

**Final Report**

Name: Aayush Tamang

Student Number: 2330458

Course: Bachelors (Hons) in Computer Science

Supervisor: Mr. Bipul Bahadur Pradhan

Title and Declaration

**Abstract**

DevX is an innovative web-based solution to connect employers with developers. It utilizes NLP techniques specifically BERT model for performing skill based match making of developers to jobs. It uses semantic understanding of the resumes to create recommendations based on job requirements. It includes three key features: managing your profile as a developer, posting and managing job ads and a way for anyone to give feedback to improve trust on the platform.

DevX is built using Agile SCRUM, so it is delivered in frequent phases and always improves based on teamwork. The agile SCRUM framework directs the software development lifecycle through an adaptive framework which produces continuous improvement of collaboration and delivers iterative results for effective management. The application uses React together with Tailwind CSS for frontend development along with Node.js and Express.js as backend infrastructure and MongoDB for database management and Python for AI model building through lightweight API deployment.

The DevX platform combines AI-driven automation systems with current software development approaches which work together to modernize recruitment methods in the tech sector. DevX stands ready to transform talent recruitment in the digital era through its solution of matching between employers and developers.

**Table of Contents**

[1. Introduction 1](#_Toc194330276)

[1.1. Project Briefing 1](#_Toc194330277)

[1.2. Aims 2](#_Toc194330278)

[1.3. Objectives 2](#_Toc194330279)

[1.4. Artefacts 2](#_Toc194330280)

[1.5. Academic Question 4](#_Toc194330281)

[1.6. Scope and Limitation 4](#_Toc194330282)

[1.6.1. Scope 4](#_Toc194330283)

[1.6.2. Limitations 4](#_Toc194330284)

[1.7. Report Structure 5](#_Toc194330285)

[2. Literature Review 6](#_Toc194330286)

[2.1. Attention Is All You Need 6](#_Toc194330287)

[2.2. Bidirectional Encoder Representations from Transformers 7](#_Toc194330288)

[2.3. Mining and Utilization of English Learning Resources Using the Python NLTK 7](#_Toc194330289)

[2.4. Study of Tesseract OCR 8](#_Toc194330290)

[2.5. Efficient Text Preprocessing for Enhanced Classification 9](#_Toc194330291)

[2.6. Bridging the Gap between Human and Machine Translation 9](#_Toc194330292)

[3. Project Methodology 10](#_Toc194330293)

[3.1. Why SCRUM? 10](#_Toc194330294)

[3.2. Gantt Chart 11](#_Toc194330295)

[3.3. Tools and Technology 11](#_Toc194330296)

[3.3.1. Frontend 11](#_Toc194330297)

[3.3.2. Backend 11](#_Toc194330298)

[3.3.3. Database 11](#_Toc194330299)

[3.3.4. AI Models 12](#_Toc194330300)

[3.3.5. Authentication & Security 12](#_Toc194330301)

[3.3.6. Development Tools 12](#_Toc194330302)

[4. Artefact Designs 13](#_Toc194330303)

[4.1. User Registration and Authentication 13](#_Toc194330304)

[4.1.1. SRS 13](#_Toc194330305)

[4.1.2. UMLS 15](#_Toc194330306)

[4.1.3. Testing 18](#_Toc194330307)

[4.2. Resume-Based Job Recommendation System 22](#_Toc194330308)

[4.2.1. SRS 22](#_Toc194330309)

[4.2.2. UMLS 23](#_Toc194330310)

[4.2.3. Testing 25](#_Toc194330311)

[4.3. Job Management System 27](#_Toc194330312)

[4.3.1. SRS 27](#_Toc194330313)

[4.3.2. UMLS 29](#_Toc194330314)

[4.3.3. Testing 30](#_Toc194330315)

[4.4. Review Management System 34](#_Toc194330316)

[4.4.1. SRS 34](#_Toc194330317)

[4.4.2. UMLS 35](#_Toc194330318)

[4.4.3. Testing 36](#_Toc194330319)

[4.5. Chat System 39](#_Toc194330320)

[4.5.1. SRS 39](#_Toc194330321)

[4.5.2. UMLS 40](#_Toc194330322)

[4.5.3. Testing 40](#_Toc194330323)

[4.6. Schema Design Diagram 43](#_Toc194330324)

[4.7. Wireframes and Designs 44](#_Toc194330325)

[5. Conclusion 51](#_Toc194330326)

[6. Critical Evaluation 51](#_Toc194330327)

[7. Evidence of Project Management 52](#_Toc194330328)

[7.1 Git 52](#_Toc194330329)

[7.2 Log Sheet 53](#_Toc194330330)

[8. References 65](#_Toc194330331)

# Introduction

## Project Briefing

The rapid growth and popularity of computer science have led to an increasingly saturated market of developers. Employers now struggle to distinguish the most suitable developers from countless others. At the same time, numerous highly skilled developers remain undiscovered, adding to the complexity of the situation. This project is designed to overcome these problems by implementing a novel approach connecting employers with developers.

DevX helps solve the issue of increased competition among developers and employers in the industry. Both employers and job seekers have difficulties finding one another among the large number of potential partners available. Traditional hiring processes, including CV screening, technical interviews, and one-on-one discussions, are effective but costly, time-consuming, and exhaustive. Furthermore, existing solutions such as LinkedIn and Upwork prioritize professional networking over finding the optimal pairing of skills and job requirements.

To address these limitations, the platform utilizes Natural Language Processing to enhance the skill matching process. DevX analyzes resumes and job descriptions and provides personalized recommendations that help developers find suitable jobs for them and create equal opportunities for everyone.

The main functionalities include profile management for developers, job posting with detailed descriptions for employers, automated recommendations, and public reviews for transparency. The platform emphasizes meeting each candidate’s abilities, ensuring that all participants experience efficient recruitment experience.

The project’s skill-based match making is totally based on supervised learning techniques. The training data consisted of labeled resumes scrapped from LinkedIn, QwikResume and similar platforms. Supervised learning allows the model to identify the underlying patterns in the resumes with contextual understanding to provide accurate matching of skills to the job description.

DevX's skill-matching system utilizes a sophisticated Natural Language Processing architecture named BERT. This process begins by converting input text, such as resumes, into numerical embeddings via WordPiece Tokenization, capturing semantic and contextual meaning. BERT's Transformer architecture then employs self-attention mechanisms, calculated through a scaled dot-product formula, to discern relationships between tokens, while positional encodings retain word order information. BERT is initially pre-trained using Masked Language Modeling and Next Sentence Prediction, enabling it to grasp bidirectional context. For skill-matching, a pre-trained BERT model is fine-tuned with labeled data, utilizing a classification head to predict resume and job description compatibility. Finally, cosine similarity calculates the alignment between resume and job description embeddings, measuring semantic closeness for precise skill matching. This robust mathematical foundation enables DevX to effectively connect developers and employers.

## Aims

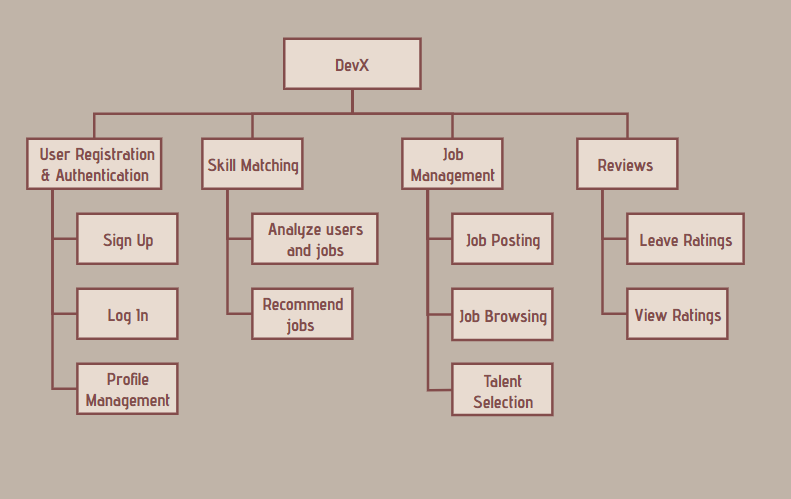
* Address challenges faced while hiring a developer
* Address challenges faced by a developer
* Explore the possible method of solving/ automating the process of hiring a developer
* Develop a platform to bridge employers and developers

