JOB SEARCH PLATFORM

A Project Report for

CSE 6324 ADVANCED TOPICS IN SOFTWARE ENGINEERING

Fall 2023

(Maters in Computer Science)

By

GROUP – 1

MUNI SAI KALYAN TEJA DUDI - 1002104402
RICHITHAREDDY GORLAGUNTA - 1002126204

Under The Esteemed Guidance Of
Mrs. FARNAZ FARAHANIPAD
Professor, Department of Computer Science

UNIVERSITY OF TEXAS AT ARLINGTON

TABLE OF CONTENTS

ABSTARCT

SUMMARY

In today's dynamic and ever-evolving job market, the need for an efficient and user-friendly job search platform has never been more crucial. With the global workforce becoming increasingly mobile and diverse, job seekers, employers, and recruiters alike require a centralized and intuitive solution to connect with each other.

This proposal outlines the development of a cutting-edge "Job Search Platform," designed to revolutionize the job search and hiring process. Our platform aims to provide a seamless experience for job seekers, offering a comprehensive database of job listings, personalized recommendations, and powerful search tools. Simultaneously, it empowers employers and recruiters with the means to efficiently post job openings, review applicants, and make informed hiring decisions.

The "Job Search Platform" project is an ambitious endeavor aimed at creating a comprehensive and transformative solution for job seekers, employers, and recruiters in the contemporary job market. This platform will serve as a bridge connecting job seekers to their dream careers and employers to the ideal candidates they seek.

OBJECTIVES

The primary objective of this project is to address these pain points by developing a Job Search Platform that offers the following key benefits

- **Efficiency**: Simplify the job search process for job seekers by centralizing job listings from various sources.
- **Personalization**: Provide personalized job recommendations based on user profiles and preferences.
- **Ease of Use**: Create an intuitive and user-friendly interface for both job seekers and employers.
- **Advanced Tools**: Offer advanced tools for employers to streamline the hiring process, including applicant tracking and communication features.

- **Market Competitiveness**: Position the platform as a strong contender in the job search industry, offering unique features and a superior user experience.

KEY OUTCOMES

The project will encompass a wide range of features to meet the diverse needs of our users. Key features of the Job Search Platform include:

For Job Seekers:

- **Job Search**: A powerful search engine that enables users to find relevant job listings based on keywords, location, industry, and other criteria.
- **Personalized Recommendations**: Utilizing advanced algorithms to provide users with tailored job recommendations based on their profiles and preferences.
- **User Profiles**: Users can create and manage detailed profiles, including resumes, cover letters, and professional portfolios.
- **Application Tracking**: A centralized dashboard for tracking job applications, interview invitations, and offers.
- **Communication Tools**: Messaging and notification systems to facilitate communication between job seekers and employers.

For Employers and Recruiters:

- **Job Posting**: An intuitive job posting tool that allows employers and recruiters to create, edit, and manage job listings with ease.
- **Applicant Tracking**: Advanced applicant tracking features, including resume parsing and candidate evaluation tools.
- **Communication**: In-platform messaging and scheduling capabilities for seamless communication with job seekers.
- **Analytics**: Data analytics tools to track the performance of job listings and the effectiveness of the hiring process.

SIGNIFICANCE

The Job Search Platform addresses the pressing need for a modern and efficient solution in the job search and hiring sector. In a world where job markets are increasingly global and competitive, this platform will provide the following benefits:

- **Empowerment**: Job seekers gain access to a vast database of job opportunities, making it easier to discover and pursue their desired careers.
- **Efficiency**: Employers and recruiters can optimize their hiring processes, reducing the time and resources spent on candidate sourcing and assessment.
- **Streamlined Communication**: Seamless communication tools enhance the interaction between job seekers and employers, leading to more successful hires.
- **Market Innovation**: The Job Search Platform aims to stand out in the market by introducing innovative features and providing a superior user experience.

This project represents a significant opportunity to transform the job search and hiring landscape, simplifying the process for all parties involved and contributing to the success of job seekers and employers alike.

CONCLUSION

With a clear understanding of the project's objectives and the critical issues it aims to address, we are prepared to embark on the development journey of this innovative Job Search Platform. Through careful planning, cutting-edge technology, and user-centric design, we are confident in our ability to create a platform that will have a significant impact on the job market.

INTRODUCTION

OVERVIEW

The Job Search Platform will be a web-based application accessible from various devices, including desktop computers, laptops, tablets, and smartphones. It will feature a clean and intuitive user interface designed to streamline the job search and hiring processes.

OBJECTIVES

The "Job Search Platform" project is driven by a set of well-defined objectives aimed at achieving both functional and user-oriented goals. These objectives guide the development process and serve as the foundation for our project's success.

1. **Market Differentiation**: To position the Job Search Platform as a standout solution in the competitive job search and hiring sector by offering unique features and a superior user experience.
2. **Scalability**: To build a platform that can scale efficiently to accommodate a growing user base and adapt to evolving market trends and user needs.
3. **Market Expansion**: To explore opportunities for regional or global expansion, enabling the platform to serve a diverse range of job markets and industries.
4. **Revenue Generation**: To explore monetization strategies, such as premium job listings, subscription plans, or targeted advertising, to sustain the platform's growth and development.

By pursuing these objectives, we aim to create a Job Search Platform that not only addresses the immediate needs of job seekers, employers, and recruiters but also positions itself as a leader in the industry, driving positive change in the job search and hiring process.

PROBLEM STATEMENT

The traditional job search process can be time-consuming and frustrating for job seekers, often requiring them to navigate multiple job boards, websites, and application processes. Employers and recruiters, on the other hand, face challenges in efficiently sourcing and assessing suitable candidates from diverse talent pools.

PROJECT SCOPE

SCOPE OF WORK

The scope of work for the development of the "Job Search Platform" project is defined to ensure clarity in project boundaries, deliverables, and limitations. This section outlines what will and will not be included in the project.

FEATURES FUNCTIONALITIES AND CONSTRAINTS

Inclusions

The following components and activities are within the scope of work for the project:

1. **Platform Development**: This includes the design, development, testing, and deployment of the web-based Job Search Platform.

2. **User Management**: Creation of user registration and authentication systems for job seekers, employers, and recruiters.

3. **Job Listings**: Implementation of a job listing database with advanced search capabilities.

4. **User Profiles**: Development of user profiles for job seekers and the ability to manage resumes, cover letters, and portfolios.

5. **Personalization**: Incorporation of algorithms for personalized job recommendations based on user profiles and preferences.

6. **Application Tracking**: Creation of dashboards for job seekers to track job applications, interview invitations, and offers.

7. **Communication Features**: Integration of in-platform messaging and notification systems for user communication.

