

# **SMART RESUME SELECTOR**

## **A MAIN PROJECT REPORT**

submitted in partial fulfillment of the requirements for the  
award of the Degree of

**Master Of Computer Applications**

**UNDER**

**APJ Abdul Kalam Technological University**

**BY**

**MUHAMMED SHABEER M P**

**MES21MCA-2027**



**DEPARTMENT OF COMPUTER APPLICATIONS**

**MES COLLEGE OF ENGINEERING**

**KUTTIPPURAM, MALAPPURAM - 679 582**

**May 2023**

# Declaration

I undersigned hereby declare that the main project report **SMART RE-SUME SELECTOR** submitted in partial fulfillment of the requirements for the award of *Degree of Master of Computer Applications* under *APJ Abdul Kalam Technological University* is a bona fide record done by me under the supervision of **Vasudevan T V**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and / or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Sincerely,  
**MUHAMMED SHABEER M P**  
**MES21MCA-2027**

Place: Kuttippuram

Date:

**MES COLLEGE OF ENGINEERING**  
**KUTTIPPURAM, KERALA - 679 582**  
**(A NAAC ACCREDITED INSTITUTION WITH NBA**  
**ACCREDITED DEPARTMENTS,**  
**APPROVED BY AICTE AND AFFILIATED TO APJ ABDUL**  
**KALAM TECHNOLOGICAL UNIVERSITY )**



**DEPARTMENT OF COMPUTER APPLICATIONS**

**CERTIFICATE**

This is to certify that the main project report entitled **SMART RESUME SELECTOR** is a bona fide record of the main project work carried out by **MUHAMMED SHABEER M P (MES21MCA-2027 )**, fourth semester student of the department, during the academic year 2022-23, in partial fulfillment of the requirements for the award of *Degree of Master of Computer Applications* under *APJ Abdul Kalam Technological University*. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s)

External Supervisor(s)

External Examiner

Head of the Department

# Acknowledgement

My endeavor stands incomplete without dedicating my gratitude to a few people who have contributed towards the successful completion of my main project.

I pay my gratitude to the Almighty for His invisible help and blessing for the fulfillment of this work .

At the outset I express my heart full thanks to our **Head of the Department, Prof.Hyderali.K** for permitting me to do this main project.

I would like to express my sincere gratitude to , our Project coordinator,**Vasudevan T V**, Assistant Professor for his exceptional support and encouragement throughout this project.

I take this opportunity to express my profound gratitude to **Vasudevan T V**, my **guide**, for his valuable support and help in my main project.

I am also grateful to all our teaching and non teaching staff for their encouragement, guidance and whole-hearted support.

Last but not least, I am gratefully indebted to my family and friends, who gave me their precious help in my main project.

Sincerely,  
**Muhammed Shabeer M P**  
**MES21MCA-2027**

# Synopsis

Job is an important phase of a person's life as it brings out the true personality and tests you in pressure handling situations. It is often seen that the correct person is not placed for the position and later organization find that the selected employee does not have the required skills and qualification which results in dissatisfaction with employee. Resumes plays an important role in the recruitment process, it decides the candidate's first impression on the recruiter. Resumes are a form of unstructured data, it is really hard to programmatically read a resume and extract information from it. This system for automatically choose the right candidate for a right job profile which reduces the effort, time and extra cost spend. This system help to short-list the candidate initially based on their resume for further process. It will automate the process of resume screening making it easy and time-efficient. The resume is then converted into tokens which are compared with the defined array of information and skill sets as per the requirement. It focuses on matching occupational category and job requirements with the candidate resume which results in easy management.

# Table Of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	2
1.1.1	Motivation . . . . .	2
1.2	Objective . . . . .	3
1.3	Contribution . . . . .	3
1.4	Report Organisation . . . . .	4
<b>2</b>	<b>Literature Survey</b>	<b>5</b>
<b>3</b>	<b>Methodology</b>	<b>6</b>
3.1	Introduction . . . . .	6
3.1.1	NLP (Natural Language Processing) . . . . .	6
3.1.2	Naïve Bayes classifier . . . . .	7
3.1.3	Developing Environment . . . . .	8
3.1.4	Module Description . . . . .	8
3.2	Workflow . . . . .	9
3.3	Data Flow Diagram . . . . .	11
3.4	Implementation . . . . .	13
<b>4</b>	<b>Agile Methodology</b>	<b>15</b>
4.1	Introduction . . . . .	15
4.2	User Story . . . . .	16
4.3	Product Backlog . . . . .	17
4.4	Project Plan . . . . .	19
4.5	Sprint Backlog (Plan) . . . . .	20
4.6	Sprint Backlog (Actual) . . . . .	22
4.7	Product Backlog Review . . . . .	24
4.8	Sprint Review . . . . .	26
4.9	Testing and Validation . . . . .	29
4.10	Git . . . . .	32
<b>5</b>	<b>Conclusion</b>	<b>33</b>
	<b>BIBLIOGRAPHY</b>	<b>34</b>
<b>A</b>	<b>Source Code</b>	<b>35</b>
<b>B</b>	<b>Database Design</b>	<b>39</b>

<b>C Output</b>	<b>42</b>
<b>D Git History</b>	<b>49</b>

# List of Figures

3.1	level 0 . . . . .	11
3.2	level 1 . . . . .	11
3.3	level 2 . . . . .	12
3.4	level 3 . . . . .	12
C.1	Candidate Registration . . . . .	42
C.2	Company Registration . . . . .	42
C.3	Login . . . . .	43
C.4	Admin Home . . . . .	43
C.5	Verify Company . . . . .	43
C.6	View Candidate . . . . .	44
C.7	Manage Videos . . . . .	44
C.8	Manage Tips . . . . .	44
C.9	View Compliant . . . . .	45
C.10	View Feedback . . . . .	45
C.11	Company Home . . . . .	45
C.12	Manage Vacancy . . . . .	45
C.13	Add Vacancy . . . . .	46
C.14	Add Questions . . . . .	46
C.15	Post Compliant . . . . .	46
C.16	View Resume Score . . . . .	46
C.17	View Matching Job . . . . .	47
C.18	View Vacancy . . . . .	47
C.19	Upload Resume . . . . .	47
C.20	Attend Exam . . . . .	47
C.21	View Tips . . . . .	48
C.22	View Videos . . . . .	48
C.23	View Result . . . . .	48
C.24	Post Feedback . . . . .	48
D.1	Git Repository . . . . .	49
D.2	Git Commits . . . . .	49



# List of Tables

4.1	User Story . . . . .	17
4.2	Product Backlog . . . . .	18
4.3	Project plan . . . . .	19
4.4	Sprint Backlog (Plan)-Sprint 1 . . . . .	20
4.5	Sprint Backlog (Plan)-Sprint 2 . . . . .	20
4.6	Sprint Backlog (Plan)-Sprint 3 . . . . .	21
4.7	Sprint Backlog (Plan)-Sprint 4 . . . . .	21
4.8	Sprint Backlog (Actual)-Sprint 1 . . . . .	22
4.9	Sprint Backlog (Actual)-Sprint 2 . . . . .	22
4.10	Sprint Backlog (Actual)-Sprint 3 . . . . .	23
4.11	Sprint Backlog (Actual)-Sprint 4 . . . . .	23
4.12	Product backlog review sprint 1 . . . . .	24
4.13	Product backlog review sprint 2 . . . . .	24
4.14	Product backlog review sprint 3 . . . . .	25
4.15	Product backlog review sprint 4 . . . . .	25
4.16	Sprint review sprint 1 . . . . .	26
4.17	Sprint review sprint 2 . . . . .	27
4.18	Sprint review sprint 3 . . . . .	27
4.19	Sprint review sprint 4 . . . . .	28
4.20	Test Sprint 1 . . . . .	29
4.21	Test Sprint 2 . . . . .	30
4.22	Test Sprint 3 . . . . .	31
4.23	Test Sprint 4 . . . . .	32
B.1	Login . . . . .	39
B.2	Company . . . . .	39
B.3	Candidate . . . . .	39
B.4	vacancy . . . . .	40
B.5	Applications . . . . .	40
B.6	testquestion . . . . .	40
B.7	testresult . . . . .	40
B.8	complaint . . . . .	41
B.9	feedback . . . . .	41
B.10	tips . . . . .	41
B.11	videos . . . . .	41

# Chapter 1

## Introduction

The hiring process is a critical aspect of any organization's success, as finding the right talent can greatly impact business outcomes. However, sifting through a large number of resumes and assessing candidates' qualifications can be a time-consuming and challenging task for recruiters. To address this issue, we present an innovative project named "Smart Resume Selector" that combines Naive Bayes-based vacancy prediction, NLP-based resume scoring, and MCQ test ranking to streamline the hiring process and enhance candidate selection.

The first component of the Smart Resume Selector project is the Naive Bayes-based vacancy prediction. By leveraging historical data and training a Naive Bayes classifier, the system learns the patterns and relationships between resume features and job vacancies. This algorithm calculates the probability of a candidate being a good fit for a specific vacancy based on the available resume information. The Naive Bayes-based prediction serves as an initial filter, allowing recruiters to focus their attention on the most suitable candidates and reducing the manual effort required for initial screening.

The second component of the project is NLP-based resume scoring. Natural Language Processing techniques are employed to extract and analyze relevant information from resumes, including skills, experience, education, and achievements. By applying advanced algorithms such as named entity recognition, part-of-speech tagging, and sentiment analysis, the system assigns a score to each resume based on its relevance to the job requirements. This automated scoring process enables recruiters to objectively evaluate and compare candidates' qualifications, further streamlining the selection process.

To further enhance candidate assessment, the Smart Resume Selector project incorporates an MCQ (Multiple Choice Questions) test based on the job vacancy. The MCQ test is designed to evaluate candidates' knowledge and expertise related to the specific job requirements. By including domain-specific questions, recruiters can gain valuable insights into candidates' subject matter understanding and their ability to apply knowledge in practical scenarios. Finally, the project combines the resume score and MCQ test results to generate a comprehensive ranking of candidates. By considering both the qualifications highlighted in the resume and the performance in the domain-specific MCQ test, the system provides a holistic

assessment of each candidate's suitability for the vacancy. This combined ranking approach enables recruiters to make more informed decisions and prioritize candidates who possess the right combination of skills, experience, and subject matter expertise.

