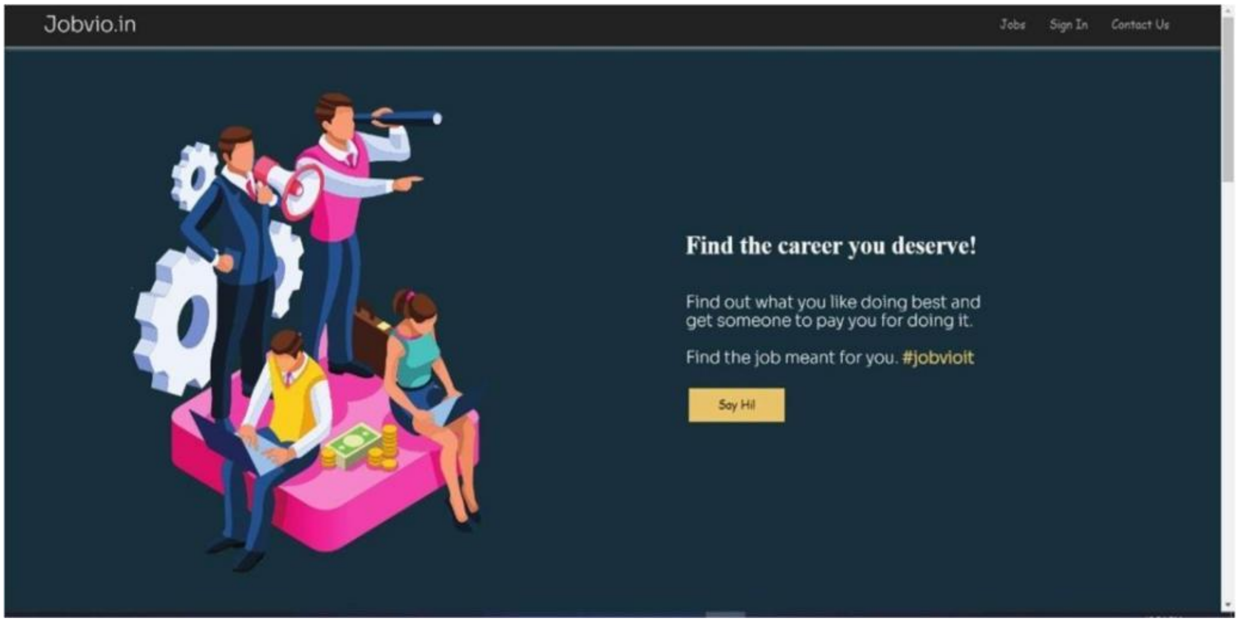
RESULTS

5.1Screens and Explanations

This chapter will include all the screens available in the project such as home page, registration page, login page, start new interview, interview customization, interview screen and response screen along with detailed explanation of each screen and its functionality. Screens available in the system are as follows:

Screen 1: Home Page

Screen 1 is the home page of the website which displays the basic information about the home page of the Online Job Search Portal web application offers a welcoming introduction to the platform. It features a visually engaging hero section with the salon's name and tagline, followed by key features highlighting appointment scheduling, client management, and more. Strategic call-to-action buttons prompt users to sign up or explore further. Contact information and a footer with essential links complete the page, creating a compelling overview of the platform's capabilities and encouraging user engagement.



Find the career you deserve!

Find out what you like doing best and get someone to pay you for doing it.

Find the job meant for you. #jobvioit

Fig.5.1 Home Page

Screen 2: SignIn Screen

Screen 2 is the log in and the registration page. The admin sign-in page of the Online Job Search Portal web application serves as the gateway for administrators to access the system's backend functionalities.