8. **Job Posting**: Development of an easy-to-use job posting tool for employers and recruiters.

9. **Applicant Tracking and Evaluation**: Inclusion of applicant tracking features with resume parsing and candidate evaluation tools for employers and recruiters.

10. **User Experience Design**: Emphasis on user-centric design to ensure an engaging and accessible interface.

Exclusions

The following components and activities are not within the scope of work for this project:

1. **Content Creation**: The project will not include the creation of job listings or user-generated content; it will rely on external data sources for job listings.

2. **Third-Party Integration**: While the project may utilize third-party services (e.g., payment processing or analytics), the development of these third-party services is not included.

3. **Legal and Regulatory Compliance**: Ensuring legal and regulatory compliance, such as data privacy and labor laws, is the responsibility of the platform users (employers, recruiters, and job seekers).

4. **Market Research**: In-depth market research and analysis, including competitor analysis, is not included in this project and will be conducted separately.

5. **Marketing and Promotion**: Marketing strategies, campaigns, and promotional activities are outside the project scope and will be addressed separately.

Constraints

The project is subject to the following constraints:

1. **Budget Constraints**: The project budget is fixed and should be managed within the allocated resources.

2. **Time Constraints**: The project timeline is defined with specific milestones and deadlines that should be adhered to.

3. **Technology Stack**: The project will utilize the chosen technology stack (as outlined in Section 5) with limited flexibility for changing technologies.

4. **Geographical Scope**: The initial release of the platform will focus on a specific geographical region, with potential expansion in subsequent phases.

5. **Data Security and Privacy**: The project must adhere to data security and privacy regulations, which may affect the collection and handling of user data.

By defining the scope of work, we aim to ensure a clear understanding of project boundaries and expectations among all stakeholders involved in the development of the Job Search Platform.

TARGET AUDIENCE AND LIMITATIONS

The target audience for our Job Search Platform includes:

- **Job Seekers**: Individuals actively seeking job opportunities or exploring career options.
- **Employers**: Companies of all sizes looking to hire qualified talent.
- **Recruiters**: Professional recruiters and staffing agencies helping companies find the right candidates.

With a clear understanding of the project's objectives and the critical issues it aims to address, we are prepared to embark on the development journey of this innovative Job Search Platform. Through careful planning, cutting-edge technology, and user-centric design, we are confident in our ability to create a platform that will have a significant impact on the job market.

REQUIREMENT ANALYSIS

REQUIREMENT GATHERING

SURVEYS

Surveys are a useful technique for getting user feedback on job search platforms. Inquiries may concern user happiness, usability, search filter efficacy, and general user experience.

Surveys are a useful tool for platforms to find areas for development and learn about the requirements and preferences of its users.

INTERVIEWS

In the context of job search sites, "interviews" may refer to user interviews or even practice interviews for prospective candidates. Platforms can learn more about consumers' preferences, problems, and most helpful features by conducting user interviews. The platform may include mock interviews as a feature to aid job searchers in honing their interviewing techniques.

DOCUMENTATION

On a job search platform, documentation is essential for employers and job seekers alike. It is crucial to provide job seekers with clear instructions on how to make a profile, look for opportunities, and apply for openings. Documentation is required by employers in order to manage the hiring process, evaluate applications, and promote job openings.

FUNCTIONAL REQUIREMENTS

1. **Efficient Job Search**: To create a platform that streamlines the job search process, allowing job seekers to find relevant job listings quickly and easily.

2. **Personalization**: To implement advanced algorithms for personalized job recommendations, enhancing the user experience by presenting relevant opportunities based on user profiles and preferences.

3. **User-Friendly Interface**: To design an intuitive and user-friendly interface that ensures a seamless and engaging experience for both job seekers and employers.

4. **Comprehensive User Profiles**: To provide job seekers with the tools to create and manage detailed profiles, including resumes, cover letters, and portfolios.

5. **Application Tracking**: To develop a centralized dashboard for job seekers, enabling them to track job applications, interview invitations, and offers conveniently.

6. **Effective Communication**: To incorporate in-platform messaging and notification systems that facilitate communication between job seekers and employers, ensuring a smooth interaction.

7. **Efficient Job Posting**: To offer employers and recruiters an easy-to-use job posting tool that simplifies the process of creating, editing, and managing job listings.

8. **Applicant Tracking and Evaluation**: To provide advanced applicant tracking features for employers and recruiters, including resume parsing and candidate evaluation tools.

NON FUNCTIONAL REQUIREMENTS

Performance:

- Fast response times for search queries and page loads.
- Scalability to handle a large number of users and job listings.

Security:

- Secure user authentication and authorization mechanisms.
- Encryption of sensitive user data.

Reliability:

- High availability to ensure the platform is accessible when needed.
- Regular backups and data recovery procedures.

Usability:

- Intuitive user interface and user experience design.
- Accessibility features to accommodate users with disabilities.

Scalability:

- Ability to handle increasing numbers of users and job listings.

- Efficient resource utilization.

USER STORIES

1. **Enhanced User Experience**: To prioritize user experience design to ensure that the platform is visually appealing, easy to navigate, and accessible across various devices.
2. **User Empowerment**: To empower job seekers with the tools and information they need to make informed career decisions and to assist employers in finding the right talent efficiently.
3. **Transparent and Trustworthy**: To establish a platform that fosters trust and transparency between job seekers and employers through clear communication and ethical practices.

PROJECT PLANNING

WORK BREAKDOWN STRUCTURE (WBS):

A Work Breakdown Structure (WBS) is a hierarchical decomposition of the project into smaller, manageable work packages. Here's a simplified WBS for the Job Search Platform project:

1. Project Initiation

1.1. Project Proposal

1.2. Feasibility Study

1.3. Cost and Time Estimation

2. Design and Development

2.1. Landing Page Development

2.2. Registration and Login Pages

2.3. Employer Dashboard

2.3.1. Job Listing Creation

2.3.2. Job Listing Editing

2.3.3. Job Listing Deletion

2.4. Employee Dashboard

2.4.1. Job Search Functionality

2.5. User Profile Pages (Future Phase)

2.6. Notifications System (Future Phase)

3. Testing and Quality Assurance

3.1. Unit Testing

3.2. User Acceptance Testing

3.3. Bug Fixing and Issue Resolution

4. Deployment

4.1. Prepare for Production Deployment

4.2. Choose Hosting Platform

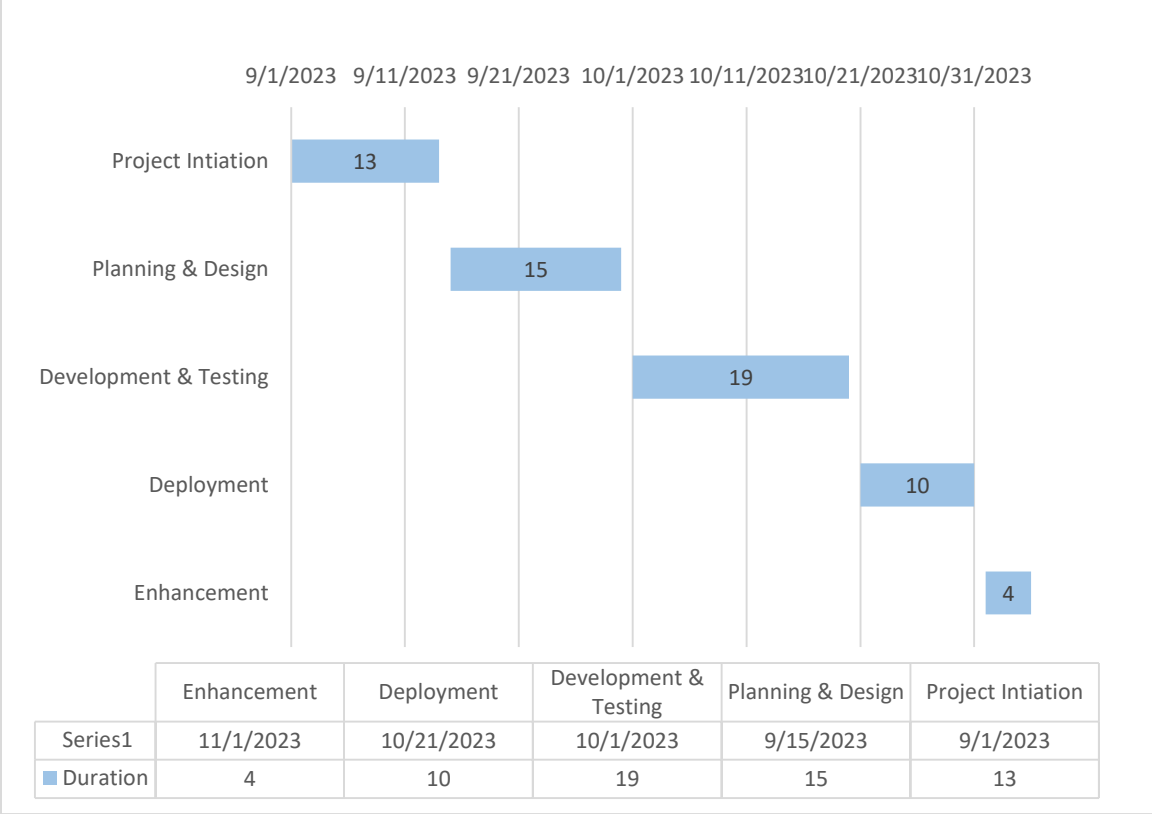
4.3. Deploy Web Application

5. Documentation and User Guides

5.1. Create User Documentation

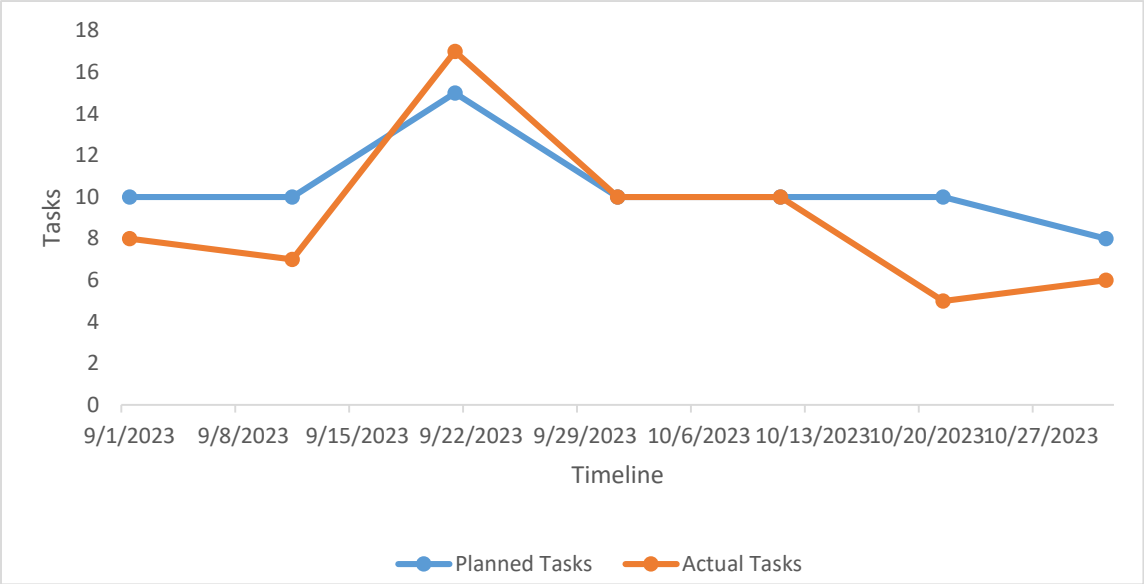
5.2. Develop User Guides

GANNT CHART



From the above timeline you can see that we have started our Project Intiation and completed in 13 days for the next phase i.e, Planning & Design was completed in the next 15 days, the next phase Development and testing were we developed and tested the application is done in 19 days after the second phase, Deployment phase is done in 10 days and the last 4 days were used for Enhancements.

BURNDOWN CHART



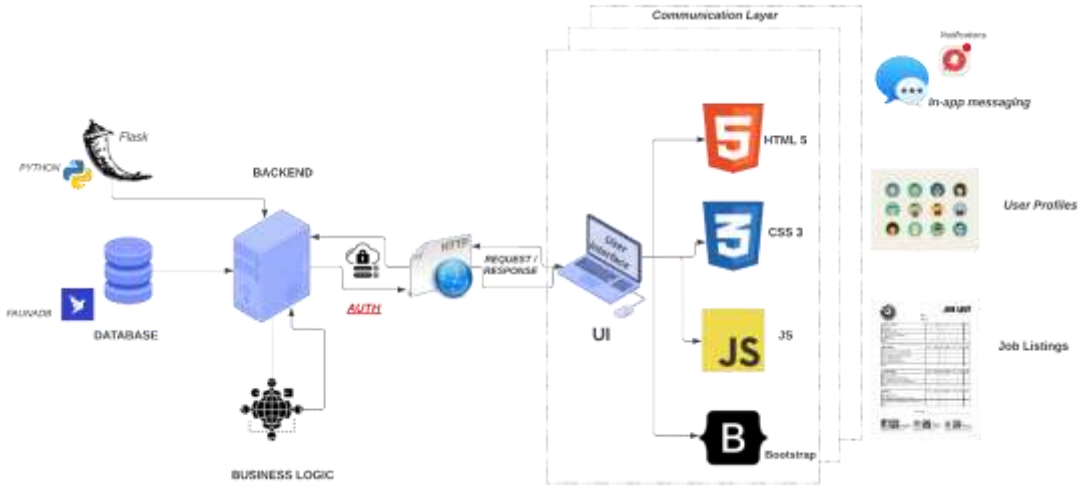
The above Burndown chart is the representation of the tasks we did in the entire project, here we used Waterfall Model. We started with Intiation Phase and went on a flow and in the development phase we spent more time to compute it and managed to complete on time. Lastly, we completed all the tasks in the time.