The Smart Resume Selector project aims to revolutionize the hiring process by leveraging advanced technologies and intelligent algorithms. By automating resume screening, objectively scoring resumes, and incorporating a domain-specific MCQ test, the system enhances the efficiency and effectiveness of candidate selection. This project holds the potential to significantly reduce manual effort, save time, and improve the overall quality of hires.

## 1.1 Background

The Smart Resume Selector project was developed in response to the challenges faced by recruiters in the hiring process. Traditional methods of resume screening and candidate evaluation often involve a laborious manual review, which can be time-consuming and prone to biases. To overcome these limitations, the project combines advanced technologies such as Naive Bayes-based vacancy prediction, Natural Language Processing (NLP)-based resume scoring, and MCQ test ranking. By leveraging these techniques, the Smart Resume Selector aims to streamline the hiring process, enhance the accuracy of candidate evaluation, and ultimately improve the efficiency and effectiveness of talent acquisition. This innovative project brings together the power of machine learning, NLP, and intelligent algorithms to revolutionize the way organizations select and hire candidates for their vacancies.

### 1.1.1 Motivation

The motivation behind the development of the Smart Resume Selector project stems from the need to address the inefficiencies and challenges faced by recruiters in the hiring process. Traditional methods of resume screening and candidate evaluation often lead to time-consuming and subjective decision-making, which can result in overlooking qualified candidates or biases in the selection process. By automating and enhancing the screening process with advanced technologies, the Smart Resume Selector aims to save recruiters' time, improve the quality of candidate evaluation, and ensure a fair and efficient selection process.

Furthermore, the increasing volume of resumes received for each job opening makes it even more challenging for recruiters to identify the most suitable candidates. With the advent of technology and the availability of vast amounts of data, it has become crucial to leverage intelligent algorithms to analyze resumes and extract valuable insights. The Smart Resume Selector project was motivated by the opportunity to harness the power of Naive Bayes-based vacancy prediction, NLP-based resume scoring, and MCQ test ranking to provide recruiters with a comprehensive and efficient tool for candidate evaluation. By empowering recruiters with advanced technologies, the project aims to overcome the limitations of traditional

methods and revolutionize the hiring process, ultimately leading to better hiring decisions and successful outcomes for organizations.

## 1.2 Objective

This Project is to revolutionize the hiring process by leveraging advanced technologies such as Naive Bayes-based vacancy prediction, NLP-based resume scoring, and MCQ test ranking. The primary goal is to streamline and automate the resume screening and candidate evaluation process, saving time for recruiters and increasing the efficiency of talent acquisition. By improving the accuracy and objectivity of candidate evaluation through intelligent algorithms, the project aims to enhance the quality of hires and ensure that the selected candidates possess the right combination of skills, experience, and domain-specific knowledge. Additionally, the project aims to reduce bias in the hiring process by providing standardized and objective evaluation criteria, promoting fairness and equal opportunities for all candidates. Ultimately, the objective is to transform the way organizations select and hire candidates, resulting in improved hiring decisions and successful outcomes.

## 1.3 Contribution

The Smart Resume Selector project makes several key contributions to the field of hiring and candidate selection:

- **Automation and Efficiency:** By leveraging advanced technologies such as Naive Bayes-based vacancy prediction, NLP-based resume scoring, and MCQ test ranking, the project automates and streamlines the resume screening and candidate evaluation process. This significantly reduces manual effort, saves time for recruiters, and increases the overall efficiency of the hiring process.
- **Enhanced Candidate Evaluation:** The project improves the accuracy and objectivity of candidate evaluation by leveraging intelligent algorithms. The Naive Bayes-based vacancy prediction and NLP-based resume scoring provide standardized and objective criteria for assessing candidate qualifications and relevance to specific job requirements. The inclusion of a domain-specific MCQ test further enhances the evaluation process by assessing candidates' knowledge and expertise.
- **Comprehensive Evaluation Approach:** By combining multiple evaluation criteria, including resume scores, vacancy prediction, and MCQ test results, the project offers a comprehensive assessment of candidates. This holistic approach provides recruiters with a more complete picture of each candidate's suitability for a specific job vacancy, enabling them to make more informed decisions during the selection process.
- **Bias Reduction:** The Smart Resume Selector project aims to reduce bias in the hiring process. By leveraging objective evaluation criteria and automated

algorithms, the project minimizes subjective judgments and biases that may arise during manual resume screening. This contributes to a fairer and more equal opportunity for all candidates, regardless of personal characteristics or backgrounds.

- **Improved Hiring Decisions:** By automating and enhancing the candidate evaluation process, the project aims to improve the quality of hires. Recruiters can make more informed decisions based on standardized criteria and a comprehensive assessment of each candidate's qualifications, reducing the chances of costly hiring mistakes and increasing the likelihood of successful outcomes.

Overall, the Smart Resume Selector project's contributions lie in its ability to automate and streamline the hiring process, enhance candidate evaluation accuracy, provide a comprehensive evaluation approach, reduce bias, and improve the overall quality of hiring decisions. By incorporating advanced technologies and intelligent algorithms, the project brings significant advancements to the field of candidate selection, benefiting both recruiters and organizations in their talent acquisition efforts.

## 1.4 Report Organisation

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

# Chapter 2

## Literature Survey

In 2021, Nandhini S, Gomathi S, and Lavanya S published "Automated Resume Screening Using Natural Language Processing" in the International Journal of Advanced Research in Computer Science and Software Engineering. The study proposes an automated resume screening system that extracts data from resumes using NLP techniques and ranks them based on how well they match the job description."Resume Screening using Natural Language Processing and Machine Learning" was published by Kondapalli Sai Pranay in the International Journal of Current Technology and Engineering in 2020. The method outlined in the study uses NLP and machine learning to screen resumes and match them to job descriptions.

In 2019, "Automated Resume Screening System Using Machine Learning and Natural Language Processing" by Shweta Agrawal and Sumit Gupta was published in the International Journal of Innovative Technology and Exploratory Engineering. Resume Screening and Ranking using Machine Learning Techniques by A. Kumar and S. Sharma. This paper focuses on the application of machine learning techniques for resume screening and ranking. The authors discuss various machine learning algorithms, including Naive Bayes, Support Vector Machines, and Random Forest, applied to the task of resume analysis. They evaluate the performance of these algorithms using metrics such as accuracy, precision, and recall. The paper also explores feature selection methods to identify the most relevant resume attributes for effective screening. The findings highlight the effectiveness of machine learning techniques in automating the resume screening process and improving the efficiency of candidate selection.

Enhanced Resume Scoring using Natural Language Processing by B. Patel and R. Gupta. This paper investigates the use of natural language processing techniques to enhance resume scoring in the hiring process. The authors propose a framework that extracts various resume features, such as skills, experience, and education, using NLP algorithms. They discuss the importance of feature weighting and ranking methods to assign scores to resumes accurately. The paper also explores the incorporation of domain-specific knowledge and ontologies to improve the accuracy and relevance of resume scoring. The study demonstrates the efficacy of NLP-based approaches in providing a more comprehensive and informative assessment of candidate resumes.

# Chapter 3

## Methodology

### 3.1 Introduction

The methodology employed in the Smart Resume Selector project encompasses a series of steps designed to automate and enhance the resume screening process. The project combines Naive Bayes-based vacancy prediction, NLP-based resume scoring, and an MCQ test tailored to the job vacancy. The initial steps involve collecting a diverse dataset of job vacancies and resumes, followed by preprocessing the resumes to standardize their format. The Naive Bayes classifier is then employed to predict the most suitable job vacancy for each candidate, leveraging the extracted features from the resumes. Subsequently, NLP techniques are applied to analyze the resumes and assign scores based on the relevance and quality of the extracted information. Additionally, an MCQ test specifically designed for the job vacancy is administered to assess the candidates' knowledge and aptitude. The final step involves combining the resume scores and MCQ test results to generate rankings, facilitating the identification of the most qualified candidates. By incorporating these methodologies, the Smart Resume Selector project aims to streamline the hiring process, improve efficiency, and ensure a more objective evaluation of candidates.

#### 3.1.1 NLP (Natural Language Processing)

NLP (Natural Language Processing) is a subfield of artificial intelligence and linguistics that focuses on the interaction between computers and human language. It involves the development of algorithms and models that enable machines to understand, interpret, and generate human language in a meaningful way. NLP techniques are widely used in various applications, including text classification, sentiment analysis, machine translation, information extraction, and speech recognition.

In the context of the Smart Resume Selector project, NLP plays a crucial role in analyzing and extracting relevant information from resumes. By employing NLP techniques, the system can automatically process and understand the textual content of resumes, enabling the extraction of key details such as skills, education, experience, and certifications. These extracted features provide valuable insights into the qualifications and suitability of candidates for specific job vacancies.

NLP-based resume scoring further enhances the evaluation process by assigning scores to different aspects of a candidate's resume. By leveraging techniques such as named entity recognition, part-of-speech tagging, and syntactic parsing, the system can identify and evaluate the importance and relevance of various resume components. This allows for a more objective and comprehensive assessment of candidates, beyond simple keyword matching or manual review.

Moreover, NLP techniques can also be used to improve the overall user experience in the Smart Resume Selector system. For example, natural language understanding and dialogue systems can enable candidates to interact with the system using their own language, simplifying the application process and providing a more intuitive interface.

In summary, NLP is a fundamental component of the Smart Resume Selector project, empowering the system to effectively process, understand, and score resume content. By leveraging NLP techniques, the system can automate the analysis of resumes, extract meaningful information, and enhance the overall evaluation process. This enables recruiters to make more informed decisions and efficiently identify the most qualified candidates for specific job vacancies.

### 3.1.2 Naïve Bayes classifier

Naive Bayes is a popular probabilistic classification algorithm that is commonly used in machine learning and natural language processing tasks. It is based on Bayes' theorem and assumes independence among the features (hence the term "naive").

The general equation for Naive Bayes can be represented as follows:

$$P(C \mid X) = (P(X \mid C) * P(C)) / P(X)$$

Where:

- $P(C \mid X)$  is the posterior probability of class  $C$  given the input  $X$ .
- $P(X \mid C)$  is the likelihood of observing the input  $X$  given class  $C$ .
- $P(C)$  is the prior probability of class  $C$ .
- $P(X)$  is the probability of observing the input  $X$ .

In the context of the Smart Resume Selector project, Naive Bayes is used for vacancy prediction. The algorithm predicts the most suitable job vacancy for each candidate based on the features extracted from their resume. The features may include qualifications, skills, experience, and other relevant information.