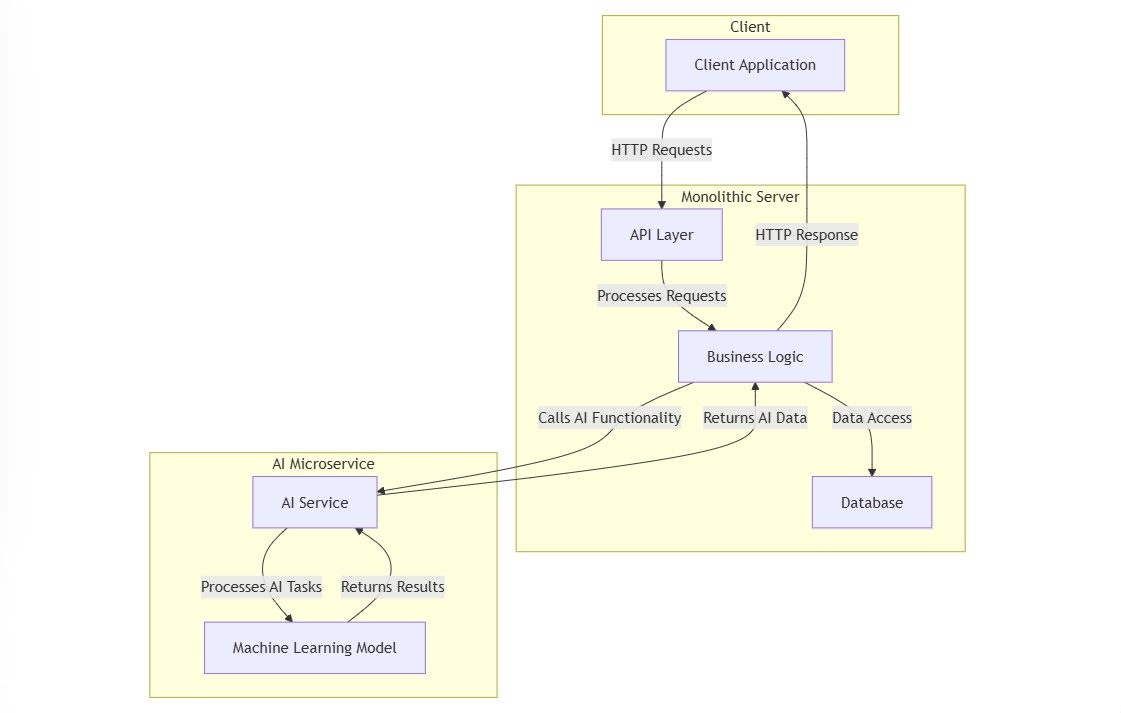
## Objectives

* Identify key challenges employers face during the developer recruitment process.
* Investigate difficulties developers encounter while job hunting.
* Propose innovative solutions for skill matching using machine learning or AI driven algorithms.
* Design and implement an interactive and user-friendly platform that enhances employer-developer collaboration.
* Evaluate the effectiveness of the developed platform through testing and feedback.

## Artefacts

DevX uses high end technologies and structured processes to prioritize the skill matching feature unlike other platforms which focus on the business aspect of freelancing and talent hiring. Users on the platform can manage their profile and upload their resume. The jobs posted can also be personalized through a brief description from the client. The developers and the jobs are then analyzed to provide a proper recommendation of jobs creating equality for all. But the final decision all comes upon the client who can select from the range of the developers and the developers who can apply for their recommended jobs. Also, the client reviews publicly to ensure fairness and transparency . The system can be divided into four main components as follows

*Figure 1: Function Decomposition Diagram*

*Figure 2: System Architecture*

## Academic Question

How can you bridge the gap between employers and developers by addressing all the challenges such as skill matching?

The academic question of the project focuses on understanding how to bridge the connection between employers and developers. It focuses on addressing the critical challenges in the hiring process, especially skill matching. It aims to explore the hurdles that prevent a good and effective connection. The question also explores to identify the innovative methods to create seamless and efficient hiring process that is advantageous for both, employers and developers.

## Scope and Limitation

### Scope

* Skill-Focused Matching: The platform's primary function is to connect developers with clients based on a robust skill-matching algorithm that analyzes profiles and job postings. This feature allows skill-based job recommendation to the developers allowing a level of satisfaction to the clients.
* Developer Selection and Review: Clients can review recommended developers, browse profiles, make their final selection, and publicly rate their performance.
* Technology Driven: AI models and processes are used to prioritize skill relevance and efficient matching which differentiates the platform from others.
* Basic Communication & Features: The platform includes real- time chat and video call features for easy communication between the clients and the talents.

### Limitations

* No Payment & Project Management: The platform does not include any payment processing or integrated project management tools. This means payment arrangements and project collaboration are handled outside the platform.
* Limited Skill Matching Scope: The skill-based matching functionality is initially limited to a pre-defined set of classes/categories for job titles, it will not support all job title classifications from the start.
* Limited Support & Verification: The platform does not provide developer background verification, advanced client verification, or full 24/7 customer support.

## Report Structure

The report starts with an **Introduction** which provides comprehensive details about the project briefing as well as the **Aims** and **Objectives** along with explanations about **Artefacts** and **Academic** **Question** and the **Scope and Limitation** of the DevX platform. The Literature Reviewconsists of research on base works for NLP along with present-day technology developments regarding Transformer architecture, BERT, OCR with Tesseract, effective text preprocessing protocols and the link between human and machine translation mechanisms. The **Project Methodology** includes the agile SCRUM framework adoption with timeline details in a Gantt chart as well as descriptions of technologies used across frontend, backend, database and AI models, authentication and security and development tools. The report examines artefact designs in great detail through Software Requirements Specifications (SRS), Unified Modeling Language diagrams (UMLs) and testing procedures for five essential components which start from user registration and authentication systems and extend to the resume-based job recommendation system alongside job and review management features as well as a chat system. The section includes extensive diagrams of the schema design and multiple wireframes and design representations that visually explain the system architecture. Finally the paper provides an overview which analyzes the project outcomes and also proofs regarding project management used during development.

# Literature Review

## Attention Is All You Need

The NLP field was transformed following the introduction of the Transformer architecture described in the paper “Attention is All You Need” (Vaswani, 2017). Before Transformers, RNNs with LSTM and GRU variants were commonly used for handling such problems as sequence transduction and machine translation. Consequently, scaling RNNs up to long sequences was challenging from both a computational and a time perspective. RNNs were unable to take full advantage of the parallel computing capabilities offered by GPUs. Additionally, RNNs were not well-suited for comprehending long-range dependencies due to the need to transmit information along multiple timesteps, which could result in gradient instability. The Transformer solved the issues caused by recurrency and dependence exclusively by utilizing the attention mechanism. By embracing this novel method, the model was able to analyze all tokens at once, resulting in enormous improvements in parallelization and a considerable reduction in training time. The central element of the Transformer’s design is self-attention, a technique that enables the model to evaluate how important different elements of an input sequence are during the processing of each word or token separately. This approach allows the model to comprehend the relationships between words at a variety of distances within a sentence, thereby helping to overcome the difficult problem of long-range dependency modeling. The incorporation of scaled dot-product attention and multi-head attention helps the model extract the most important information from the input while being able to evaluate varous characteristics of the data. Transformer architectures featuring an encoder-decoder design have consistently outperformed previous models on various NLP tasks and now serve as the basis for further development in the field. The move from recurrence to attention has transformed NLP, paving the way for the creation of more effective and resource-efficient models.

## Bidirectional Encoder Representations from Transformers

## Built upon the foundational Transformer model, the BERT model unveiled by Google AI in 2018 completely transformed the field of natural language processing by highlighting the importance of bidirectional pre-training in building powerful and accurate language representations. BERT introduced a novel bi-directional pre-training scheme where it aims to understand language from two directions through the MLM task. The approach randomizes the selection of tokens and trains the model to recover the deleted ones using the available linguistic context. Training BERT in both directions enables the model to develop a richer and more accurate understanding of words and how they relate to one another inside sentences. BERT implemented the Next Sentence Prediction (NSP) task as a technique to teach the model to identify situations when two sentences follow each other in the original text. Thanks to these abilities, BERT excels at tasks such as question answering and natural language inference, which require an understanding of how different sentences relate to one another. Fusing MLM and NSP with a Transformer backbone allowed BERT to secure substantial improvements in performance across multiple NLP tasks, like GLUE, MultiNLI and SQuAD. Pre-training and fine-tuning quickly became the dominant approach in NLP, allowing researchers to harness huge amounts of unlabeled text to build potent language models and fine-tune them to tackle challenging downstream problems using only a small amount of labeled data. BERT’s outstanding performance highlighted the power of transfer learning in NLP, effectively amplifying results for many different downstream tasks and alleviating the challenge of limited data availability (Google AI, 2018).

