A Project Report

On

“JOB SEARCH PORTAL APP”

Submitted for partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

Ms. Saba Farheen (160320737004)

Ms. Sumiya Amreen Sultana (160320737005)

Ms. Nayela Qudsiya (160320737008)

Under the guidance of

Mrs. Sameena Siddiqui

Assistant Professor

Department of I.T

DCET, Hyderabad.

Department of Information Technology

Deccan College of Engineering and Technology

(Affiliated to Osmania University)

Hyderabad

CERTIFICATE

This is to certify that the project work entitled **“JOB SEARCH PORTAL APP”** is a bonafide work carried out by **Ms. Saba Farheen (160320737004), Ms. Sumiya Amreen Sultana (160320737005), Ms. Nayela Qudsiya (160320737008)** in partial fulfillment of the requirement for the award of degree of **BACHELOR OF ENGINEERING IN INFORMATION TECHNOLOGY** by the **OSMANIA UNIVERSITY**, Hyderabad, under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide	**Project Coordinator**	**Head of Department**
Mrs. Sameena Siddiqui	Dr. Waseema Masood	Dr. Ayesha Ameen
Assistant Professor	Associate Professor	Professor and HOD
Department of I. T	Department of I. T	Department of I. T
DCET, Hyderabad.	DCET, Hyderabad.	DCET, Hyderabad.

External Examiner

DECLARATION

This is to certify that the work reported in the present project entitled "**JOB SEARCH PORTAL APP**" is a record of work done by us in the Department of Information Technology, Deccan College of Engineering and Technology, Osmania University. The reports are based on the project work done entirely by us and not copied from any other source.

Ms. Saba Farheen-(160320737004)

Ms. Sumiya Amreen Sultana-(160320737005)

Ms. Nayela Qudsiya-(160320737008)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude and indebtedness to my project supervisor **Mrs. Sameena Siddiqui** for her valuable suggestions and interest throughout the course of the project.

I am thankful to the Head of Department **Dr. Ayesha Ameen** for providing excellent infrastructure and a nice atmosphere for complementing this project successfully.

I convey my heartfelt thanks to the lab staff for allowing me to use the required equipmentwhenever needed.

Finally, I would like to take this opportunity to thank my family for their support through thework. I sincerely acknowledge and thank who gave directly or indirectly their support in this project .

Sincerely,

Ms. Saba Farheen-(160320737004)

Ms. Sumiya Amreen Sultana-(160320737005)

Ms. Nayela Qudsiya - (160320737008)

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

JOB SEARCH PORTAL APP – Job search browser mobile application

The job search portal application aims to enhance efficiency, foster transparency, and create a dynamic space that catalyses meaningful connections between talent and opportunities in the ever-evolving job market. By prioritizing user experience, leveraging advanced technology, and providing valuable resources and insights, the platform is poised to revolutionize the way individuals approach their professional journey. At the heart of the application lies a commitment to user experience. Recognizing the diverse needs and preferences of job seekers and employers alike, the platform offers a range of intuitive features and functionalities that cater to individual requirements. Job seekers can leverage advanced search filters to narrow down their job preferences based on criteria such as industry, location, salary range, and job type. This ensures that users can quickly identify relevant opportunities tailored to their specific interests and qualifications. Through sophisticated algorithms and machine learning techniques, the application delivers personalized job recommendations to users based on their profile, search history, and preferences. This proactive approach streamlines the job search process, presenting relevant opportunities that may have otherwise gone unnoticed.

The platform provides real-time updates on job listings across diverse industries, ensuring that users have access to the latest opportunities as soon as they become available. This dynamic feature enables job seekers to stay ahead of the curve and react promptly to new openings. For employers, the portal streamlines the recruitment process by offering a robust platform for posting job listings, applicant screening, and efficient management of the entire hiring workflow. This not only saves time and resources but also ensures that employers can identify and attract top talent with ease. The application fosters transparency in the job market by providing valuable insights into industry trends, salary benchmarks, and skill demand. This data-driven approach equips both job seekers and employers with the information they need to make info

decisions and stay competitive in their respective fields Networking tools and professional development resources further enrich the platform, creating an ecosystem

that supports continuous growth and connectivity within the workforce. Whether it's connecting with industry peers, accessing online courses, or attending virtual events, users have access to a wide range of opportunities for personal and professional development. Top of Form

TABLE OF CONTENTS

Page No.

CHAPTER I

INTRODUCTION

1.1 Introduction

In the contemporary landscape of employment, the Job Search Portal Application emerges as a transformative force, redefining the dynamics between job seekers and employers. This sophisticated platform serves as a digital nexus, seamlessly bridging the gap between talent and opportunity in a rapidly evolving job market. With a user-centric design, the application becomes a one-stop solution for job seekers, offering intuitive features such as advanced search functionalities, personalized job recommendations, and real-time updates on diverse employment opportunities. Its commitment to empowering individuals extends beyond mere job matching, encompassing tools for resume refinement, career guidance, and skill development, thereby enriching the professional journey of users. Simultaneously, the portal revolutionizes the recruitment process for employers, providing a streamlined interface for posting job listings, applicant screening, and efficient workflow management. This innovative tool is not just a job-matching platform; it is a comprehensive ecosystem fostering transparency through insights into industry trends, salary benchmarks, and skill demand. The introduction of networking tools and professional development resources further positions the job search portal as a catalyst for meaningful connections, creating a dynamic space where talent and opportunities converge harmoniously in the modern employment landscape.

Job Search: Empowering Your Career Journey! Our cutting-edge mobile application developed using React Native seamlessly connects job seekers with their dream opportunities on both iOS and Android platforms, With user friendly interfaces and advanced algorithms Catalyst Job Search offers personalized job recommendations real-time notifications and interactive features to streamline your job search process, Explore endless possibilities discover tailored job listings and embark on your career adventure with confidence, Your dream job is just a tap away.

1.2 Objective

A job search portal application serves as a comprehensive platform designed to facilitate the employment-seeking process for individuals and streamline the recruitment process for employers. The primary objectives of such an application are manifold. Firstly, it aims to provide job seekers with a user-friendly and centralized platform to search for employment opportunities across various industries and sectors. This includes features such as advanced search filters, personalized job recommendations, and real-time updates on new job postings.

For individuals thinking about new job, new career, or new direction, it helps you explore the possibilities and find the opportunities that are right for you. Whether you are looking for a job or a new member of your staff team, you definitely have the right place. The best way to change your career or get the right people for the jobs is quite easy now.

For employers, the job search portal aims to simplify the recruitment process by offering a robust platform to post job listings. It provides employers with access to a diverse pool of talent, thereby increasing the likelihood of finding suitable candidates for their job openings.

The application also facilitates seamlessly connects jobseekers with their dream opportunities on both IOS and Android Platforms, with user friendly interfaces and advanced algorithms catalyst.

Ultimately, the objectives of a job search portal application revolve around bridging the gap between job seekers and employers, optimizing the recruitment process, and contributing to the overall efficiency and dynamism of the job market.

1.3 Domain

Android mobile development

Android mobile development has been Kotlin-first since Google I/O in 2019.

Over 50% of professional Android developers use Kotlin as their primary language, while only 30% use Java as their main language. 70% of developers whose primary language is Kotlin say that Kotlin makes them more productive.

Using Kotlin for Android development, you can benefit from:

- **Less code combined with greater readability**. Spend less time writing your code and working to understand the code of others.
- **Fewer common errors**. Apps built with Kotlin are 20% less likely to crash based on Google's internal data.
- **Kotlin support in Jetpack libraries**. Jetpack Compose is Android's recommended modern toolkit for building native UI in Kotlin. KTX extensions add Kotlin language features, like coroutines, extension functions, lambdas, and named parameters to existing Android libraries.
- **Support for multiplatform development**. Kotlin Multiplatform allows development for not only Android but also iOS, backend, and web applications. Some Jetpack libraries are already multiplatform. Compose Multiplatform, JetBrains' declarative UI framework based on Kotlin and Jetpack Compose, makes it possible to share UIs across platforms – iOS, Android, desktop, and web.
- **Mature language and environment**. Since its creation in 2011, Kotlin has developed continuously, not only as a language but as a whole ecosystem with robust tooling. Now it's seamlessly integrated into Android Studio and is actively used by many companies for developing Android applications.
- **Interoperability with Java**. You can use Kotlin along with the Java programming language in your applications without needing to migrate all your code to Kotlin.
- **Easy learning**. Kotlin is very easy to learn, especially for Java developers.
- **Big community**. Kotlin has great support and many contributions from the community, which is growing all over the world. Over 95% of the top thousand Android apps use Kotlin.

Many startups and Fortune 500 companies have already developed Android applications using Kotlin, see the list on the Google website for Android developers.

To start using Kotlin for:

- Android development, read Google's documentation for developing Android apps with Kotlin.

- Developing cross-platform mobile applications, see Get started with Kotlin Multiplatform for Android and iOS.

Introduction to React Native

If you want to build mobile apps for both Android and iOS. What should you learn? The individual native languages for each app I .e, Java for Android and Swift/Objective-C for iOS?, Actually NO. Native Android and iOS development are quite different and can be expensive – first, the language itself is quite different, and second, all the underlying APIs are different – the way of using the GPS is different, the way to create animations is different, the way you make network calls is different.

We're always looking for shorter development cycles, quicker time to deployment, and better app performance. And there are so many hybrid mobile frameworks such as Native Script, React Native, Ionic, Xamarin, PhoneGap, etc.

React Native: It is a framework developed by Facebook for creating native-style apps for iOS & Android under one common language, JavaScript. Initially, Facebook only developed React Native to support iOS. However, with its recent support of the Android operating system, the library can now render mobile UIs for both platforms.

Basic Knowledge of HTML, CSS and JS.

Basic Knowledge of ReactJS.

Node J s should be installed in your system.

Building with React Native is extremely efficient and highly addictive but getting started can be a little tricky. React Native uses Node.js, a JavaScript runtime, to build your JavaScript code. If you don't already have Node.js installed, it's time to get it!

Installation: Here we will use the Expo CLI version that will much smoother to run your React Native applications. Follow the below steps one by one to setup your React native environment.

Step 1: Open your terminal and run the below command.

npm install -g expo-cli

Step 2: Now expo-cli is globally installed so you can create the project folder by running the below command.

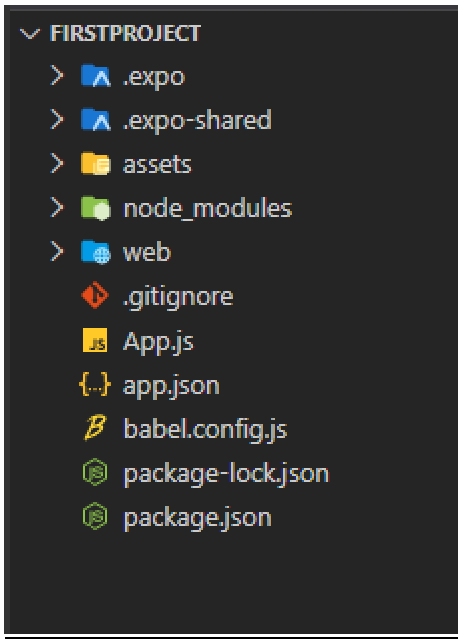
expo init "projectName"

Step 3: Now go into the created folder and start the server by using the following command.

cd "projectName"

npm start web

Project Structure:



```
FIRSTPROJECT
  > .expo
  > .expo-shared
  > assets
  > node_modules
  > web
    .gitignore
    App.js
    app.json
    babel.config.js
    package-lock.json
    package.json
```

Example:

// File name App.js

import React from 'react';

import { Text, View, Style Sheet } from 'react-native';

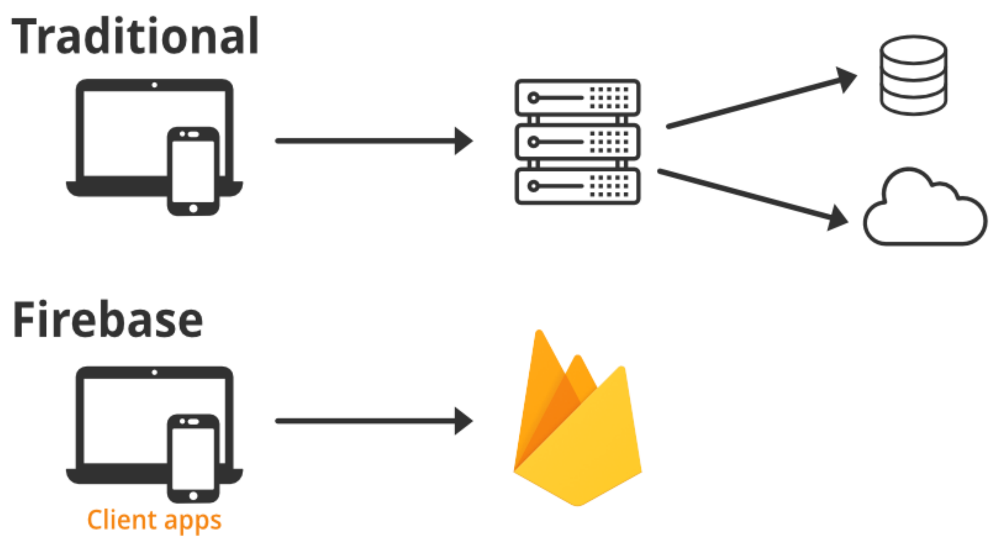
```
import Constants from 'expo-constants';

const Home=()=>{

  return (

      <Text style={{

                    marginTop:300,

                    marginLeft:10}}>

          Geeks for geeks

      </Text>

  )

}

export default function App() {

  return (

    <View>

          <Home/>

    </View>

  );

}
```

Advantages over other frameworks and Languages: Whenever there is an update for apps written in Swift/Objective-C or Java, the whole app needs to be recompiled and a new version has to be distributed to the App Store again. All this can take a few weeks depending on the App Store review process.

To avoid this hassle, React Native apps work in a different way, a native app is able to locate specific JavaScript code, which is later downloaded and compiled when the app is launched on an actual device. By this, updating the app can be done instantly without needing to submit a new version to the App Store again and again.

Firebase – Introduction

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, iOS, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

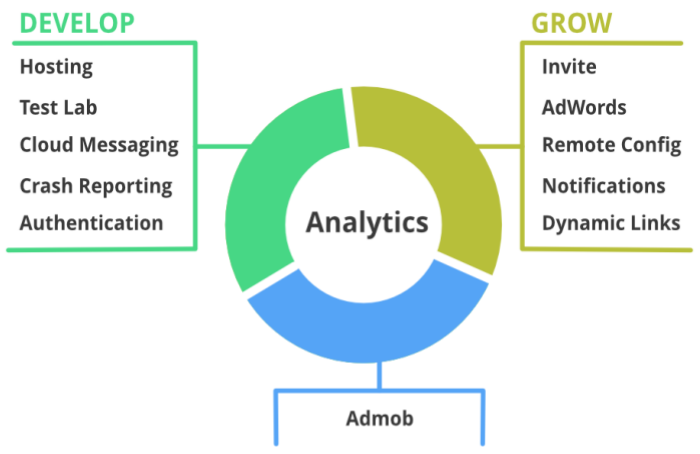


Brief History of Firebase:

Firebase initially was an online chat service provider to various websites through API and ran with the name **Envolve**. It got popular as developers used it to exchange application data like a game state in real time across their users more than the chats. This resulted in the separation of the Envolve architecture and it's chat system. The Envolve architecture was further evolved by it's founders James Tamplin and Andrew Lee, to what modern day Firebase is in the year 2012.

Features of Firebase:

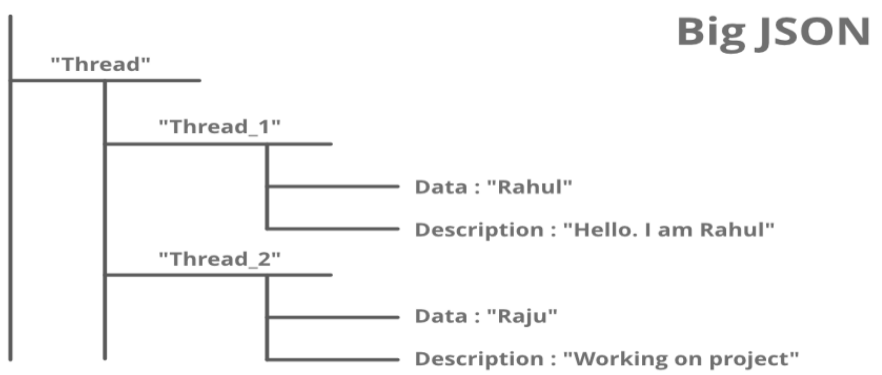
Mainly there are 3 categories in which firebase provides its services.