To apply Naive Bayes, the algorithm first estimates the prior probabilities  $P(C)$  for each job vacancy category. It then calculates the likelihood  $P(X \mid C)$  by



assuming independence among the features. This means that the presence or absence of one feature does not affect the likelihood of other features.

Once the likelihoods and prior probabilities are estimated, the algorithm can determine the posterior probability  $P(C \mid X)$  for each job vacancy category. The candidate is then assigned to the job vacancy category with the highest posterior probability.

The Naive Bayes algorithm is known for its simplicity, efficiency, and ability to handle large feature spaces. However, its assumption of feature independence may not hold in all cases, and it may struggle with rare or unseen feature combinations. Despite these limitations, Naive Bayes remains a widely used and effective algorithm for various classification tasks, including vacancy prediction in the Smart Resume Selector project.

### 3.1.3 Developing Environment

- Front End : HTML, CSS, BOOTSTRAP
- Back End : MySQL, Python
- IDE : PyCharm community
- Operating System : Windows 8 or above

### 3.1.4 Module Description

#### Admin

- Login
- Verify company
- View complaint and send reply
- Manage Tips
- Manage Videos
- View feedbacks

#### Company

- Registration
- Login
- Manage vacancies
- Upload MCQ test question
- View result

- View ranking
- View Resume Score
- View candidate resume
- Send compliant and view reply

#### Candidate

- Registration
- Login
- View Matching Vacancy
- Apply for vacancy
- Upload resume
- Attend test
- View tips
- View videos
- Send feedback
- View result
- View resume score

## 3.2 Workflow

The workflow of the Smart Resume Selector project involves several key steps that integrate Naive Bayes-based vacancy prediction, NLP-based resume scoring, and an MCQ test to rank candidates based on a combination of resume score and test results. Here is an overview of the workflow:

1. Data Collection: The project begins with the collection of relevant data, including job vacancies and corresponding resumes. This data serves as the foundation for training and testing the system
2. Training Phase:
  - (a) Naive Bayes-based Vacancy Prediction: The Naive Bayes algorithm is trained using the collected data to learn the patterns and relationships between job vacancies and the features extracted from resumes. This enables the algorithm to predict the most suitable job vacancy category for a given candidate based on their resume.
  - (b) NLP-based Resume Scoring: NLP techniques are employed to analyze the textual content of resumes. Relevant information such as skills, education, experience, and achievements are extracted and assigned scores based on their importance and relevance to the job vacancy.

### 3. Candidate Evaluation

- (a) . Resume Scoring: Each candidate's resume is scored using the NLP-based approach developed in the training phase. The system evaluates the candidate's qualifications and suitability based on the extracted information and assigns a score to their resume.
- (b) MCQ Test: A tailored MCQ test is administered to each candidate, focusing on the specific job requirements and domain knowledge. The test assesses the candidate's understanding, problem-solving abilities, and critical thinking skills.

### 4. Ranking and Selection:

- (a) Combination of Resume Score and MCQ Test Results: The scores obtained from the resume evaluation and MCQ test are combined using a predefined weighting mechanism. The combination represents the overall performance of each candidate
  - (b) Ranking: Based on the combined scores, candidates are ranked in descending order. The top-ranked candidates are considered the most qualified and suitable for the job vacancy.
5. Candidate Shortlisting: Recruiters or hiring managers can then review the ranked list of candidates and shortlist those who meet the desired criteria and qualifications.

The workflow of the Smart Resume Selector project provides an automated and efficient approach to resume screening, vacancy prediction, and candidate ranking. By integrating Naive Bayes-based vacancy prediction, NLP-based resume scoring, and MCQ testing, the system significantly improves the efficiency and effectiveness of the hiring process, ensuring that the most qualified candidates are identified and considered for the job vacancy.

### 3.3 Data Flow Diagram

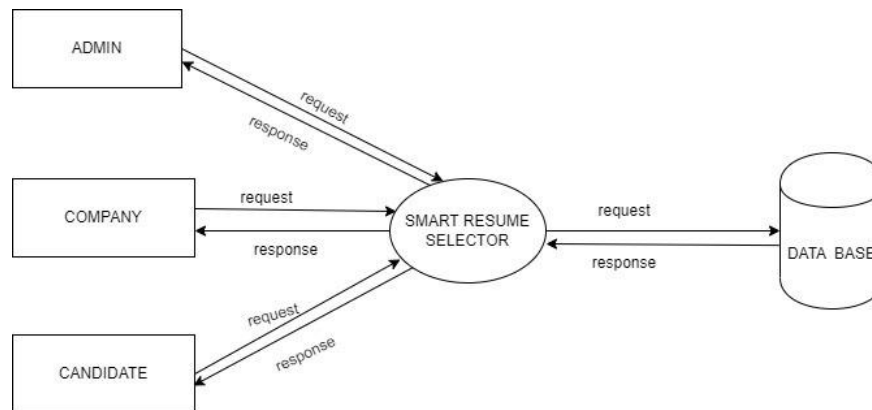


Figure 3.1: level 0

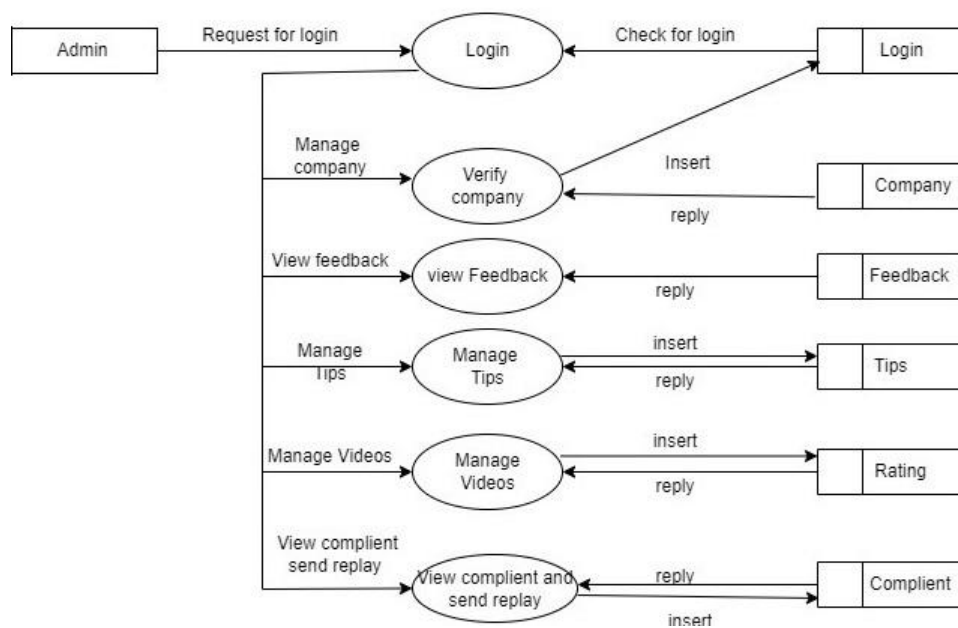


Figure 3.2: level 1

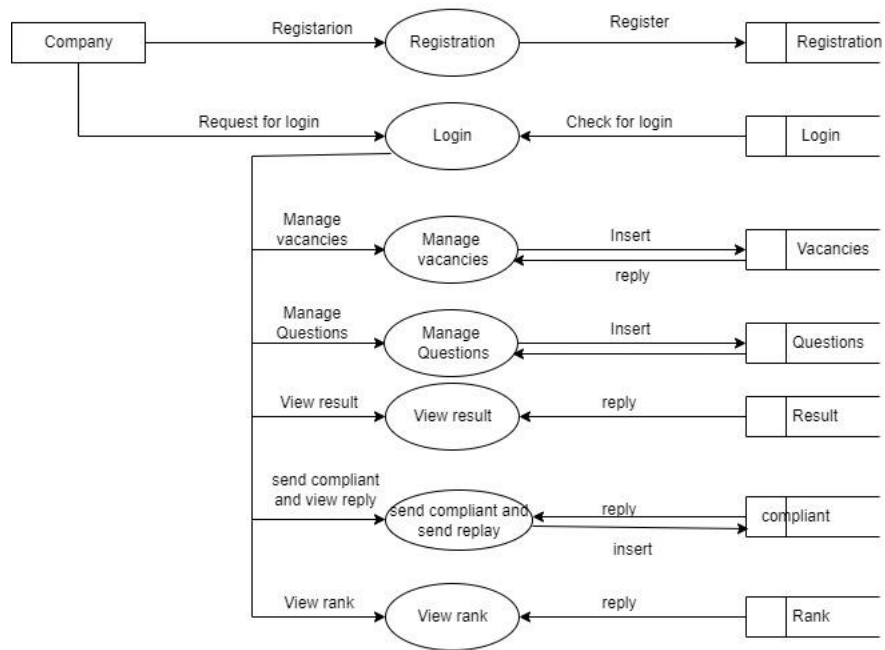


Figure 3.3: level 2

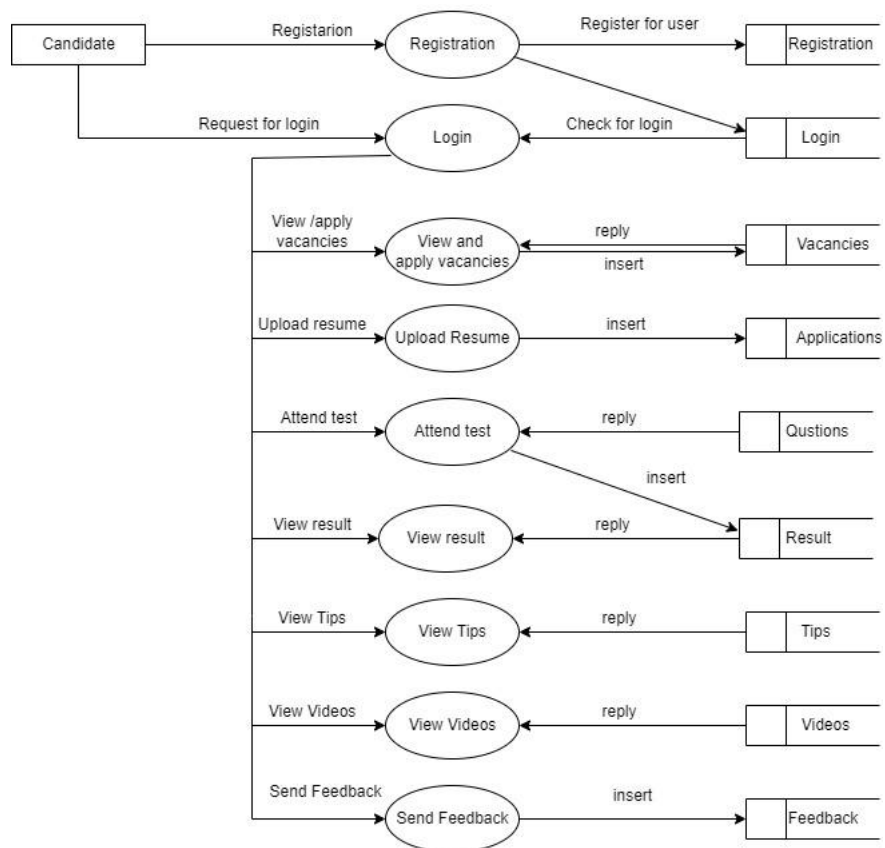


Figure 3.4: level 3

## 3.4 Implementation

The implementation of the Project involves several components and technologies to effectively integrate Naive Bayes-based vacancy prediction, NLP-based resume scoring, and the MCQ test. Here is an overview of the implementation process:

1. Data Preprocessing:

- Resumes: The collected resumes undergo preprocessing to clean and normalize the text. This may include removing special characters, converting text to lowercase, and tokenizing the text into words or phrases.
- Job Vacancies: The job vacancy data is also preprocessed to extract relevant features and create a structured representation of the vacancies.

2. Naive Bayes-based Vacancy Prediction:

- Training: The preprocessed job vacancy data is used to train the Naive Bayes classifier. The classifier learns the statistical patterns and relationships between job vacancy features and categories.
- Testing: During the testing phase, the trained classifier predicts the most suitable job vacancy category for each candidate's resume based on the extracted features.

3. NLP-based Resume Scoring:

- Feature Extraction: NLP techniques are employed to extract relevant features from the preprocessed resumes. This may involve techniques such as named entity recognition, part-of-speech tagging, and keyword extraction.
- Resume Scoring: The extracted features are then assigned scores based on their importance and relevance to the job vacancy. This can be achieved using predefined criteria or machine learning algorithms.

4. MCQ Test Administration:

- Designing the MCQ Test: The company designing the test creates a set of multiple-choice questions that assess the domain-specific knowledge, problem-solving abilities, and critical thinking skills required for the job vacancy.
- Test Administration: The software system provides an interface for the company to administer the MCQ test to candidates. The candidates can access the test, answer the questions, and submit their responses.

5. Ranking and Selection:

- Combination of Scores: The scores obtained from the resume evaluation and MCQ test are combined using a predefined weighting mechanism. The combined scores represent the overall performance of each candidate.

- Ranking: Based on the combined scores, the system ranks the candidates in descending order. Recruiters can review the ranked list and shortlist candidates based on their qualifications and suitability for the job vacancy..

The implementation of the Smart Resume Selector project requires coordination between software developers, data collection, and the company administering the MCQ test. By integrating Naive Bayes-based vacancy prediction, NLP-based resume scoring, and an MCQ test, the system provides a comprehensive and efficient approach to resume screening and candidate evaluation, ultimately simplifying and enhancing the hiring process for companies.

# Chapter 4

## Agile Methodology

### 4.1 Introduction

After the initial studies it is found that agile model of software development is suitable and is the best method for the development of this system. Agile methodology mainly focused on the client satisfaction through continuous delivery. Also it sets a minimum number of requirements and turns them in to a deliverable product. As this project has many individual requirements which can be delivered in parts and the user can gradually improve their work efficiency. Agile methodology has a family of methods of which scrum is selected for the development of this project. Scrum is process framework that has been used to manage complex product development. It is not a process or technique for building products rather it is a framework within which various processes can be employed. Also it is suitable method to support the development process. It focuses on lean software development and has in building better software effectively and efficiently. Agile is one of the most widely used and recognized software development framework. The methodology those experts agreed upon was described as ‘lightweight’ and fast. Agile is also about being the adaptive and continuous improvement, as much as it is about constant feedback and speed of delivery

“Agile is a software development approach where a self-sufficient and cross-functional team works on making continuous deliveries through iterations and evolves throughout the process by gathering feedback from the end users.” The major rules in scrum methodology are

- The product owner (PO) : Who represents the stakeholder and the business.
- The scrum master : Ensures the process followed, removes obstructions, and protects the development system
- Development team: Cross functional, self organizing team who actually do the actual analysis, design implementation and testing process.

They work together in iterative time boxed duration called sprints. The first step is the creation of the product backlog by the PO. It’s a to-do list of stuff to be done by the scrum team. Then the scrum team selects the top priority items and tries to finish them within the time box called a sprint. An easier way to remember all of this is to memorize the 3-3-5 framework. It means that a scrum project has 3 roles, 3 artifacts, and 5 events



These are:-

- Roles : Product Owner, Scrum Master, and development team.
- Artifacts : Product Backlog, Sprint Backlog and Product Increment.
- Events : Sprint, Sprint planning, Daily Scrum, Sprint review and Sprint retrospective

The framework begins with a simple premise start with what can be seen or known. After that the progress is tracked and tweak as necessary. The three pillars of scrum are transparency, inspection and adaptation. In scrum everyone has a role.

## 4.2 User Story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work — which drives collaboration, creativity, and a better product overall.

UserstoryID	As a type of user	I want to	So that I can
1	Admin	Login for admin	Login successful with correct username and password.
2	Admin	Verify Company	Verify the company details and approve for login.
3	Admin	View feedbacks	View feedback given by the users.
4	Admin	View complaints and send reply	View complaints of users and post reply.
5	Admin	Manage tips	Manage Tips add/delete

UserstoryID	As a type of user	I want to	So that I can
6	Admin	Manage videos	Manage Videos add/delete.
7	Company.	Registration for Company	Create a profile.
8	Company	Login for Company	Login successful with correct username and password.
9	Company	Manage vacancies	Add/delete job vacancies.
10	Company	Manage Test question	Add/delete Test question.
11	Company	Send complaint and view reply	send complaint and view reply.
12	Company	View result	View Test result of candidates.
13	Company	View Resume Score	View Resume Score of candidate.
14	Company	View ranking	View rank of candidates.
15	Company	View Candidate resume	View Candidate resume
16	Candidate	Registration for users	Create a profile.
17	Candidate	Login for Candidate	Login successful with correct username and password.
18	Candidate	View vacancies	View job vacancies.
19	Candidate	Apply for vacancies	Apply for vacancies.
20	Candidate	Add resume	Upload resume
21	Candidate	Attend test	Attend MCQ test.
22	Candidate	View test result	View MCQ test result.
23	Candidate	View tips	View tips from admin.
24	Candidate	View videos	View videos from admin.
25	Candidate	View Resume score	View Resume score for specific job.
26	Candidate	Send feedbacks	Send feedback to admin.

Table 4.1: User Story

### 4.3 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome. The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't

on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment. It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories. The product backlog of the system is given in Table

UserstoryID	Name	Priority (High/Medium/Low)
1	Login for admin	High
2	Verify Company	High
3	View feedbacks	Low
4	View complaints and send reply	High
5	Manage Tips	Low
6	Manage Videos	Low
7	Company Registrationr	High
8	Login for company	High
9	Manage vacancies	High
10	Manage Test question	High
11	Send complaint and view re- ply	Low
12	View MCQ test result	High
13	View resume score	High
14	View ranking	High
15	View candidate resume	Medium
16	Candidate Registration	High
17	Login for Candidate	High
18	View vacanciese	Medium
19	Apply for vacancies	Medium
20	Upload resume	Medium
21	Attend test	Medium
22	View test result	Medium
23	View tips	Low
24	View videos	low
25	View resume score	High
26	Send feedback	Low

Table 4.2: Product Backlog

## 4.4 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it.

UserstoryID	Sprint	StartDate	EndDate	Day	Status
1 2 3 4 5 6 7	sprint 1	07/02/2023	03/03/2023	25	Completed
8 9 10 11 12 13 14	sprint 2	04/03/2023	23/03/2023	20	Completed
15 16 17 18 19 20	Sprint 3	24/03/2023	16/04/2023	24	Completed
21 22 23 24 25 26	Sprint 4	17/04/2023	15/05/2023	29	Completed

Table 4.3: Project plan

## 4.5 Sprint Backlog (Plan)

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

Backlog item	Status and completion date	original Estimate (hours)	Day 1 07/02	Day 2 08/02	Day 3 09/02	Day 4 10/02	Day 5 11/02	Day 6 12/02	Day 7 13/02	Day 8 14/02	Day 9 15/02	Day 10 16/02	Day 11 17/02	Day 12 18/02	Day 13 19/02	Day 14 20/02	Day 15 21/02	Day 16 22/02	Day 17 23/02	Day 18 24/02	Day 19 25/02	Day 20 26/02	Day 21 27/02	Day 22 28/02	Day 23 01/03	Day 24 02/03	Day 25 03/03
UI designing	10/02/23	4	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coding	21/02/23	8	0	0	0	0	1	2	0	2	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Testing	03/02/23	6	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2	2	0	0	0	0	0	0	0	0
Total		18	2	1	1	0	1	2	0	2	0	2	1	1	0	1	0	2	2	0	0	0	0	0	0	0	0

Table 4.4: Sprint Backlog (Plan)-Sprint 1

Backlog item	Status and completion date	original Estimate (hours)	Day 1 06/03	Day 2 07/03	Day 3 08/03	Day 4 09/03	Day 5 10/03	Day 6 11/03	Day 7 12/03	Day 8 13/03	Day 9 14/03	Day 10 15/03	Day 11 16/03	Day 12 17/03	Day 13 18/03	Day 14 19/03	Day 15 20/03	Day 16 21/03	Day 17 22/03	Day 18 23/03
UI designing	08/03/23	4	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coding	17/03/23	12	0	0	0	3	1	2	0	3	1	2	0	0	0	0	0	0	0	0
Testing	23/03/23	4	0	0	0	0	0	0	0	0	0	0	1	1	0	2	0	0	0	0
Total		20	1	2	1	3	1	2	0	3	1	2	1	1	0	2	0	0	0	0