## Mining and Utilization of English Learning Resources Using the Python NLTK

(Xiaohong Zhou, 2023) delves into the potential of Python's Natural Language Toolkit (NLTK) to revolutionize English language learning and research. NLTK provides a robust platform for analyzing vast text corpora, enabling researchers to delve into the intricate nuances of language. By leveraging NLTK's powerful tools, researchers can conduct in-depth analyses of lexical richness, syntactic structures, and the contextual nuances of language use within various genres. This study builds upon existing research that underscores the crucial role of text-based corpora in language learning and the significant benefits of employing computational tools for linguistic analysis. Specifically, this research aims to demonstrate how NLTK can be effectively utilized in several key areas. First, it can be used to analyze lexical frequency and identify key vocabulary, offering valuable insights into vocabulary acquisition and the development of language proficiency. Second, by leveraging NLTK’s parsing capabilities, it is possible to investigate syntactic patterns and grammatical structures. This allows for the analysis of sentence construction, the identification of grammatical errors, and a deeper understanding of the rules that govern language. Finally, NLTK can help explore the relationship between language use and different text genres. This includes analyzing variations in language across literary texts, news articles, and social media posts, which can reveal important stylistic and linguistic characteristics unique to each genre. (Xiaohong Zhou, 2023) endeavors to showcase the transformative potential of NLTK in enhancing English language learning and research by providing a robust framework for data-driven analysis and a deeper understanding of the complexities of human language.

## Study of Tesseract OCR

Tesseract OCR has become integral to the process of converting hardcopy documents into digital text. Tesseract is designed to extract text from scanned images in numerous languages, including many Indian languages. Tesseract assumes images as its input, unlike other commercial OCR engines. It processes binary images with or without mathematically specified text fields without relying on sophisticated built-in page layout recognition during the initial stages. Connected component analysis first isolates character outlines in a binary image, making it possible to identify and extract both white-integer dots and black-on-white text. The separated components are subsequently arranged into blobs and are clustered into lines using spatial information. Subsequently, word segmentation is performed by analyzing white space and periods between characters for letters that share the same width and by distinguishing definite spaces and uncertain spacing for variable character spacing. Tesseract uses a two-pass recognition method to improve the system’s precision. A preliminary recognition process employs identified words as input for a customizable classifier, enabling it to adapt to the unique qualities of a particular document. The adaptive classifier is applied during the second pass to refine the identified text. The text is then enhanced by deleting extra spaces and accurately detecting features such as x-height used for specialized formatting. Tesseract first loads an image (70 DPI or higher for best results), determines whether text extraction is appropriate, crops a relevant area, then matches recognized characters against a character database to retrieve text. Features such as optional page layout analysis, adaptive line and word detection in tilted text and handling of fixed-pitch and proportional fonts are also available. We split compound characters and match split characters using A\* algorithm. While evaluations have shown reasonable accuracy in some cases, with examples showing around 89% accuracy based on word count, performance can vary depending on image quality and text complexity. While Tesseract remains a valuable open-source tool, its reliance on older techniques means it has been surpassed by newer deep-learning based OCR systems, highlighting the rapid advancements in the field. (Joshi, 2024)

## Efficient Text Preprocessing for Enhanced Classification

(Lijie Zhu, 2023) introduces a novel approach to text classification by prioritizing data preprocessing. Recognizing that traditional methods often focus solely on optimizing classification algorithms, this study emphasizes the critical role of feature engineering in improving both accuracy and efficiency. (Lijie Zhu, 2023) proposes three innovative preprocessing methods (NP1, NP2, NP3) that combine established techniques such as tokenization, lowercase conversion, and stopword removal.

* Tokenization is the initial step that involves breaking down the raw text into individual words or sub-word units (tokens). This process is crucial as it prepares the text for further analysis.
* Lowercase conversion transforms all characters in the text to lowercase. This step helps to standardize the text by treating words like "Hello" and "hello" as the same, reducing the number of unique terms and simplifying the analysis.
* Stopword removal eliminates common words that have little or no semantic meaning, such as "the," "a," "is," and "and." These words occur frequently but do not contribute significantly to the overall meaning of the text. By removing them, we can reduce noise and improve the focus on more informative terms.

These preprocessing techniques, when combined in different ways within the proposed methods (NP1-NP3), significantly impact the feature space of the text data. By effectively reducing the number of features and removing irrelevant information, these methods contribute to improved classification accuracy and efficiency.

## [Bridging the Gap between Human and Machine Translation](https://paperswithcode.com/paper/googles-neural-machine-translation-system)

The WordPiece approach to subword tokenization has played a crucial role in the development of highly effective Neural Machine Translation models. WordPiece overcomes the difficulty faced by word-based models when processing uncommon or previously unreached words by breaking words down into more frequently occurring subword pieces. It reduces the adverse effects of the OOV issue, in which the model comes across words that are new to it at test time. WordPiece achieves better generalization and improves the model’s ability to deal with rare or unseen words by breaking down words into smaller units. For example, the word “unprecedented” could be decomposed into subwords such as “un,” “prec,” “ed,” and “##ent” where "##" signifies that the subsequent subword is part of a larger word. The split of words into subparts improves the model’s ability to process complex grammar and abundant compound words in languages with rich morphology. In addition, the use of WordPiece results in a smaller vocabulary size, which can lower computational costs and enhance the speed at which the model can reach accurate translations. WordPiece has gained widespread use in the building of high-performance NMT models because it strikes a delicate equilibrium between the flexibility of character-based approaches and the efficiency of word-based representations. (Yonghui Wu, 2016)

# Project Methodology

Choosing the optimal project methodology ensures efficient management and uninterrupted development processes in every development project. The project utilized SCRUM as it proved to be a system that matches how efficient and consistent software can be accomplished in the SDLC.

## Why SCRUM?

SCRUM breaks up work into easily manageable parts during development by deploying sprints as part of its organized procedure. The project would greatly benefit from deploying structured work iterations since its features require them to stay aligned and determine the team’s progress effectively. SCRUM uses iterative development to ensure that features can be easily changed should new requirements emerge throughout the project. The project team tackles any feature changes on a case-by-case basis, making sure to keep track of all modifications to meet new needs. SCRUM enhances teamwork and information flow within the project as core strengths. The project will have multiple chances to promote effective communication with the client thanks to the reviews held after every sprint and routine meetings. they're able to share important suggestions that allow the project to evolve in line with their company objectives. SCRUM facilitates effective risk management by anchoring quality assurance across every step of the Software Development Life Cycle. Developers complete exhaustive and proactive tests early in the development process to identify bugs and improve overall software quality.

## Gantt Chart

A chart with green rectangular bars

AI-generated content may be incorrect.

## Tools and Technology

### Frontend

* + **React**: The primary library for building user interfaces. It allows for a dynamic and responsive user experience.
  + **Tailwind CSS**: For styling and building modern UI components.

### Backend

* + **Node.js**: The runtime environment for executing JavaScript on the server.
  + **Express.js**: A web application framework for Node.js that simplifies routing and server-side logic.
  + **Mongoose**: An ODM (Object Data Modeling) library for MongoDB and Node.js, helping in data validation and schema management.

### Database

* + **MongoDB**: A NoSQL database that allows for flexible data modeling, ideal for storing user data, application states, and other dynamic content.

### AI Models

* + **Python**: For building and training AI models. Python libraries like TensorFlow, PyTorch, or scikit-learn can be used for machine learning.
  + **Flask or FastAPI**: A lightweight web framework to create RESTful APIs in Python for serving your AI models to the Node.js backend.

### Authentication & Security

* + **JSON Web Tokens (JWT)**: For secure user authentication and session management in your application.
  + **bcrypt**: For hashing passwords securely.

### Development Tools

* + **Postman**: For testing APIs during development.
  + **Git**: For version control.
  + **VSCode**: Preferred IDE.