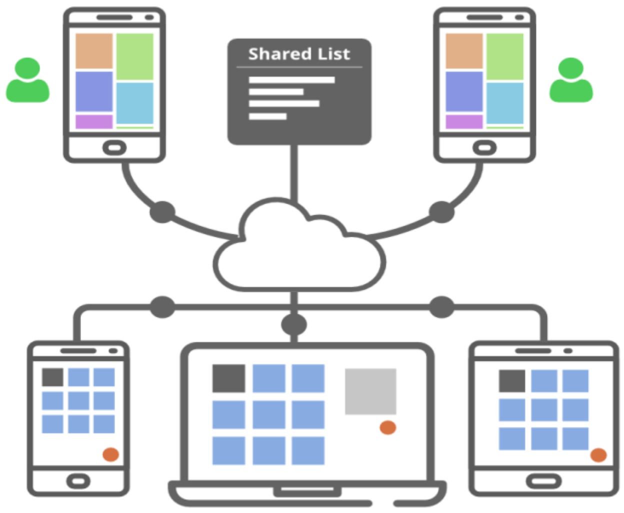
Build better applications

This feature mainly includes backend services that help developers to build and manage their applications in a better way. Services included under this feature are :

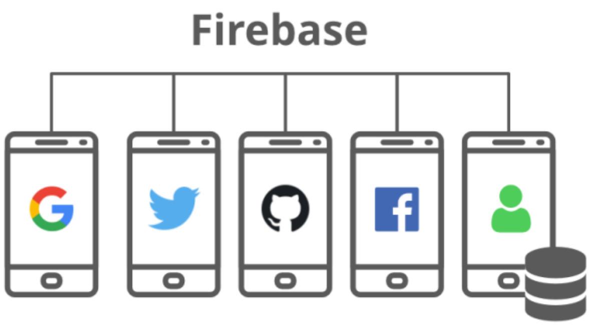
- **Realtime Database:** The Firebase Realtime Database is a cloud-based NoSQL database that manages your data at the blazing speed of milliseconds. In simplest term, it can be considered as a big JSON file.


Big JSON
"Thread"
"Thread_1"
Data : "Rahul"
Description : "Hello. I am Rahul"
"Thread_2"
Data : "Raju"
Description : "Working on project"

- **Cloud Fire store:** The cloud Fire store is a NoSQL document database that provides services like store, sync, and query through the application on a global scale. It stores data in the form of objects also known as Documents. It has a key-value pair and can store all kinds of data like, strings, binary data, and even JSON trees.



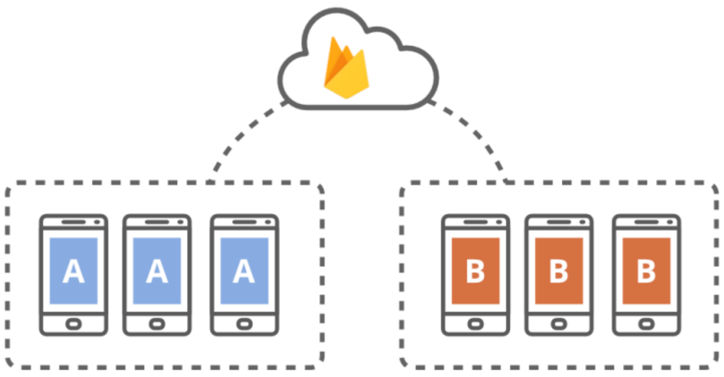
- **Authentication:** Firebase Authentication service provides easy to use UI libraries and SDKs to authenticate users to your app. It reduces the manpower and effort required to develop and maintain the user authentication service. It even handles tasks like merging accounts, which if done manually can be hectic.



- **Remote Config:** The remote configuration service helps in publishing updates to the user immediately. The changes can range from changing components of the UI to changing the behaviour of the applications. These are often used while publishing seasonal offers and contents to the application that has a limited life.

MY APP
Hi Geeks
MY APP
Hi Geeks

- **Hosting:** Firebase provides hosting of applications with speed and security. It can be used to host Stati or Dynamic websites and microservices. It has the capability of hosting an application with a single command.
- **Firebase Cloud Messaging(FCM):** The FCM service provides a connection between the server and the application end users, which can be used to receive and send messages and notifications. These connections are reliable and battery-efficient.



Improve app quality:

Here majorly all the application performance and testing features are provided. All the features required to check and manage before launching your application officially are provided in this section. Services included are:

- **Crashlytics:** It is used to get real-time crash reports. These reports can further be used to improve the quality of the application. The most interesting part of this service is that it gives a detailed description of the crash which is easier to analyze for the developers.

- **Performance monitoring:** This service gives an insight to the performance characteristics of the applications. The performance monitoring SDK can be used to receive performance data from the application, review them, and make changes to the application accordingly through the Firebase console.
- **Test lab:** This service helps to test your applications on real and virtual devices provided by Google which are hosted on the Google Datacenters. It is a cloud-based app-testing infrastructure which supports testing the application on a wide variety of devices and device configurations
- **App Distribution:** This service is used to pre-release applications that can be tested by trusted testers. It comes in handy as decreases the time required to receive feedback from the testers.

Grow your app:

- This feature provides your application analytics and features that can help you to interact with your user and make predictions that help you to grow your app. Services provided are:

- **Google analytics:** It is a Free app measurement service provided by Google that provides insight on app usage and user engagement. It serves unlimited reporting for up to 500 distinct automatic or user-defined events using the Firebase SDK.
- **Predictions:** Firebase Predictions uses machine learning to the application’s analytics data, further creating dynamic user segments that are based on your user’s behavior. These are automatically available to use for the application through Firebase Remote Config, the Notifications composer, Firebase In-App Messaging, and A/B Testing.
- **Dynamic Links:** Deeps Links are links that directly redirects user to specific content. Firebase provides a Dynamic linking service that converts deep links into dynamic links which can directly take the user to a specified content inside the application. Dynamic links are used for converting web users to Native app users. It also increases the conversion of user-to-user sharing. In

addition, it can also be used to integrate social media networks, emails, and SMS to increase user engagement inside the application.

- **A/B Testing:** It is used to optimize the application's experience by making it run smoothly, scaling the product, and performing marketing experiments.

Pros and Cons of Using Firebase:

Below listed are the advantages and disadvantages of using a Firebase backend:

Pros:

- Free plans for beginners.
- Real-time database is available.
- Growing Community.
- Numerous services are available.

Cons:

- It uses NoSQL so, people migrating from SQL might feel difficulty.
- It is still growing so, it is not tested to an extent.

Companies using Firebase

Below are some reputable organizations that rely on a firebase backend for its functioning:

- The New York Times
- Alibaba.com
- Gameloft
- Duolingo
- Trivago
- Venmo
- Lyft

1.4 Synopsis

In today's competitive environment, getting jobs and searching for candidates assumes greater importance. Earlier, the advertisements of jobs were limited to news papers But now the emerging economy demands the situation to change. We have various job portals , online employment exchanges, consultancies, company websites etc.

In this situation recruitment should be time saving, cost effective and at the same time should search out the qualified candidates .By using the efficient methods and techniques the recruitment process is made easy. The employers and candidates see things not in terms of what they need, but in terms of best things they want. The client hunted a job portal which is meant for Employers, jobseekers, and also which guides the candidates in various areas of job search and related things. Finding and recruiting the best quality candidates seems to get more complicated in the coming days. So the proposed system aimed to provide best class of employment services to job seekers, employers, and recruiters. The system should provide free job search and services. It should be quick, safe and easy to use. The system is meant for the end users -the employer and the candidate. The end users may be registered users or not registered.

CHAPTER II

LITERATURE SURVEY

2.1 Literature Survey

A comprehensive literature survey on job search portal applications reveals a growing body of research and scholarly work addressing various aspects of these platforms. Researchers have explored the impact of job search portals on the dynamics of the labor market, emphasizing their role in connecting job seekers with diverse employment opportunities. Studies have delved into the technological aspects of these applications, focusing on the integration of artificial intelligence and machine learning algorithms for improved job matching and recommendation systems. The effectiveness of features such as advanced search filters, personalized job recommendations, and real-time updates has been a subject of investigation. Additionally, scholars have examined the social and economic implications of job search portals, considering factors such as employment rates, job satisfaction, and the influence of these platforms on traditional recruitment methods.

Literature also highlights challenges and limitations associated with job search portals, including concerns related to data privacy, algorithmic biases, and the need for continuous adaptation to emerging job market trends. Researchers have proposed frameworks for evaluating the user experience and effectiveness of these platforms, emphasizing the importance of user feedback and continuous improvement.

- **Van Hoye, G., & Lievens, F. (2007)**. Networking as a job search behavior: A social network perspective. Journal of Occupational and Organizational Psychology, 80(4), 661-682. (2007) - Examines the impact of job search portal apps on networking behavior among job seekers, emphasizing the role of social networks.
- **Rappa, M. (2009)**. Business models on the web: Managing the digital enterprise. In Business models on the web (pp. 1-17). Springer, Boston, MA. (2009) - Discusses the transition of job search from traditional methods to digital platforms, laying the foundation for the development of job search

portal apps.

- **Vosko, L. F. (2010).** Managing the margins: Gender, citizenship, and the international regulation of precarious employment. OUP Oxford. (2010) - Offers insights into employer perspectives on the use of job search portal apps for hiring and managing a diverse workforce.
- **Dey, S., & Ghattas, J. (2015)**. A study of job search mobile applications. In 2015 8th International Conference on Human System Interactions (HSI) (pp. 346-353). IEEE. (2015) - Analyzes the features and usability of job search mobile applications, highlighting the importance of user-centric design.
- **Marston, S., Samson, K., & Burton, S. (2016)**. Improving the usability of mobile recruitment applications: The development and evaluation of a design guide. Journal of usability studies, 11(2), 67-85. (2016) - Provides insights into enhancing the usability of job search portal apps through the development of design guidelines.
- **van Laar, E., van Deursen, A. J., van Dijk, J. A., & de Haan, J. (2017)**. The relation between 21st-century skills and digital skills: A systematic literature review. Computers in Human Behavior, 72, 577-588. (2017) - Explores the relationship between digital skills and user experience in the context of job search portal apps.
- **Coelho, D., Soares, M., Cardoso, G., & Ribeiro, J. (2018)**. The impact of using social media for job search on employability. The International Journal of Human Resource Management, 29(14), 2157-2186. (2018) - Explores the effects of utilizing social media platforms integrated into job search portal apps on job search outcomes and employability.
- **Dwivedi, Y. K., Hughes, D. L., Coombs, C., Constantinou, I., Duan, Y., Edwards, J. S., ... & Raman, R. (2019)**. Impact of digital technologies on citizen well-being: A literature review. International Journal of Information Management, 47, 36-42. (2019) - Discusses the potential impact of emerging digital technologies on the future development of job search portal apps and their role in enhancing citizen well-being
- **Sharma, A., & Gupta, R. (2020)**. User Satisfaction of Job Portal Mobile Application: A Case Study of Naukri.com. In Proceedings of the 2020 5th

International Conference on Communication and Electronics Systems (ICCES) (pp. 664-669). IEEE. (2020) - Investigates user satisfaction with specific job portal mobile applications, offering insights into factors influencing user experience.

.

Overall, the literature survey underscores the multidimensional impact of job search portal applications, shedding light on their technological, social, and economic implications. Researchers continue to contribute to this evolving field, providing valuable insights that inform the design, functionality, and overall effectiveness of job search portals in the ever-dynamic landscape of employment.

CHAPTER III

PROBLEM SPECIFICATION

3.6 SYSTEM REQUIREMENT SPECIFICATION

3.6.1 Hardware Requirements:

❖ **System:**

- Pentium Dual Core Processor**:** This processor type provides sufficient computing power for running the application smoothly. It balances performance and cost-effectiveness, making it suitable for most users

❖ **Hard Disk:**

- 120GB Storage: The 120GB hard disk offers ample storage space for the application, its associated files, and any user-generated content. It ensures that users can store necessary data without encountering storage constraints.

❖ **Monitor:**

- 15’’ LED Monitor: A 15-inch LED monitor provides a comfortable viewing experience for users interacting with the application. LED technology offers energy efficiency and crisp display quality, enhancing the user's visual experience.

❖ **Input Devices:**

- Keyboard and Mouse: These standard input devices enable users to interact with the application efficiently. The keyboard facilitates text input, while the mouse allows for precise navigation and selection within the application's interface.

❖ **RAM:**

- 1GB RAM: With 1GB of RAM, the system can adequately handle the application's memory requirements. While this may be considered a modest amount by today's standards, it remains sufficient for the smooth operation of the application.

3.6.2 Software Requirements

❖ **Operating System:**

- Pentium Dual Core Process or: This processor type provides sufficient computing power for running the application smoothly. It balances performance and cost-effectiveness, making it suitable for most user

- Windows 10: The choice of Windows 10 as the operating system ensures compatibility with a wide range of hardware configurations and provides a familiar environment for users. Additionally, Windows 10 offers robust security features and ongoing support, contributing to the application's stability and reliability.

❖ **Tools:**

- Android Studio and its tools: Android Studio is the primary integrated development environment (IDE) used for developing Android applications. It provides a comprehensive set of tools for designing, coding, debugging, and testing mobile applications, making it an ideal choice for building the job search portal application.

❖ **Database:**

- Firebase: Firebase serves as the database backend for the application, offering real-time data synchronization, user authentication, and cloud storage capabilities. Its integration with Android Studio simplifies the development process and enables seamless data management for the job search portal.

3.2 SYSTEM ANALYSIS & DISCUSSION

3.2.1 Objective

The objectives of system analysis in the context of a job search portal app are multi-faceted, aiming to ensure the efficient design and functionality of the application. Firstly, system analysis seeks to comprehensively understand the requirements of both job seekers and employers. This involves conducting thorough interviews, surveys, and stakeholder discussions to gather insights into user expectations, desired features, and industry-specific needs. Through this process, the analysis aims to identify key functionalities, such as advanced search capabilities, resume parsing, and real-time notifications, to optimize the user experience.

Another critical objective is to assess the existing system and determine areas for improvement or innovation. This involves evaluating the usability of the current application, identifying bottlenecks, and understanding technological advancements that could enhance the platform. Additionally, system analysis aims to define the scope of the job search portal app, outlining the features and functionalities that will be prioritized in the development process.

Furthermore, the analysis phase plays a pivotal role in establishing system requirements and specifications. This includes defining the technical architecture, database structure, and integration points with external services. Through careful documentation and validation with stakeholders, the analysis sets the foundation for the subsequent phases of design and development.

In the context of a job search portal app, specific considerations during system analysis may include ensuring secure handling of personal information, implementing effective algorithms for job matching, and evaluating the scalability of the system to accommodate a growing user base. Overall, the objectives of system analysis in the development of a job search portal app are geared towards creating a robust, user-centric, and technologically advanced platform that addresses the evolving dynamics of the job market.

3.5 Modules

1.Job seeker

2.Search job

3.Search job seeker

1.<u>Job Seeker</u>

- ➢ The Module is again divided in to the following sub modules
- ➢ <u>Update personal Details:</u> the job seeker can update his personal details
- ➢ <u>Update Qualification Details:</u> the job seeker updates his qualification details like qualification, year of passed out.
- ➢ <u>Update skill details</u>: in these he updates his skill details like java. net and update his experience and current company details
- ➢ <u>Job search</u>: in these he/she can search the job related to his skills, experience and company.

2. Search jobs

A registered job seeker can search jobs by applying several filtering criteria. The job seeker can save his job preference, so that he can get mail alerts if any job is posted by the registered job providers.

3.Search job seekers

A registered job provider can search job seeker by applying several filtering criteria. The job provider can save the profile preference, so that he can get mail alerts if any new profile is posted by the registered job seekers.

3.2.3 Problem Specification:

The problem specification for the system analysis of a job search portal app involves a comprehensive examination of challenges and inefficiencies in the existing system that the app aims to address. One key issue is the fragmentation of job information across various platforms and sources, making it cumbersome for job seekers to access a consolidated view of available opportunities. Additionally, the lack of advanced algorithms for job matching often results in mismatches between candidate skills and job requirements. The application process itself may be convoluted, with multiple steps and manual data entry causing user fatigue and hindering the application experience. Moreover, the absence of robust networking features limits the ability of job seekers to connect with professionals in their industry. Security concerns, such as data privacy during the application process, also pose significant challenges. Addressing these issues through the implementation of an integrated job search portal app involves designing a user-friendly interface, incorporating advanced algorithms for precise job matching, streamlining the application process, fostering networking opportunities, and ensuring robust data privacy measures. The development of the app must carefully consider these challenges to deliver a solution that enhances the overall efficiency and experience of the job-seeking process for users.

3.2.4 Proposed System

The proposed system for a job search portal application builds upon the strengths of existing platforms while introducing innovative features to address evolving needs in the employment landscape. This advanced system aims to provide a more personalized,

efficient, and interactive experience for both job seekers and employers.
One key aspect of the proposed system is the integration of artificial intelligence and machine learning algorithms to enhance job matching. These algorithms will not only consider traditional criteria like skills and experience but also factor in soft skills, cultural fit, and individual preferences to generate highly accurate job recommendations. This personalized approach ensures that job seekers are presented with opportunities that align not only with their qualifications but also with their career aspirations and workplace preferences.

To further empower job seekers, the proposed system includes a comprehensive suite of career development tools. Beyond resume building and skill enhancement, it introduces features like virtual mentorship programs, industry-specific webinars, and personalized career coaching. This holistic approach aims to not only assist in securing a job but also in fostering long-term career growth and adaptability in a rapidly changing job market.