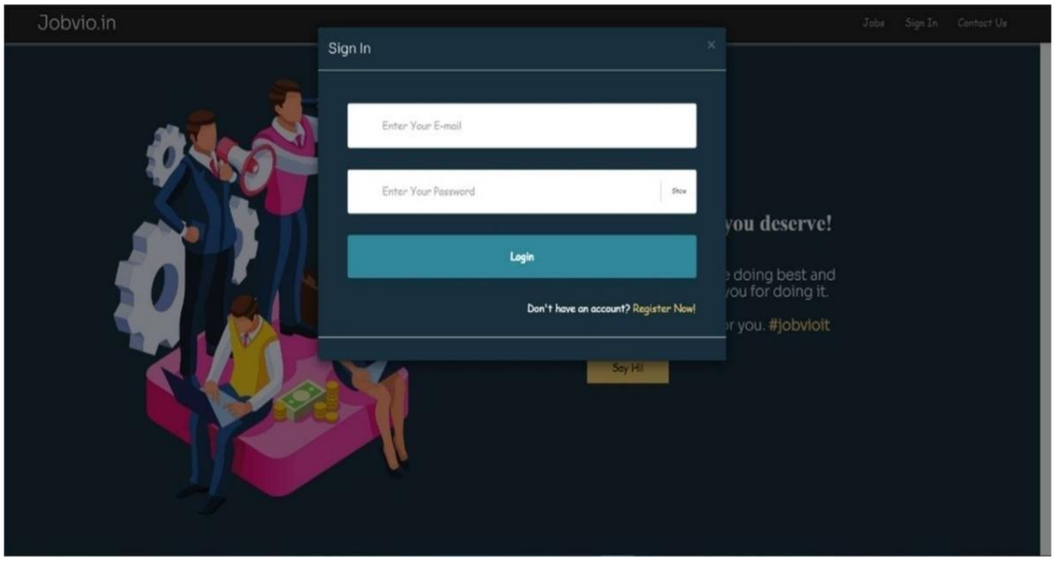


Fig 5.2 SignIn Page

Screen 3: Employer Registration page

Screen 3 is the page The dashboard of the Online Job Search Portal web application serves as a centralized hub where salon administrators and staff can access key information and perform various tasks efficiently.

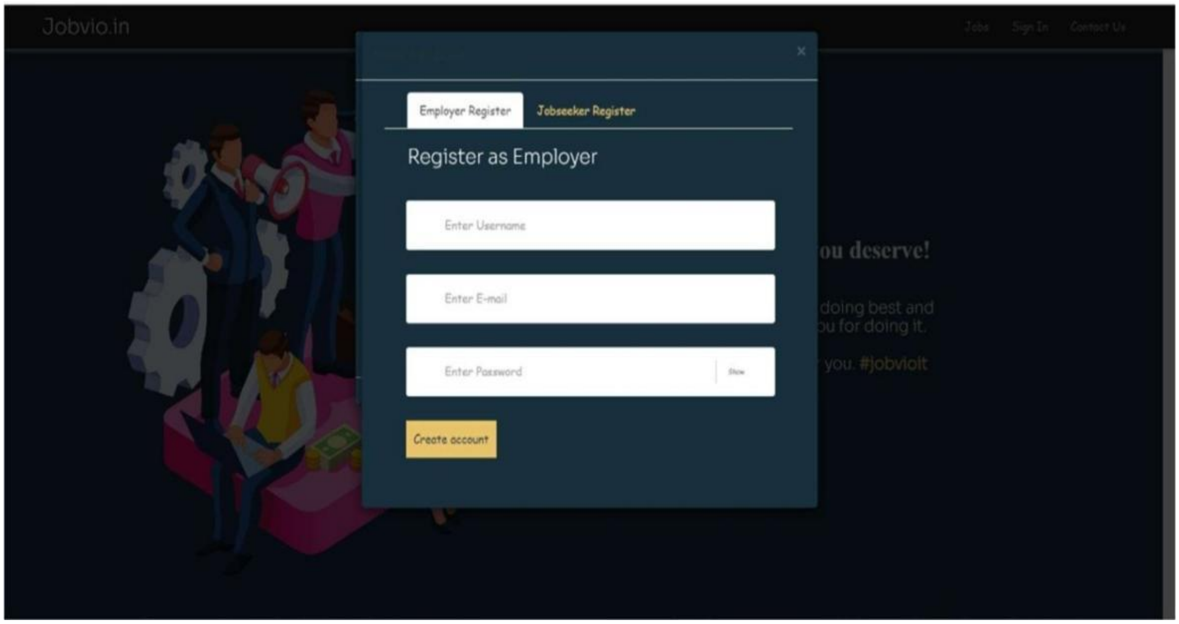
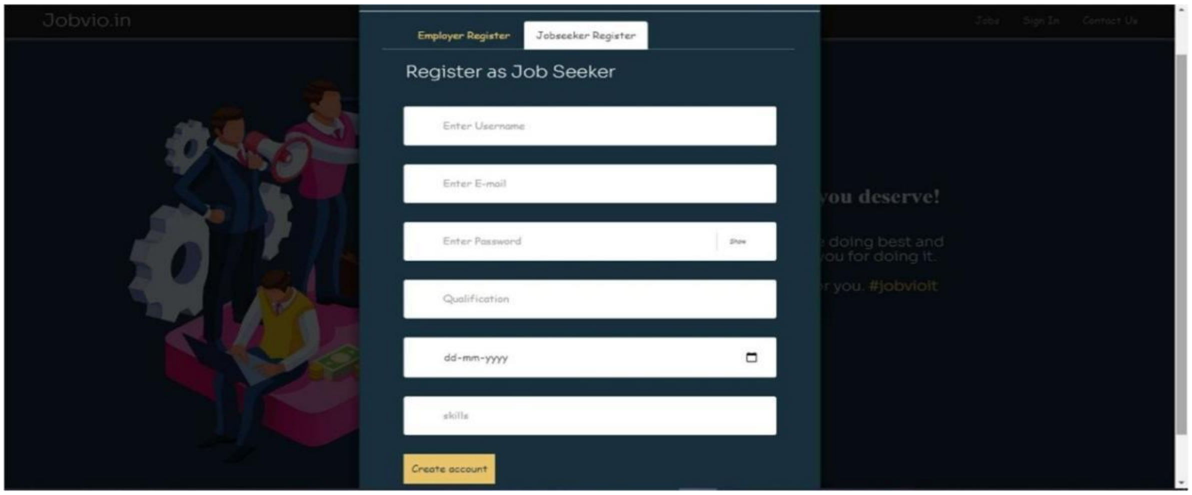