# Artefact Designs

## User Registration and Authentication

### SRS

|  |  |  |
| --- | --- | --- |
| **Req. Code** | **Req. Description** | **Use Case** |
| **URA-F-1.0** | The system shall allow new users to register an account. | User Registration |
| **URA-F-1.1** | The system shall require a unique email for each user account. | User Registration |
| **URA-F-1.2** | The system shall require a password that meets defined complexity criteria (e.g., minimum length, uppercase/lowercase, numbers, special characters). | User Registration |
| **URA-F-1.3** | The system shall require password confirmation during registration. | User Registration |
| **URA-F-1.4** | The system shall validate all registration fields to ensure required information is provided and is in the correct format. | User Registration |
| **URA-F-1.5** | The system shall store user registration information securely in a database. | User Registration |
| **URA-F-1.6** | The system shall send an email to the registered email address. | User Registration |
| **URA-UR-1.7** | The system shall provide appropriate error messages to the user during registration if validation fails. | User Registration |
| **URA-F-1.8** | The system shall allow registered users to log in with their email and password. | User Login |
| **URA-F-1.9** | The system shall verify user credentials against stored data. | User Login |
| **URA-F-1.10** | The system shall allow users to login using password-based authentication. | User Login |
| **URA-NF-1.11** | The system shall implement secure password handling by not storing passwords in plaintext. | User Login |
| **URA-F-1.12** | Upon successful login, the system shall create a session for the logged-in user. | User Login |
| **URA-F-1.13** | Upon successful login, the system shall redirect the user to the appropriate default page. | User Login |
| **URA-F-1.14** | The system shall allow logged-in users to log out securely. | User Logout |
| **URA-F-1.15** | The system shall invalidate user session after logging out. | User Logout |
| **URA-UR-1.16** | The system shall display an error message to the user if login fails (invalid username/password). | User Login |
| **URA-F-1.17** | The system shall provide a "Forgot Password" feature. | Password Reset |
| **URA-F-1.18** | The system shall send a password reset link (or code) to the user's registered email address. | Password Reset |
| **URA-F-1.19** | The system shall allow users to reset their password using the received link (or code). | Password Reset |
| **URA-F-1.20** | The system shall ensure the password reset link is time limited. | Password Reset |
| **URA-NF-1.21** | The system shall encrypt sensitive data, like passwords, at rest and in transit. | Security |
| **URA-NF-1.22** | The system shall protect against common security vulnerabilities (e.g., SQL injection, cross-site scripting). | Security |
| **URA-NF-1.23** | The system shall not store passwords or email reset links in plain text | Security |
| **URA-NF-1.24** | The system shall protect access to user accounts through strong authentication and authorization | Security |
| **URA-NF-1.25** | The system shall respond to user requests (e.g., login, registration) | General |
| **URA-UR-1.26** | The system shall provide clear visual feedback to the user during any processing that takes longer than **1 second**. | General |
| **URA-UR-1.27** | The user interface shall adapt to various screen sizes (desktop, tablet, mobile) without loss of functionality or usability. | General |

### UMLS

A diagram of a diagram

Description automatically generatedA diagram of a user registration

Description automatically generated

A diagram with text and arrows

Description automatically generated

### Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req. Code | Req. Description | Test ID | Test Case Description | Expected Result | Status |
| URA-F-1.0 | The system shall allow new users to register an account. | TC-URA-F-1.0\_001 | Verify successful user registration with valid input data. | New user account is successfully created. Confirmation email is sent. | Pass |
| URA-F-1.1 | The system shall require a unique email for each user account. | TC-URA-F-1.1\_002 | Attempt to register a new user with an email already associated with another user. | System should display an error message stating email is already registered. Registration should fail. | Pass |
| URA-F-1.2 | The system shall require a password that meets defined complexity criteria. | TC-URA-F-1.2\_003 | Attempt to register with a password that is too short or does not contain required characters (uppercase, lowercase, numbers, special characters). | System should display an error message stating password does not meet complexity requirements. Registration should fail. | Pass |
| URA-F-1.2 | The system shall require a password that meets defined complexity criteria. | TC-URA-F-1.2\_004 | Attempt to register with a valid password meeting all complexity requirements. | System should allow user creation with successful password. | Pass |
| URA-F-1.3 | The system shall require password confirmation during registration. | TC-URA-F-1.3\_005 | Attempt to register with password and confirmation fields not matching. | System should display an error message stating passwords do not match. Registration should fail. | Pass |
| URA-F-1.3 | The system shall require password confirmation during registration. | TC-URA-F-1.3\_006 | Attempt to register with password and confirmation fields matching. | System should allow password confirmation. | Pass |
| URA-F-1.4 | The system shall validate all registration fields to ensure required information is provided and is in the correct format. | TC-URA-F-1.4\_007 | Attempt to register with missing required fields (email, password etc.) or incorrect format data. | System should display appropriate error message stating the invalid field. Registration should fail. | Pass |
| URA-F-1.4 | The system shall validate all registration fields to ensure required information is provided and is in the correct format. | TC-URA-F-1.4\_008 | Attempt to register with all required fields with correct format and proper input data. | User registration should be successful. | Pass |
| URA-F-1.5 | The system shall store user registration information securely in a database. | TC-URA-F-1.5\_009 | Verify data stored in the database after successful registration, including hashing of password. | User data is correctly stored in the database with hashed password. | Pass |
| URA-F-1.6 | The system shall send an email to the registered email address. | TC-URA-F-1.6\_010 | Register a new user and verify the confirmation email is received. | Email should be delivered to the specified email address upon successful registration. | Pass |
| URA-UR-1.7 | The system shall provide appropriate error messages to the user during registration if validation fails. | TC-URA-UR-1.7\_011 | Verify all error messages during registration and provide meaningful feedback on the cause of failure. | Relevant and understandable error messages displayed for each registration validation failure. | Pass |
| URA-F-1.8 | The system shall allow registered users to log in with their email and password. | TC-URA-F-1.8\_001 | Attempt to log in with a registered user's valid email and password. | User should be successfully logged in and directed to the default page. | Pass |
| URA-F-1.9 | The system shall verify user credentials against stored data. | TC-URA-F-1.9\_002 | Attempt to log in with an invalid email or password. | System should display an error message stating invalid credentials and login should fail. | Pass |
| URA-F-1.10 | The system shall allow users to login using password-based authentication. | TC-URA-F-1.10\_003 | Attempt to login using valid credentials and password authentication method. | User should be successfully logged in. | Pass |
| URA-NF-1.11 | The system shall implement secure password handling by not storing passwords in plaintext. | TC-URA-NF-1.11\_004 | Verify password storage method in the database after successful registration. | Passwords should be stored using a strong hashing method. | Pass |
| URA-F-1.12 | Upon successful login, the system shall create a session for the logged-in user. | TC-URA-F-1.12\_005 | Log in with a valid user and verify the presence of an active session. | Session for user is created and maintained. | Pass |
| URA-F-1.13 | Upon successful login, the system shall redirect the user to the appropriate default page. | TC-URA-F-1.13\_006 | Log in with a valid user and verify user is redirected to the default homepage after successful login. | User is redirected to the correct default page after successful login. | Pass |
| URA-F-1.14 | The system shall allow logged-in users to log out securely. | TC-URA-F-1.14\_001 | Log in and then log out and ensure log out process is successful. | User is logged out successfully and is no longer an active session. | Pass |
| URA-F-1.15 | The system shall invalidate user session after logging out. | TC-URA-F-1.15\_002 | Log in and then log out, then try accessing authorized pages. | User should not be able to access authorized pages after session invalidation. | Pass |
| URA-UR-1.16 | The system shall display an error message to the user if login fails (invalid username/password). | TC-URA-UR-1.16\_007 | Attempt login with invalid credentials and verify an error message is displayed. | Error message should be displayed indicating login failed and should indicate if it is due to incorrect credentials. | Pass |
| URA-F-1.17 | The system shall provide a "Forgot Password" feature. | TC-URA-F-1.17\_001 | Request a password reset and verify the reset password link email is received. | User should receive an email with the password reset link. | Pass |
| URA-F-1.18 | The system shall send a password reset link (or code) to the user's registered email address. | TC-URA-F-1.18\_002 | Request a password reset and verify the reset link received to users mail. | Password reset email with valid reset link is delivered to the specified email address. | Pass |
| URA-F-1.19 | The system shall allow users to reset their password using the received link (or code). | TC-URA-F-1.19\_003 | Reset password using the link received to email and ensure successful password reset and that they are logged in with the updated password. | User can successfully reset their password and log in with the new password. | Pass |
| URA-F-1.20 | The system shall ensure the password reset link is time limited. | TC-URA-F-1.20\_004 | Request a password reset and wait beyond the time limit of the reset link and then attempt to reset the password using the same link. | Password reset using an expired link should fail and display error message to the user. | Pass |
| URA-NF-1.21 | The system shall encrypt sensitive data, like passwords, at rest and in transit. | TC-URA-NF-1.21\_001 | Verify all sensitive data (passwords, tokens) are encrypted both in the database and during network communication. | Sensitive data is encrypted in both database at rest and during transit. | Pass |
| URA-NF-1.22 | The system shall protect against common security vulnerabilities (e.g., SQL injection, cross-site scripting). | TC-URA-NF-1.22\_002 | Conduct vulnerability scans and penetration testing to ensure system is resistant to common security vulnerabilities. | System should be resistant to common security vulnerabilities and should not be easily attacked. | Pass |
| URA-NF-1.23 | The system shall not store passwords or email reset links in plain text. | TC-URA-NF-1.23\_003 | Verify password and email reset link storage in the database, ensure plain text is not being used. | Passwords and reset links should be stored in a secure, encrypted format, never in plain text. | Pass |
| URA-NF-1.24 | The system shall protect access to user accounts through strong authentication and authorization. | TC-URA-NF-1.24\_004 | Test access to user accounts with valid user credentials, ensure unauthorized users cannot access others accounts or functionalities. | Only authorized users should have access to their own accounts and functionalities. | Pass |
| URA-NF-1.25 | The system shall respond to user requests (e.g., login, registration). | TC-URA-NF-1.25\_001 | Perform multiple login, registration, password reset requests to ensure system does respond to user requests. | System responds to all user requests with appropriate time. | Pass |
| URA-UR-1.26 | The system shall provide clear visual feedback to the user during any processing that takes longer than 1 second. | TC-URA-UR-1.26\_002 | Perform actions like registration and password reset that take more than 1 second and verify the loading indicator is provided. | Loading animation/indicator should be present if any user request is taking longer than 1 second. | Pass |
| URA-UR-1.27 | The user interface shall adapt to various screen sizes | TC-URA-UR-1.27\_003 | Test on different screen sizes to ensure the responsive | Application UI should adapt to different screen size without any loss of functionality and usability. | Pass |