Table 4.5: Sprint Backlog (Plan)-Sprint 2

Backlog Item	Status Completi on Date	Original Estimate Date (Hours)	Day1 24/0 3	Day 2 26/0 3	Day3 28/0 3	Day 4 30/ 03	Day 5 03/ 04	Day 6 05/0 4	Day 7 06/0 4	Day 8 07/ 04	Day 9 08/0 4	Day 10 10/ 04	Day 11 11/ 04	Day 12 13/ 04	Da y13 14/ 04	Day 14 16/ 04
UI designing	26/03/23	4	2	2	0	0	0	0	0	0	0	0	0	0	0	0
Coding	07/04/23	16	2	2	2	2	2	2	2	2	0	0	0	0	0	0
Testing	16/04/23	8	0	0	0	0	0	0	0	0	1	0	2	0	1	2
Total		28	4	4	2	2	2	2	2	2	1	0	2	0	1	2

Table 4.6: Sprint Backlog (Plan)-Sprint 3

Backlog Item	Status Completi on Date	Original Estimate Date (Hours)	Day1 18/0 4	Day 2 19/0 4	Day3 22/0 4	Day 4 24/ 04	Day 5 26/ 04	Day 6 27/0 4	Day 7 28/0 4	Day 8 02/ 05	Day 9 04/0 5	Day 10 06/ 05	Day 11 09/ 05	Day 12 11/ 05	Da y13 13/ 05	Day 14 15/ 05
UI designing	24/03/23	8	2	2	2	2	0	0	0	0	0	0	0	0	0	0
Coding	10/05/23	18	2	2	2	2	2	2	2	2	2	0	0	0	0	0
Testing	15/05/23	8	0	0	0	0	0	0	0	0	1	0	2	0	1	2
Total		34	4	4	4	4	4	2	2	2	3	2	2	0	1	2

Table 4.7: Sprint Backlog (Plan)-Sprint 4

## 4.6 Sprint Backlog (Actual)

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below.

Backlog item	Status and completion date	original Estimate (hours)	Day 1 07/02	Day 2 08/02	Day 3 09/02	Day 4 10/02	Day 5 11/02	Day 6 12/02	Day 7 13/02	Day 8 14/02	Day 9 15/02	Day 10 16/02	Day 11 17/02	Day 12 18/02	Day 13 19/02	Day 14 20/02	Day 15 21/02	Day 16 22/02	Day 17 23/02	Day 18 24/02	Day 19 25/02	Day 20 26/02
UI designing	10/02/23	4	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coding	21/02/23	8	0	0	0	0	1	2	0	2	0	2	1	0	0	0	0	0	0	0	0	0
Testing	03/02/23	6	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2	2	0	0	0
Total		18	2	1	1	0	1	2	0	2	0	2	1	1	0	1	0	2	2	0	0	0

Table 4.8: Sprint Backlog (Actual)-Sprint 1

Backlog item	Status and completion date	original Estimate (hours)	Day 1 06/03	Day 2 07/03	Day 3 08/03	Day 4 09/03	Day 5 10/03	Day 6 11/03	Day 7 12/03	Day 8 13/03	Day 9 14/03	Day 10 15/03	Day 11 16/03	Day 12 17/03	Day 13 18/03	Day 14 19/03
UI designing	08/03/23	4	1	2	1	0	0	0	0	0	0	0	0	0	0	0
Coding	17/03/23	12	0	0	0	3	1	2	0	3	1	2	0	0	0	0
Testing	23/03/23	4	0	0	0	0	0	0	0	0	0	0	1	1	0	2
Total		20	1	2	1	3	1	2	0	3	1	2	1	1	0	2

Table 4.9: Sprint Backlog (Actual)-Sprint 2

Backlog Item	Status Completi on Date	Original Estimate Date (Hours)	Day1 24/0 3	Day 2 26/0 3	Day3 28/0 3	Day 4 30/ 03	Day 5 03/ 04	Day 6 05/0 4	Day 7 06/0 4	Day 8 07/ 04	Day 9 08/0 4	Day 10 10/ 04	Day 11 11/ 04	Day 12 13/ 04	Da y13 14/ 04	Day 14 16/ 04
UI designing	26/03/23	4	2	2	0	0	0	0	0	0	0	0	0	0	0	0
Coding	07/04/23	16	2	2	2	2	2	2	2	2	0	0	0	0	0	0
Testing	16/04/23	8	0	0	0	0	0	0	0	0	1	0	2	0	1	2
Total		28	4	4	2	2	2	2	2	2	1	0	2	0	1	2

Table 4.10: Sprint Backlog (Actual)-Sprint 3

Backlog Item	Status Completi on Date	Original Estimate Date (Hours)	Day1 18/0 4	Day 2 19/0 4	Day3 22/0 4	Day 4 24/ 04	Day 5 26/ 04	Day 6 27/0 4	Day 7 28/0 4	Day 8 02/ 05	Day 9 04/0 5	Day 10 06/ 05	Day 11 09/ 05	Day 12 11/ 05	Da y13 13/ 05	Day 14 15/ 05
UI designing	24/03/23	8	2	2	2	2	0	0	0	0	0	0	0	0	0	0
Coding	10/05/23	18	2	2	2	2	2	2	2	2	2	0	0	0	0	0
Testing	15/05/23	8	0	0	0	0	0	0	0	0	1	0	2	0	1	2
Total		34	4	4	4	4	4	2	2	2	3	2	2	0	1	2

Table 4.11: Sprint Backlog (Actual)-Sprint 4



## 4.7 Product Backlog Review

### REVIEW FORM SPRINT 1

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
1	A single login form for admin,candidate ,company	easy login.
2	Unverified companies cannot be entered into the sytem.	it ensures system security.
3	View feedbacks from candidate.	feedback monitoring in goodmanner.
4	view compliants to be sent.	complaints can be monitered in good manner.
5	Share tips for candidates.	easy monitoring.
6	Share videos for candidates.	easy monitoring.
7	Add few more details to company registration.	easy verification for company.

Table 4.12: Product backlog review sprint 1

### REVIEW FORM SPRINT 2

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
8	A single login form for admin,candidate ,company	easy login.
9	Verified company can manage vacancy	easy access.
10	Add test questions as MCQ	easy monitoring.
11	Complaints from company can only vew admin.	easily reply to complaints.
12	View test result by candidate name	easy monitoring.
13	Calculate resume score out of hundred	easy analysing.
14	Prepare rank by combining resume score and test result	easy analysing.

Table 4.13: Product backlog review sprint 2

## REVIEW FORM SPRINT 3

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
15	Company can download candidate resume	easy analysing.
16	Add few more details to candidate registration.	easy verification for candidate.
17	A single login form for admin, candidate, company	easy login.
18	View matching vacancies based on resume	easily view vacancies
19	Apply for vacancy	easy access.
20	Upload resume in pdf	easy analysing resume.

Table 4.14: Product backlog review sprint 3

## REVIEW FORM SPRINT 4

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
21	Candidate can attend MCQ test	easy access.
22	Candidate can view MCQ test result	Test result monitoring in good manner.
23	View all tips added by admin can view all candidate.	easy access to tips for every candidate
24	View all videos added by admin can view all candidate.	easy access to videos for every candidate
25	Candidate can view uploaded resume score for each vacancy	resume score monitoring in good manner
26	Feedback from candidate can only view admin.	easily View feedback.

Table 4.15: Product backlog review sprint 4

## 4.8 Sprint Review

At the end of each sprint a Sprint Review meeting is held. During this meeting the Scrum Team shows which Scrum Product Backlog items they completed (according to the Definition of Done) during the sprint. This might take place in the form of a demo of the new features. Backlog items that are not completed shall not be demonstrated. Otherwise this might suggest that these items are finished as well. Instead incomplete items/remaining activities shall be taken back into the Scrum Product Backlog, re-estimated and completed in one of the following sprints. The Sprint Review meeting should be kept very informal. No PowerPoint slides should be used and time for preparation and performing the meeting should be limited. During the meeting the Scrum Product Owner inspects the implemented backlog entries and accepts the solution or adds new stories to the Scrum Product Backlog to adapt the functionality. Participants in the sprint review typically include the Scrum Product Owner, the Scrum Team and the Scrum Master. Additionally management, customers, and developers from other projects might participate as well.

REVIEW FORM SPRINT 1

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
1	A single login form for admin,candidate ,company	easy login.
2	Unverified companys cannot be entered into the sytem.	it ensures system security.
3	View feedbacks from candidate.	feedback monitoring in goodmanner.
4	Only view compliants to be sent.	complaints can be monitered in good manner.
5	Share tips for smart resume selector.	easy monitoring.
6	Share videos for smart resume selector.	easy monitoring.
7	Add few more details to company registration.	easy verification for company.

Table 4.16: Sprint review sprint 1

## REVIEW FORM SPRINT 2

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
8	A single login form for admin,candidate ,company	easy login.
9	Verified company can manage vacancy	easy access.
10	Add test questions as MCQ	easy monitoring.
11	Complaints from company can only view admin.	easily reply to complaints.
12	View test result by candidate name	easy monitoring.
13	Calculate resume score out of hundred	easy analysing.
14	Prepare rank by combining resume score and test result	easy analysing.

Table 4.17: Sprint review sprint 2

## REVIEW FORM SPRINT 3

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
15	Company can download candidate resume	easy analysing.
16	Add few more details to candidate registration.	easy verification for candidate.
17	A single login form for admin,candidate ,company	easy login.
18	View matching vacancies based on resume	easily view vacancies
19	Apply for vacancy	easy access.
20	Upload resume in pdf	easy analysing resume.

Table 4.18: Sprint review sprint 3

## REVIEW FORM SPRINT 4

UserstoryID	Comments From Scrum master if any	Comments From Product Owner if any
21	Candidate can attend MCQ test	easy access.
22	vCandidate can view MCQ test result	Test result monitoring in good manner.
23	View all tips added by admin can view all candidate.	easy access to tips for every candidate
24	View all videos added by admin can view all candidate.	easy access to videos for every candidate
25	Candidate can view uploaded resume score for each vacancy	resume score monitoring in good manner
26	Feedback from candidate can only view admin.	easily View feedback.

Table 4.19: Sprint review sprint 4

## 4.9 Testing and Validation

According to Agile approaches, a working product is supposed to be delivered at the end of the Sprint, which means that it is tested and accepted by the Product Owner: from a testing perspective this means that all tests (unit test, system test, acceptance test) take place during the Sprint.

