

AI ENABLED MOCK INTERVIEW SYSTEM

A PROJECT REPORT

for

Project (KCA451)

Session (2023-24)

Submitted by

Palak Singh

2200290140104

Manmohan Dwivedi

2200290140086

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

**Under the Supervision of
Dr. Amit Kumar Gupta
(Professor)**



Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS

KIET Group of Institutions, Ghaziabad

Uttar Pradesh-201206

MAY 2024

CERTIFICATE

Certified that **Palak Singh 2200290140104, Manmohan Dwivedi 2200290140086** have carried out the project work having “**Mock Master**”(Major Project- **KCA451**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Palak Singh 2200290140104

Manmohan Dwivedi 2200290140086

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

Date:

Dr. Amit Kumar Gupta
Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

AI Enabled Mock Interview System

Palak Singh, Manmohan Dwivedi

ABSTRACT

The AI-Enabled Mock Interview System operates like a supportive mentor, leading individuals through the often daunting process of preparing for job interviews. This innovative platform aims to simplify and improve the experience of practicing for interviews by using the capabilities of smart technology known as artificial intelligence.

Consider a virtual space that serves as a helpful companion, specifically designed to help job seekers improve their interview skills. This system has been painstakingly designed to bridge the gap between traditional interview preparation methods and the ever-changing job-seeker requirements in today's competitive job market.

The system prioritises user convenience and effectiveness at its core. It starts with a simple process in which users upload their resumes and specify the job fields they prefer. This customisation allows the system to personalise the interview practice sessions based on the user's career goals and skill levels, making the experience more relevant and impactful. The interactive AI-driven virtual interviewer is what truly distinguishes this system. Consider it to be a friendly computer character who engages users in realistic interview simulations. This virtual interviewer evaluates the user's responses in real-time and provides constructive feedback by utilising advanced technologies such as natural language processing (NLP) and machine learning (ML). It's as if you have a supportive mentor guiding you through the interview, highlighting your strengths and pointing out areas for improvement.

The system's extensive question bank is an important component. This repository has been meticulously curated to contain a wide range of industry-specific interview questions. The question bank covers a wide range of job sectors, from technical questions to behavioural scenarios, ensuring that users have a diverse set of questions to practise regardless of their career interests.

The AI-Enabled Mock Interview System is, in short, more than just a piece of technological advancement. In order to make interview preparation less stressful for users, it is a helpful ally that is accessible, efficient, and customised to fit each user's unique path to professional achievement.

ACKNOWLEDGEMENTS

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

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

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

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

Palak Singh

Manmohan Dwivedi

TABLE OF CONTENTS

	Certificate	ii
	Abstract	iii
	Acknowledgements	iv
	Table of Contents	v
	List of Tables	vii
	List of Figures	viii
1	Introduction	1-4
	1.1 Overview	1
	1.2 Motivation	2-3
	1.3 Problem Statement	3-4
	1.4 Expected Outcome	4-5
2	Literature Survey	6-9
3	Design	9-17
	3.1 Data Flow Diagram	9-10
	3.1.1 Level 0 DFD	9
	3.1.2 Level 1 DFD	10
	3.2 Data Dictionary	11
	3.3 Sequence Diagram	11-15
	3.3.1 Sequence Diagram of Registration	11-12
	3.3.2 Sequence Diagram of Login Process	12-13
	3.3.3 Sequence Diagram of Domain Customization	13-14
	3.3.4 Sequence Diagram of Interview and Feedback	14-15
	3.4 Use Case Diagram	15-18
	3.4.1 Use Case Description of Registration	16
	3.4.2 Use Case Description of Login	16
	3.4.3 Domain Customization	16-17
	3.4.4 Use Case Description of Interview	17