For employers, the proposed system focuses on optimizing the recruitment process. Advanced applicant tracking systems, automated interview scheduling, and collaborative hiring tools are integrated to streamline the workflow. Additionally, the system incorporates data analytics tools that provide in-depth insights into talent trends, helping employers make informed decisions about their hiring strategies.

In terms of user interface, the proposed system emphasizes a seamless and intuitive design. Advanced search features are augmented with natural language processing capabilities, allowing users to input queries in a conversational manner. This enhances user experience and ensures that even those with limited technical expertise can navigate the platform effortlessly.
Moreover, the proposed system promotes community building through enhanced networking features. Virtual job fairs, industry-specific forums, and collaborative projects are introduced to facilitate meaningful connections between professionals, fostering a vibrant and supportive ecosystem within the job search portal.

In conclusion, the proposed job search portal application represents a next-generation platform that leverages cutting-edge technologies and a holistic approach to address the multifaceted needs of the modern job market. By prioritizing personalization, career development, and streamlined recruitment processes, this system aims to redefine the job search experience, providing a dynamic and adaptive solution for individuals and employers alike.

- Job search portals provide access to a vast and diverse range of job opportunities across various industries and geographic locations, offering users a comprehensive platform to explore potential careers.
- Portals often feature advanced search filters, enabling users to tailor their job searches based on specific criteria such as industry, job type, experience level, and location, resulting in more relevant job matches.
- Job seekers receive real-time updates on new job listings, ensuring that they stay informed about the latest opportunities and can promptly apply to positions that match their skills and preferences.
- The intuitive and user-friendly interfaces of job search portals simplify the navigation and application process, enhancing the overall experience for both job seekers and employers.
- Many portals employ algorithms to provide personalized job recommendations, increasing the likelihood of matching individuals with positions that align with their skills, qualifications, and career goals.
- The majority of job search portals offer mobile applications, providing users with the flexibility to search for jobs, apply, and stay updated on the go, enhancing accessibility for a broader audience.
- Job search portals save time and reduce costs for both job seekers and employers by providing a centralized platform for job hunting and recruitment, eliminating the need for extensive physical efforts and resources.
- Job seekers can access reviews and insights about companies, helping them make informed decisions about potential employers and ensuring a better cultural fit.
- Some portals leverage emerging technologies, such as artificial intelligence and machine learning, to enhance features like resume matching, thereby improving the accuracy of job recommendations and recruitment processes.

- Job search portal aims to provide a seamless experience for both job seekers and employers, facilitating efficient job hunting and recruitment processes.
- The system encompasses various essential features, including user authentication and profile management, a comprehensive database of job listings with advanced search and recommendation capabilities, application management tools allowing users to apply for jobs directly within the app and track their application statuses.
- Additionally, the portal incorporates messaging and communication functionalities to enable seamless interaction between employers and job seekers, along with dedicated employer accounts for job posting and candidate management.
- Analytics and reporting tools offer insights into user engagement and job application trends, empowering both parties to make informed decisions.
- Ensuring mobile responsiveness, accessibility, security, and privacy protection are paramount, along with avenues for feedback and support.
- A well-structured monetization strategy, including premium features and potential partnerships, enhances sustainability.
- Integration with external platforms, such as social media and third-party services, enriches the portal's functionality, ultimately providing a robust and user-centric solution to the job search process.
- In the realm of data analytics, the portal empowers users with actionable insights, leveraging metrics to optimize their strategies and improve outcomes. Mobile responsiveness ensures accessibility across devices, while robust security protocols safeguard sensitive information and instill trust. User feedback mechanisms and responsive support channels demonstrate a commitment to continuous improvement and user satisfaction.
- A well-structured monetization strategy not only ensures the portal's sustainability but also opens avenues for value-added services and strategic partnerships. Integration with external platforms further enriches the user experience, amplifying reach and functionality.
- In essence, the proposed job search portal transcends mere functionality to become a catalyst for empowerment, connecting talent with opportunity in a dynamic and inclusive digital ecosystem.

3.4 Applications And Features

- Real- time
- Notification
- Regularly updated job news
- Connecting direct to the recruiters
- B2B Model for direct university engagement notification
- Social Media integration

The applications of a job search portal app are diverse and impactful, revolutionizing the way individuals search for employment and organizations find suitable candidates. Firstly, it serves as a centralized hub, aggregating job opportunities from various industries and locations, providing users with a comprehensive overview of the job market. For job seekers, the app facilitates a more streamlined and accessible job search experience, allowing them to browse, apply, and track applications in real-time. Employers, on the other hand, benefit from a broad reach to a pool of potential candidates, enabling them to efficiently manage recruitment processes. Additionally, the app often incorporates advanced features such as resume parsing, personalized job recommendations, and skill assessments, enhancing the matching accuracy between candidates and job postings. The convenience of mobile access ensures that users can engage with the job market anytime, anywhere. Furthermore, the app fosters networking opportunities, enabling professionals to build connections and stay informed about industry trends. Overall, the applications of a job search portal app extend beyond traditional job hunting, encompassing features that promote efficiency, connectivity, and effectiveness in the dynamic landscape of employment.

- **Comprehensive Job Search**: Job search portal apps serve as centralized platforms, offering a wide array of job opportunities from diverse industries and locations.
- **Efficient Job Matching**: Advanced algorithms enable personalized job recommendations based on user profiles, skills, and preferences, improving the accuracy of job matching.
- **Real-Time Job Notifications**: Users receive instant notifications about new job postings, ensuring timely awareness of relevant opportunities.

- **Mobile Accessibility**: The convenience of mobile access allows users to engage with the job market anytime, anywhere, enhancing flexibility and responsiveness.
- **Streamlined Application Process**: Job seekers can easily apply to multiple positions with streamlined application processes, reducing time and effort in submitting applications.
- **Skill Assessments**: Some apps incorporate skill assessment tools, allowing candidates to showcase their abilities and helping employers identify the most qualified candidates.
- **Networking Opportunities**: Job search portals often include features for professional networking, enabling users to connect with industry peers, mentors, and potential employers.
- **Career Development Resources**: Many apps provide resources such as articles, webinars, and courses to support ongoing career development and skill enhancement.
- **Industry-Specific Portals**: Some job search portals cater to specific industries, providing specialized platforms for professionals in fields like IT, healthcare, finance, and more.
- **Global Job Market Access**: Job seekers can explore opportunities in the global job market, expanding their horizons and considering international career options.
- **Freelance and Remote Opportunities**: The app may feature freelance and remote job opportunities, catering to individuals seeking flexible work arrangements.
- **Integration with social media**: Integration with social media platforms allows users to leverage their professional networks for job referrals and recommendations.

CHAPTER IV

SYSTEM DESIGN

4.1 System Architecture:

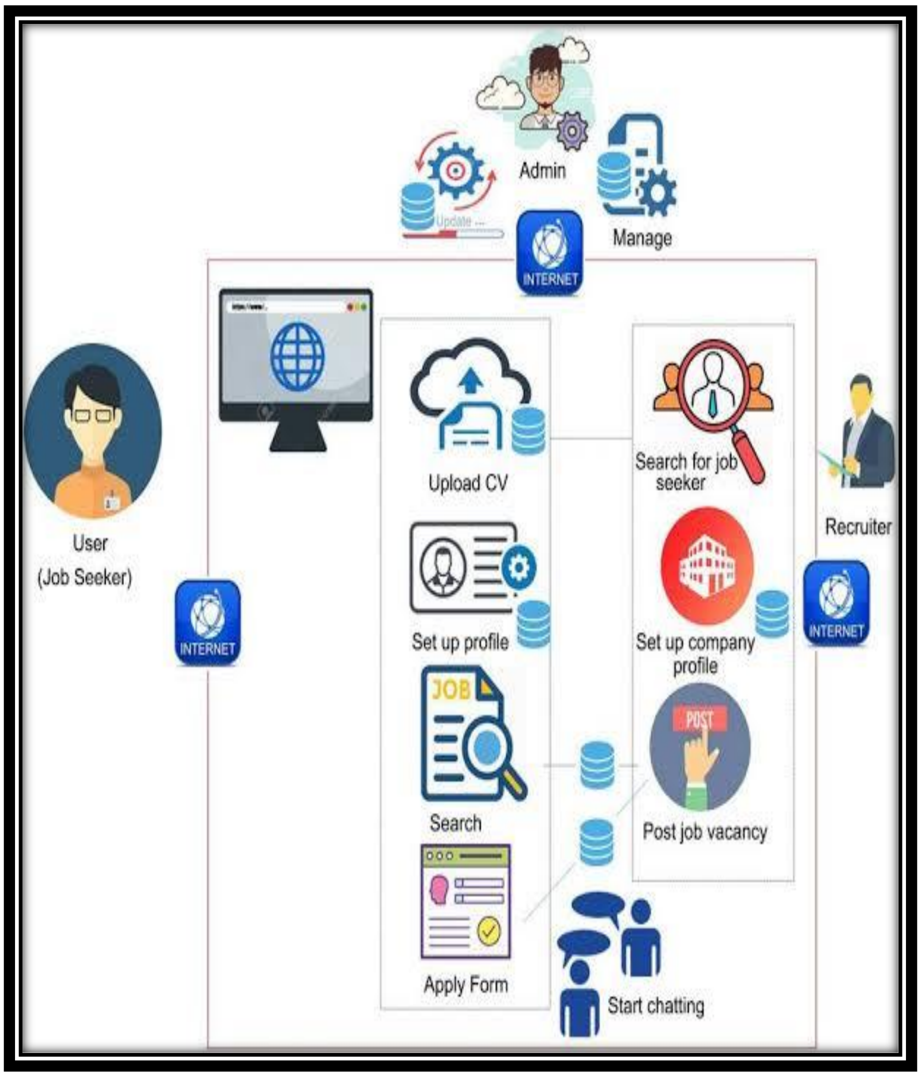


Figure 4.1-Architecture

The architecture of a job search portal application involves a multifaceted structure that seamlessly integrates various components to deliver a robust and user-friendly experience. At its core, the system typically comprises three main layers: the presentation layer, the application layer, and the data layer. The presentation layer

involves the user interface where job seekers and employers interact with the platform. This layer incorporates features like search functionalities, personalized dashboards, and communication tools, ensuring an intuitive and engaging experience for users. The application layer is the engine that powers the portal, managing the logic behind job matching algorithms, applicant tracking systems, and other functionalities crucial for efficient recruitment. This layer often utilizes technologies like artificial intelligence for resume parsing and matching. Lastly, the data layer involves the storage and management of vast datasets, including job listings, user profiles, and company information. Database systems play a pivotal role in ensuring data integrity, scalability, and quick retrieval. Additionally, security measures are integrated throughout the architecture to safeguard user information and maintain the platform's credibility. The architecture of a job search portal application is designed for flexibility and adaptability, enabling seamless integration of new features and technologies to keep pace with the dynamic landscape of the job market.
Modern employment markets are constantly shifting as digital technology changes our world and job seekers and employers search for easier ways to fulfill their respective needs. As a result, job portal apps are revolutionizing recruitment and the search for jobs.

Job portal apps offer job seekers and employers an effective platform for discovering potential employment. This blog will outline how to design a user-friendly job portal application that benefits employers and job seekers, making this guide an invaluable tool when seeking customized solutions.

For a job portal to be successful, it must have several elements integrated. This includes robust filtering and search options, as well as notification systems and networking functions. Advanced algorithms that match users to job postings that match their skills, location, and experience can enhance user satisfaction. To protect sensitive data effectively, it's also crucial that a secure authentication system and robust data protection policy exist

Adapting the app for mobile devices is crucial since many job portal users use smartphones or tablets. Hire job portal developers to design apps compatible with multiple devices and operating systems. Compatibility should always be top of mind in their designs if they wish for users to enjoy a smooth user experience. Integrating a payment system for premium services. such as enhanced profiles or featured job listings can also monetize an app.

The development of job portal apps offers many benefits to the job seeker. This app accessibility ,efficiency ,convenience while saving the employers money and giving them access to an extensive talented pool. In todays competitive job market , job portal is a valuable tool for bridging gaps between employers and job seekers. They also improve the experience of job scaling and job recruitment. By incorporating the key features in our app development, overcome the challenges and address them properly.

4.2 Key Features of Job search Portal Applications:

1.Filters and Job Search :

Job portals must offer effective job searching and filtering features. Job seekers should be able to filter jobs based on keywords, location, and industry. They can also search by salary, experience level, keyword, etc. Users can narrow their searches and locate relevant job listings by using advanced filtering options.

2.Resume Upload and Builder

The job seeker should have the option to upload their CVs or to use an integrated resume maker to produce a professional CV. The feature streamlines the application process and provides employers comprehensive information on candidates' qualifications.

3.Notifications

Users are kept informed of job openings, the status of their applications, and any messages received from employers through notifications. Job seekers and employers alike should be notified in real-time about new listings for jobs, application status, and communications on the platform. The notifications increase user engagement by keeping them informed.

4.Algorithm for Matching

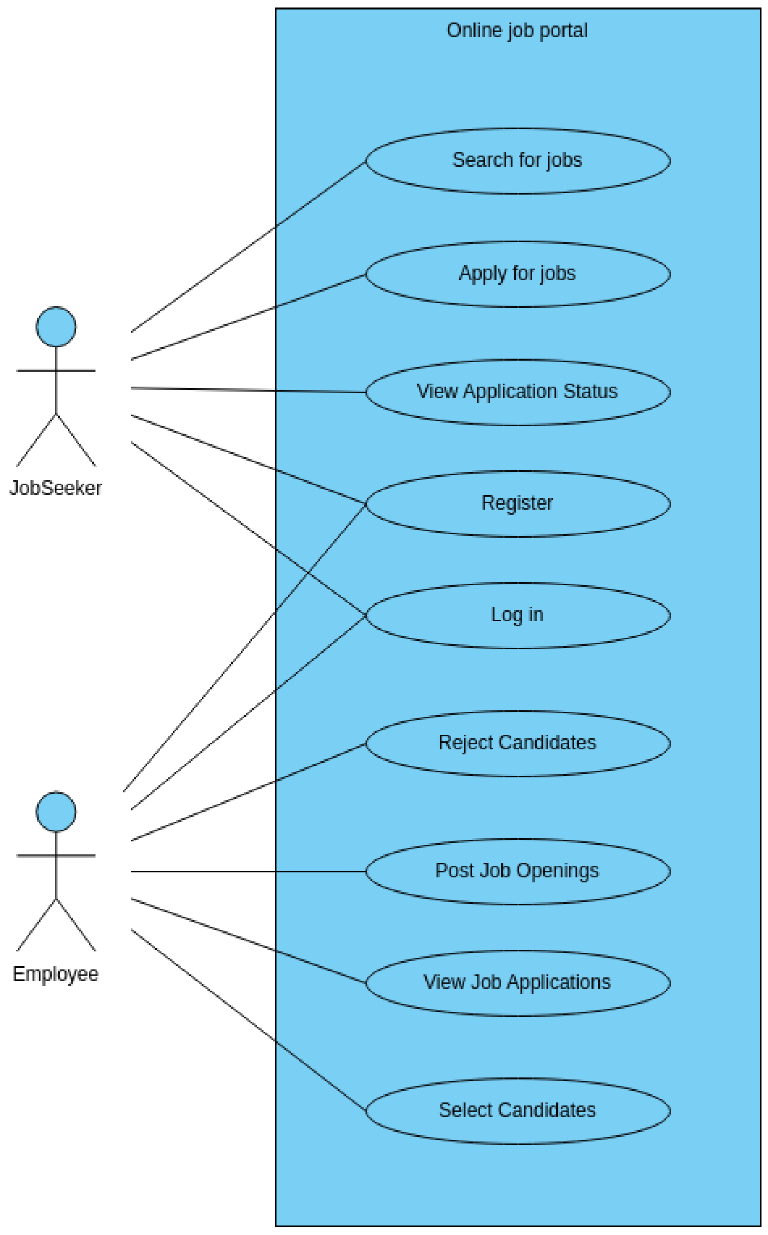
A robust algorithm for matching is critical. The technology is based on data collected from the user profile to provide employers with suitable candidates and job seekers with relevant listings. This algorithm considers skills, location, preferences, and experience to increase the likelihood of a successful match.