Fig 5.3 Employer Registration

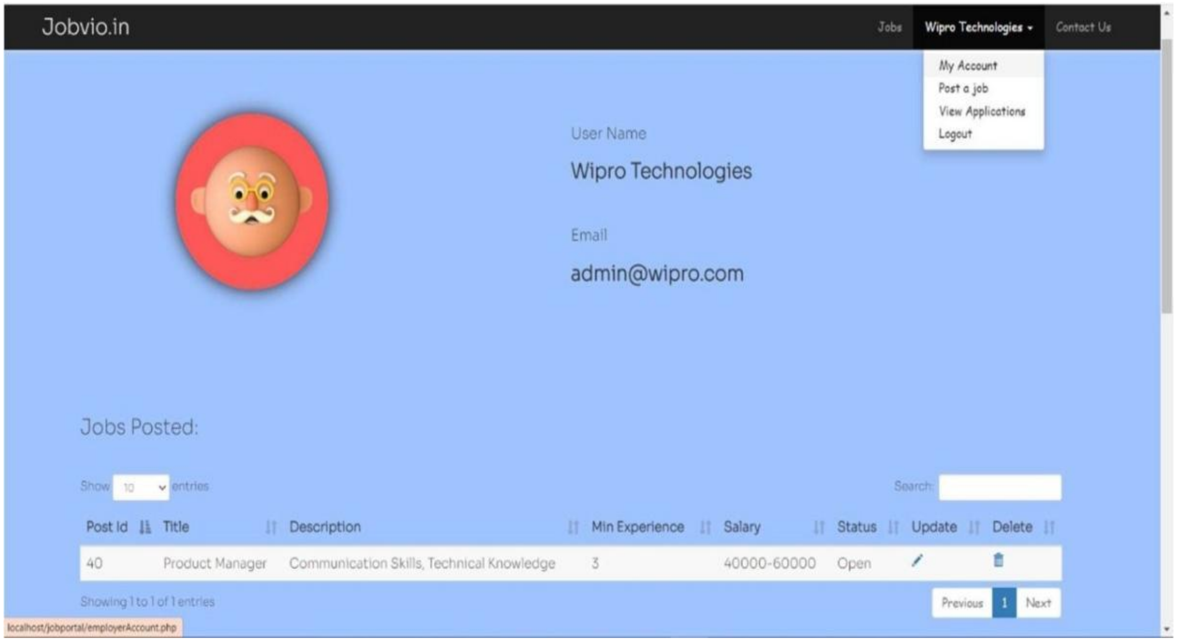
Screen 4: JobSeeker Registration Screen

Screen 4 is It includes features such as appointment scheduling, client and employee management, inventory tracking, billing and invoicing, reporting and analytics, marketing



Screen 5: Employer Dashboard

Screen 5 is the screen of the appointments feature serves as a centralized tool for managing the salon's schedule. Staff members can view their upcoming appointments, check availability, and make adjustments as needed.