## Resume-Based Job Recommendation System

### SRS

|  |  |  |
| --- | --- | --- |
| **Req. Code** | **Req. Description** | **Use Case** |
| **JRS-F-1.0** | The system shall allow users to upload their resume in a common format (e.g., PDF). | Resume Upload |
| **JRS-F-1.1** | The system shall parse the uploaded resume to extract text content. | Resume Parsing |
| **JRS-F-1.2** | The system shall identify and extract relevant skills mentioned in the resume using Natural Language Processing (NLP) techniques (e.g., Named Entity Recognition, keyword extraction). | Skills Extraction |
| **JRS-F-1.3** | The system shall categorize skills into predefined categories (e.g., Programming Languages, Databases, Frameworks, Cloud Technologies, etc.). | Skills Categorization |
| **JRS-F-1.4** | The system shall analyze the extracted skills and job titles/descriptions in the resume to determine the developer type (e.g., Frontend, Backend, Full-Stack, Mobile, DevOps, etc.). | Developer Type Inference |
| **JRS-F-1.5** | The system shall create a skills profile for each user based on the extracted and categorized skills. | User Profile Generation |
| **JRS-F-1.6** | The system shall calculate the cosine similarity between the user's skills profile and job profile for all job postings in the system. | Similarity Calculation |
| **JRS-F-1.7** | The system shall rank job postings based on their cosine similarity score to a user's skill profile. | Job Ranking |
| **JRS-F-1.8** | The system shall recommend the top N job postings with highest cosine similarity score to the user. (where N is configurable) | Job Recommendation |
| **JRS-F-1.9** | The system shall display recommended job postings to the user in an understandable format, including job title, company, short description, and a link to the full posting. | Job Display |
| **JRS-F-1.10** | The system shall allow user to see the similarity score between his profile and the job recommended. | Job Display |

### UMLS

A screenshot of a computer screen

Description automatically generated

A diagram of a diagram

Description automatically generatedA diagram of a software project

Description automatically generated with medium confidence

### Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req. Code | Req. Description | Test ID | Test Case Description | Expected Result | Status |
| JRS-F-1.0 | The system shall allow users to upload their resume in a common format (e.g., PDF). | TC-JRS-F-1.0\_001 | Attempt to upload a resume in PDF format. | Resume is uploaded successfully. | Pass |
| JRS-F-1.0 | The system shall allow users to upload their resume in a common format (e.g., PDF). | TC-JRS-F-1.0\_002 | Attempt to upload a resume in a non-supported format (e.g., DOCX). | System displays an error message indicating the file format is not supported. | Pass |
| JRS-F-1.1 | The system shall parse the uploaded resume to extract text content. | TC-JRS-F-1.1\_003 | Upload a resume and verify that the text content is extracted correctly. | Text content of the resume is extracted without any errors. | Pass |
| JRS-F-1.1 | The system shall parse the uploaded resume to extract text content. | TC-JRS-F-1.1\_004 | Upload a resume with image-based text and verify the parsing failure. | Text content is not parsed or parsed with errors. | Pass |
| JRS-F-1.2 | The system shall identify and extract relevant skills mentioned in the resume using Natural Language Processing (NLP) techniques (e.g., Named Entity Recognition, keyword extraction). | TC-JRS-F-1.2\_005 | Upload a resume with clearly listed technical skills. | All listed technical skills are identified and extracted correctly. | Pass |
| JRS-F-1.2 | The system shall identify and extract relevant skills mentioned in the resume using Natural Language Processing (NLP) techniques (e.g., Named Entity Recognition, keyword extraction). | TC-JRS-F-1.2\_006 | Upload a resume with skills mentioned in various contexts (e.g., project descriptions, job roles). | Skills mentioned in different contexts are identified and extracted accurately. | Fail |
| JRS-F-1.3 | The system shall categorize skills into predefined categories (e.g., Programming Languages, Databases, Frameworks, Cloud Technologies, etc.). | TC-JRS-F-1.3\_007 | Upload a resume with varied skills and validate categorization. | Extracted skills are categorized into predefined categories correctly. | Pass |
| JRS-F-1.3 | The system shall categorize skills into predefined categories (e.g., Programming Languages, Databases, Frameworks, Cloud Technologies, etc.). | TC-JRS-F-1.3\_008 | Upload a resume with skills that don't fit predefined categories. | Skills that don't fit predefined categories are either categorized into a default category or are flagged. | Fail |
| JRS-F-1.4 | The system shall analyze the extracted skills and job titles/descriptions in the resume to determine the developer type (e.g., Frontend, Backend, Full-Stack, Mobile, DevOps, etc.). | TC-JRS-F-1.4\_009 | Upload a resume with frontend-related skills and job titles. | Developer type should be identified as Frontend Developer. | Pass |
| JRS-F-1.4 | The system shall analyze the extracted skills and job titles/descriptions in the resume to determine the developer type (e.g., Frontend, Backend, Full-Stack, Mobile, DevOps, etc.). | TC-JRS-F-1.4\_010 | Upload a resume with skills and titles related to both frontend and backend. | Developer type should be identified as Full-Stack Developer. | Pass |
| JRS-F-1.5 | The system shall create a skills profile for each user based on the extracted and categorized skills. | TC-JRS-F-1.5\_011 | Upload a resume and verify the user profile generated. | A user skills profile is generated based on the extracted and categorized skills. | Pass |
| JRS-F-1.5 | The system shall create a skills profile for each user based on the extracted and categorized skills. | TC-JRS-F-1.5\_012 | Upload multiple resumes and verify the latest user profile. | Latest profile is updated based on latest uploaded resume. | Pass |
| JRS-F-1.6 | The system shall calculate the cosine similarity between the user's skills profile and job profile for all job postings in the system. | TC-JRS-F-1.6\_013 | Upload a resume, then compare the profile with various job postings. | Cosine similarity scores are calculated correctly for all job postings. | Pass |
| JRS-F-1.6 | The system shall calculate the cosine similarity between the user's skills profile and job profile for all job postings in the system. | TC-JRS-F-1.6\_014 | Upload a resume and verify cosine similarity with no skills match. | Cosine similarity score of 0 should be given to unmatched job postings. | Pass |
| JRS-F-1.7 | The system shall rank job postings based on their cosine similarity score to a user's skill profile. | TC-JRS-F-1.7\_015 | Verify job posting ranking based on cosine similarity scores. | Job postings are ranked based on the cosine similarity in descending order. | Pass |
| JRS-F-1.8 | The system shall recommend the top N job postings with highest cosine similarity score to the user. (where N is configurable) | TC-JRS-F-1.8\_016 | Upload a resume and verify recommended job postings (N=3). | Top 3 jobs with highest cosine similarity scores are recommended. | Pass |
| JRS-F-1.8 | The system shall recommend the top N job postings with highest cosine similarity score to the user. (where N is configurable) | TC-JRS-F-1.8\_017 | Upload a resume and verify recommended job postings (N=5). | Top 5 jobs with highest cosine similarity scores are recommended. | Pass |
| JRS-F-1.9 | The system shall display recommended job postings to the user in an understandable format, including job title, company, short description, and a link to the full posting. | TC-JRS-F-1.9\_018 | Upload a resume and verify job posting display format. | Job postings are displayed with job title, company, short description, and link. | Pass |
| JRS-F-1.9 | The system shall display recommended job postings to the user in an understandable format, including job title, company, short description, and a link to the full posting. | TC-JRS-F-1.9\_019 | Upload a resume and verify the UI in mobile layout. | Job postings are displayed well on mobile layout | Pass |
| JRS-F-1.10 | The system shall allow user to see the similarity score between his profile and the job recommended. | TC-JRS-F-1.10\_020 | Upload resume, verify the similarity score displayed in UI with recommendation. | Cosine similarity score is displayed alongside the recommended jobs. | Pass |