ROLES & RESPONSIBILITIES

MUNI SAI KALYAN TEJA DUDI	RICHITHAREDDY GORLAGUNTA
• Gathering Requirements • Backend Development (Authentication, Functionality) • Database setup to server • Testing • Documentation	• Developing Project Scope & Plan • Frontend Development (Home page, listing page, Employee and Employer pages) • UI Design • Database Connectivity • Documentation

DESIGN & ARCHITECTURE

SOFTWARE ARCHITECTURE



Architecture Components:

1. Frontend Layer:

- HTML, CSS, JS, Bootstrap: These technologies create the user interface (UI) of the platform, allowing users to interact with the application via web browsers on various devices.

2. Backend Layer:

- Flask (Python): Flask serves as the backend web framework responsible for handling HTTP requests, routing, and business logic.

3. Database Layer:

- FaunaDB: FaunaDB is used as the NoSQL database to store user profiles, job listings, communication data, and more.

4. Authentication and Authorisation:

- User authentication and authorisation are handled securely by the backend using Flask's authentication mechanisms and tokens to ensure data privacy and security.

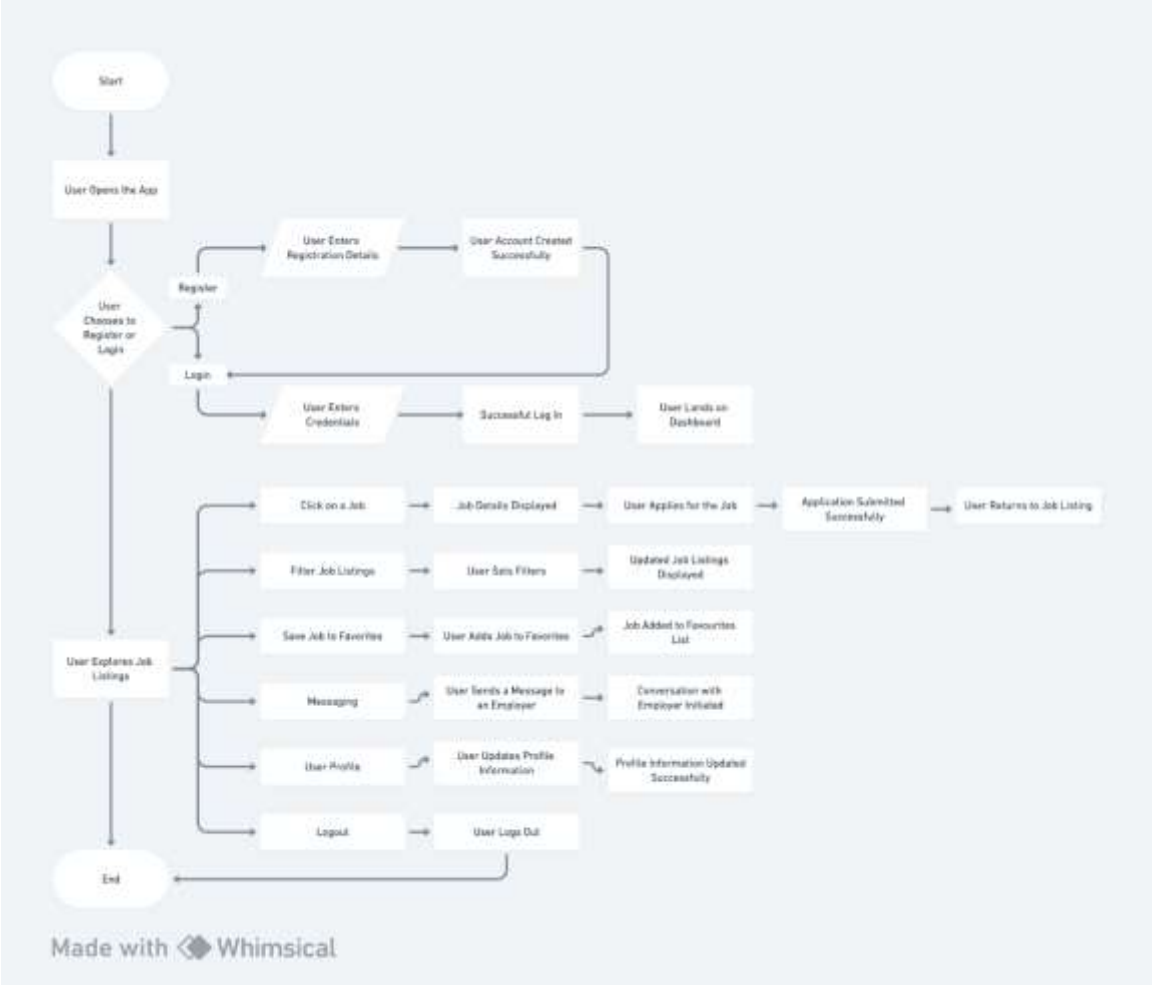
5. Business Logic:

- The backend layer contains the core business logic responsible for features like job recommendations, user profiles, job posting, and messaging.

6. Communication Layer:

- In-app messaging and notifications are managed by the backend, facilitating communication between job seekers and employers/recruiters.

FLOW CHART



Architecture Flow:

1. User Interaction:

- Users interact with the frontend layer, where they can register, log in, browse job listings, apply for jobs, and communicate with employers/recruiters.

2. Frontend-Backend Communication:

- User interactions trigger HTTP requests to the Flask backend, which processes the requests and communicates with FaunaDB for data retrieval and storage.

3. Database Management:

- FaunaDB stores and manages user profiles, job listings, messages, and other data securely in a distributed and scalable manner.

4. Authentication and Authorisation:

- User authentication and authorisation are performed by Flask, ensuring that only authorised users access specific functionalities.

5. Business Logic Execution:

- The backend layer executes business logic for features like job recommendations and applicant tracking, utilising data stored in FaunaDB.

6. Messaging and Notifications:

- The communication layer handles in-app messaging and notifications, facilitating real-time communication between users.