4.3 UML Diagrams:

A UML use case diagram can create a broad, high-level view of the relationship between use cases, actors involved, and systems being performed**.**

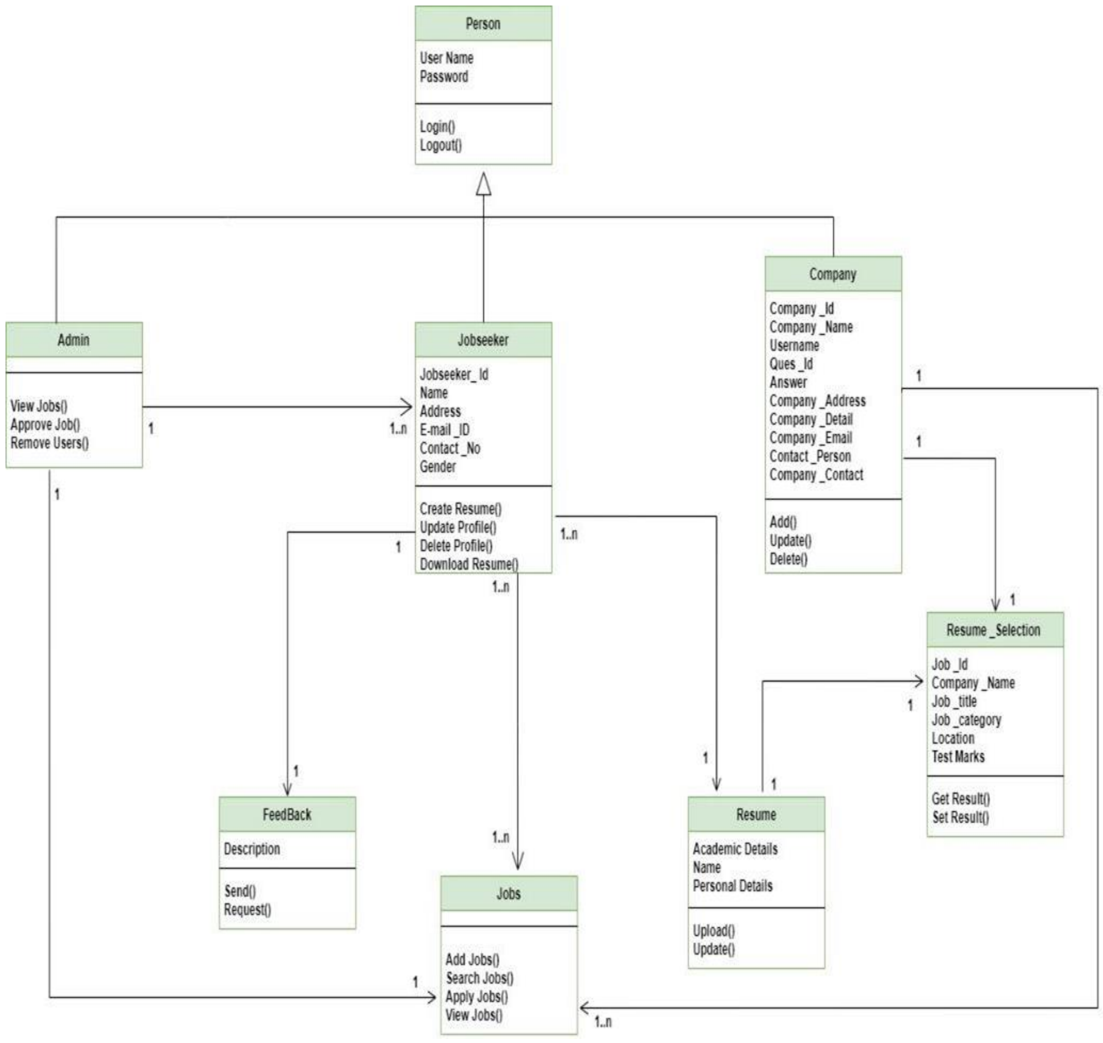
- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Component Diagram
- Activity Diagram

4.3.1 Use Case Diagram:



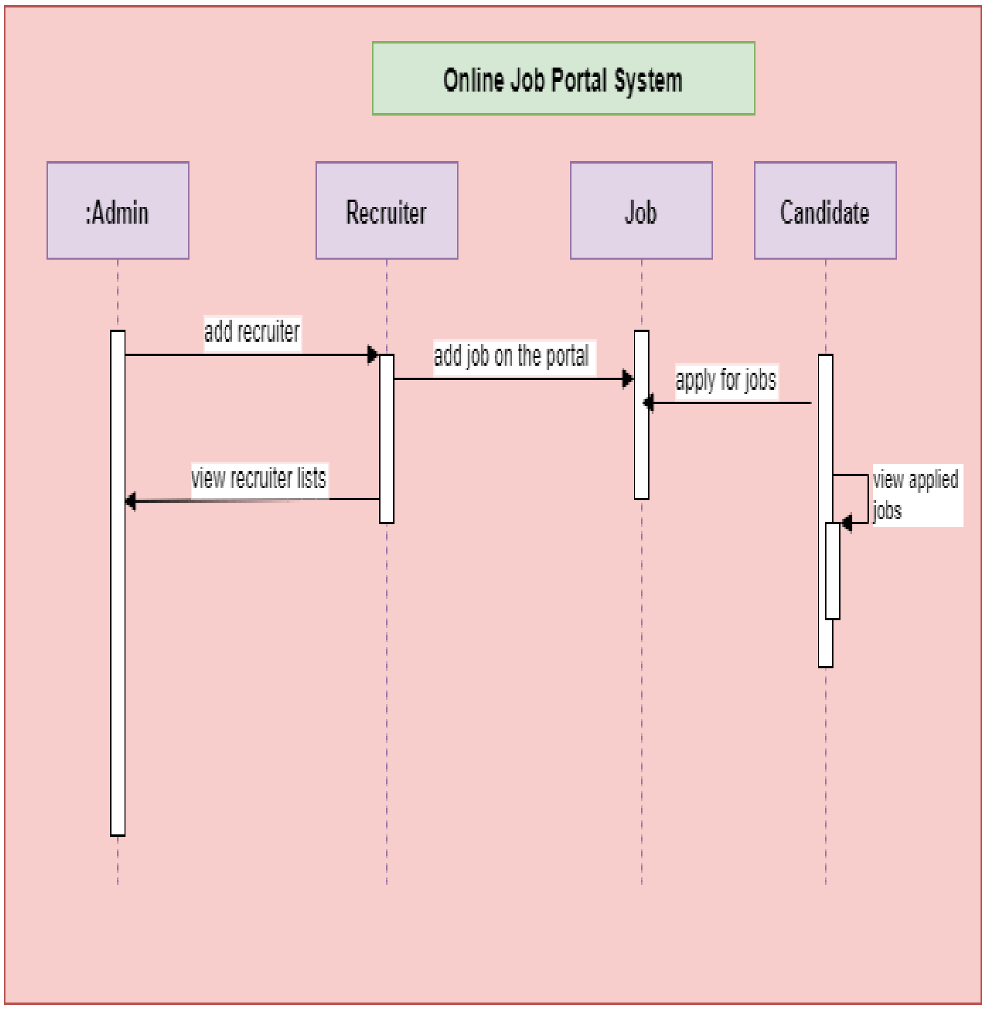
4.3.2 Class Diagram:

The component diagram below shows the structural relations between components in an Online Job Portal System. The connected components by lines represent relationships within the systems.



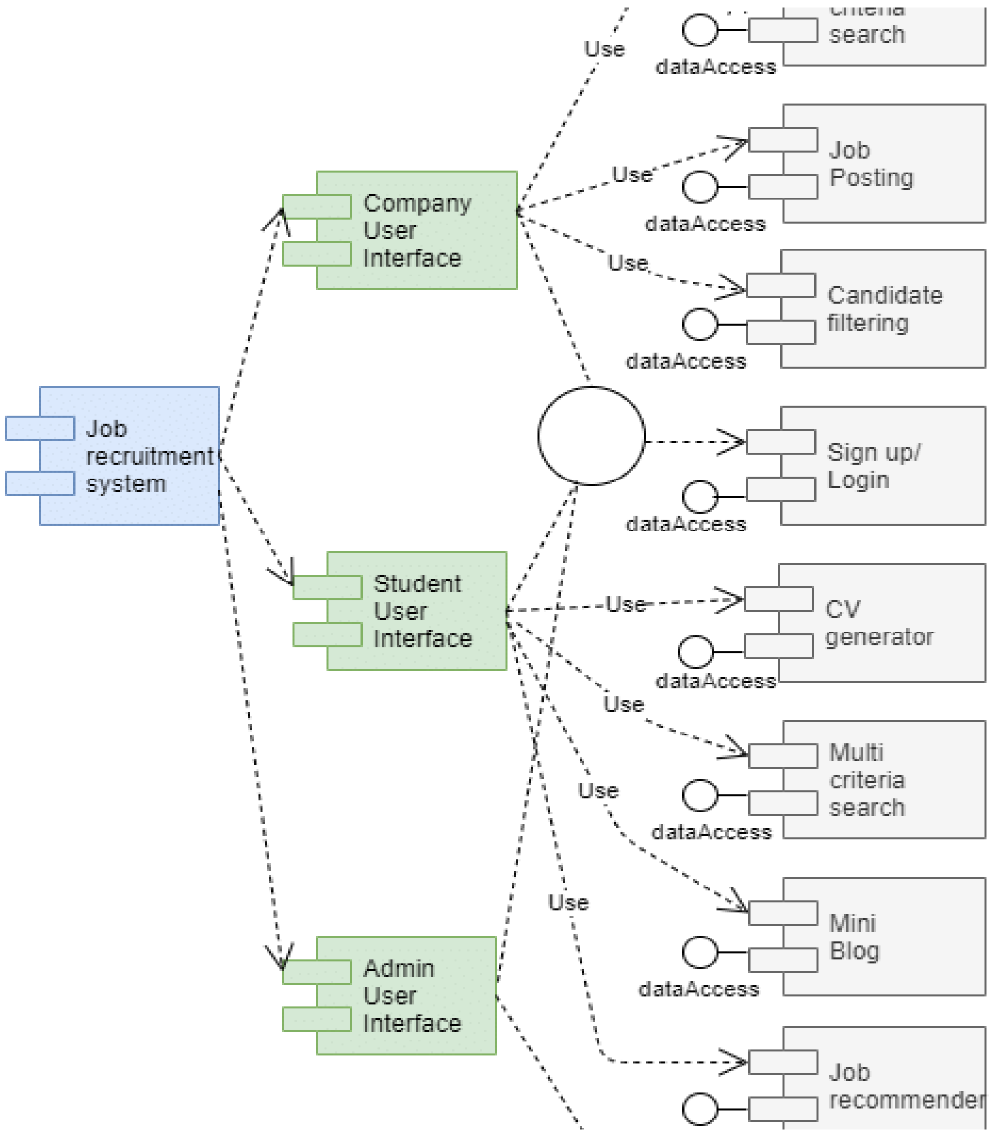
4.3.3 Sequence Diagram:

Sequence diagrams in UML are used to illustrate the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines and the messages that they exchange overtime during the interaction.



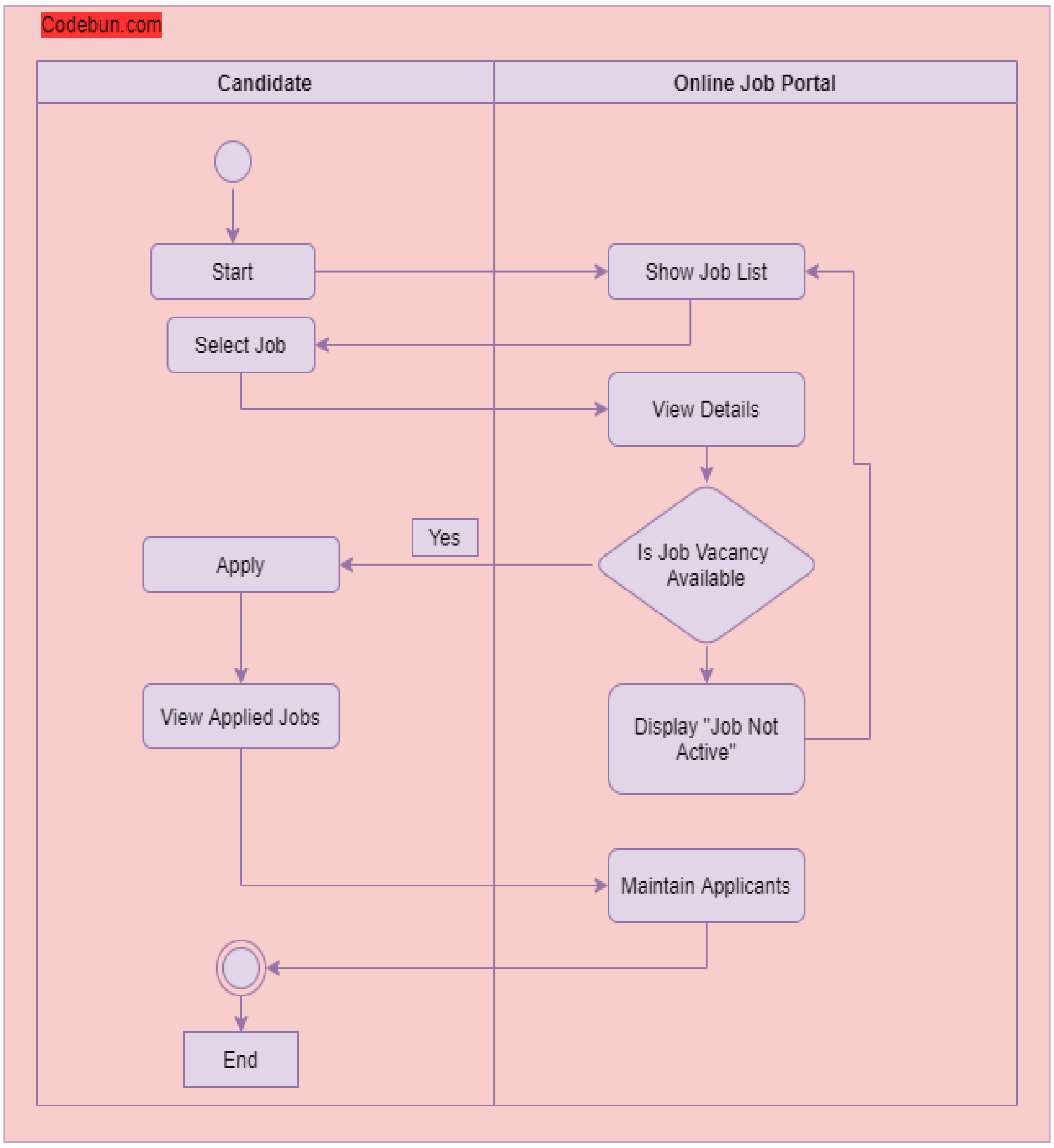
4.3.4 Component Diagram:

The component diagram below shows the structural relations between components in an Online Job Portal System. The connected components by lines represent relationships within the systems.



4.3.5 Activity Diagram

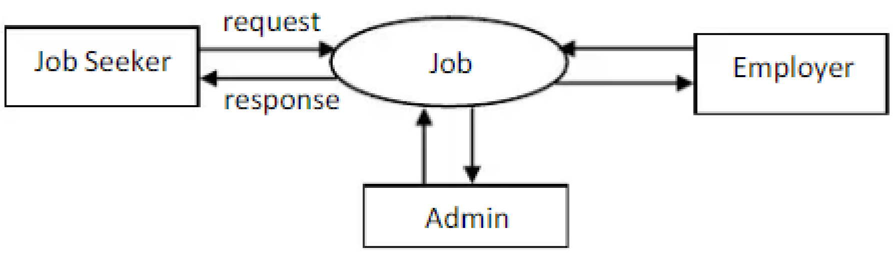
Activity diagrams in UML display the functionalities of various activities and flow in management processes and software systems. A recruiter can View/Confirm/Cancel job applications, manage recruiter/job categories, details, and vacancies can manage applicants,reports,etc.



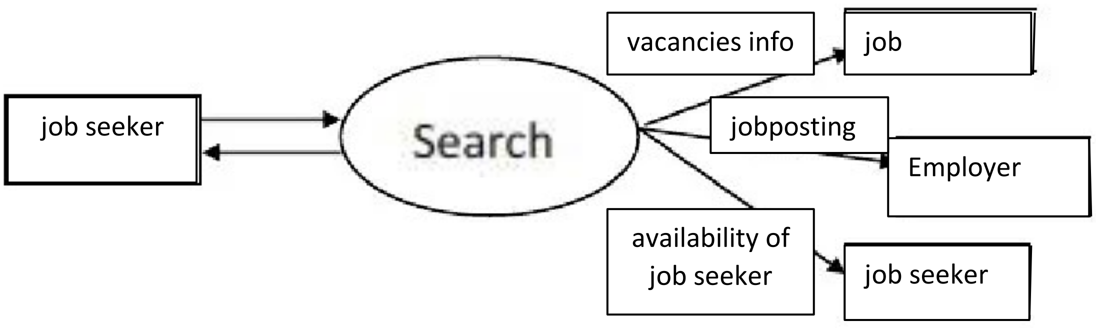
4.3.6 Data Flow Diagram:

A data flow diagram represents the flow of information for any process or system. It shows the system with its relationship to external entities.