User Name

Wipro Technologies

Email

admin@wipro.com

Jobs Posted:

Show 10 entries

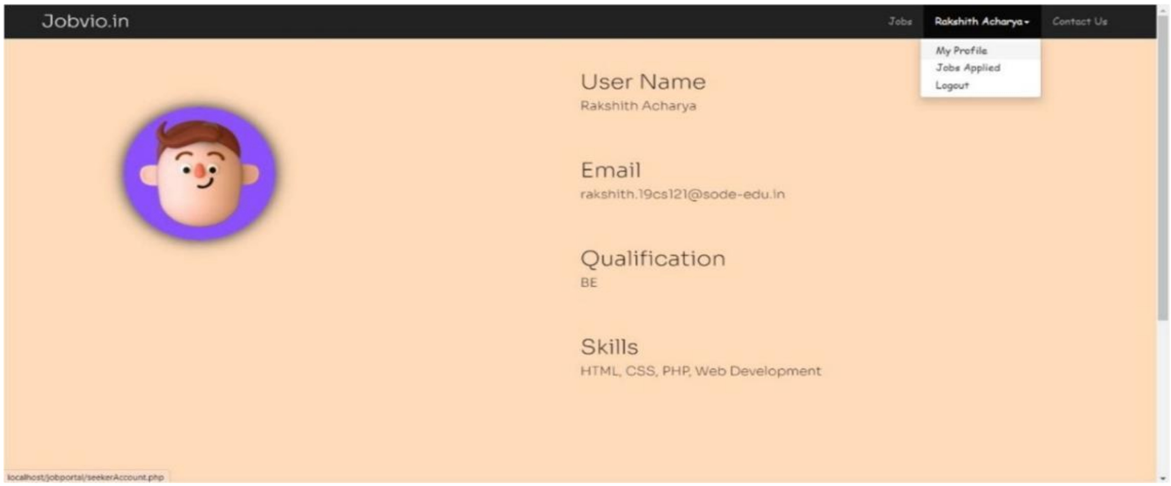
Search:

Post Id	Title	Description	Min Experience	Salary	Status	Update	Delete
40	Product Manager	Communication Skills, Technical Knowledge	3	40000-60000	Open	✎	🗑

Showing 1 to 1 of 1 entries

Fig 5.5 Employer Dashboard

Screen 6: Employer Account



User Name

Rakshith Acharya

Email

rakshith.19cs121@sode-edu.in

Qualification

BE

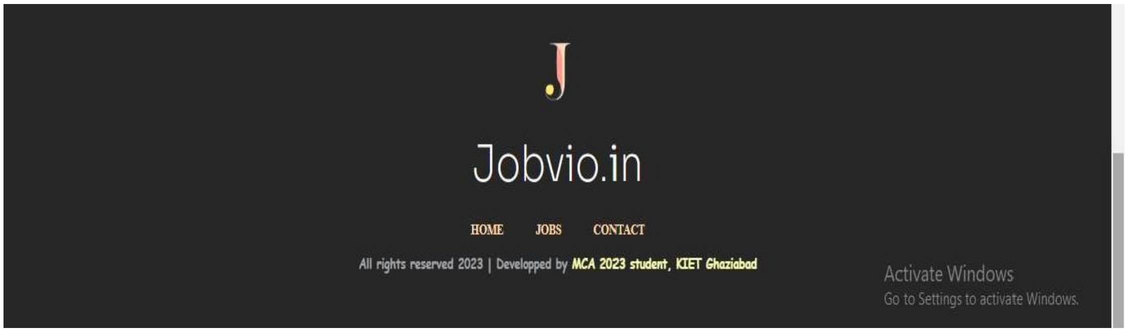
Skills

HTML, CSS, PHP, Web Development

Fig 5.6 Employer Account

Screen 7: Footer

Footer contains “HOME”, “JOBS”, “CONTACT” options. Where “HOME” is linked to the home page of the system. “JOBS” is linked to the job posts where all the job postsare available. “CONTACT” is linked to the admin’s email address.



All rights reserved 2023 | Developped by MCA 2023 student, KIET Ghaziabad

Activate Windows
Go to Settings to activate Windows.