7. Feedback and Iteration:

- User interactions and feedback collected through the frontend UI are used to iterate and enhance the platform's features and user experience.

DATABASE SCHEMA AND RELATIONSHIPS

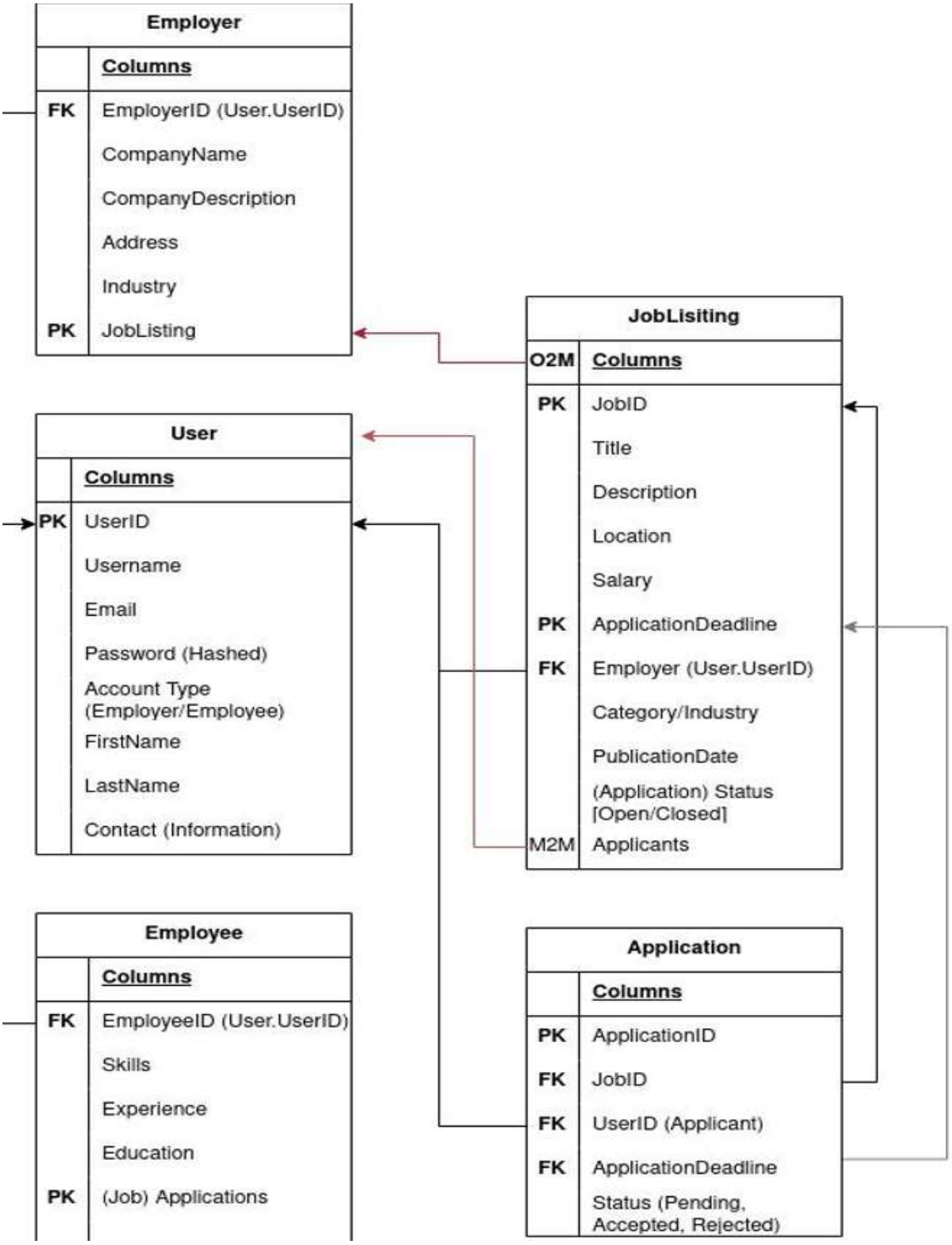
- FaunaDB's server-less and globally distributed architecture enables the platform to scale efficiently as the user base grows, ensuring optimal performance across different regions.

- The Flask backend can be horizontally scaled to accommodate increased traffic and usage.

This architecture provides a high-level overview of how the various components of the "Job Search Platform" interact and work together to deliver a seamless and secure experience for job seekers, employers, and recruiters. Keep in mind that in a real-world

application, there would be more intricate details, security measures, and redundancy considerations to ensure high availability and data integrity.

ER DIAGRAM



IMPLEMENTATION

The technology stack chosen for the development of the "Job Search Platform" has been thoughtfully selected to ensure a robust, scalable, and user-friendly application. This stack encompasses both frontend and backend technologies, as well as the database management system.

Frontend Technologies

- **HTML (HyperText Markup Language)**: HTML will serve as the backbone for structuring the content and layout of web pages.
- **CSS (Cascading Style Sheets)**: CSS will be used to define the visual presentation of web pages, ensuring a consistent and appealing user interface.
- **JavaScript (JS)**: JavaScript will enable dynamic and interactive features on the platform, enhancing user engagement and functionality.
- **Bootstrap**: Bootstrap will be employed for responsive web design, streamlining the development of a mobile-friendly and visually cohesive user interface.

Backend Technologies

- **Flask**: Flask, a micro web framework for Python, will form the core of the backend. Its simplicity and flexibility make it an ideal choice for building web applications.
- **Python**: Python will serve as the primary programming language for server-side development, offering versatility and a vast ecosystem of libraries and frameworks.

Database Management

- **FaunaDB**: FaunaDB is chosen as the database management system for the "Job Search Platform." It offers a serverless, globally distributed, and scalable NoSQL database solution. FaunaDB's capabilities align with our project's requirement for data storage, retrieval, and scalability.

MODULES

We have designed two modules they are:

1. Employer

2. Employee

1. Employer

Register and Login.

Add a job with complete information.

2. Employee

Register and Login.

Search for Job and apply.

CHALLENGES

The choice of this technology stack is driven by the following considerations:

- **Scalability**: FaunaDB's serverless architecture allows us to scale our database needs efficiently as the platform grows, ensuring optimal performance.
- **User Experience**: The frontend technologies, including HTML, CSS, JavaScript, and Bootstrap, will contribute to an engaging and responsive user interface.
- **Development Efficiency**: Flask, known for its simplicity and Python's readability, will facilitate rapid development and easy maintenance of the backend.
- **Community Support**: Python, Flask, and Bootstrap boast active and extensive developer communities, offering resources and libraries to aid in development.
- **Security**: The selected technologies have a strong track record of security, which is paramount in handling user data and ensuring the platform's trustworthiness.