### SPRINT 1

Test	Date	Action	Expected Result	Actual Result	Pass ? Yes/ No
1	21/02/23	Login for admin	Admin should login into their accounts.	Admin can successfully login into their accounts.	yes
2	24/02/23	Verify/Block registered companies	Admin should block or unblock users.	Successfully Block or unblock registered users.	yes
3	29/02/23	View feed-backs	Admin should View feedbacks of candidate.	Successfully view feedbacks.	yes
4	27/02/23	View complaints and send reply	Admin should View complaints of companies and post reply.	Successfully view complaints of companies and post reply.	yes
5	28/03/23	manage Tips.	Admin should manage tips.	Successfully manage tips.	yes
6	01/03/23	manage videos.	Admin should manage videos.	Successfully manage videos.	yes
7	02/03/23	Company registration	Company should create accounts.	Company can successfully create profile.	yes

Table 4.20: Test Sprint 1

## SPRINT 2

Test	Date	Action	Expected Result	Actual Result	Pass ? Yes/ No
8	18/03/23	Login for company	company should login into their accounts.	company can successfully login into their accounts.	yes
9	20/03/23	Manage Vacancy details	Company should Manage vacancy details.	Successfully manage vacancy details.	yes
10	20/03/23	Manage MCQ test details	Company should Manage MCQ test details.	Successfully manage MCQ test details.	yes
11	22/03/23	Manage complaint	Send complaints and view the responses of complaint.	Successfully send complaint and view response..	yes
12	23/03/23	View MCQ test result	Company should view the result of MCQ test.	Company can successfully view MCQ test result.	yes
13	22/03/23	View resume score.	Company should view the resume score of candidate.	Successfully view the resume score.	yes
14	23/03/23	View candidate rank .	Company should view the candidate rank .	Successfully view the rank .	yes

Table 4.21: Test Sprint 2

## SPRINT 3

Test	Date	Action	Expected Result	Actual Result	Pass ? Yes/ No
15	07/04/23	View candidate resume	company should view candidate resume.	successfully view the resume.	yes
16	07/04/23	Registration	Candidate should create accounts.	Candidate can successfully create profile.	yes
17	10/04/23	Login	Candidate should login into their account.	Login successful with correct username and password.	yes
18	13/04/23	View vacancy	View the fitness vacancy added by the companies .	Successfully view vacancy details.	yes
19	14/04/23	Apply for vacancy.	Candidate can apply for vacancy.	successfully applied for vacancy.	yes
20	16/04/23	Upload resume	candidate can upload resume .	resume successfully uploaded.	yes

Table 4.22: Test Sprint 3



## SPRINT 4

Test	Date	Action	Expected Result	Actual Result	Pass ? Yes/ No
21	10/05/23	Attend MCQ Tesst	candidate can attend the MCQ test.	Successfully attend MCQ test.	yes
22	11/05/23	View test result.	candidate can view test result	Successfully view test result.	yes
23	12/05/23	View tips	candidate can view all tips posted by admin.	Successfully view tips.	yes
24	13/05/23	View Videos	candidate can play all videos posted by admin.	Successfully view videos.	yes
25	15/05/23	view re-sume score	candidate can view uploaded resume score	successfully view resume score	yes
26	13/05/23	Send feed-backs	On the basis of their experience post feedbacks	Successfully send feedbacks.	yes

Table 4.23: Test Sprint 4

## 4.10 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. To show the continuous development of the project the Gitlab histories is shown in Appendix from(Fig C.1 to Fig C.2).

# Chapter 5

## Conclusion

In conclusion, the Smart Resume Selector project presents a powerful and innovative solution to revolutionize the traditional hiring process. By incorporating Naive Bayes-based vacancy prediction, NLP-based resume scoring, and an MCQ test, the system enables recruiters to efficiently screen resumes, predict job vacancies, and rank candidates based on a combination of resume scores and test results.

The implementation of the Smart Resume Selector project offers several key advantages. Firstly, it saves significant time and resources by automating the initial screening process, allowing recruiters to focus their efforts on evaluating the most qualified candidates. Secondly, the integration of NLP-based resume scoring ensures a more objective evaluation of candidates, going beyond simple keyword matching to extract and assess the relevant information from resumes. Thirdly, the MCQ test tailored to each job vacancy provides a comprehensive assessment of candidates' domain-specific knowledge and problem-solving abilities, further enhancing the accuracy of candidate evaluation.

Overall, the Smart Resume Selector project has the potential to transform the hiring process, making it more efficient, objective, and streamlined. By leveraging advanced technologies, recruiters can make better-informed decisions and identify the most suitable candidates for specific job vacancies. The project opens new avenues for enhancing recruitment practices and improving the overall quality and effectiveness of the hiring process.

# BIBLIOGRAPHY

- [1] Nandhini S, Gomathi S, Lavanya S , Automated Resume Screening Using Natural Language Processing International Journal of Advanced Research in Computer Science and Software Engineering
- [2] Shweta Agrawal , Sumit Gupta , Automated Resume Screening System Using Machine Learning and Natural Language Processing Engineering. International Journal of Innovative Technology and Exploratory
- [3] Ms. Y. Sowjanya , Mareddy Keerthana , Pulluri Suneeksha Dorgipati Sai Sri Harsha, *Smart Resume Analyser*. International Journal of Research in Engineering and Science (IJRES)
- [4] Patrick Nyanumba Mwaro , Dr. Kennedy Ogada , Prof. Wilson Cheruiyot *Applicability of Naïve Bayes Model for Automatic Resume Classification*. International Journal of Computer Applications Technology and Research

# Appendix A

## Source Code

```
import operator
import time
from datetime import datetime
from django.contrib import auth
from django.contrib.auth.decorators import login_required
from django.core.files.storage import FileSystemStorage
from django.http import HttpResponse
from django.shortcuts import render, redirect
from .predictionfile_nb import predictfn
    from pyresparser import ResumeParser
from pdfminer3.layout import LAParams, LTTextBox
from pdfminer3.pdfpage import PDFPage
from pdfminer3.pdfinterp import PDFResourceManager
from pdfminer3.pdfinterp import PDFPageInterpreter
from pdfminer3.converter import TextConverter
import io, random
from selector.dt import predict1
from selector.models import candidate, login
from selector.models import *
    Create your views here.
def main(request):
    return render(request,"loginindex.html")
def logincode(request):
    uname=request.POST['username']
    password=request.POST['pwd']
    try:
        ob=login.objects.get(username=uname,password=password)
        if ob.utype == 'admin':
            ob1=auth.authenticate(username='admin',password='admin')
            if ob1 is not None:
                auth.login(request,ob1)
                return HttpResponse('''<script>alert("welcome admin");
window.location='/adminhome'</script>''')
            elif ob.utype == 'verified':
                request.session['lid']=ob.id
```

```

        ob1 = auth.authenticate(username='admin', password='admin')
        if ob1 is not None:
            auth.login(request, ob1)
            return HttpResponseRedirect(''''<script>alert("welcome company");
window.location='/cmphome'</script>''')
        elif ob.utype == 'candidate':
            request.session['lid'] = ob.id
            request.session['cnt'] = 0
            ob1 = auth.authenticate(username='admin', password='admin')
            if ob1 is not None:
                auth.login(request, ob1)
                return HttpResponseRedirect(''''<script>alert("welcome candidate")
;window.location='/cndhome'</script>''')
            elif ob.utype == 'careerguidance':
                request.session['lid'] = ob.id
                ob1 = auth.authenticate(username='admin', password='admin')
                if ob1 is not None:
                    auth.login(request, ob1)
                    return HttpResponseRedirect(''''<script>alert("welcome
careerguidance");window.location='/careerhome'</script>''')
            else:
                return HttpResponseRedirect(''''<script>alert("invalid username or password");
window.location='/'</script>''')
        except:
            return HttpResponseRedirect(''''<script>alert("invalid username or password");
window.location='/'</script>''')
    /reply.html",{ 'val':ob})
@login_required(login_url='/')
def uploadresume(request):
    ob1=applied.objects.filter(candidate_id__lid__id=request.session['lid'],
vid__id=request.session['vid'])
    if len(ob1) == 0:
        obv=vacancy.objects.get(id=request.session['vid'])
        req=obv.requirments.lower().replace("\r","").split('\n')
        print(req,"=====")
        resume = request.FILES['file']
        userexp=request.POST['userexp']
        if obv.experiance <= int(userexp):
            fs = FileSystemStorage()
            fp = fs.save(resume.name, resume)
        save_image_path = "media/"+fp
        resume_text = pdf_reader(save_image_path).lower()
        resume_score = 0
        print("RT \n", resume_text)
        if 'objective' or 'OBJECTIVE' in resume_text:
            resume_score = resume_score + 5
            a = '[+] Awesome! You have added Objective'
        else:

```

```

        a = '[-] According to our recommendation please add your
career objective, it will give your career intension to the Recruiters.'
if 'declaration' in resume_text or 'DECLARATION' in resume_text:
    resume_score = resume_score + 5
    b = '[+] Awesome! You have added Declaration'
    else:
    b = '[+] According to our recommendation please add Declaration.
    obb=vacancy.objects.get(id=request.session['vid'])
    rcs=60/len(req)
count=0
    for cr in req:
        if cr in resume_text :
            resume_score = resume_score + rcs
            h = '[+] Awesome! You have added your requirements'
            count=count+1
        else:
            h = '[-] According to our recommendation  add requirements '
            h = '[+] You have '+str(count)+' requirements matches '
if 'hobbies' in resume_text or 'HOBBIES' in resume_text or
'Interests' in resume_text or 'INTERESTS' in resume_text:
    resume_score = resume_score + 5 c = '[+] You have added your Hobbies'
    else:
    c = '[-] According to our recommendation please add Hobbies
It will show your persnality
    if 'achievements' in resume_text or 'ACHIEVEMENTS' in resume_text:
    resume_score = resume_score + 10 d = '[+] added your Achievements
    else:
d = '[-] According to our recommendation please add Achievements
It will show that you are capable for the required position.'

        if 'projects' in resume_text or 'project' in resume_text:
            resume_score = resume_score + 15
            e = '[+] Awesome! You have added your Projects'

        f = '**Resume Score'
        score = resume_score
    ob = applied()
        ob.predicted_score=int(score)
        ob.status='predicted'
        ob.resume = fp
    ob.candidate_id = candidate.objects.get(lid_id=request.session['lid'])
        ob.vid = vacancy.objects.get(id=request.session['vid'])
        ob.save()
        return HttpResponse(''<script>alert("uploaded");
window.location='/viewvacancy'</script> ''')
    else:
        return HttpResponse(''<script>alert("The application doesnot
met the minimum Experiance");