Fig 5.7 Footer

CHAPTER 6

DISCUSSIONS

The Discussions section of this report delves into crucial aspects of the Online Job Search Portal. Online Job Search Portal web applications offer numerous benefits for salon owners and clients alike, but they also pose challenges related to internet dependency, customization limitations, data security, and integration complexity. However, with ongoing advancements in technology and a focus on user experience, the future prospects for Online Job Search Portals are promising, paving the way for greater efficiency, innovation, and growth in the salon industry.

6.1 Performance

Responsiveness: One aspect of performance is how quickly the application responds to user actions. Users expect a snappy and responsive interface, especially when scheduling appointments, viewing client information, or processing payments. Slow response times can frustrate users and negatively impact their experience.

Scalability: The application should be able to handle increasing numbers of users, appointments, and transactions without significant degradation in performance. Scalability ensures that the system can grow with the business and accommodate peak usage periods without crashing or becoming unresponsive.

Reliability: The system should be reliable and available whenever salon staff need to access it. Downtime or system failures can disrupt operations, lead to missed appointments, and affect customer satisfaction. High availability and fault tolerance are essential for ensuring uninterrupted service.

Response Time: Performance testing should measure the response time for key operations within the application, such as loading the appointment calendar, searching for client records, or generating reports. Short response times contribute to a smooth and efficient user experience, while long response times can lead to frustration and decreased productivity.

Database Performance: The performance of the database backend is critical for the overall performance of the application. Database queries should be optimized for efficiency, indexes should be used appropriately, and database connections should be managed effectively to minimize latency and maximize throughput.

Security Performance: While not traditionally considered a performance metric, the effectiveness of security measures can indirectly impact performance. Security controls such as encryption, authentication, and access controls should be implemented in a way that minimally impacts system performance while effectively protecting sensitive data

6.2 Limitations of the System

Internet Dependency: Since Online Job Search Portals are web-based, they rely on a stable internet connection. Interruptions or slow internet speeds can hinder access to the system, affecting productivity and causing inconvenience for salon staff.

Limited Customization: Off-the-shelf Online Job Search Portals may offer limited customization options. While they provide a range of features and functionalities, they may not fully align with the unique needs and preferences of every salon. Customization options may be limited to branding and basic configurations.

Initial Setup and Training: Implementing a new Online Job Search Portal requires time and resources for initial setup and staff training. Salon staff may need to familiarize themselves with the new software, which can temporarily disrupt workflow and productivity during the transition period.

Integration Challenges: Integrating a Online Job Search Portal with existing software or hardware infrastructure can be challenging. Compatibility issues may arise when attempting to connect the system with third-party services, such as payment processors or accounting software. This can require additional development effort and technical expertise

Data Security Concerns: Online Job Search Portals store sensitive client information, including personal details and financial data. Ensuring the security and privacy of this data is essential to protect against data breaches and unauthorized access. However, maintaining robust security measures requires ongoing vigilance and investment in cybersecurity practices.

Cost Considerations: While Online Job Search Portals can provide long-term cost savings through improved efficiency and productivity, there are upfront costs associated with purchasing and implementing the software. Additionally, subscription fees or licensing costs may be required for ongoing access to the system, which can impact the salon's budget.

Scalability Issues: Some Online Job Search Portals may have limitations in scalability, particularly for small salons that experience rapid growth or expansion. As the salon grows and the volume of clients and transactions increases, the system may struggle to accommodate higher demands, leading to performance issues or the need for upgrades.

User Experience Challenges: Despite efforts to design intuitive user interfaces, some Online Job Search Portals may still face usability challenges. Complex workflows, cluttered interfaces, or unfamiliar terminology can make it difficult for salon staff to navigate the system efficiently, leading to frustration and reduced productivity.

Maintenance and Support Dependence: Ongoing maintenance and technical support are essential for ensuring the smooth operation of a Online Job Search Portal. Dependence on software vendors or service providers for updates, bug fixes, and troubleshooting can pose a risk if adequate support is not available or if the vendor discontinues support for the product.

6.3 Testing Of System

Testing is asset of activities that can be planned in advanced and conducted systematically. For this reason, a template for software testing a set into which we can specifically test case design techniques and testing methods should be defined for the software process.

6.3.1 Types of Testing

1. Alpha Testing: -

Testing after code is mostly complete or contains most of the functional and prior to end user being involved. More often this testing will be performed in house or by an outside testing firm in close cooperation with the software engineering department.

2. Beta Testing: -

Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is released.

3. Functional Testing: -

Testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the modules perform its intended functions as stated in the specification and establishing confidence that a program does what it is supposed do.