By leveraging this technology stack, we aim to create a robust, scalable, and user-centric Job Search Platform that meets the project's objectives and user expectations.

THIRD PARTY LIBRARIES & FRAMEWORKS

Browser: To display the application we used browser.

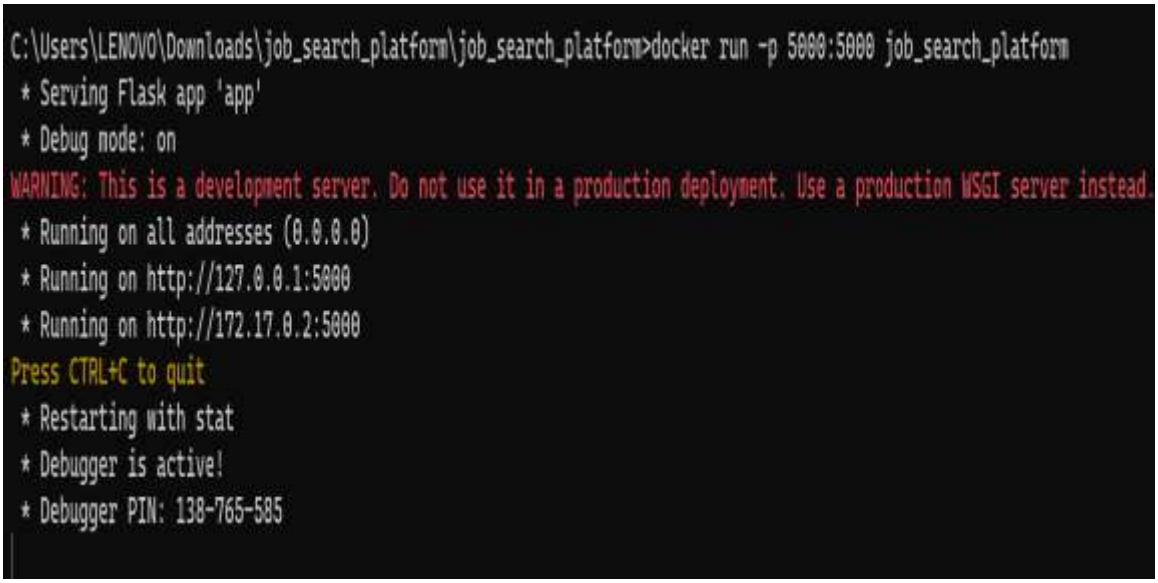
Flask: To develop web application using Python we used Flask in our project.

JavaScript: We used JavaScript for making web pages interactive.

TESTING

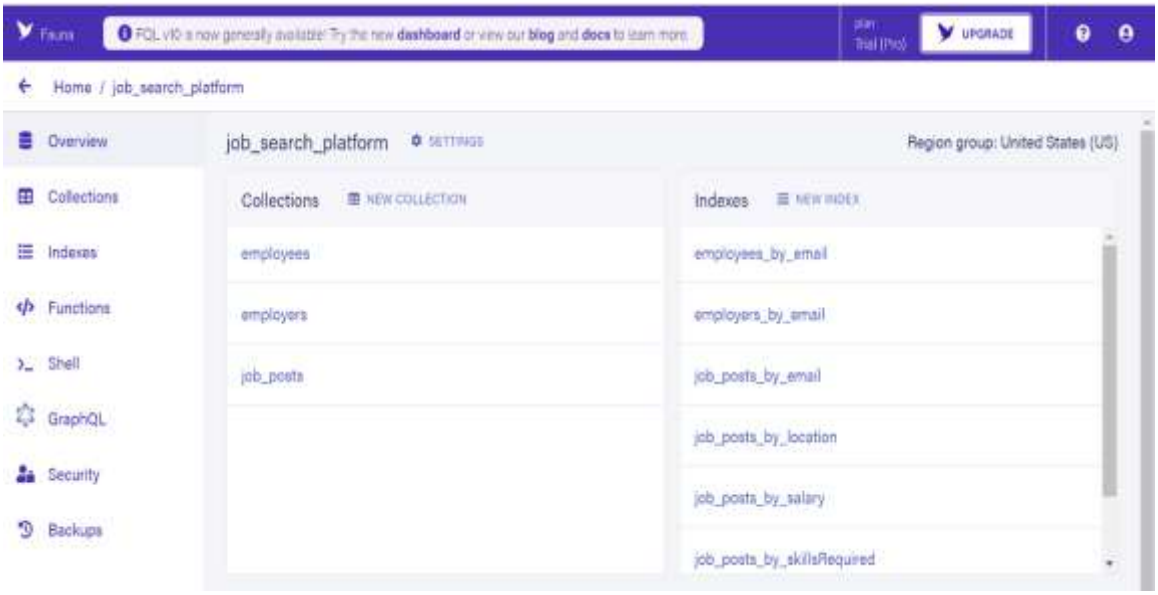
UNIT TESTING

PyTest, a testing framework for Python, will be used for unit testing to ensure the correctness of our code.



INTEGRATION TESTING

Testing the interaction between the frontend and backend, as well as external services like FaunaDB.



USER ACCEPTANCE TESTING

It will be employed for automated UI testing to verify the functionality and user experience of the web application.

Tester: Varshinee Duddi Mahendra

Team: Team 17

1. User Authentication and Authorization: Secure user registration and authentication mechanisms will ensure data privacy and access control.

Result: We are successfully provided security for Register and Login for Employee and Employer.

2. Responsive Design: The platform will feature responsive design, making it accessible and user-friendly on a variety of devices, including desktops, tablets, and smartphones.

Result: We tested our application in different devices comparing sizes of screens.

3. Search and Filter Options: Advanced search and filtering options will allow users to refine job search results based on their specific preferences.

Result: The search option in our application provides us to search a job on the basis of salary, location, skills and many other functionalities.

4. Analytics and Reporting: Data analytics tools will provide users with insights into the performance of their job listings and the effectiveness of the hiring process.

Result: In this test we will conclude whether the job is filled or not.

5. User Support: A support system will be in place to assist users with any inquiries, issues, or feedback they may have.