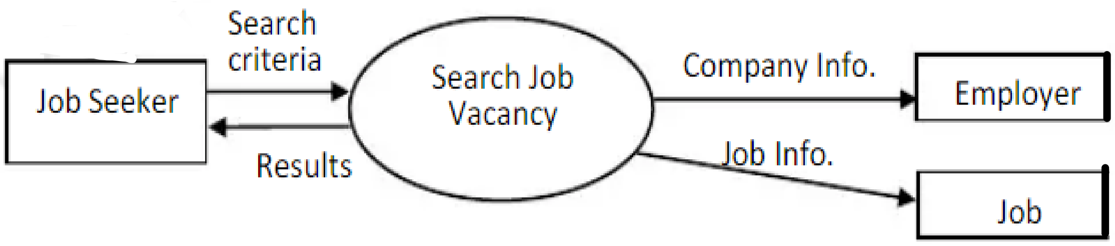
LEVEL 0



LEVEL 1



LEVEL 2



CHAPTER V

IMPLEMENTATION

5.1 Sample Code

```
import {
  View,
  Text,
  StyleSheet,
  ActivityIndicator,
  FlatList,
} from "react-native";
import React, { useState } from "react";
import useRequest from "../../hook/useRequest";
import { COLORS, FONTS, SIZES } from "../../constants";
import MyJobCard from "../cards/my-job-card";

export default function MyJobs() {
  const [selectedJob, setSelectedJob] = useState(null);

  const { data, isLoading, error } = useRequest("search", {
    query: "Javascript Developer in Hyderabad",
    page: "1",
  });

  return (
    <View style={styles.container}>
      <Text style={styles.title}>Jobs for you</Text>

      <View style={styles.jobsContainer}>
        {isLoading ? (
          <ActivityIndicator size={"small"} color={COLORS.primary} />
        ) : error ? (
          <Text>Something went wrong</Text>
        ) : (
          <FlatList
            data={data}
            renderItem={({ item }) => (
              <MyJobCard
                item={item}
                selectedJob={selectedJob}
                setSelectedJob={setSelectedJob}
              />
            )}
            keyExtractor={(item) => `job-${item.job_id}`}
            contentContainerStyle={{ columnGap: SIZES.medium }}
            scrollEnabled={false}
nestedScrollEnabled={true}
/>
)}
</View>
</View>
);
}

const styles = StyleSheet.create({
container: {
marginTop: SIZES.xLarge,
},
title: {
fontSize: SIZES.xLarge,
fontFamily: FONTS.bold,
color: COLORS.primary,
},
jobsContainer: {
marginTop: SIZES.medium,
},
});
// 5.1.2//
import {
View,
Text,
SafeAreaView,
FlatList,
StyleSheet,
ActivityIndicator,
TouchableOpacity,
Image,
} from "react-native";
import React, { useEffect, useState } from "react";
import { COLORS, FONTS, SIZES, icons } from "../../constants";
import { Stack, useGlobalSearchParams, useRouter } from "expo-router";
import { HeaderBtn } from "../../components";
import useRequest from "../../hook/useRequest";
import JobCard from "../../components/cards/job-card";
import axios from "axios";

export default function Search() {
const router = useRouter();
const params = useGlobalSearchParams();

const [data, setData] = useState([]);
const [isLoading, setIsLoading] = useState(false);
const [error, setError] = useState(null);
const [page, setpage] = useState(1);
const handleSearch = async () => {
  setIsLoading(true);
  setData([]);
  try {
    const option = {
      method: "GET",
      url: `https://jsearch.p.rapidapi.com/search`,
      headers: {
        "X-RapidAPI-Key":
          "fb0aa64274mshe95d14dacea4b3bp15e1efjsn5ac73e5a80b7", //tokenchange
        "X-RapidAPI-Host": "jsearch.p.rapidapi.com",
      },
      params: {
        query: params.id,
        page: page.toString(),
      },
    };

    const { data: res } = await axios.request(option);
    setData(res.data);
  } catch (error) {
    setError(error);
    console.log(error);
  } finally {
    setIsLoading(false);
  }
};

const handlePagination = (direction) => {
  if (direction === "left" && page > 1) {
    setpage(page - 1);
    handleSearch();
  } else if (direction === "right") {
    setpage(page + 1);
    handleSearch();
  }
};

useEffect(() => {
  handleSearch();
}, []);

return (
  <SafeAreaView style={{ flex: 1, backgroundColor: COLORS.lightWhite }}>
    <Stack.Screen
      options={{
        headerStyle: { backgroundColor: COLORS.lightWhite },
        headerShadowVisible: false,
        headerTitle: "",
        headerLeft: () => (
          <HeaderBtn
            icon={icons.left}
            dimensions={"60%"}
            onPress={() => router.back()}
          />
        ),
      }}
    />

    <FlatList
      data={data}
      renderItem={({ item }) => (
        <JobCard
          item={item}
          onPress={() => router.push(`/details/${item.job_id}`)}
        />
      )}
      keyExtractor={(item) => `search-job-${item.job_id}`}
      contentContainerStyle={{ padding: SIZES.medium, rowGap: SIZES.medium }}
      ListHeaderComponent={() => (
        <>
          <View style={styles.container}>
            <Text style={styles.searchTitle}>{params.id}</Text>
            <Text style={styles.noOfSearchedJobs}>Job Opportunities</Text>
          </View>
          <View style={styles.loaderWrapper}>
            {isLoading ? (
              <ActivityIndicator size={"small"} color={COLORS.primary} />
            ) : (
              error && <Text>Something went wrong</Text>
            )}
          </View>
        </>
      )}
      ListFooterComponent={() => (
        <View style={styles.footerContainer}>
          <TouchableOpacity
            style={styles.paginationBtn}
            onPress={() => handlePagination("left")}
          >
            <Image
              style={styles.paginationIcon}
              source={icons.chevronLeft}
              resizeMode="contain"
            />
```

```
            </TouchableOpacity>
            <View style={styles.paginationTextWrapper}>
              <Text style={styles.paginationText}>{page}</Text>
            </View>
            <TouchableOpacity
              style={styles.paginationBtn}
              onPress={() => handlePagination("right")}
            >
              <Image
                style={styles.paginationIcon}
                source={icons.chevronRight}
                resizeMode="contain"
              />
            </TouchableOpacity>
          </View>
        )}
      />
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  container: {
    width: "100%",
  },
  searchTitle: {
    fontFamily: FONTS.bold,
    fontSize: SIZES.xLarge,
    color: COLORS.primary,
  },
  noOfSearchedJobs: {
    marginTop: 2,
    fontFamily: FONTS.medium,
    fontSize: SIZES.small,
    color: COLORS.primary,
  },
  loaderWrapper: {
    marginTop: SIZES.medium,
  },
  footerContainer: {
    marginTop: SIZES.small,
    justifyContent: "center",
    alignItems: "center",
    flexDirection: "row",
    gap: 10,
  },
  paginationBtn: {
    width: 30,
    height: 30,
    borderRadius: 5,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: COLORS.tertiary,
  },
  paginationIcon: {
    width: "60%",
    height: "60%",
    tintColor: COLORS.white,
  },
  paginationTextWrapper: {
    width: 30,
    height: 30,
    borderRadius: 2,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: COLORS.white,
  },
  paginationText: {
    fontFamily: FONTS.bold,
    fontSize: SIZES.medium,
    color: COLORS.primary,
  },
});
  //5.1.3//
import { useEffect, useState } from "react";
import axios from "axios";

export default function useRequest(endpoint, query) {
  const [data, setData] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [error, setError] = useState(null);

  const option = {
    method: "GET",
    url: `https://jsearch.p.rapidapi.com/${endpoint}`,
    headers: {
      "X-RapidAPI-Key": "fb0aa64274mshe95d14dacea4b3bp15e1efjsn5ac73e5a80b7",
//tokenchange
      "X-RapidAPI-Host": "jsearch.p.rapidapi.com",
    },
    params: { ...query },
  };

  const fetchData = async () => {
    setIsLoading(true);
    try {
      const { data: res } = await axios.request(option);
      setData(res.data);
    } catch (error) {
      setError(error);
      console.log(error);
    } finally {
      setIsLoading(false);
    }
  };

  useEffect(() => {
    fetchData();
  }, []);

  const refetch = () => {
    setIsLoading(true);
    fetchData();
  };

  return { data, isLoading, error, refetch };
}
```

CHAPTER VI

RESULTS

6.1. Introduction

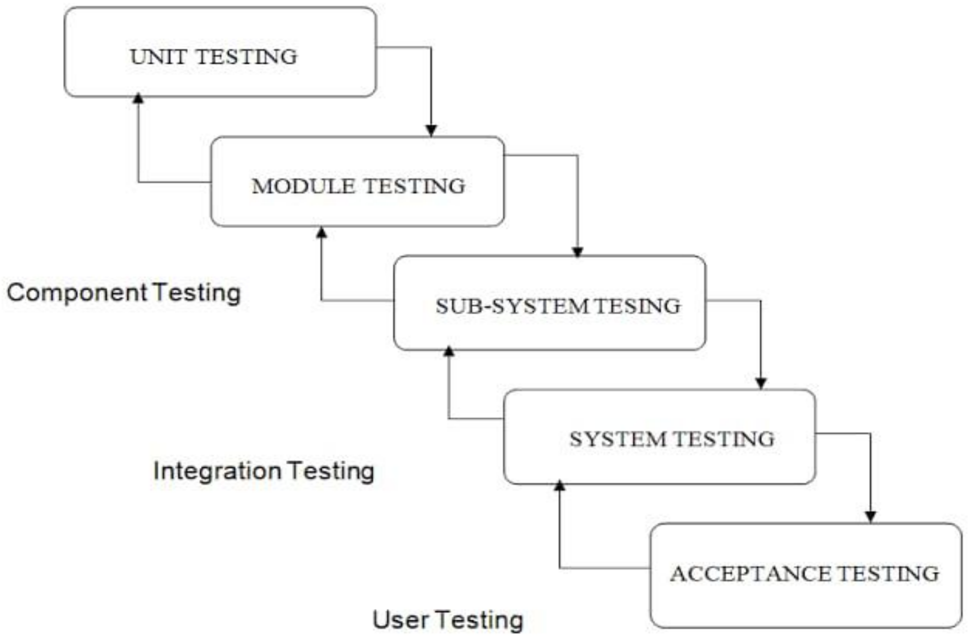
Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

6.2. Strategic Approach To Software Testing

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.



System Security

Introduction

The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural Disaster is known as System Security.

System Security can be divided into four related issues:

- Security
- Integrity
- Privacy
- Confidentiality

System Security refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

Data Security is the protection of data from loss, disclosure, modification and destruction.

System Integrity refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

Privacy defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

6.3 Module Testing

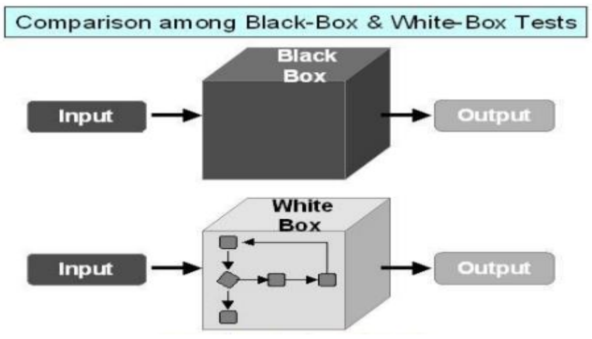


Figure 6.2 *Black Box & White Box Testing*

Each Module can be tested using the following two Strategies:

- Black Box Testing
- White Box Testing

6.3.1 Black Box Testing

Black box testing is a software testing technique in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal

Figure 6.3.1 *Black Box Testing*

paths of the software. This type of testing is based entirely on the software requirements and specifications. In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

The above Black Box can be any software system you want to test. For example: an operating system like Windows, a website like Google, a database like Oracle or even yourown custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Black box testing – Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

Types of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones

- **Functional testing** –This black box testing type is related to functional requirements of a system; it is done by software testers.
- **Non-functional testing –**This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- **Regression testing –**Regression testing is done after code fixes, upgrades or another system maintenance to check the new code has not affected the

existing code.

6.3.2 White Box Testing

White Box Testing is the testing of a software solution's internal coding and **Regression testing** –Regression testing is done after code fixes, upgrades or another system maintenance to check the new code has not affected the existing code infrastructure.It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as clear,open, structural, and glass box testing. It is one of two parts of the "box testing" approachof software testing. Its counterpart, Black box testing, involves testing from an external orend-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing. The term "White box" wasused because of the see-through box concept. The clear box or White box name symbolizesthe ability to see through the software's outer shell (or "box") into its inner workings.

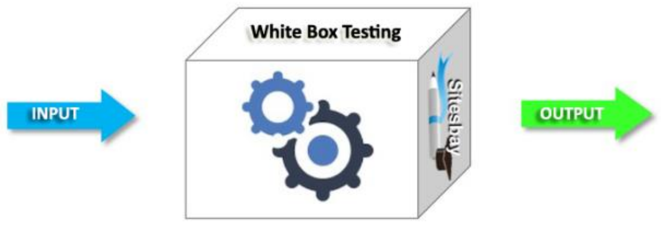


Figure 6.3.2 *White Box Testing*

Likewise, the "black box" in "black box testing" symbolizes not being able to see the innerworkings of the software so that only the end-user experience can be tested.

What do you verify in White Box Testing?

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding process
- The flow of specific inputs through the code

- Expected output
- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of Whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so thatwhen a specific input does not result in the expected output, you have encountered a bug.

How do you perform White Box Testing?

To give you a simplified explanation of white box testing, we have divided it into two basicsteps. This is what testers do when testing an application using the white box testing technique:

Understand the Source Code

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used inthe applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

Create Test Cases and Execute

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

Application versus embedded testing

Embedded systems software testing shares much in common with application software testing. Thus, much of this two-part article is a summary of basic testing concepts and terminology. However, some important differences exist between application testing and embedded systems testing. Embedded developers often have access to hardware-based testtools that are generally not used in application development. Also, embedded systems oftenhave unique characteristics that should be reflected in the test plan. These differences tendto give embedded systems testing its own distinctive flavor. This article covers the basics of testing and test case development and points out details unique to embedded systems work along the way.

Why Test?

Before you begin designing tests, it’s important to have a clear understanding of why you are

testing. This understanding influence which tests you stress and (more importantly) howearly you begin testing. In general, you test for four reasons:

- To find bugs in software (testing is the only way to do this)
- To reduce risk to both users and the company
- To reduce development and maintenance costs
- To improve performance
- To Find the Bugs

One of the earliest important results from theoretical computer science is a proof (knownas the Halting Theorem) that it’s impossible to prove that an arbitrary program is correct. To Reduce Costs.

The classic argument for testing comes from Quality Warsby Jeremy Main. In

1990, HP sampled the cost of errors in software development during the year. The answer,

$400 million, shocked HP into a completely new effort to eliminate mistakes in writing software.

The $400M waste, half of it spent in the labs on rework and half in the field to fix the mistakes that escaped from the labs, amounted to one-third of the company’s total R&D budget and could have increased earnings by almost 67%. The earlier a bug is found, the less expensive it is to fix. The cost of finding errors and bugs in a released product is significantly higher than during unit testing, for example

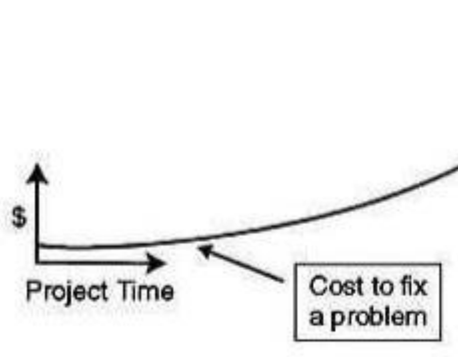


Figure 6.3.3 *Relationship between project time and cost to fix a problem*

Figure: The Cost to Fix a Problem. Simplified graph showing the cost to fix a problem asa function of the time in the product life cycle when the defect is found. The costs associated with finding and fixing the Y2K problem in embedded systems is a close approximation to an infinite cost model.

6.4 Design Of Test Cases And Scenarios

Various levels of testing are :

1. White Box Testing
2. Black Box Testing
3. Unit Testing
4. Functional Testing
5. Performance Testing
6. Integration Testing
7. Objective
8. Validation Testing
9. System Testing
10. Structure Testing
11. Output Testing
12. User Acceptance Testing

1) <u>White Box Testing</u> :-

1. Search Component:
   - Verify that the search input field component is correctly implemented.
   - Verify that the search button component is correctly implemented.
   - Verify that the search functionality is implemented correctly.
2. My Job Card Component:
   - Verify that the employer logo component is correctly implemented.
   - Verify that the job title component is correctly implemented.
   - Verify that the navigation functionality is correctly implemented.
3. Job Tabs Component:
   - Verify that all tabs are correctly implemented.
   - Verify that the state management for active tab is correctly implemented.
   - Verify that clicking on a tab triggers the correct state update.
4. Job Component:
   - Verify that the company logo component is correctly implemented.
   - Verify that the job title, company name, and location components are correctly implemented.
5. My Jobs Component:
   - Verify that the data fetching mechanism is correctly implemented.
   - Verify that the loading indicator component is correctly implemented.
   - Verify that the error message component is correctly implemented.
   - Verify that the job card component is correctly implemented.
   - Verify that the navigation functionality is correctly implemented.
6. Search Component:
   - Verify that the search input field component is correctly implemented.
   - Verify that the search button component is correctly implemented.
   - Verify that the search functionality is correctly implemented.
7. Header Button Component:
   - Verify that the button component is correctly implemented.
   - Verify that the action triggered by clicking the button is correctly implemented.