4. Configuration Testing: -

Testing to determine how well the product works with a broad of the hardware/peripheral equipment configurations as on the different operating systems and software.

5. Pilot Testing: -

Testing that involves the users just before actual release to ensure that users become familiar with the release contents and ultimately accept it. Typically involves many users, is conducted over a short period of time and is tightly controlled.

6. System Integration Testing: - Testing a specific hardware/software installation. This is typically performed on a COTS system or any other system comprised or the disparate parts where custom configurations and /or unique installation are the norm .

7. Software Testing: - The process of exercising software is with the intent of ensuring that the software system meets its requirements and the user expectations and doesn’t file in an unacceptable manner.

8. Security testing: - Testing of database and network software in order to keep company data and resources from mistaken/ accidental users, hackers and other malevolent attackers.

9. Installation Testing: - Testing with the intent of determining if the product will install on a variety of platforms and how easily it installs.

10. Compatibility Testing: - Testing used to determine whether other system software components such as browsers, utilities and competing software would conflict with the software being tested.

CHAPTER 7

CONCLUSION

In conclusion, the development of an online job search portal web application offers a promising solution to the evolving needs of job seekers and employers alike. Through comprehensive research, analysis, and implementation of key features such as user-friendly interfaces, advanced search functionalities, and robust security measures, this project aims to bridge the gap between job seekers and employers efficiently.

With the potential to streamline the job search process, enhance candidate-employer connections, and contribute to economic growth, this web application stands as a valuable asset in the digital landscape of recruitment and employment.

Customer registration functionality streamlines the process of on boarding new clients, capturing essential information, preferences, and contact details. By maintaining a centralized database of client profiles, salons can personalize services, send targeted promotions, and foster long-term customer relationships.

Services management modules empower salon staff to efficiently manage appointments, allocate resources, and track service availability. This ensures optimal utilization of salon resources while minimizing scheduling conflicts and maximizing revenue potential.

The invoicing module automates the billing process, generating accurate invoices based on services rendered, products sold, and applicable taxes or discounts. By streamlining billing and payment processing, salons can improve cash flow, reduce administrative overhead, and enhance customer satisfaction.

Reporting functionalities provide salon owners with actionable insights into business performance, client demographics, service utilization, and revenue trends. By analyzing these reports, salons can identify areas for improvement, optimize service offerings, and develop targeted marketing strategies to attract and retain customers.

The dashboard module provides a centralized platform for monitoring key performance metrics, appointment schedules, and revenue trends in real-time, facilitating informed decision-making and efficient resource allocation.

In addition to facilitating job discovery and application processes, the online job search portal also serves as a centralized platform for employers to showcase their opportunities and connect with qualified talent. By incorporating features such as resume parsing, personalized job recommendations, and communication tools, the application strives to enhance user experience and maximize engagement.

Furthermore, the implementation of analytics and reporting functionalities provides valuable insights into user behavior, job market trends, and recruitment metrics, empowering stakeholders to make data-driven decisions and optimize their hiring strategies. Through continuous iteration and improvement, the web application aims to adapt to evolving industry dynamics and user preferences, ensuring its relevance and effectiveness in the competitive landscape of online recruitment.

In conclusion, a well-designed Online Job Search Portal with modules encompassing dashboard, customer registration, services management, invoicing, and reporting offers numerous benefits for salon owners and staff, including enhanced operational efficiency, improved customer engagement, and increased profitability. By leveraging technology to streamline salon operations and deliver personalized experiences, salons can differentiate themselves in a competitive market landscape and achieve sustainable growth in the long term.

CHAPTER 8

REFERENCES & BIBLIOGRAPHY

[1] Wikipedia. “Requirements analysis” Internet: http://en.wikipedia.org/wiki/Requirements_analysis [May 14 2017]

[2] Accurate iOneSoft (Pvt.) Ltd. “JustBookSalon”. Internet: http://justbooksalon.com [May 10 2017]

[3] Tyner Blain. “Ten Requirements Gathering Techniques”. Internet: http://tynerblain.com/blog/2006/11/21/ten-requirements-gathering-techniques/ [May 21 2017]

[4] PCMag Digital Group. “Definition of: Web application”. Internet: http://www.pcmag.com/encyclopedia/term/54272/web-application [May 28 2017]

[5] Wikipedia. “Use Case Diagram”. Internet: http://en.wikipedia.org/wiki/Use_Case_Diagram [June 06 2017]