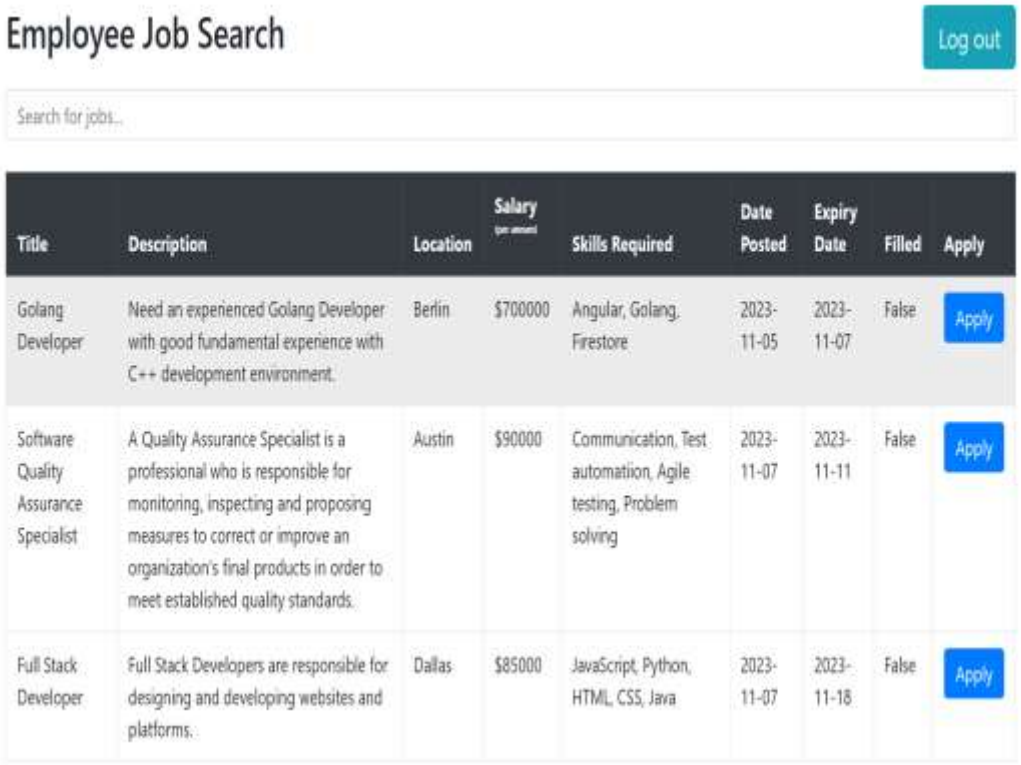
Result: If anything goes wrong while using the application, a pop up will be prompted saying that, something went wrong. Please try again.

UI SCREENSHOTS

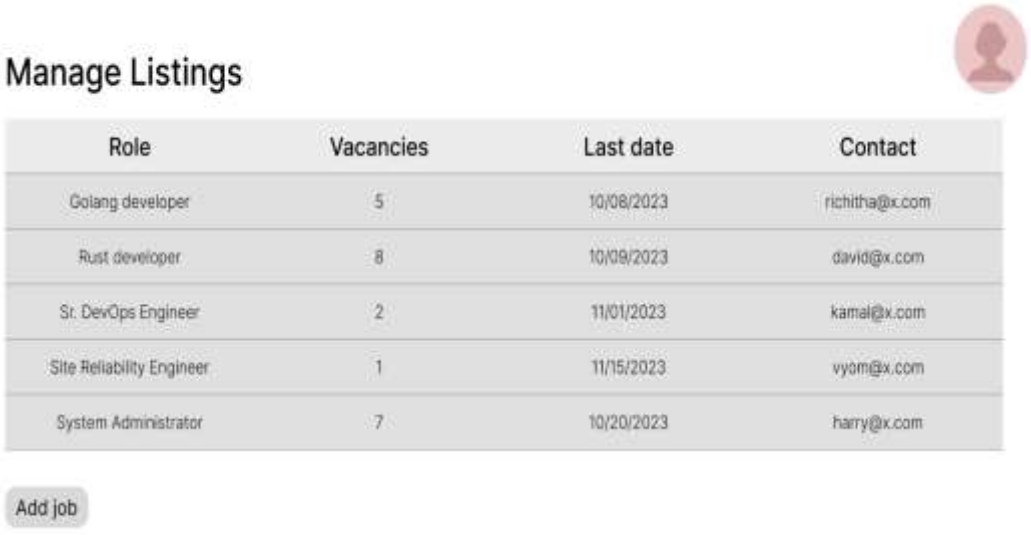
Landing Page

Registration & Login Pages

Employee Dashboard



Employer Dashboard



TESTCASES

SPECIFICATION	EXAMPLE	SUCCESS/ FAILURE
Name (must be a CHAR)	1.JOHN 2.JOHN12	1. SUCCESS 2. FAILURE
Mail ID (must be a VARCHAR)	1. john12@gmail.com 2. john2@uff.allours,com	1. SUCCESS 2. FAILURE
Password (must be a VARCHAR)	1. PASSWORD123 2. PASSWORD	1. SUCCESS 2. FAILURE

ISSUES & BUGS

When employee is applying a job they're allowed to send their resume multiple times. So that, employer is getting the application multiple times.

RESOLUTION

When the employee is applying the job multiple times .It will be rejected from the admin side So that, they can't apply the same job again.(This will avoids duplication)

MAINTENANCE

MAINTENANCE PLAN

UPDATE AND PATCH MANAGEMENT

Regular Platform Updates: Schedule regular updates to the job search platform to introduce new features, improve existing functionalities, and enhance the overall user experience.

Security Patches: Implement a robust process for promptly applying security patches to protect user data and maintain the integrity of the platform.

Versioning for Platform Releases: Establish a versioning strategy to clearly communicate updates to both job seekers and employers, ensuring a smooth transition with backward compatibility.

Automated Platform Updates: Consider implementing an automated update mechanism to efficiently distribute new features and improvements, minimizing disruption to users.

Rollback Procedures: Develop rollback procedures to quickly address any issues that may arise from updates, ensuring the platform's continuous availability.

BUG FIX MANAGEMENT

Issue Tracking System Integration: Integrate a robust issue tracking system into the platform's development workflow to efficiently manage and prioritize bug reports.

Regular Bug Fix Releases: Schedule regular bug fix releases, focusing on resolving critical issues that impact the functionality of the job search platform.

User Feedback Loop: Establish a user feedback loop to capture and prioritize bug reports based on user experiences, ensuring continuous improvement.

Testing Protocols: Implement rigorous testing protocols for bug fixes to maintain the stability and reliability of the platform.

SCALABILITY CONSIDERATION

Performance Monitoring: Integrate performance monitoring tools to continuously assess and optimize the platform's performance, particularly during peak usage times.

Load Testing for Scalability: Conduct regular load testing to evaluate the platform's scalability and identify potential bottlenecks, ensuring it can handle increased user activity.

Scalable Infrastructure: Design the platform's infrastructure to be scalable, leveraging cloud services and containerization for flexible resource allocation.

Resource Scaling Procedures: Develop procedures for scaling resources, such as servers and database capacity, to accommodate the growing user base.