## Job Management System

### SRS

|  |  |  |
| --- | --- | --- |
| **Req. Code** | **Req. Description** | **Use Case** |
| **JMS-F-1.0** | The system shall allow registered clients to post new job openings. | Job Posting |
| **JMS-F-1.1** | Job postings shall include a job title, description, required skills, salary range. | Job Posting |
| **JMS-F-1.2** | The system shall allow employers to edit existing job postings. | Job Editing |
| **JMS-F-1.3** | Clients shall be able to modify all fields of a job posting. | Job Editing |
| **JMS-F-1.4** | The system shall allow clients to delete job postings. | Job Deletion |
| **JMS-F-1.5** | The system shall provide confirmation before permanently deleting a job posting. | Job Deletion |
| **JMS-F-1.6** | The system shall store job postings in a database. | Job Management |
| **JMS-UR-1.7** | The system shall display job postings in a user-friendly format to job seekers. | Job Browsing |
| **JMS-F-1.8** | Job seekers should be able to filter job listings based on various criteria. | Job Browsing |
| **JMS-F-1.9** | The system shall support pagination for long lists of job postings. | Job Browsing |
| **JMS-F-1.10** | The system shall allow employers to browse talent profiles. | Talent Viewing |
| **JMS-F-1.11** | The system shall allow employers to search talent profiles based on criteria. | Talent Viewing |
| **JMS-F-1.12** | Employers shall be able to see talent profiles which includes all information | Talent Viewing |
| **JMS-F-1.13** | The system shall allow job seekers to bid on job postings. | Job Bidding |
| **JMS-F-1.14** | Job seekers shall be able to submit a bid with a personalized message (cover letter). | Job Bidding |
| **JMS-F-1.15** | The system shall notify employers of new bids on their job postings. | Job Bidding |
| **JMS-F-1.16** | Job seekers shall be able to view all the jobs on which they bid. | Job Bidding |
| **JMS-F-1.17** | The system shall allow employers to view bids on their job postings. | Hiring Process |
| **JMS-F-1.18** | Employers shall be able to view job seeker profiles associated with bids. | Hiring Process |
| **JMS-F-1.18** | The system shall allow employers to select a job seeker for hiring. | Hiring Process |
| **JMS-F-1.19** | The system shall notify job seekers if they have been selected for a job. | Hiring Process |
| **JMS-F-1.20** | The system shall notify other job seekers of the job being filled. | Hiring Process |
| **JMS-F-1.21** | The system shall store information about the hiring process. | Hiring Process |
| **JMS-F-1.22** | The system shall implement authentication for employers and job seekers using a unique identifier and password. | Authentication |
| **JMS-NF-1.23** | The system shall implement data validation and sanitization for all user inputs. | Security |

### UMLS

A diagram with text and black lines

Description automatically generated with medium confidence

### Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req. Code | Req. Description | Test ID | Test Case Description | Expected Result | Status |
| JMS-F-1.0 | The system shall allow registered clients to post new job openings. | TC-JMS-F-1.0\_001 | Registered client attempts to post a new job. | Job opening is successfully posted. | Pass |
| JMS-F-1.0 | The system shall allow registered clients to post new job openings. | TC-JMS-F-1.0\_002 | Unregistered user attempts to post a new job. | System should display an error indicating user is not authorized. | Pass |
| JMS-F-1.1 | Job postings shall include a job title, description, required skills, salary range. | TC-JMS-F-1.1\_003 | Post a job with job title, description, required skills and salary range. | Job posting is created successfully and include all specified details. | Pass |
| JMS-F-1.1 | Job postings shall include a job title, description, required skills, salary range. | TC-JMS-F-1.1\_004 | Post a job without some required fields. | System should display error indicating required fields are missing. | Pass |
| JMS-F-1.2 | The system shall allow employers to edit existing job postings. | TC-JMS-F-1.2\_005 | Registered client edits an existing job posting. | Job posting is updated successfully. | Pass |
| JMS-F-1.2 | The system shall allow employers to edit existing job postings. | TC-JMS-F-1.2\_006 | Registered client edits a job posting not owned by them. | System should display error indicating user is not authorized. | Pass |
| JMS-F-1.3 | Clients shall be able to modify all fields of a job posting. | TC-JMS-F-1.3\_007 | Client modifies all fields of a job posting (title, description, skills, salary). | All fields are updated successfully. | Pass |
| JMS-F-1.4 | The system shall allow clients to delete job postings. | TC-JMS-F-1.4\_008 | Client attempts to delete their own job posting. | Job posting is deleted successfully. | Pass |
| JMS-F-1.4 | The system shall allow clients to delete job postings. | TC-JMS-F-1.4\_009 | Client attempts to delete another client's job posting. | System displays an error indicating user is not authorized. | Pass |
| JMS-F-1.5 | The system shall provide confirmation before permanently deleting a job posting. | TC-JMS-F-1.5\_010 | Client tries to delete job posting and checks the confirmation message. | System displays a confirmation before the job is deleted permanently. | Pass |
| JMS-F-1.6 | The system shall store job postings in a database. | TC-JMS-F-1.6\_011 | Post a job, verify it is stored in database. | Job posting is stored in the database successfully. | Pass |
| JMS-UR-1.7 | The system shall display job postings in a user-friendly format to job seekers. | TC-JMS-UR-1.7\_012 | Browse job postings as a job seeker. | Job postings are displayed with relevant details in a clear format. | Pass |
| JMS-F-1.8 | Job seekers should be able to filter job listings based on various criteria. | TC-JMS-F-1.8\_013 | Job seeker filters job postings by job title. | Job postings are filtered based on provided title successfully. | Pass |
| JMS-F-1.8 | Job seekers should be able to filter job listings based on various criteria. | TC-JMS-F-1.8\_014 | Job seeker filters job postings by salary range. | Job postings are filtered based on provided salary range successfully. | Pass |
| JMS-F-1.8 | Job seekers should be able to filter job listings based on various criteria. | TC-JMS-F-1.8\_015 | Job seeker filters job postings by skills. | Job postings are filtered based on provided skills successfully. | Pass |
| JMS-F-1.9 | The system shall support pagination for long lists of job postings. | TC-JMS-F-1.9\_016 | Browse a long list of job postings and check pagination. | Job postings are displayed with pagination controls. | Pass |
| JMS-F-1.10 | The system shall allow employers to browse talent profiles. | TC-JMS-F-1.10\_017 | Employer attempts to browse talent profiles. | Employer is able to browse the list of talent profiles. | Pass |
| JMS-F-1.10 | The system shall allow employers to browse talent profiles. | TC-JMS-F-1.10\_018 | Job Seeker attempts to browse talent profiles. | System should display error indicating user is not authorized. | Pass |
| JMS-F-1.11 | The system shall allow employers to search talent profiles based on criteria. | TC-JMS-F-1.11\_019 | Employer searches talent profiles based on skills. | Talent profiles are filtered based on the skill input. | Pass |
| JMS-F-1.11 | The system shall allow employers to search talent profiles based on criteria. | TC-JMS-F-1.11\_020 | Employer searches talent profiles based on name. | Talent profiles are filtered based on the name input. | Pass |
| JMS-F-1.12 | Employers shall be able to see talent profiles which includes all information | TC-JMS-F-1.12\_021 | Employer views a talent profile and all information is present. | Talent profile is displayed with complete set of information. | Pass |
| JMS-F-1.13 | The system shall allow job seekers to bid on job postings. | TC-JMS-F-1.13\_022 | Job seeker bids on a job posting. | Job seeker is able to bid on job postings successfully. | Pass |
| JMS-F-1.13 | The system shall allow job seekers to bid on job postings. | TC-JMS-F-1.13\_023 | Job seeker tries to bid on a job posting twice. | System should display error indicating job seeker already applied for the position. | Pass |
| JMS-F-1.14 | Job seekers shall be able to submit a bid with a personalized message (cover letter). | TC-JMS-F-1.14\_024 | Job seeker bids with a cover letter message. | Cover letter is successfully saved with the bid. | Pass |
| JMS-F-1.15 | The system shall notify employers of new bids on their job postings. | TC-JMS-F-1.15\_025 | Job seeker bids on job, employer gets the notification. | Employer gets notification on a new bid successfully. | Pass |
| JMS-F-1.16 | Job seekers shall be able to view all the jobs on which they bid. | TC-JMS-F-1.16\_026 | Job seeker browses jobs which they have bid on. | Job seeker can view all jobs where they have bids. | Pass |
| JMS-F-1.17 | The system shall allow employers to view bids on their job postings. | TC-JMS-F-1.17\_027 | Employer views bids on their job postings. | Employer is able to see bids on their job posting. | Pass |
| JMS-F-1.18 | Employers shall be able to view job seeker profiles associated with bids. | TC-JMS-F-1.18\_028 | Employer views a job seeker profile associated with a bid. | Employer is able to see job seeker profile associated with the bid. | Pass |
| JMS-F-1.18 | The system shall allow employers to select a job seeker for hiring. | TC-JMS-F-1.18\_029 | Employer selects a job seeker for hiring for a specific job. | Employer is able to hire a job seeker. | Pass |
| JMS-F-1.19 | The system shall notify job seekers if they have been selected for a job. | TC-JMS-F-1.19\_030 | A selected job seeker gets the notification. | Selected job seeker receives the notification. | Pass |
| JMS-F-1.20 | The system shall notify other job seekers of the job being filled. | TC-JMS-F-1.20\_031 | Other job seekers receives the notification that job is filled. | Other job seeker receives notification job position is filled. | Pass |
| JMS-F-1.21 | The system shall store information about the hiring process. | TC-JMS-F-1.21\_032 | Hiring process is carried out and information stored. | Hiring process is successfully tracked and stored in database. | Pass |
| JMS-F-1.22 | The system shall implement authentication for employers and job seekers using a unique identifier and password. | TC-JMS-F-1.22\_033 | Login with valid user credentials | Users are able to login with valid username and password successfully. | Pass |
| JMS-F-1.22 | The system shall implement authentication for employers and job seekers using a unique identifier and password. | TC-JMS-F-1.22\_034 | Login with invalid user credentials | User is not able to login and gets error message. | Pass |