	3.4.5 Use Case Description of Feedback Generation	17-18
4	Proposed Work	19-27
	4.1 Dataset Description	19
	4.2 Technology Description	19
	4.3 Approach Used	20
	4.4 Dependencies Required	20-22
	4.5 Algorithms and Flowchart	22-27
5	Results	28-32
	5.1 Screens and Explanation	28-32
6	Discussions	33-40
	6.1 Performance	33
	6.2 Limitations of the System	33-34
	6.3 Future Research Directions	34-36
7	Conclusion	37-38
8	References	39-40
9	Bibliography	41-42
	9.1 Online Websites	41
	9.2 Reference Books	41-42

LIST OF TABLES

Table No.	Name of Table	Page
3.1	Data Dictionary	11
4.1	Dataset Description	19

LIST OF FIGURES

Figure No.	Name of Figure	Page No.
1.1	Basic Architecture of Mock Interview System	2
2.1	System Architecture	5
2.2	Technology Acceptance Model	8
3.1	Level 0 DFD	9
3.2	Level 1 DFD	10
3.3	Sequence Diagram of Registration	12
3.4	Sequence Diagram of Login	13
3.5	Sequence Diagram of Domain Customization	14
3.6	Sequence Diagram of Interview and Feedback	14
3.7	Use Case Diagram of Mock Interview System	15
4.1	Interconnection of Technologies	20
4.2	Flowchart of signup	23
4.3	Flowchart of Login	24
4.4	Flowchart of Database Connectivity	25
4.5	Flowchart of API Connectivity	26
4.6	Flowchart of Feedback Generation	27
5.1	Home Page	28
5.2	Login and Registration Screen	29
5.3	Start New Interview Screen	30
5.4	Interview Customization screen	30
5.5	Interview Screen	31

CHAPTER 1

INTRODUCTION

1.1 Overview

The innovative AI-Enabled Mock Interview System aims to transform and update the conventional method of interview preparation. This cutting-edge platform seeks to provide users with a customized and immersive experience to hone their interview abilities by utilizing cutting-edge technologies like artificial intelligence (AI) and machine learning (ML). The method primarily focuses on customizing mock interview scenarios to fit each person's skill level and career goals, creating a simulated but authentic interview setting. Users interact in real-time through dynamic question-and-answer sessions with AI-driven virtual interviewers, receiving immediate feedback to help them improve. Practice sessions are varied and pertinent because of the platform's extensive question library, which covers multiple industry domains. To further support ongoing skill development, comprehensive performance statistics let users monitor advancement and pinpoint areas for improvement. The system's ultimate goal is to close the knowledge gap between theory and practice by giving users more self-assurance, flexibility, and success in actual job interviews."

The technology integration, tailored learning, simulated interview setting, and the system's potential to revolutionize interview preparation techniques are highlighted in this overview, which sums up the essence of the AI-Enabled Mock Interview System. Depending on what the project requires, changes can be made to highlight certain features of the system or incorporate specific project data.

1.2 Motivation

As a college student who is looking for job opportunities and wants to learn several aspects of an interview such as written test, technical interview, or HR interview. We as students also watch so many videos of how to crack a particular interview but all of these are waste if we won't implement it and practice it. But we do not have any proper platform where we can practice our interview without any hesitation. As some applications or websites are available which provide such services of taking interviews but all of them were with the real person which is a good practice but for a candidate who is new and wants to try the experience of interview may feel hesitant to talk to real person. That is why we came up with the idea of taking an interview with the help of an AI and get the feedback also on your interview.