FUTURE PROOFING

Technology Stack Evaluation: Regularly evaluate the technology stack to ensure it remains up-to-date, secure, and aligned with industry standards, incorporating new technologies when beneficial.

Modular Platform Architecture: Design the platform with a modular architecture to facilitate the seamless integration of new features and the replacement or upgrade of individual components.

Adaptability to Industry Trends: Stay informed about emerging trends in the job search and recruitment industry, and ensure the platform can adapt to evolving user needs and expectations.

This tailored maintenance plan aims to keep the job search platform robust, secure, and responsive to both user feedback and industry changes.

CONCLUSION

In conclusion, the development and launch of the "Job Search Platform" represent a significant opportunity to revolutionize the job search and hiring process. This proposal has outlined the vision, objectives, and strategies for creating a platform that empowers job seekers, simplifies the hiring process for employers and recruiters, and addresses the challenges of the modern job market.

Our commitment to user-centric design, robust technology, and a systematic approach to development ensures that this platform will not only meet but exceed the expectations of its users. By incorporating advanced features, personalized recommendations, and user-friendly interfaces, we aim to create a comprehensive solution that streamlines job searches, enhances communication, and fosters transparency in the hiring process.

We recognize that the success of this project depends on collaboration, transparency, and a relentless commitment to improvement. User testing and feedback will be instrumental in shaping the platform's evolution and ensuring its continued relevance in the dynamic job market landscape.

As we move forward, we remain dedicated to delivering a high-quality "Job Search Platform" that not only addresses the needs of today's job seekers and employers but also anticipates and adapts to the changing demands of tomorrow. This platform is not just a project; it is a mission to empower individuals in their pursuit of meaningful careers and assist organizations in finding the right talent.

RECOMMENDATIONS

FUTURE WORK

This platform have the potential to shape the future of software engineering careers, making the hiring process more efficient, transparent, and empowering for both job seekers and employers alike. In future it can also be developed in Android Application to make things ease.

CITATIONS & REFERENCES

1. Design and implementation of job-search system based on javaEE | IEEE Conference Publication | IEEE Xplore

2. https://ieeexplore.ieee.org/document/6959916

APPENDICES

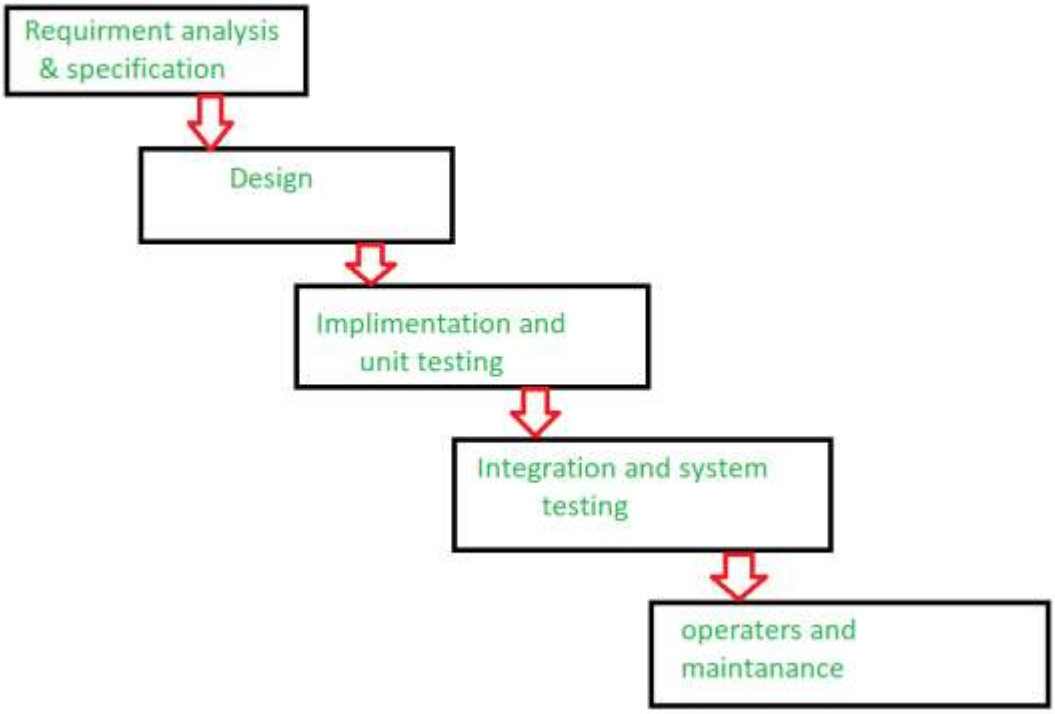
Code Snippet

```
import os
import bcrypt
import datetime
from dotenv import load_dotenv
from faunadb import query as q
from faunadb.client import FaunaClient
from faunadb import errors as fauna_errors
from flask import Flask, render_template, request, redirect, url_for, flash, session

# Load the env-vars from `.env` file
load_dotenv()

# Initialize the client
client = FaunaClient(secret=os.getenv("FAUNADB_SECRET"))

# Initialize the flask-app
app = Flask(__name__)
app.secret_key = os.getenv("FLASH_SECRET").encode()

app.config['SESSION_COOKIE_SECURE'] = False
app.config['SESSION_COOKIE_HTTPONLY'] = True
app.config['SESSION_COOKIE_SAMESITE'] = 'Lax'
app.config['PERMANENT_SESSION_LIFETIME'] = datetime.timedelta(hours=1)

hash_password = lambda p: bcrypt.hashpw(p.encode('utf-8'), bcrypt.gensalt()).decode('utf-8')
check_password = lambda hp, p: bcrypt.checkpw(p.encode('utf-8'), hp.encode('utf-8'))

@app.route('/')
def index():
    return render_template('index.html')
```

Code Snippet

```python
@app.route('/<user_type>', methods=['GET'])
def register_login(user_type):
    return render_template('register_login.html', user_type=user_type)

@app.route('/<user_type>/register', methods=['POST'])
def register(user_type):
    # Extract data from <form>
    name = request.form.get('name')
    email = request.form.get('email')
    password = request.form.get('password')
    confirm_password = request.form.get('confirm_password')

    # Validate confirmation
    if password != confirm_password:
        flash("Passwords doesn't match! Please try again.", 'error')
        return redirect(f'/{user_type}')

    # Hash password
    hashed_password = hash_password(password)

    # Save to database
    try:
        result = client.query(
            q.create(
                q.collection(f'{user_type}s'),
                {"data": dict(name=name, email=email, password=hashed_password)}
            )
        )
    except fauna_errors.BadRequest as ex:
        flash(f'Account already exists! Please login with <{email}>', 'warning')
```

Waterfall Model:



We have used waterfall model for our application. This is the process that we've followed.