```

```
window.location='/viewvacancy'</script> ''')
    else:
        return HttpResponse(''<script>alert("Already applied");
window.location='/viewvacancy'</script> ''')

@login_required(login_url='/')
def uploadresume1(request):

    resume = request.FILES['file']
    userexp=request.POST['userexp']
    fs = FileSystemStorage()
    fp = fs.save(resume.name, resume)

    save_image_path = "media/"+fp
    resume_text = pdf_reader(save_image_path).lower()
    res=predictfn(resume_text)
    ob=vacancy.objects.filter(vacancy=res,experiance__lte=int(userexp))
    return render(request, "candidate/predjob.html", {'val': ob})

def logout(request):
    auth.logout(request)
    return HttpResponse(''<script>alert("logout")\newline;
window.location='/'</script> ''')
```

# Appendix B

## Database Design

Name	Type	
lid	int	Primary key
username	varchar	not null
password	varchar	not null
type	varchar	not null

Table B.1: Login

Name	Type	
tid	int	Primary key
cname	varchar	notnull
place	varchar	notnull
post	varchar	notnull
mail	varchar	notnull
lid	int	foreign key

Table B.2: Company

Name	Type	
lid	int	foreign key
id	int	primary key
name	varchar	not null
place	varchar	not null
gender	varchar	not null
addressr	varchar	not null
phoner	int	not null
mailr	varchar	not null

Table B.3: Candidate



Name	Type	
id	int	primary key
cid	int	foreign key
vacancy	varchar	not null
requirments	varchar	not null
novacancy	int	not null
experiance	int	not null

Table B.4: vacancy

Name	Type	
id	int	primary key
candidateid	int	foreign key
vid	int	foreign key
resume	varchar	not null
predictedstore	int	not null

Table B.5: Applications

Name	Type	
id	int	primary key
vid	int	foreign key
question	varchar	not null
option1	varchar	not null
option2	varchar	not null
option3	varchar	not null
option4	varchar	not null
answer	varchar	not null

Table B.6: testquestion

Name	Type	
id	int	primary key
questionid	int	foreign key
candidateid	int	foreign key
mark	int	not null
date	date	not null

Table B.7: testresult

Name	Type	
id	int	primary key
complaint	varchar	notnull
reply	varchar	notnull
cid	int	foreign key
date	date	not null

Table B.8: complaint

Name	Type	
id	int	primary key
feedback	varchar	notnull
cid	int	foreign key

Table B.9: feedback

Name	Type	
id	int	primary key
tips	varchar	notnull
date	date	notnull

Table B.10: tips

Name	Type	
id	int	primary key
videos	varchar	notnull
date	date	notnull

Table B.11: videos

# Appendix C

## Output

CANDIDATE REGISTRATION	
Candidate Name	<input type="text"/>
Place	<input type="text"/>
Address	<input type="text"/>
Gender	<input type="radio"/> male <input type="radio"/> female
Phone	<input type="text"/>
Email	<input type="text"/>
Username	<input type="text"/>
Password	<input type="text"/>
<input type="button" value="SUBMIT"/>	

Figure C.1: Candidate Registration

Company Registration	
Company Name	<input type="text"/>
Place	<input type="text"/>
PIN/POST	<input type="text"/>
Phone	<input type="text"/>
Email	<input type="text"/>
Username	<input type="text"/>
password	<input type="text"/>
<input type="button" value="SUBMIT"/>	

Figure C.2: Company Registration

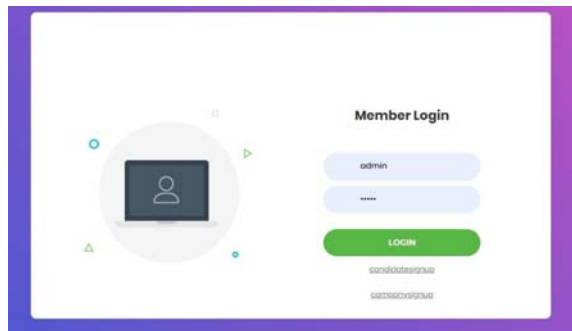


Figure C.3: Login

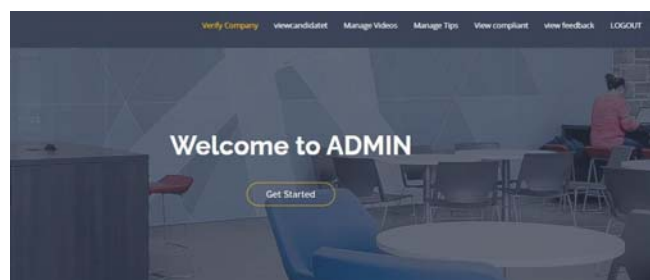


Figure C.4: Admin Home

ADMIN						
Verify Company						
COMPANY_ID	COMPANY NAME	PLACE	MAIL	Post	Approve/Reject	
1	ics	Kochi	ics@gmail.com	6765m	verified	
3	abc	malappuram	abc@gmail.com	676506	verified	
10	abc	malappuram	abct@gmail.com	pazhamalur	verified	
11	abctech	malappuram	abctech	pazhamalur	verified	
12	junaid	kuttipuram	junaid@123	6764546	verified	

Figure C.5: Verify Company

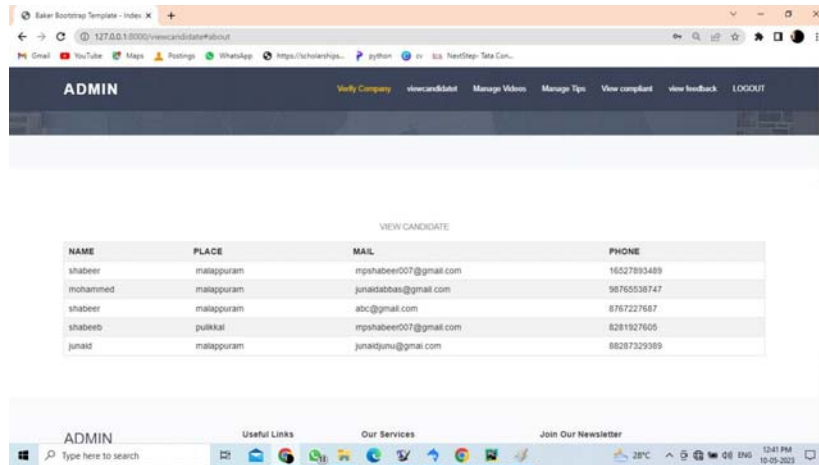


Figure C.6: View Candidate

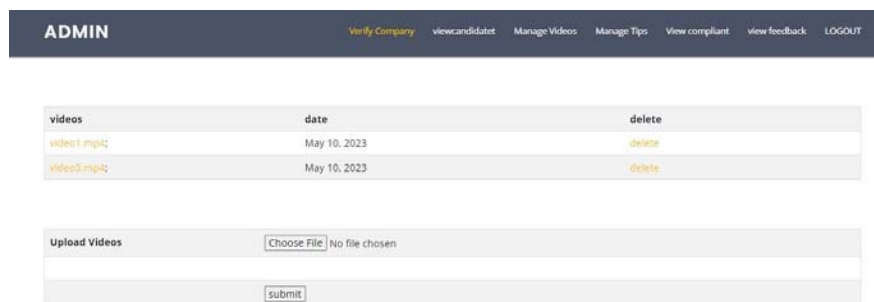


Figure C.7: Manage Videos

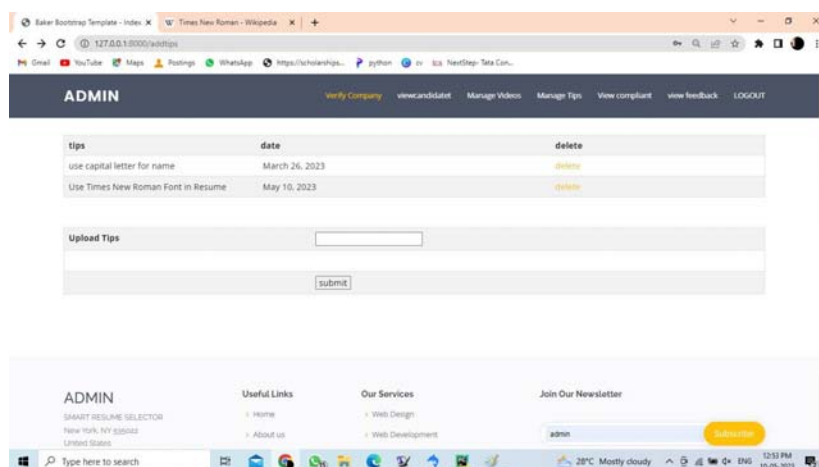


Figure C.8: Manage Tips

ADMIN			
<a href="#">Verify Company</a> <a href="#">view candidate</a> <a href="#">Manage Videos</a> <a href="#">Manage Tips</a> <a href="#">View compliant</a> <a href="#">view feedback</a> <a href="#">LOGOUT</a>			
View Complaint			
company name	Compliant	Date	reply
abc	admin	March 28, 2023	
abc	cant login some times	March 28, 2023	we are trying to solve the problem
tcs	we caant login at some times	April 7, 2023	<a href="#">reply</a>
abctech	system not work properly	April 26, 2023	<a href="#">reply</a>

Figure C.9: View Compliant

ADMIN	
<a href="#">Verify Company</a> <a href="#">view candidate</a> <a href="#">Manage Videos</a> <a href="#">Manage Tips</a> <a href="#">View compliant</a> <a href="#">view feedback</a> <a href="#">LOGOUT</a>	
candidate name	Feedback
shabeer	okay
shabeer	I got placed at tcs