[6] Wikipedia. “Class Diagram”. Internet: http://en.wikipedia.org/wiki/Class_diagram [June 10 2017]

[7] Wikipedia. “Activity Diagaram”. Internet: http://en.wikipedia.org/wiki/Activity_diagra [June 10 2017]

[8] Wikipedia. “Entity Relationship Diagram”. Internet: http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model [June 16 2017]

[9]Tutorials Point. “Angular 2”. Internet: https://www.tutorialspoint.com/angular2/angular2_overview.htm [June 03 2017]

[10]MSDN. “Model View Viewmodel”, Internet: https://code.msdn.microsoft.com/silverlight/How-to-implement-MVVM 71a6544 [June 12 2017]

[11] Tec Target. “SQL Server 2012”. Internet: http://whatis.techtarget.com/definition/SQL-Server-2012 [June 02 2017]1

[12] Angular.io. “Form Validation”. Internet: https://angular.io/guide/formvalidation [August 26 2017]

[13] GitHub. “mydatepicker”. Internet: https://github.com/kekeh/mydatepicker

[14] NPM Package. “JS PDF”. Internet: https://www.npmjs.com/package/pdf-viewer

[15] Daypilot. "Angular 2 Scheduler". Internet: https://doc.daypilot.org/scheduler/angular [August 17 2017] Online Online Job Search Portal

[16] Wikipedia. "Unit Testing". Internet: http://en.wikipedia.org/wiki/Unit_testing [August 17 2017]

[17] Wikipedia. "Black Box Testing". Internet: http://en.wikipedia.org/wiki/Blackbox_testing [August 15 2017]

[18] Wikipedia. "White Box Testing". Internet: http://en.wikipedia.org/wiki/Whitebox_testing [August 15 2017]

[19] Wikipedia. "Integration Testing". Internet: http://en.wikipedia.org/wiki/Integration_testing [August 15 2017]

[20] Wikipedia. "System Testing". Internet: http://en.wikipedia.org/wiki/System_testing [August 15 2017]

[21] MSDN. "How to Install and Configure Internet Information Services". Internet: http://msdn.microsoft.com/en-us/library/hh167503(v=nav.71).aspx

8.1 Online Websites

The following are the AI enabled mock interview system websites that we had analyzed for ours:

- https://www.greetai.co/practice
- https://beta.interviewai.in/
- https://www.acetheinterview.app/
- https://geekflare.com/ai-powered-interview-preparation-platforms/
- https://interviewly.ai/
- https://www.myamcat.com/products/mock-ai

8.2 Reference Books

Following are the books that we had referred for our project Mock Interview System:

1. J. Dorn and T. Naz, "Integration of Job portals by Meta-search," in Proc. 3rd International Conf. on Interoperability for Enterprise Software and Applications, Funchal, Portugal, 2007, pp. 401-412.
2. M. Mansourvar and N. Y. Mohd, "Web portal as a knowledge management system inthe universities," World Academy of Science, Engineering and Technology, vol. 70,pp. 968-974, 2010.
3. M. Gangle, "The only way is up? Employment protection and job mobility among recent entrants to European labour markets," European Sociological Review, vol. 19,pp. 429, 2007
4. M. Mansourvar, Development of a Job Web Portal to Capture Industry's Needs, 2011.
5. Weber and H. Mahringer, "Choice and success of job search methods," Empirical Economics, vol. 35, no. 1, pp. 153-178, 2008.
6. Marjan Mansourvar and Norizan Binti Mohd Yasin, "Development of a job web portal to improve education quality," International Journal of Computer Theory and Engineering, Vol. 6, No. 1, February 2014.
7. Vivek Kumar Sehgal Akshay Jagtiani, Meha Shah, Anupriya Sharma, Arpit Jaiswal and Dhananjay Mehta, "Job Portal – A web application for geographically distributed multiple clients," 2013 First International Conference on Artificial Intelligence, Modelling & Simulation.

8. Pooja T. Killewale and Prof. A.R. Mune, “Job Portal – A web application for distributed clients,” International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 5, May 2017.
9. Malgorzata Mochol, Holger Wache and Lyndon Nixon, “Improving the accuracy ofjob search with semantic techniques,” Conference Paper, April 2007.
10. Abu bucker Sam sudeen Shaffi and Mohaned Al-Obaidy, “Analysis and comparative study of traditional and web information systems development methodology (WISDM) towards web development applications,” International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 11, November 2013.