2) <u>Black Box Testing:</u>-

Test Case	Check Field	Expected Results
TC-01	Search Component	- Verify that the search input field is rendered. - Verify that the search button is rendered. - Verify that clicking the search button triggers the search functionality with the correct search term.
TC-02	My Job Card Component	- Verify that the employer logo is rendered. - Verify that the job title is displayed correctly. - Verify that clicking on the job card triggers the correct navigation.
TC-03	Job Tabs Component	- Verify that all tabs are rendered. - Verify that clicking on a tab updates the active tab state correctly.
TC-04	Job Component	- Verify that the company logo is rendered. - Verify that the job title, company name, and location are displayed correctly.
TC-05	My Jobs Component	- Verify that the correct data is fetched from the API. - Verify that the loading indicator is displayed while fetching data. - Verify that an error message is displayed if there's an error in fetching data. - Verify that each job card is rendered

		with the correct data. - Verify that clicking on a job card triggers the correct navigation.
TC-06	Search Component	- Verify that the search input field is rendered. - Verify that the search button is rendered. - Verify that typing in the search field updates the state correctly. - Verify that clicking the search button triggers the search functionality with the correct search term.
TC-07	Header Button Component	- Verify that the button is rendered. - Verify that clicking the button triggers the correct action.
TC-001	Search and Filtering	-Test the search functionality with various keywords. -Check the accuracy and relevance of search results. -Test advanced search options like job type, location, and salary range filters. -Confirm that users can sort search results by relevance, date, or other criteria. -Ensure that search results pagination works correctly. -Test the “Save Search” feature for registered users.

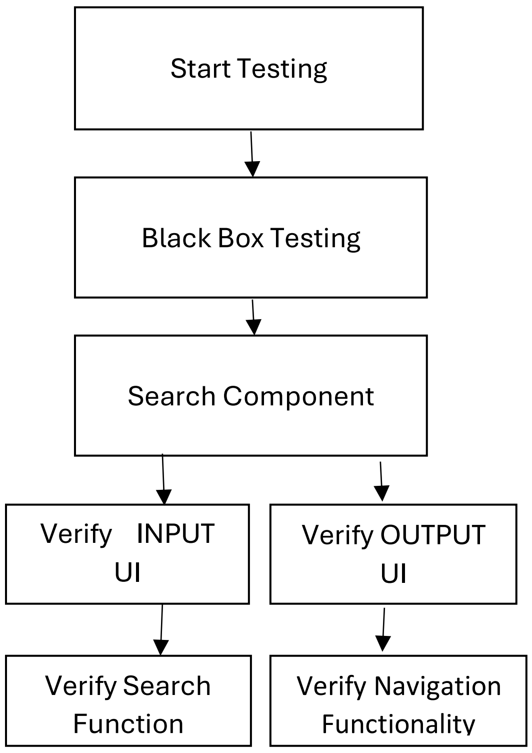
TC-002	Job Posting	-Test the job posting process for employers. -Verify that job details, requirements, and location are accurately displayed. -Ensure that posted jobs appear in search results. -Test the editing and removal of job postings. -Confirm that employers receive notifications for new job applications.
TC-003	Notifications and Alerts	-Test email and mobile app notifications for job recommendations. -Verify notifications for new messages, application status updates, and job posting approvals. -Test the option to customize notification preferences. -Ensure that users receive alerts for expired job postings.
TC-004	Mobile Responsiveness	-Test the website's responsiveness on various mobile devices and screen sizes. -Verify that touch gestures (e.g., pinch-to-zoom) work as expected. -Test landscape and portrait orientations on mobile devices. -Confirm that mobile navigation menus and buttons are accessible and functional.

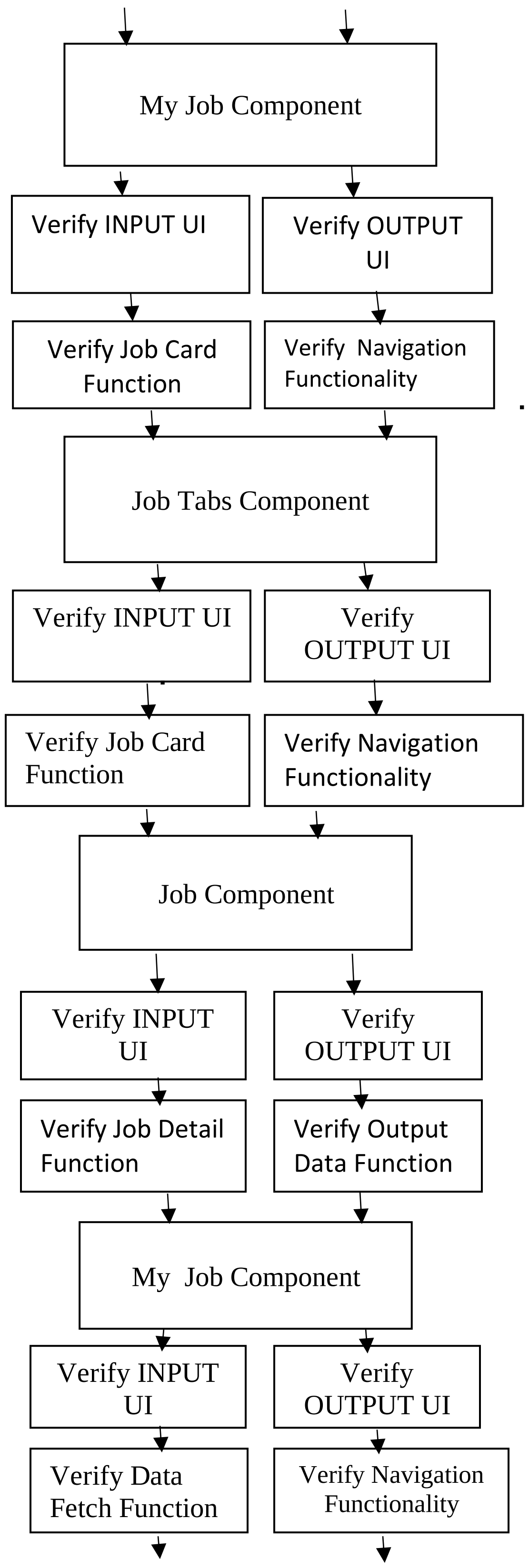
TC-005	Performance Testing	-Perform load testing to assess the portal’s performance under peak loads. -Monitor response times, server capacity, and scalability during load testing. -Verify that the website remains responsive during peak traffic.
TC-006	Security Testing	-Test for vulnerabilities like SQL injection and cross-site scripting (XSS). -Ensure the security of user data and transactions. -Test the password reset process for security and validation. -Verify that user sessions are securely managed and terminated
TC-007	Resume Parsing	-Test the resume parsing feature to extract information accurately from uploaded resumes. -Verify that parsed information is correctly populated in user profiles. -Ensure that users can edit and refine parsed information as needed.
TC-008	Company Profiles	-Test the creation and management of company profiles by employers. -Confirm that company logos and descriptions display correctly. -Test the verification process for company profiles (if applicable). -Ensure that users can follow or favorite companies for updates.

TC-009	Job Recommendation	-Verify the accuracy of job recommendations based on user profiles and preferences. -Test the recommendation algorithm’s ability to suggest relevant jobs. -Confirm that users receive personalized job recommendations in their dashboards.
TC-010	Social Media Integration	-Test the integration of social media sharing options for job postings. -Confirm that users can share job listings on platforms like LinkedIn, Twitter, and Facebook. -Verify that shared job postings include relevant information and images.
TC-011	Accessibility Testing	-Perform accessibility testing to ensure the website is usable by individuals with disabilities. -Verify that all website features are accessible via keyboard navigation. -Test for proper ARIA attributes and alt text for images.
TC-012	Mobile App Testing	-Test the functionality and usability of the mobile app version of the job portal. -Verify that app notifications are synchronized with the website. -Test app performance, including load times and responsiveness.

TC-013	Browser Capability	-Test the job portal website on different web browsers (e.g., Chrome, Firefox, Safari, Edge). -Verify that all features work consistently across supported browsers. -Check for any rendering or layout issues in specific browsers.
TC-014	Data Backup and Recovery	-Test the system's ability to backup user data regularly. -Verify that users can recover their accounts and data in case of a system failure. -Test the data restoration process for accuracy.

Below is a flowchart representing the testing process for both black box and white box testing of the React Native project:




My Job Component
Verify INPUT UI
Verify OUTPUT UI
Verify Job Card Function
Verify Navigation Functionality
Job Tabs Component
Verify INPUT UI
Verify OUTPUT UI
Verify Job Card Function
Verify Navigation Functionality
Job Component
Verify INPUT UI
Verify OUTPUT UI
Verify Job Detail Function
Verify Output Data Function
My Job Component
Verify INPUT UI
Verify OUTPUT UI
Verify Data Fetch Function
Verify Navigation Functionality

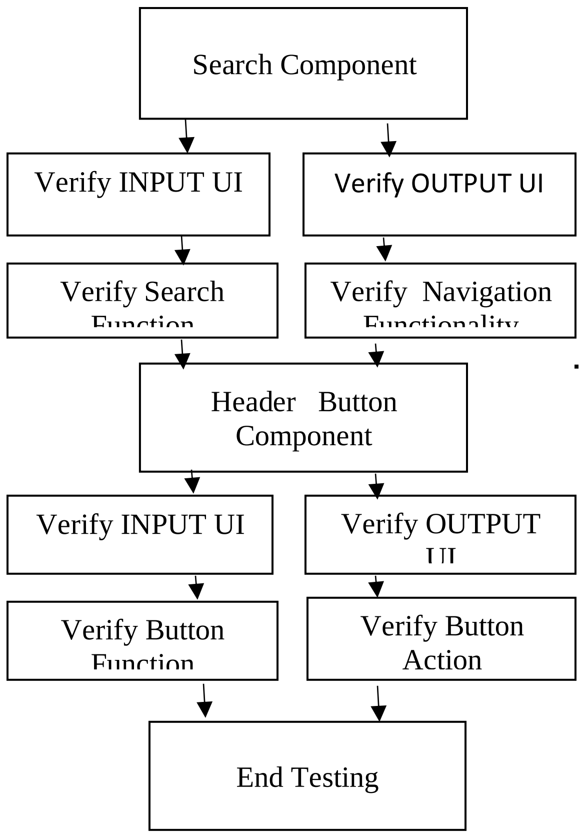


Figure : 6.4 *Test Process Flow Chart*

3) <u>Unit Testing</u>

Unit testing, also known as Module Testing, focuses verification efforts on the module. The module is tested separately and this is carried out at the programming stage itself. Unit Test comprises of the set of tests performed by an individual programmer before integration of the unit into the system. Unit test focuses on the smallest unit of software design- the software component or module. Using component level design, important control paths are tested to uncover errors within the boundary of the module. Unit test is white box oriented and the step can be conducted in parallel for multiple components.

4) <u>Functional Testing</u>

Functional test cases involve exercising the code with normal input values for which the expected results are known, as well as the boundary values.

5) <u>Performance Testing</u>

Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization of the program unit. It occurs throughout all steps in the testing process.

6) <u>Integration Testing</u>

It is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with in the interface. It takes the unit tested modules and builds a program structure. All the modules are combined and tested as a whole. Integration of all the components to form the entire system and a overall testing is executed.

7) <u>Validation Testing</u>

Validation test succeeds when the software functions in a manner that can be reasonably expected by the client. Software validation is achieved through a series of black box testing which confirms to the requirements. Black box testing is conducted at the software interface. The test is designed to uncover interface errors, is also used to demonstrate that software functions are operational, input is properly accepted, output are produced and that the integrity of external information is maintained.

8) <u>System Testing</u>

Tests to find the discrepancies between the system and its original objective, current specifications and system documentation.

9) <u>Structure Testing</u>

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

10) <u>Output Testing</u>

Output of test cases compared with the expected results created during design of test cases. Asking the user about the format required by them tests the output generated or displayed by the system under consideration. Here, the output format is considered into two was, one is on screen and another one is printed format. The output on the screen is found to be correct as the format was designed in the system design phase according to user needs. The output comes out as the specified requirements as the user's hard copy.

11) <u>User acceptance</u>

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

a) Two set of acceptance test to be run:

i. Those developed by quality assurance group.

ii. Those developed by customer.

12) The objective is to take unit-tested modules and build a program structure that has been dictated by design.

Validation

Validation test succeeds when the software functions in a manner that can be reasonably expected by the client. Software validation is achieved through a series of black box testing which confirms to the requirements. Black box testing is conducted at the software interface.

The test is designed to uncover interface errors, is also used to demonstrate that software functions are operational, input is properly accepted, output are produced and that the integrity of external information is maintained.

V Model

The "V" model (sometimes known as the "U" model) reflects the approach to systems development where in the definition side of the model is linked directly to the confirmation side. It specifies early testing and preparation of testing scenarios and cases before the build stage to simultaneously validate the definitions and prepare for the test stages. It is the standard for German federal government projects and is considered as much a project management method as a software development approach.

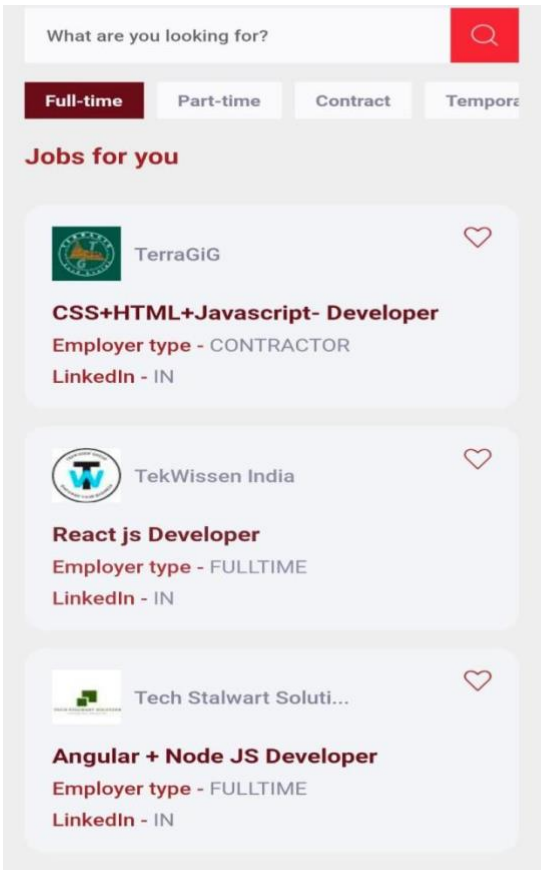
“The V Model, while admittedly obscure, gives equal weight to testing rather than 52 treating it as an afterthought. Initially defined by the late Paul Rook in the late 1980s, the V was included in the U.K.'s National Computing Centre publications in the 1990s with the aim of improving the efficiency and effectiveness of software development. It's accepted in Europe and the U.K. as a superior alternative to the waterfall model; yet in the U.S., the V Model is often mistaken for the waterfall.

“In fact, the V Model emerged in reaction to some waterfall models that showed testing as a single phase following the traditional development phases of requirements analysis, high-level design, detailed design and coding. The waterfall model did considerable damage by supporting the common impression that testing is merely a brief detour after most of the mileage has been gained by mainline development activities. Many managers still believe this, even though testing usually takes up half of the project time." (Goldsmith and Graham, "The Forgotten Phase", Software development, July 2002). As shown below, the model is the shape of the development cycle (a waterfall wrapped around) and the concept of flow down and across the phases. The V shows the typical sequence of development activities on the left-hand (downhill) side and the corresponding sequence of test execution activities on the right-hand (uphill) side. Example of a typical V Model (IEEE) The primary contribution the V Model makes is this alignment of testing and specification.

This is also an advantage to the business analyst who can use the model and approach to enforce early consideration of later testing. The V Model emphasizes that testing is done throughout the SDLC rather than just at the end of the cycle and reminds the business analyst to prepare the test cases and scenarios in advance while the solution is being defined.

The business analyst's role in the V Model is essentially the same as the waterfall. The business analyst is involved full time in the specification of the business problem and the confirmation and validation that the business problem has been solved which is done at acceptance test. The business analyst is also involved in the requirements phases and advises the system test stage which is typically performed by independent testers – the quality assurance group or someone other than the development team. The primary business analyst involvement in the system test stage is keeping the requirements updated as changes occur and providing "voice of the customer" to the testers and development team.

6.5 RESULT SNAPSHOTS


What are you looking for?