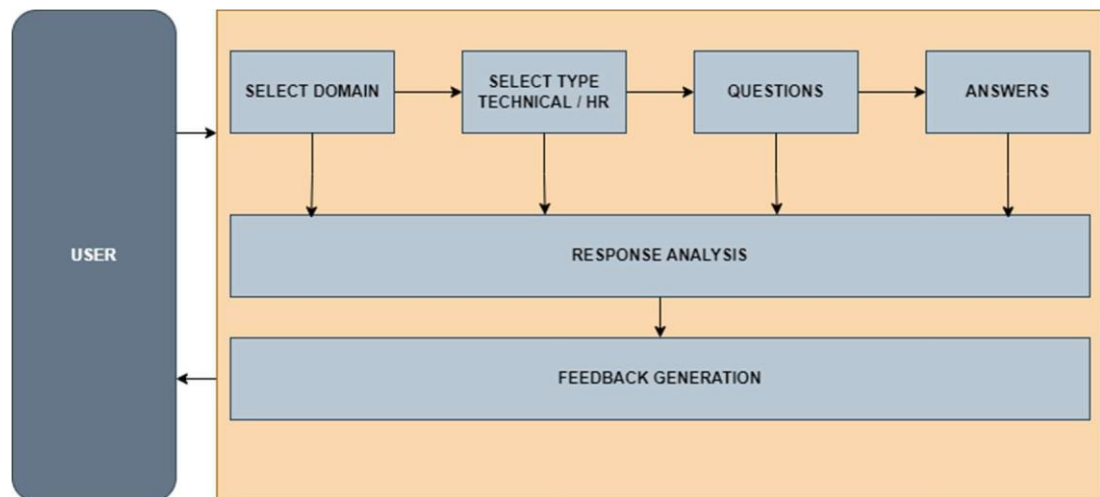


Fig. 1.1 Basic Architecture of Mock Interview System

Fig 1.1 represents the basic architecture of the mock interview system in which the user will interact with the system and will select the domain and type of interview the user wants to take then the system will ask questions to the user, user will answer those questions then all the answers will be analyzed by the AI and generate the feedback to display to the user.

Why Mock Interviews are Important

1. **Practice Makes Perfect:** Just like any skill, interviewing gets better with practice. Mock interviews provide a safe environment to refine your answers, body language, and overall presentation.

2. **Reduce Anxiety:** Interviews can be nerve-wracking. By going through mock interviews, you can become more comfortable with the format and reduce your anxiety when it comes to the real thing.
3. **Receive Constructive Feedback:** Mock interviews offer the opportunity to receive feedback on your performance. This feedback is invaluable as it can highlight your strengths and identify areas for improvement that you might not have noticed.
4. **Improve Communication Skills:** Being able to articulate your thoughts clearly and confidently is crucial in an interview. Mock interviews help you practice this, making you a more effective communicator.
5. **Enhance Problem-Solving Abilities:** Many interviews, especially for technical roles, include problem-solving questions. Mock interviews can help you develop strategies for approaching these questions logically and efficiently.

1.3 Problem Statement

The AI-Enabled Mock Interview System revolves around the inadequacy of conventional interview preparation methods to meet the evolving needs of job seekers in today's competitive job market. Traditional approaches often lack personalization, real-time feedback, and adaptability to diverse interview scenarios, leading to limited readiness and confidence among candidates.

Inadequate Interview Preparation:

- Conventional interview preparation methods, such as reading articles or attending generic interview workshops, fail to provide tailored experiences aligned with individual career aspirations, skill levels, and industry-specific requirements.
- Limited access to realistic interview simulations leads to a lack of exposure to diverse questions and scenarios, inhibiting candidates from refining their interview skills effectively.
- Feedback mechanisms in traditional methods are often absent or delayed, impeding candidates' ability to understand their strengths and areas needing improvement in real time.

Existing online resources and mock interview platforms lack the sophistication needed to provide dynamic, adaptive, and AI-driven personalized experiences. These limitations contribute to job seekers' unpreparedness, affecting their confidence and competitiveness during actual job interviews.

The AI-Enabled Mock Interview System seeks to address these shortcomings

by offering a technologically advanced platform that bridges this gap, providing personalized, adaptive, and realistic interview simulations coupled with immediate feedback to enhance users' interview readiness and confidence.

1.3 Expected Outcome

The AI-Powered Mock Interview System expects a variety of results that are intended to change the interview preparation scene. Based on innovative technology and user-centered design, the system envisions multiple significant results that will enable people looking to improve their interviewing abilities and thrive in the cutthroat job market.