## Review Management System

### SRS

|  |  |  |
| --- | --- | --- |
| **Req. Code** | **Req. Description** | **Use Case** |
| **RMS-F-1.0** | The system shall allow employers to submit reviews for developers they have hired. | Review Creation |
| **RMS-F-1.1** | Reviews shall include a rating (e.g., on a scale of 1-5 stars) and an optional text-based comment. | Review Creation |
| **RMS-F-1.2** | Reviews shall be associated with the specific developer profile they were given for. | Review Association |
| **RMS-F-1.3** | The system shall not allow users to review themselves. | Review Restriction |
| **RMS-F-1.4** | Employers shall only be able to review a developer after completing a hiring process with them. | Review Restriction |
| **RMS-F-1.5** | The system shall display the average rating for each developer profile. | Review Display |
| **RMS-F-1.6** | The system shall display all individual reviews on the developer profile page. | Review Display |
| **RMS-F-1.7** | The system shall display the reviewer’s name or alias with each review. | Review Display |
| **RMS-F-1.8** | Reviews shall be displayed in chronological order (most recent first). | Review Display |
| **RMS-F-1.9** | The system shall not allow employers to edit or delete reviews they have submitted after they have been posted. | Review Restriction |

### UMLS

A diagram of a process

Description automatically generatedA screen shot of a computer screen

Description automatically generated

A diagram of a review management system

Description automatically generated

### Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Test Case Description | Expected Result | Status |
| TC-RMS-F-1.0\_001 | Employer submits a review for a hired developer. | Review is successfully submitted. | Pending |
| TC-RMS-F-1.0\_002 | Job Seeker tries to submit a review. | System should display an error that only employers can submit review. | Pending |
| TC-RMS-F-1.1\_003 | Employer submits a review with a 3-star rating and text comment. | Review is submitted with provided star rating and comment. | Pending |
| TC-RMS-F-1.1\_004 | Employer submits a review with 5-star rating and no comment. | Review is submitted with provided star rating and no comment. | Pending |
| TC-RMS-F-1.1\_005 | Employer submits a review with invalid rating value. | System should display an error indicating that the rating value is invalid. | Pending |
| TC-RMS-F-1.2\_006 | Employer submits a review for a developer and check the profile of developer. | Review is correctly associated with the correct developer profile. | Pending |
| TC-RMS-F-1.3\_007 | Developer attempts to review their own profile. | System displays an error message that user can't review themselves. | Pending |
| TC-RMS-F-1.4\_008 | Employer tries to review a developer they have not hired. | System displays error indicating they can only review a developer after the hiring process. | Pending |
| TC-RMS-F-1.4\_009 | Employer tries to review a developer after hiring them. | System allows the employer to submit review for the developer after hiring them. | Pending |
| TC-RMS-F-1.5\_010 | Check the developer profile with one review, it should display average rating as same as the provided rating. | Average rating is displayed for developer profile correctly after one review. | Pending |
| TC-RMS-F-1.5\_011 | Check the developer profile with multiple reviews, it should display average rating as a computed average. | Average rating is displayed for developer profile correctly after multiple reviews. | Pending |
| TC-RMS-F-1.6\_012 | Check the developer profile and verify all the individual reviews are present. | All reviews given to the developer are displayed in developer profile. | Pending |
| TC-RMS-F-1.7\_013 | Check the developer profile with reviews and reviewer information. | Reviewer's name or alias is shown with every review. | Pending |
| TC-RMS-F-1.8\_014 | Check the developer profile with reviews and their order. | Reviews are displayed in chronological order with most recent first. | Pending |
| TC-RMS-F-1.9\_015 | Employer tries to edit a review after posting it. | System does not allow employer to edit their own reviews after submission. | Pending |
| TC-RMS-F-1.9\_016 | Employer tries to delete a review after posting it. | System does not allow employer to delete their own reviews after submission. | Pending |

## Chat System

### SRS

|  |  |  |
| --- | --- | --- |
| **Req. Code** | **Req. Description** | **Use Case** |
| CHAT-F-1.0 | The system shall allow users to send messages to other users. | Message Sending |
| CHAT-F-1.1 | Users shall be able to send text-based messages. | Message Sending |
| CHAT-F-1.2 | The system shall display the sender's name or alias with each message. | Message Display |
| CHAT-F-1.3 | The system shall display messages in chronological order (earliest first). | Message Display |
| CHAT-F-1.4 | The system shall provide visual notifications when new messages are received. | Message Notification |
| CHAT-F-1.5 | The system shall persist messages for future access. | Message Persistence |
| CHAT-F-1.6 | Users shall be able to search through their message history. | Message Retrieval |
| CHAT-F-1.7 | Users shall be able to initiate a chat with another user through their profile page. | Chat Initiation |
| CHAT-F-1.8 | Users shall be able to delete their own messages after sending. | Message Management |