Full-time Part-time Contract Tempora

Jobs for you

TerraGiG

CSS+HTML+Javascript- Developer

Employer type - CONTRACTOR

LinkedIn - IN

TekWissen India

React js Developer

Employer type - FULLTIME

LinkedIn - IN

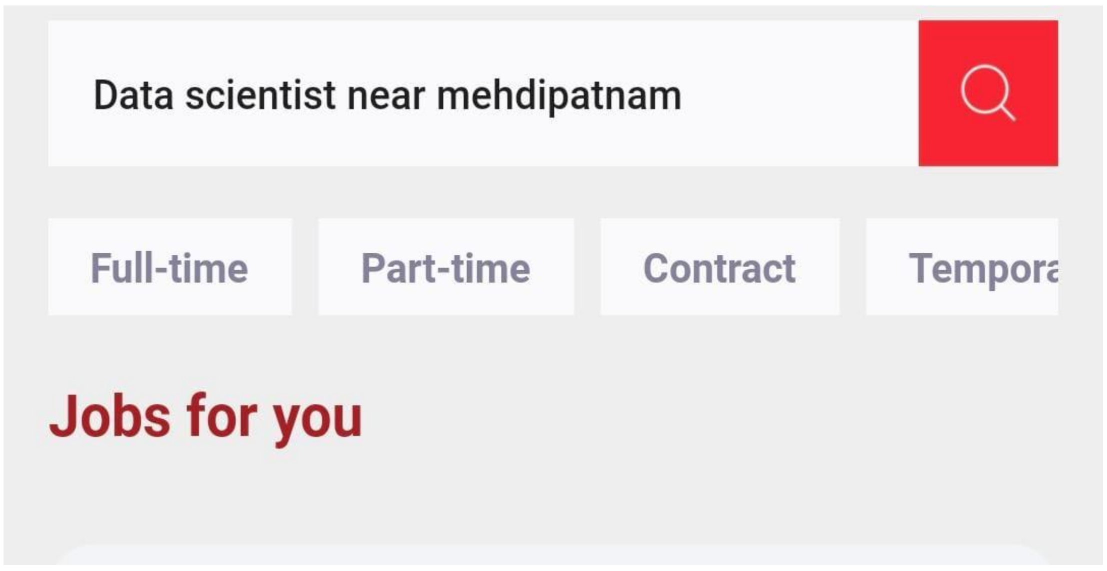
Tech Stalwart Soluti...

Angular + Node JS Developer

Employer type - FULLTIME

LinkedIn - IN

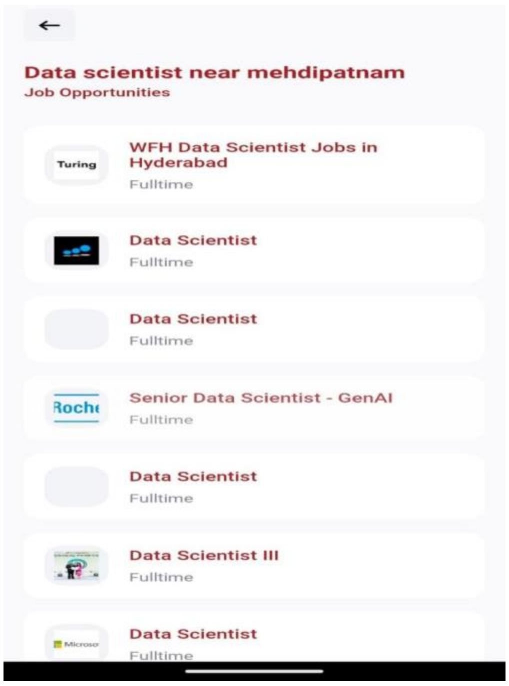
Figure: *6.5.1 Android Application Interface*


Data scientist near mehdipatnam

Full-time Part-time Contract Tempora

Jobs for you

Figure*: 6.5.2 Search for a job*



←

Data scientist near mehdipatnam
Job Opportunities

WFH Data Scientist Jobs in Hyderabad
Fulltime

Data Scientist
Fulltime

Data Scientist
Fulltime

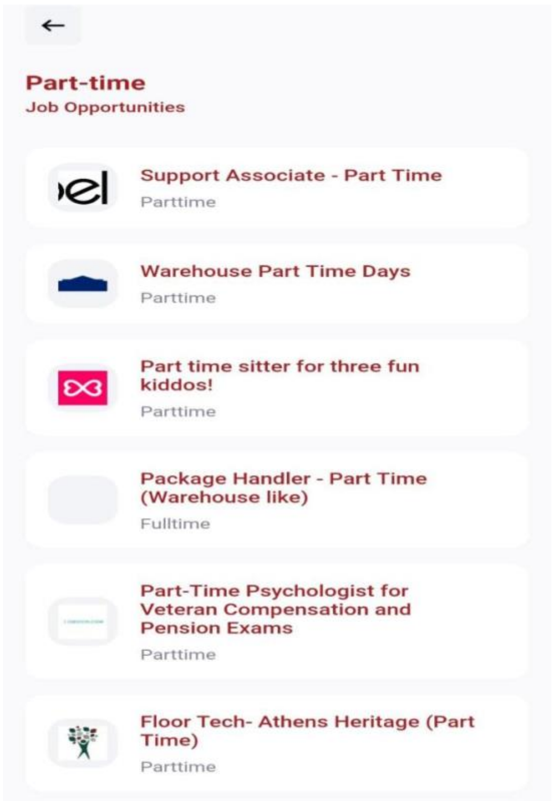
Senior Data Scientist - GenAI
Fulltime

Data Scientist
Fulltime

Data Scientist III
Fulltime

Data Scientist
Fulltime

Figure: *6.5.3 Search Results*



←

Part-time
Job Opportunities

Support Associate - Part Time
Parttime

Warehouse Part Time Days
Parttime

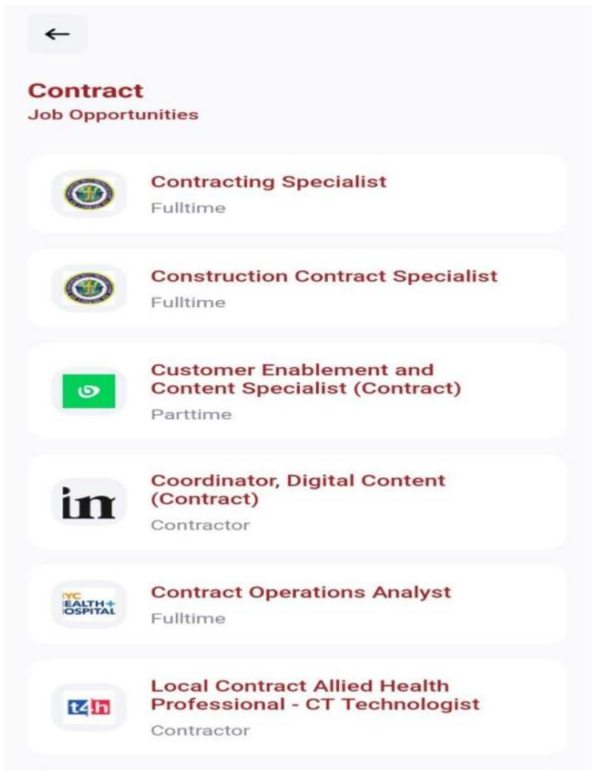
Part time sitter for three fun kiddos!
Parttime

Package Handler - Part Time (Warehouse like)
Fulltime

Part-Time Psychologist for Veteran Compensation and Pension Exams
Parttime

Floor Tech- Athens Heritage (Part Time)
Parttime

Figure: *6.5.4 Part Time Job Opportunities*


Contract
Job Opportunities

Contracting Specialist
Fulltime

Construction Contract Specialist
Fulltime

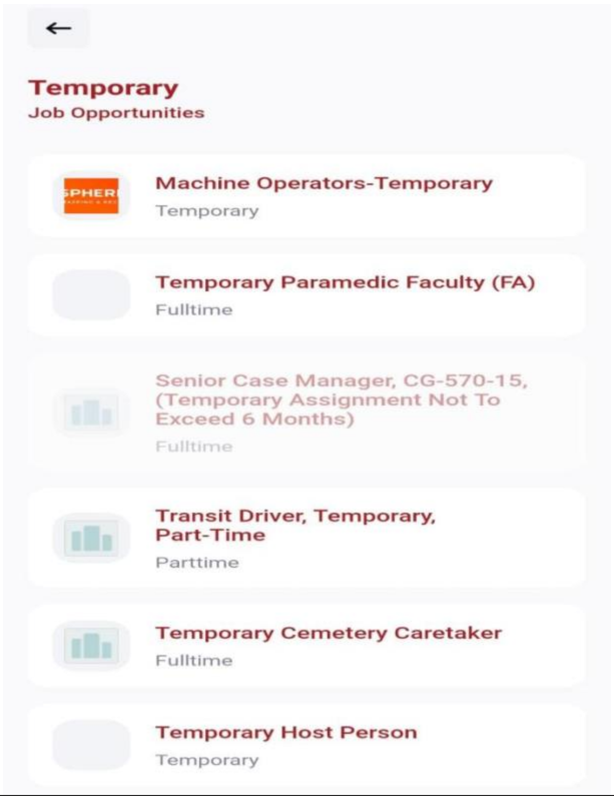
Customer Enablement and Content Specialist (Contract)
Parttime

Coordinator, Digital Content (Contract)
Contractor

Contract Operations Analyst
Fulltime

Local Contract Allied Health Professional - CT Technologist
Contractor

Figure: *6.5.5 Contract Job Opportunities*


Temporary
Job Opportunities

Machine Operators-Temporary
Temporary

Temporary Paramedic Faculty (FA)
Fulltime

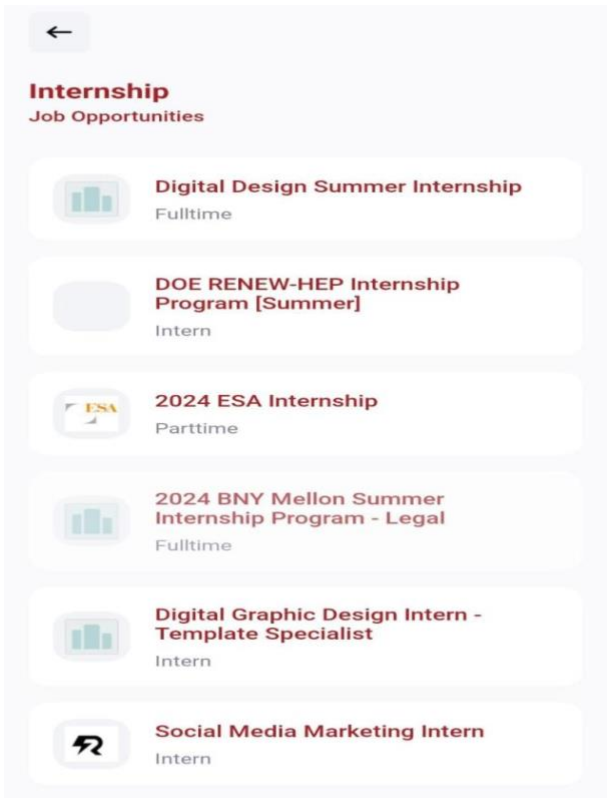
Senior Case Manager, CG-570-15, (Temporary Assignment Not To Exceed 6 Months)
Fulltime

Transit Driver, Temporary, Part-Time
Parttime

Temporary Cemetery Caretaker
Fulltime

Temporary Host Person
Temporary

Figure: *6.5.6 Temporary Job Opportunities*



Internship

Job Opportunities

Digital Design Summer Internship

Fulltime

DOE RENEW-HEP Internship Program [Summer]

Intern

2024 ESA Internship

Parttime

2024 BNY Mellon Summer Internship Program - Legal

Fulltime

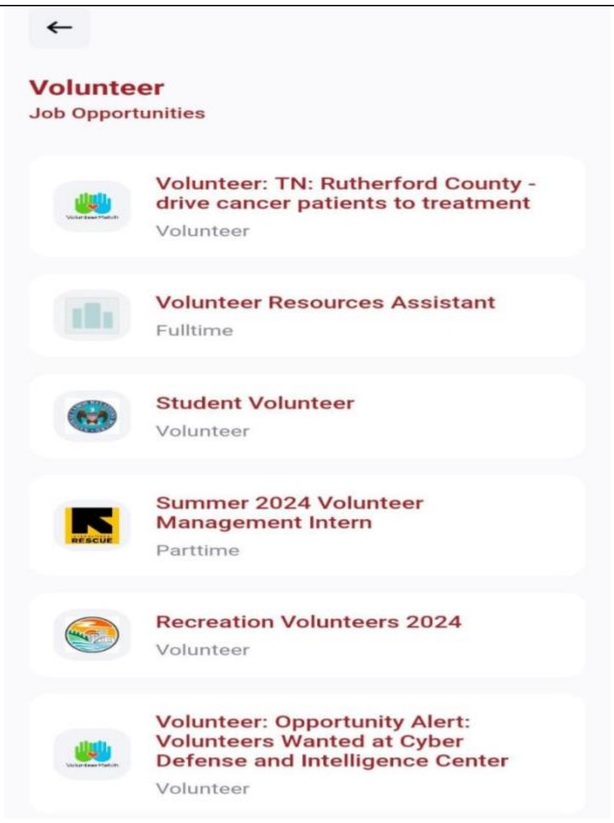
Digital Graphic Design Intern - Template Specialist

Intern

Social Media Marketing Intern

Intern

Figure: *6.5.7 Internship Job Opportunities*



Volunteer

Job Opportunities

Volunteer: TN: Rutherford County - drive cancer patients to treatment

Volunteer

Volunteer Resources Assistant

Fulltime

Student Volunteer

Volunteer

Summer 2024 Volunteer Management Intern

Parttime

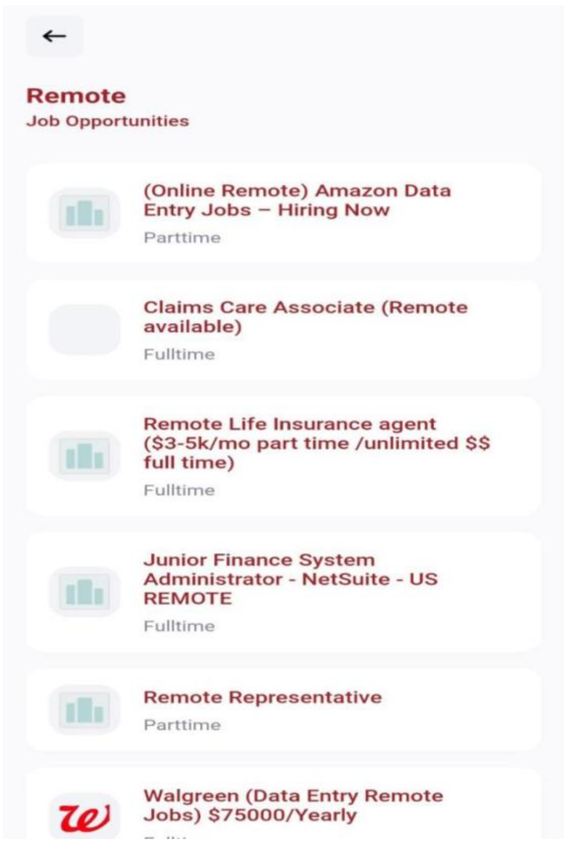
Recreation Volunteers 2024

Volunteer

Volunteer: Opportunity Alert: Volunteers Wanted at Cyber Defense and Intelligence Center

Volunteer

Figure: *6.5.8 Volunteer Job Opportunities*


Remote
Job Opportunities

(Online Remote) Amazon Data Entry Jobs – Hiring Now
Parttime

Claims Care Associate (Remote available)
Fulltime

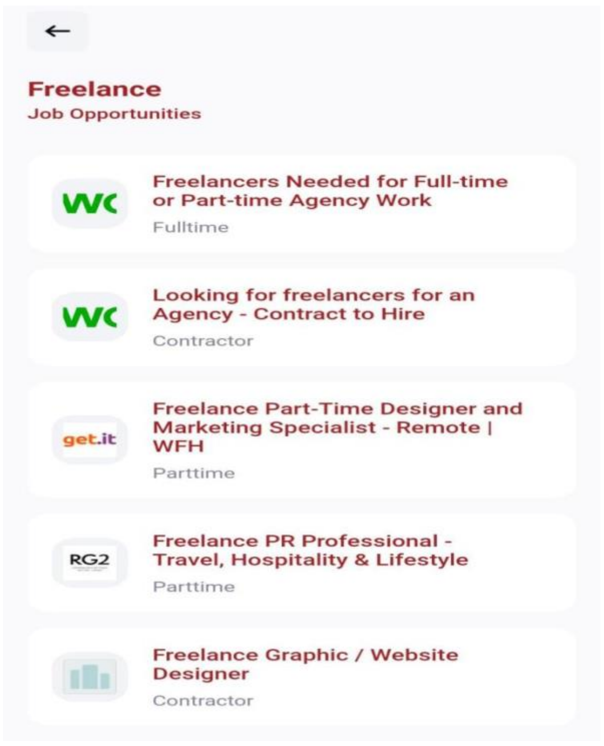
Remote Life Insurance agent ($3-5k/mo part time /unlimited $$ full time)
Fulltime

Junior Finance System Administrator - NetSuite - US REMOTE
Fulltime

Remote Representative
Parttime

Walgreen (Data Entry Remote Jobs) $75000/Yearly

Figure: *6.5.9 Remote Job Opportunities*


Freelance
Job Opportunities

Freelancers Needed for Full-time or Part-time Agency Work
Fulltime

Looking for freelancers for an Agency - Contract to Hire
Contractor

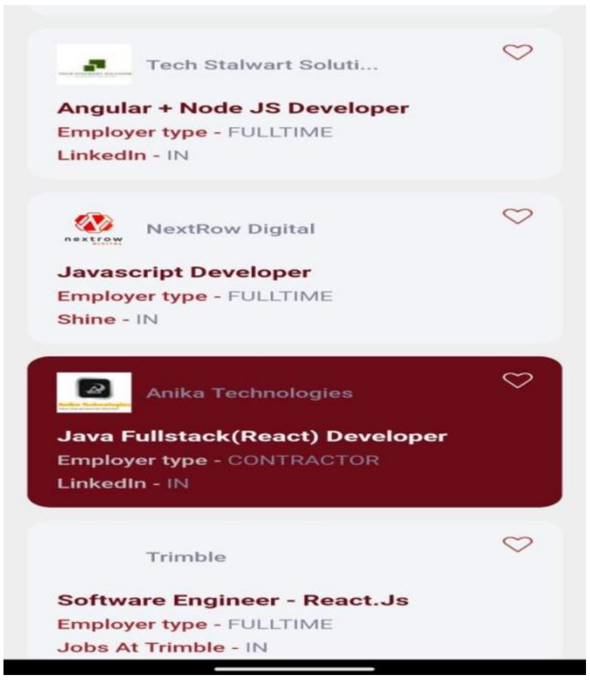
Freelance Part-Time Designer and Marketing Specialist - Remote | WFH
Parttime

Freelance PR Professional - Travel, Hospitality & Lifestyle
Parttime

Freelance Graphic / Website Designer
Contractor

Figure: *6.5.10 Freelance Job Opportunities*

Job Selection Process:



Tech Stalwart Soluti...