- **Take Interview of Candidates**

The Systems major goal is to take the interview of the registered users and the guest users also. In this Project System will take the interview in the textual and the audio format. User should be able to simply select the type and domain of the interview and then attempt the mock interview.

- **Customized Domain**

The user will be able to select amongst various options such as HR or technical interview and also the domain of interview which user wants to take in order to achieve the future goals. Based on the customized selection the interview will take place

- **Interactive User Interface**

Our system will have the interactive User Interface so that user can take interview very easily and able to use the website without any issue as the UI is not at all complicated and is created considering the college students specially, so it is even more interesting, attractive, and easy to use.

- **Response Analysis**

One of the central objectives of the system is to enhance users' accuracy by giving them correct feedback on their answers to a particular question. Even if user does not know the answer to any question, then AI will tell them the correct answer so that they can remember that answer for the future reference.

- **Efficient and Targeted Practice Sessions**

The system's diverse question bank and customization features enable users to engage in efficient and targeted practice sessions. By focusing on specific industry domains and difficulty levels aligned with users' career goals, the system ensures that practice sessions are relevant, purposeful, and conducive to

skill improvement.

- **Continuous Learning and Growth**

Beyond the immediate preparation for job interviews, the system fosters a culture of continuous learning and growth. By providing detailed performance analytics and feedback, users have the opportunity to identify areas for improvement, encouraging ongoing skill development even after securing a job.

CHAPTER 2

LITERATURE SURVEY

In [1], Helmut Prendinger and Mitsuru Ishizuka report on their efforts in developing affective character-based interfaces, i.e., interfaces that recognize and measure affective information of the user and address user affect by employing embodied characters. They describe the Empathic Companion, an animated interface agent that accompanies the user in the setting of a virtual job interview. This interface application takes physiological data (skin conductance and electromyography) of a user in real-time, interprets them as emotions, and addresses the user's affective states in the form of empathic feedback. The Empathic Companion is conceived as an educational agent that supports job seekers preparing for a job interview.

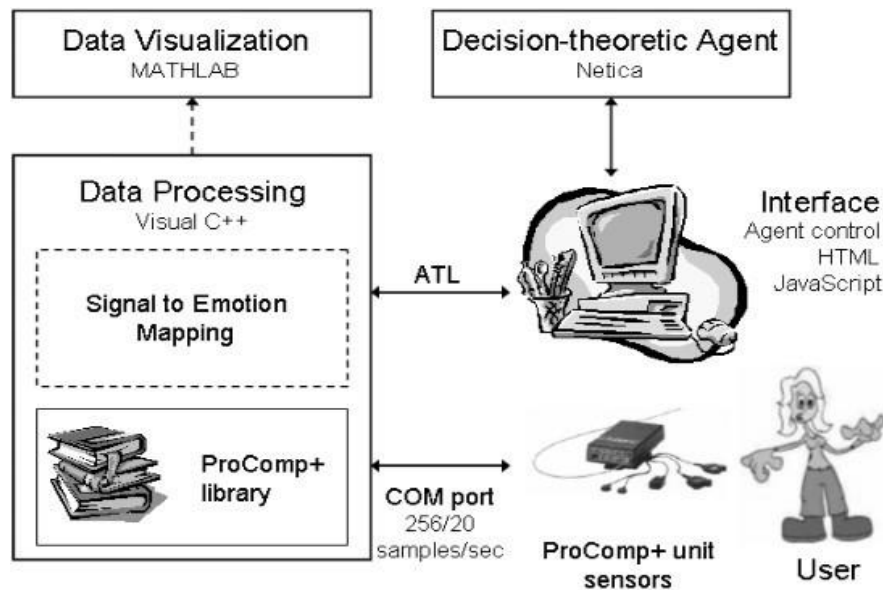


Fig. 2.1 System Architecture

They also present results from an exploratory study that aims to evaluate the impact of the Empathic Companion by measuring users' skin conductance and heart rate. While an overall positive effect of the Empathic Companion could not be shown, the outcome of the

experiment suggests that empathic feedback has a positive effect on the interviewee's stress level while hearing the interviewer question.