### UMLS

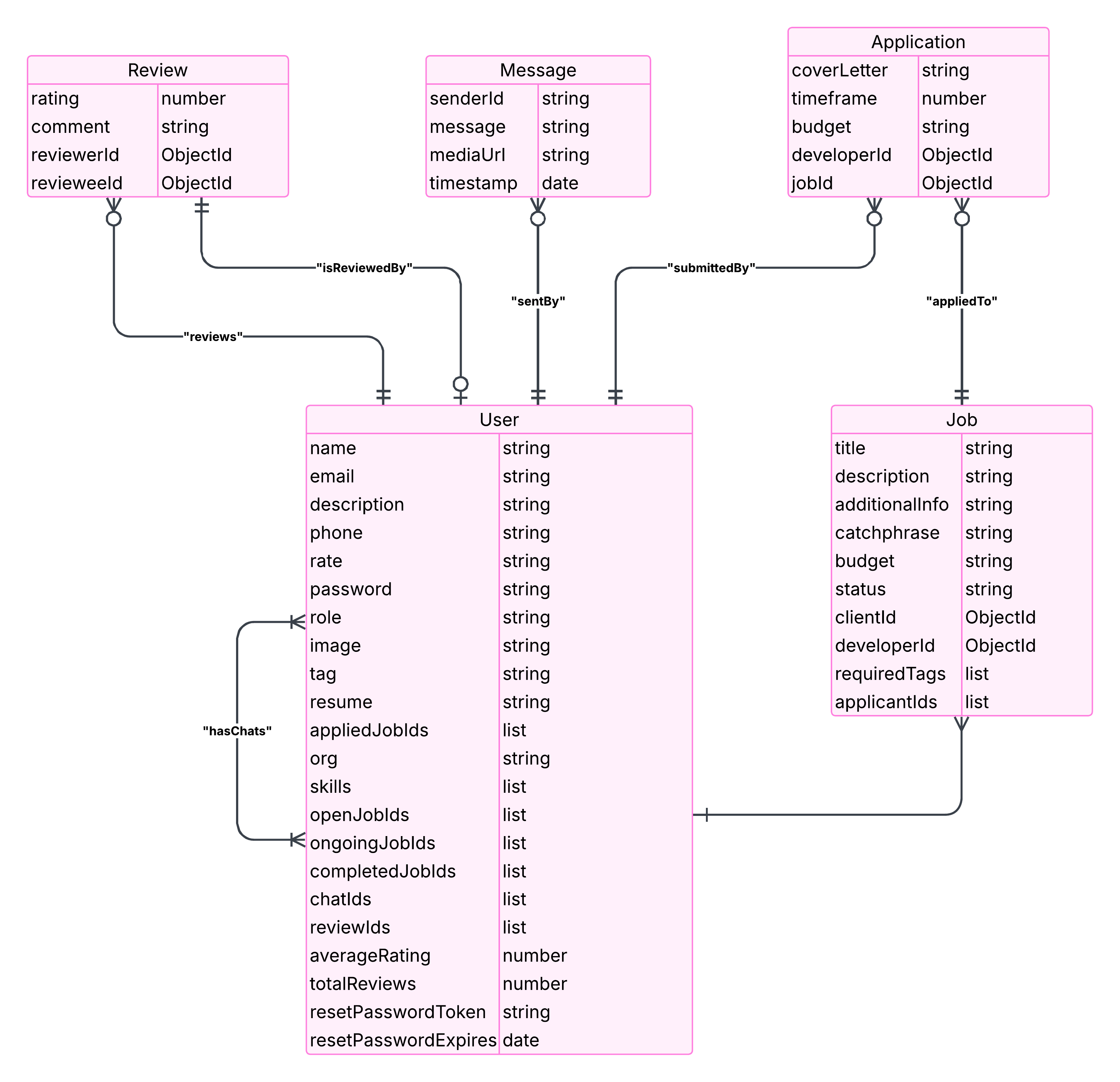
A diagram of a chat service

Description automatically generated

### Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req. Code | Req. Description | Test ID | Test Case Description | Expected Result | Status |
| CHAT-F-1.0 | The system shall allow users to send messages to other users. | TC-CHAT-F-1.0\_001 | Send a message to another user with valid credentials and content. | Message is successfully sent to the intended recipient. | Pending |
| CHAT-F-1.0 | The system shall allow users to send messages to other users. | TC-CHAT-F-1.0\_002 | Attempt to send a message to a user who doesn't exist. | System displays an appropriate error message (if applicable) or the message is not sent. | Pending |
| CHAT-F-1.1 | Users shall be able to send text-based messages. | TC-CHAT-F-1.1\_003 | Send a message containing standard text characters. | Message is sent correctly. | Pending |
| CHAT-F-1.1 | Users shall be able to send text-based messages. | TC-CHAT-F-1.1\_004 | Send a message containing special characters, emojis, or symbols. | Message is sent correctly. | Pending |
| CHAT-F-1.1 | Users shall be able to send text-based messages. | TC-CHAT-F-1.1\_005 | Send a long message that exceeds the specified length limit. | Message is not sent or an error is displayed if the message length limit is breached. | Pending |
| CHAT-F-1.2 | The system shall display the sender's name or alias with each message. | TC-CHAT-F-1.2\_006 | Verify the sender's name/alias is displayed correctly along with the message. | Sender's name/alias appears accurately next to the sent message. | Pending |
| CHAT-F-1.2 | The system shall display the sender's name or alias with each message. | TC-CHAT-F-1.2\_007 | Verify the sender's name is displayed correctly even with special characters or formatting. | Sender's name/alias appears accurately next to the sent message. | Pending |
| CHAT-F-1.3 | The system shall display messages in chronological order (earliest first). | TC-CHAT-F-1.3\_008 | Send multiple messages and verify they appear in the correct order. | Messages are displayed in chronological order, with the earliest at the top. | Pending |
| CHAT-F-1.4 | The system shall provide visual notifications when new messages are received. | TC-CHAT-F-1.4\_009 | Receive a new message and verify visual notification is displayed (e.g., badge, sound). | Visual notification is correctly displayed/delivered for new message reception. | Pending |
| CHAT-F-1.4 | The system shall provide visual notifications when new messages are received. | TC-CHAT-F-1.4\_010 | Receive multiple new messages and verify visual notification reflects the new count. | Visual notification updates with new messages. | Pending |
| CHAT-F-1.5 | The system shall persist messages for future access. | TC-CHAT-F-1.5\_011 | Send messages and verify they are still available after logging out and logging back in. | Messages are persisted and accessible after logging out and back in. | Pending |
| CHAT-F-1.5 | The system shall persist messages for future access. | TC-CHAT-F-1.5\_012 | Verify messages are persisted correctly even when application is restarted or closed. | Messages are persisted and accessible after application restart or close. | Pending |
| CHAT-F-1.6 | Users shall be able to search through their message history. | TC-CHAT-F-1.6\_013 | Search for a specific word or phrase within message history using search function. | System correctly locates and displays messages containing the search criteria. | Pending |
| CHAT-F-1.6 | Users shall be able to search through their message history. | TC-CHAT-F-1.6\_014 | Attempt to search for a term that doesn't exist in the message history. | System correctly indicates that there are no matching messages. | Pending |
| CHAT-F-1.7 | Users shall be able to initiate a chat with another user through their profile page. | TC-CHAT-F-1.7\_015 | Navigate to a user profile and click on the chat option. | Chat window opens with user profile. | Pending |
| CHAT-F-1.8 | Users shall be able to delete their own messages after sending. | TC-CHAT-F-1.8\_016 | Send a message and then delete it, verify the message is deleted from all recipients view. | Message is removed from the chat thread for both users after deletion. | Pending |
| CHAT-F-1.8 | Users shall be able to delete their own messages after sending. | TC-CHAT-F-1.8\_017 | Send a message and verify that you can't delete the message after time specified. | System throws an error that the message cannot be deleted as the time to delete the message has lapsed. | Pending |

## Schema Design Diagram



## Wireframes and Designs

A screenshot of a computer

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA person sitting in front of a computer

Description automatically generatedA screenshot of a login page

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a website

Description automatically generated

# Conclusion

The DevX platform is an effective solution for modern hiring complexities and outdated recruitment methods. The platform implements an NLP system with BERT architecture for skill base match making process. This utilization allows the recruitment procedures to be simplified and efficient. The implementation of the profile management, job recommendations, and public reviews along with chat system for efficient communication allows the system to be effective, efficient, transparent and easy to use. The project was completed by utilizing SCRUM methodology with suitable technological choice. DevX provides major benefits to users despite its current constraints of payment capabilities, limited categories and resume verification. This complete project demonstrates that AI-driven technology helps IT companies connect with developers and provides basic foundations to expand future activities in the tech industry.

# Critical Evaluation

The research outcomes underline essential insights from the system development stage and user experience and process efficiency aspects. The implementation process included orderly stages that started with research followed by planning and repeated development cycles. Implementing crucial program functions represents a substantial achievement since they fulfill the target objectives of the project. Additional testing scenarios combined with user feedback from potential users would have strengthened the findings to validate the system's operation in various utilization circumstances. The system shows effective performance according to its specified requirements and criteria. A successful testing system confirmed its reliability and user-friendliness by evaluating performance as well as usability. The system achieved its main success by perfectly integrating different components to maintain operational integrity. The system requires optimization in specific sections by improving its response speed when facing high load and implementing new features to match user requirements.

The planning system and source quality management combined effectively with organizational strategies. Project planning together with management exhibited strong execution capabilities because of a structured timeline that maintained its milestones. Reputable references formed part of an excellent selection of sources which supported the development process with solid information. The project faced time management challenges mostly in implementation phases until suitable adjustments allowed deadline compliance. The project outcomes would have been enhanced by additional detailed risk assessments and well-structured contingency plans. The experience proved exceptional for developing both career and personal abilities and skills. The initiative developed functional problem resolution techniques together with superior technical competency and improved project administration abilities. The process of working through system development challenges helped us understand better the value of research and continuous learning and adaptability. User feedback proved crucial for system development since it led to refinements that improved system performance according to users' expectations. This project functioned as an important initial step for better comprehension of software development methods and enhanced the ability to think critically about practical issues.

# Evidence of Project Management

## 7.1 Git

A screenshot of a computer

AI-generated content may be incorrect.

## 7.2 Log Sheet

A paper with writing on it

AI-generated content may be incorrect.

A paper with writing on it

AI-generated content may be incorrect.

A paper with writing on it

AI-generated content may be incorrect.

A paper with writing on it

AI-generated content may be incorrect.

A piece of paper with writing on it

AI-generated content may be incorrect.

A white sheet of paper with writing on it

AI-generated content may be incorrect.

A piece of paper with writing on it

AI-generated content may be incorrect.

A paper with writing on it

AI-generated content may be incorrect.

A paper with writing on it

AI-generated content may be incorrect.

A piece of paper with writing on it

AI-generated content may be incorrect.

A white paper with writing on it

AI-generated content may be incorrect.

A paper with writing on it

AI-generated content may be incorrect.

# References

Google AI, 2018. *BERT,* s.l.: s.n.

Joshi, K., 2024. *Study of Tessaract OCR.* s.l., GLS KALP: Journal of Multidisciplinary Studies.

Lijie Zhu, D. L., 2023. A Novel Efficient and Effective Preprocessing Algorithm for Text Classification. *Journal of Computer and Communications ,* Volume 11.

Vaswani, A. N. M. S. N. P. J. U. L. J. A. N. G. L. K. I. P., 2017. *Attention is all you need.* s.l., Neural Information Processing Systems.

Xiaohong Zhou, J. S., 2023. *Mining and Utilization of English Learning Resources Using the Python NLTK.* s.l., IEEE.

Yonghui Wu, M. S. Z. C. Q. V. L. M. N. W. M. M. K. Y. C. Q. G. K. M., 2016. *Bridging the Gap between Human and Machine Translation,* s.l.: s.n.