Angular + Node JS Developer
Employer type - FULLTIME
LinkedIn - IN

NextRow Digital

Javascript Developer
Employer type - FULLTIME
Shine - IN

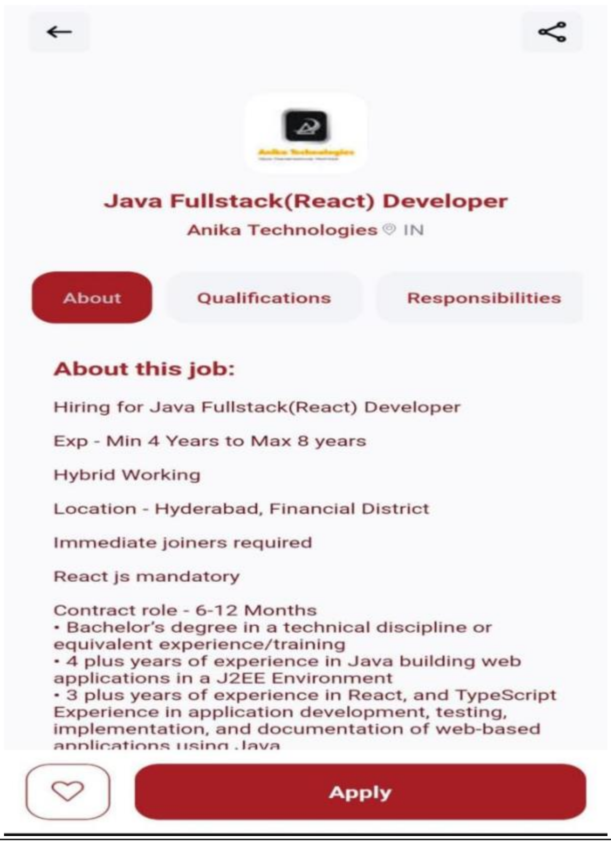
Anika Technologies

Java Fullstack(React) Developer
Employer type - CONTRACTOR
LinkedIn - IN

Trimble

Software Engineer - React.Js
Employer type - FULLTIME
Jobs At Trimble - IN

Figure: *6.5.11 Selecting a Job(also add in wish list)*



Java Fullstack(React) Developer

Anika Technologies IN

About | Qualifications | Responsibilities

About this job:

Hiring for Java Fullstack(React) Developer

Exp - Min 4 Years to Max 8 years

Hybrid Working

Location - Hyderabad, Financial District

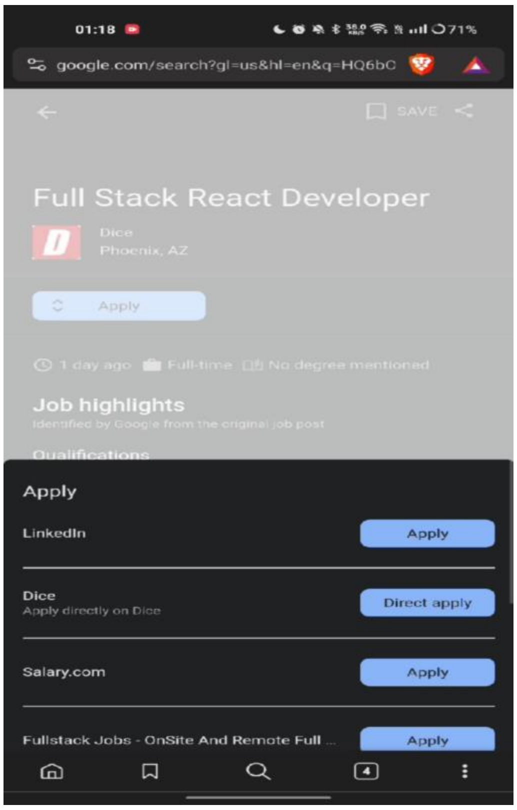
Immediate joiners required

React js mandatory

Contract role - 6-12 Months
• Bachelor's degree in a technical discipline or equivalent experience/training
• 4 plus years of experience in Java building web applications in a J2EE Environment
• 3 plus years of experience in React, and TypeScript Experience in application development, testing, implementation, and documentation of web-based applications using Java

Apply

Figure: *6.5.12 Job Description*


01:18

google.com/search?gl=us&hl=en&q=HQ6bC

SAVE

Full Stack React Developer

Dice
Phoenix, AZ

Apply

1 day ago · Full-time · No degree mentioned

Job highlights
Identified by Google from the original job post

Qualifications

Apply

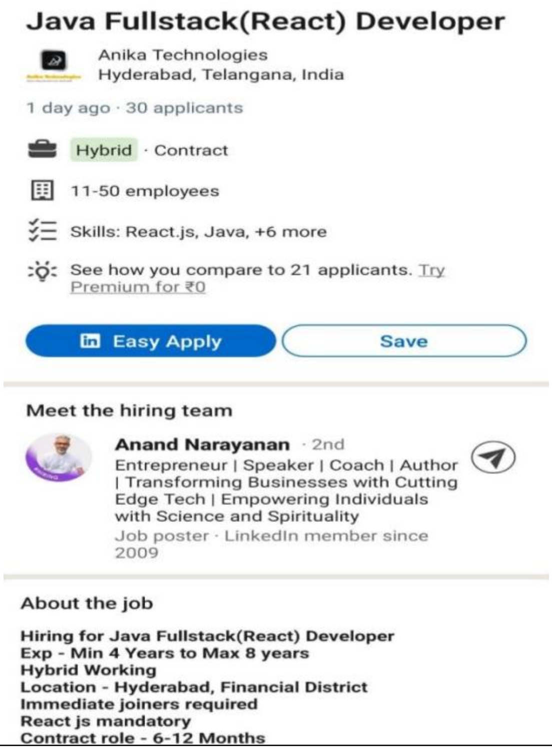
LinkedIn — Apply

Dice
Apply directly on Dice — Direct apply

Salary.com — Apply

Fullstack Jobs - OnSite And Remote Full ... — Apply

Figure: *6.5.13 Applying for a Job from Various Platforms*


Java Fullstack(React) Developer

Anika Technologies
Hyderabad, Telangana, India

1 day ago · 30 applicants

Hybrid · Contract

11-50 employees

Skills: React.js, Java, +6 more

See how you compare to 21 applicants. Try Premium for ₹0

Easy Apply | Save

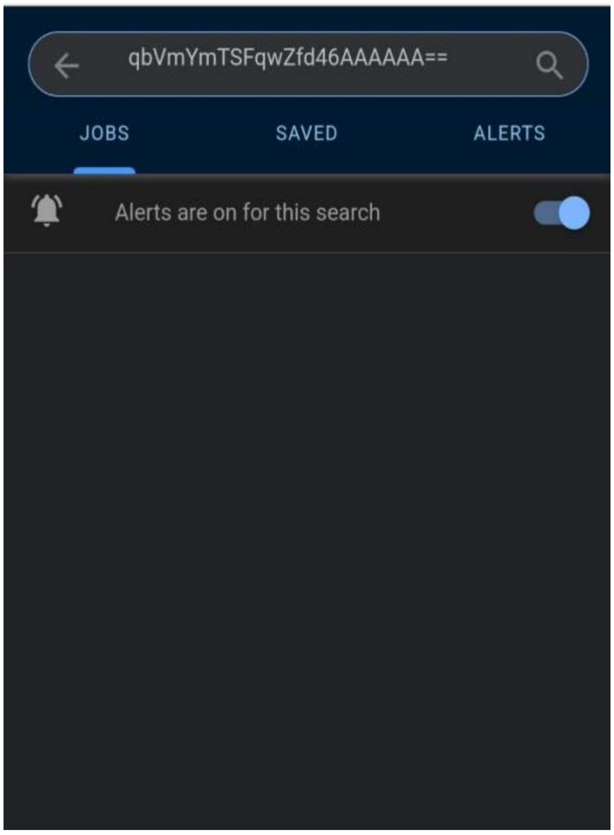
Meet the hiring team

Anand Narayanan · 2nd
Entrepreneur | Speaker | Coach | Author | Transforming Businesses with Cutting Edge Tech | Empowering Individuals with Science and Spirituality
Job poster · LinkedIn member since 2009

About the job

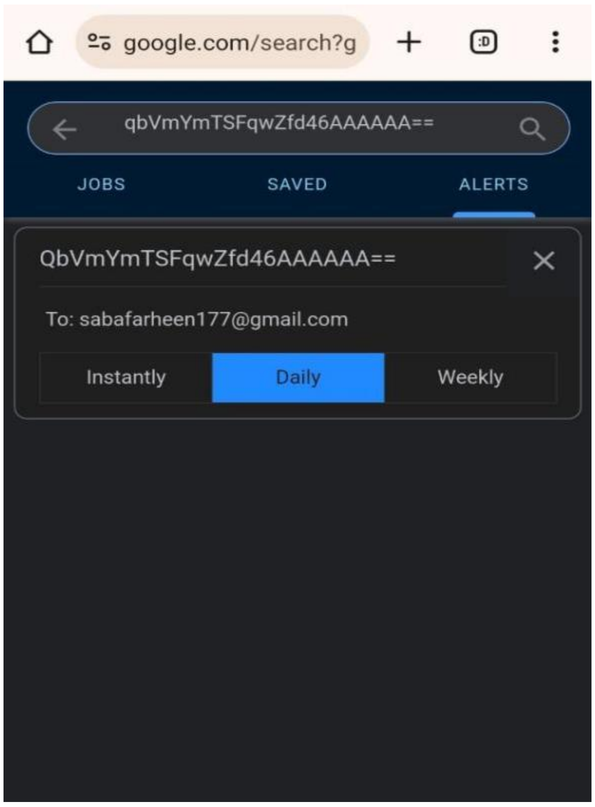
Hiring for Java Fullstack(React) Developer
Exp - Min 4 Years to Max 8 years
Hybrid Working
Location - Hyderabad, Financial District
Immediate joiners required
React js mandatory
Contract role - 6-12 Months

Figure: *6.5.14 Redirect to the chosen platform (where the Job has been Posted)*


qbVmYmTSFqwZfd46AAAAAA==

Alerts are on for this search

Figure: *6.5.15 Turn On Alert Notification (and save the job search)*



qbVmYmTSFqwZfd46AAAAAA==

QbVmYmTSFqwZfd46AAAAAA==

To: sabafarheen177@gmail.com

Instantly [✓ Daily] Weekly

Figure:*6.5.16 Get Notified Through E-mail*

CHAPTER VII

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Future Enhancement

Manual Feedback
with scorecard

Interactive community
forums

AI-powered
with scorecard

Social Networking
Features

1. AI-Powered Job Recommendations:

Enhanced User Profiling: Implement AI algorithms to analyse user behaviour, preferences, and past interactions with the platform to create more detailed user profiles.

Deep Learning Models: Utilize deep learning techniques to understand complex patterns in job listings and user profiles, enabling the system to provide highly accurate job recommendations tailored to each individual user.

Real-Time Updates: Incorporate real-time data analysis to ensure that job recommendations are constantly updated based on changing user preferences and market trends.

2. Dynamic Resume Building Tools:

AI-Driven Resume Optimization: Integrate AI-powered tools that analyse job descriptions and user profiles to offer personalized suggestions for optimizing resumes.

Natural Language Processing (NLP): Utilize NLP algorithms to parse and understand the content of resumes, enabling the system to provide targeted feedback and suggestions for improvement.

Interactive Resume Builders: Develop interactive resume-building interfaces that guide users through the process while leveraging AI to offer real-time feedback and suggestions.

3. <u>Virtual Reality (VR) for Job Previews and Interviews:</u>

Immersive Job Previews: Introduce VR technology to provide immersive job previews, allowing candidates to explore company environments, meet virtual colleagues, and get a feel for the workplace culture before applying.

VR Job Interviews: Facilitate remote job interviews through VR platforms, offering a more engaging and interactive experience for both candidates and employers while eliminating geographical barriers.

4. <u>Blockchain for Secure Credential Verification:</u>

Decentralized Credential Storage: Utilize blockchain technology to securely store and verify candidate credentials, including education certificates, work experience, and skills certifications.

Immutable Records: Leverage blockchain's immutable nature to create tamper-proof records of candidate credentials, enhancing trust and transparency in the hiring process.

Smart Contracts: Implement smart contracts to automate the verification process, reducing administrative overhead and ensuring that only verified candidates are considered for job opportunities.

5. <u>Continuous UI/UX Refinement:</u>

User Feedback Loop: Establish a feedback mechanism to gather user input and preferences, enabling continuous improvement of the application's user interface and user experience.

A/B Testing: Conduct A/B testing to experiment with different UI layouts, features, and functionalities, allowing for data-driven decisions on UI/UX enhancements.

Accessibility Considerations: Ensure that the application is accessible to users with diverse needs and preferences by incorporating accessibility features and conducting regular accessibility audits.

7.2 Conclusion

In conclusion, the job search portal app stands as a pivotal tool in reshaping the landscape of employment, providing a dynamic and efficient platform that significantly enhances the job-seeking and hiring processes. By centralizing diverse job opportunities, streamlining applications, and offering personalized recommendations, the app simplifies the often complex task of job hunting for individuals. Moreover, its robust features, such as resume parsing, skill assessments, and real-time notifications, empower users to present their profiles effectively and stay connected with the ever-evolving job market.

For employers, the app facilitates a streamlined recruitment process, allowing for efficient candidate screening and seamless communication. As the app continues to evolve, future enhancements, including artificial intelligence, predictive analytics, and immersive technologies, promise to further revolutionize the employment landscape, making the job search portal app an indispensable companion for both job seekers and employers in the pursuit of professional success.

REFERENCES

1. M Arvindhan, Abhineet Anand Scheming a proficient auto scaling technique for minimizing response time in load balancing on Amazon AWS Cloud International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India.
2. M Arvindhan, Bhanu Prakash Ande Data mining approach and security over dos attacks ICTACT journal on soft computing, 2020.
3. Abhineet Anand, Amit Chaudhary, M Arvindhan   ICTACT journal on soft computing, 2020 Advances in Communication and Computational Technology-2021.
4. N. Partheeban L. Goldin Atlas, K. P. Arjun, N. M. Sree Narayana n, M. Arvindhan Advances in Communication and Computational Technology Journal of Computational and Theoretical Nanoscience-2020/12/30.
5. M. Arvindhan Goldin Atlas, D. Gnana Jeba Das, K.P. Arjun, N.M. Sree Narayana n Detection of Botnet Activities Using Pattern Recognition Algorithms Jour of Adv Research in Dynamical & Control Systems.
6. Priya Madhya and M. Arvindhan A Framework for Analyzing Road Accidents Using Machine Learning Paradigms International Research Journal of Engineering and Technology.
7. M. Arvindhan Ahmad Ali International Research Journal of Engineering and Technology Journal of Indian University.
8. N.V. Kousik M. Arvindhan, G.S. Pradeep Ghantasala Clustering Algorithm Network Test Cost Sensitive for Specialist Division ICTACT Journal on Image and Video Processing, November 2019.
9. T Vinodh Kannan, M Arvindhan An Analysis of Cloud Computing Security Challenges a Conceptual Framework for Cloud Computing Early Adopters as a Technology Model ICTACT journal on soft computing, 2020.
10. M ARVINDHAN, B BHARATHI KANNAN, GS PRADEEP GHANTASALA Search for Harmonized Keywords Using the Voted Lab Feature and Allow Re Cryptosystem for Electronic Health Clouds International Journal of Innovative Technology and Research.
11. www.google.com
12. GS, D., Ghantasala, P., Chinnam, V., Karuk Uri, R., & Sripad, S. (2020). Of Web Crawling from Http Parse and Html Requests as Static Digraph and Web Pages.

Venkata Sai and Karuk Uri, Raju and Sripad, Sharath, Of Web Crawling from Http Parse and Html Requests as Static Digraph and Web Pages (August 8, 2020).

13. www.asp.netpractise.com