Figure C.10: View Feedback

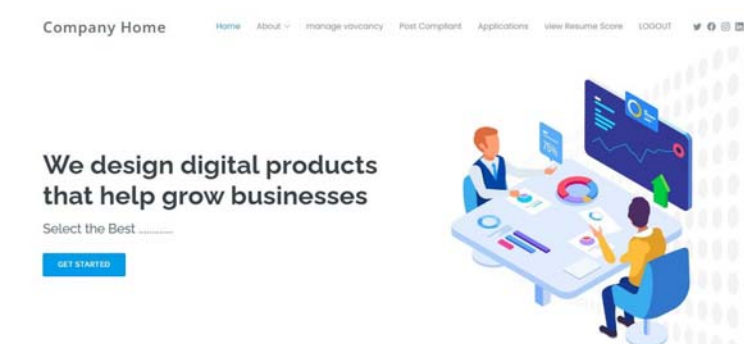


Figure C.11: Company Home

Company Home						
<a href="#">Home</a> <a href="#">About</a> <a href="#">manage vacancy</a> <a href="#">Post Complaint</a> <a href="#">Applications</a> <a href="#">view Resume Score</a> <a href="#">LOGOUT</a>						
vaccancy	No of Vaccancy	Experience	Requirements	delete	Add Question / view Question	view result
software engineer	6	1	java Html sql	DELETE	<a href="#">ADD QUESTION</a> <a href="#">Manage QUESTION</a>	<a href="#">View Test Result</a>
dot net developer	4	1	Dot net, c#, c	DELETE	<a href="#">ADD QUESTION</a> <a href="#">Manage QUESTION</a>	<a href="#">View Test Result</a>
software engineer	3	1	Js, oops	DELETE	<a href="#">ADD QUESTION</a> <a href="#">Manage QUESTION</a>	<a href="#">View Test Result</a>
django dev	3	3	Python C# Html	DELETE	<a href="#">ADD QUESTION</a> <a href="#">Manage QUESTION</a>	<a href="#">View Test Result</a>
data science	5	2	R, SQL, Data Visualization	DELETE	<a href="#">ADD QUESTION</a> <a href="#">Manage QUESTION</a>	<a href="#">View Test Result</a>
data science	7	5	R, SQL	DELETE	<a href="#">ADD QUESTION</a> <a href="#">Manage QUESTION</a>	<a href="#">View Test Result</a>
<a href="#">ADD VACANCY</a>						

Figure C.12: Manage Vacancy

The screenshot shows the 'Add Vacancy' form in the 'Company Home' dashboard. The form includes input fields for 'Vacancy', 'Requirements', 'Year of Experience', and 'No of Vacancy'. A 'Submit' button is at the bottom left. A message 'Please fill out this field' is displayed next to the 'No of Vacancy' field.

Figure C.13: Add Vacancy

The screenshot shows the 'Add Question' form in the 'Company Home' dashboard. The form includes a 'VACANCY' dropdown menu set to 'software engineer', and input fields for 'Question', 'Option1', 'Option2', 'Option3', 'Option4', and 'Answer'. A 'Submit' button is at the bottom left.

Figure C.14: Add Questions

The screenshot shows the 'Post Compliant' form in the 'Company Home' dashboard. It includes a table with columns: 'Compliant', 'Date', 'View Reply', 'reply', and 'Action'. Below the table is a 'Post Compliant' input field and a 'Submit' button.

Compliant	Date	View Reply	reply	Action
we cant login at some times	April 7, 2023		pending	<a href="#">delete</a>

Figure C.15: Post Compliant

The screenshot shows the 'View Resume Score' table in the 'Company Home' dashboard. The table has columns: Rank, Rank Score, Name, Mobile No, Email, Test Score, Resume, Date, and Resume Scores.

Rank	Rank Score	Name	Mobile No	Email	Test Score	Resume	Date	Resume Scores
1	16.0	shabeer	16527893489	<a href="mailto:mpshabeer007@gmail.com">mpshabeer007@gmail.com</a>	8	<a href="#">download</a>	May 8, 2023	40

Figure C.16: View Resume Score

Resume Selector Home Matching Jobs View Vacancy View Tips View Videos view Result POST FEEDBACK LOGOUT

VIEW VACANCY

vacancy	company name	No of Vacancy	Experiance	Rrequirements	Apply	attend test
data science	tcs	5	2	R.SQL>Data Visualization	<a href="#">apply</a>	<a href="#">attend test</a>
data science	tcs	7	5	R.SQL	<a href="#">apply</a>	<a href="#">attend test</a>

Figure C.17: View Matching Job

Resume Selector Home Matching Jobs View Vacancy View Tips View Videos view Result POST FEEDBACK LOGOUT

vacancy	company name	No of Vacancy	Experiance	Rrequirements	Apply	attend test
software engineer	tcs	6	1	Java Html sql	<a href="#">apply</a>	<a href="#">attend test</a>
dot.net developer	tcs	4	1	Dot.net, c#, c	<a href="#">apply</a>	<a href="#">attend test</a>
software engineer	tcs	3	1	Disloops	<a href="#">apply</a>	<a href="#">attend test</a>
Software Tester	abtech	4	1	one Year Experience in Testing	<a href="#">apply</a>	<a href="#">attend test</a>
django dev	tcs	3	3	Python C#s Html	<a href="#">apply</a>	<a href="#">attend test</a>
data science	tcs	5	2	R.SQL>Data Visualization	<a href="#">apply</a>	<a href="#">attend test</a>
software engineer	junaid	1	4	JAVA,Flutter	<a href="#">apply</a>	<a href="#">attend test</a>
data science	tcs	7	5	R.SQL	<a href="#">apply</a>	<a href="#">attend test</a>

Figure C.18: View Vacancy

Resume Selector Home Matching Jobs View Vacancy View Tips View Videos view Result POST FEEDBACK LOGOUT

Apply

Resume

Experience

Figure C.19: Upload Resume

Resume Selector Home Matching Jobs View Vacancy View Tips View Videos view Result POST FEEDBACK LOGOUT

Attend test

Timer  
Time Over!

A single program of OOPS contains \_\_\_\_\_ classes?

☐ Only 1  
☐ 132  
☐ 999  
☐ Any number

Figure C.20: Attend Exam





date	tips
March 26, 2023	use capital letter for name
May 10, 2023	Use Times New Roman Font in Resume

Figure C.21: View Tips



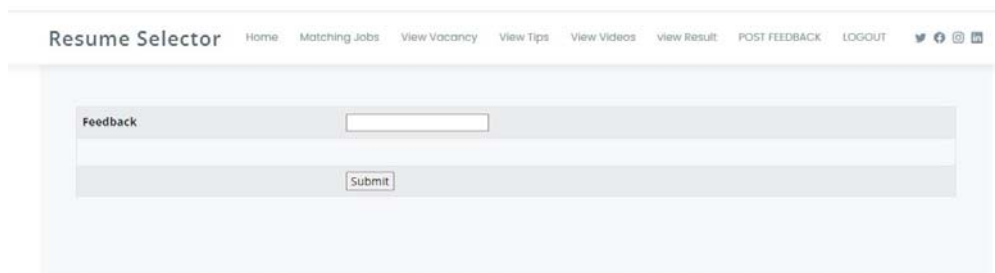
date	videos
May 10, 2023	<a href="#">video1.mp4</a>
May 10, 2023	<a href="#">video3.mp4</a>

Figure C.22: View Videos



Test Result				
vacancy	company	Mark	Date	Resume Score
software engineer	tcs	8	May 8, 2023	40.0

Figure C.23: View Result



Feedback
<input type="text"/>
<input type="button" value="Submit"/>

Figure C.24: Post Feedback

# Appendix D

## Git History

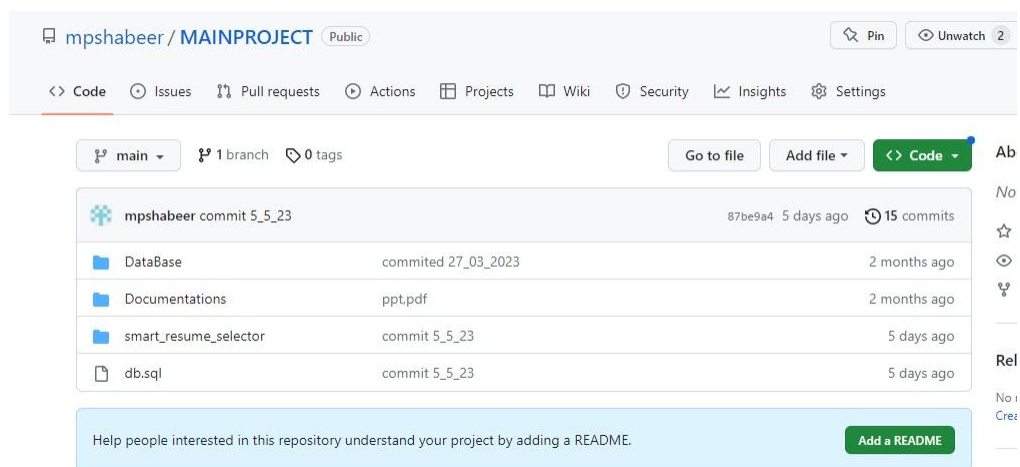


Figure D.1: Git Repository

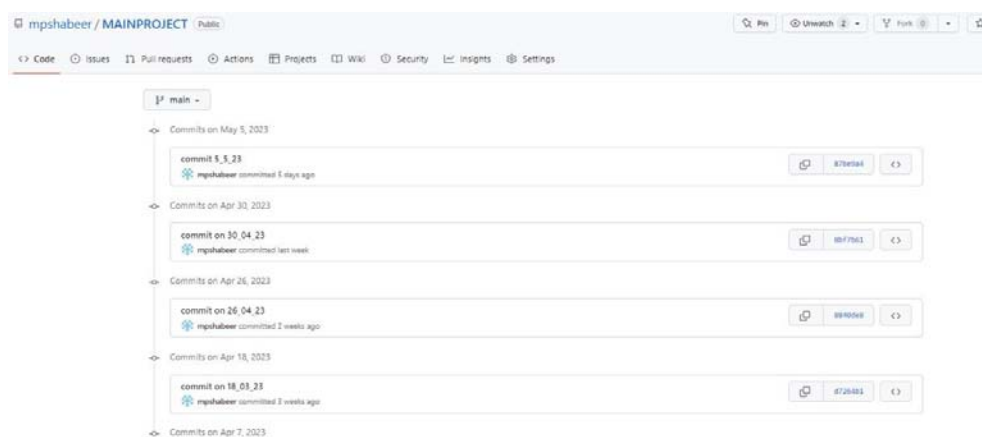


Figure D.2: Git Commits