In [2] Jay F. Nunamaker Jr., Douglas C. Derrick, Aaron C. Elkins, Judee K. Burgoon And Mark W. Patton have created an automated kiosk that uses embodied intelligent agents to interview individuals and detect changes in arousal, behavior, and cognitive effort by using psychophysiological information systems. In this paper, they describe the system and propose a unique class of intelligent agents, which are described as Special Purpose Embodied Conversational Intelligence with Environmental Sensors (SPECIES). SPECIES agents use heterogeneous sensors to detect human physiology and behavior during interactions, and they affect their environment by influencing human behavior using various embodied states (i. e., gender and demeanor), messages, and recommendations. Based on the SPECIES paradigm, we present three studies that evaluate different portions of the model, and these studies are used as foundational research for the development of the automated kiosk. The first study evaluates human-computer interaction and how SPECIES agents can change perceptions of information systems by varying appearance and demeanor. Instantiations that had the agents embodied as males were perceived as more powerful, while female embodied agents were perceived as more likable. Similarly, smiling agents were perceived as more likable than neutral demeanor agents. The second study demonstrated that a single sensor measuring vocal pitch provides SPECIES with environmental awareness of human stress and deception. The final study ties the first two studies together and demonstrates an avatar-based kiosk that asks questions and measures the responses using vocalic measurements.

In [3] Katie Maras, Jade Eloise Norris, Jemma Nicholson etc mentions that despite possessing valuable skills, social communication differences mean that autistic people are frequently disadvantaged in job interviews. We examined how autistic and non-autistic adults compared on standard (unmodified) job interview questions, and then used these findings to develop and evaluate supportive adaptations to questions. Fifty adults (25 autistic, 25 non-autistic) took part in two mock job interviews. Interview 1 provided a baseline measure of performance when answering typical, unmodified interview questions. Employment experts (unaware of participants' autism diagnoses) rated all interviewees on question-specific performance and overall impressions and then provided feedback about how interviewees could improve and how questions could be adapted to facilitate this. Interviewees also provided feedback about the interview process from their perspective. Adaptations to the questions were developed, with Interview 2 taking place approximately 6 months later. Results demonstrated that, in Interview 1, employment experts rated autistic interviewees less favorably than nonautistic interviewees. Ratings of both autistic and non-autistic participants' answers improved in Interview 2, but particularly for autistic interviewees (such that differences between autistic and non-autistic interviewees' performance reduced in Interview 2). Employers should be aware that adaptations to job interview questions are critical to level the playing field for autistic candidates.

In [4] Marcia F. Robinson said that disruptive technologies that enable knowledge work automation through artificial intelligence (AI) and robotics are changing how we do work and precipitating the redesign of organizations. Although research suggests that HR offices have traditionally lagged other functional areas in the adoption of new technology, the current talent shortage and the competitive value of big data are driving the adoption and use of AI tools in the hiring process. This qualitative research study sought to hear the voices of HR practitioners on the uses of AI technology in the hiring process and to understand the attitudes and perspectives of participants towards the adoption and use of AI in this way. For this study, HR executives (HREs), HR recruiters (HRRs), and HR information systems analysts (HRISAs) from global organizations headquartered in the Northeastern region of the United States, were interviewed.

The data revealed that while HR practitioners acknowledged the relative advantage of AI for algorithm-based-hiring, they also acknowledge the value of human contact for successful recruiting outcomes. Data analysis revealed HR practitioners' personal beliefs and feelings about AI and framed their perspectives through organizational change experiences, social or environmental observations, and the uses of the technology itself. As technologies continue to usher in the automation of knowledge work, HR practitioners will need both academic and professional development training to design and support the automated workplaces of the future where human and artificial intelligence work side by side.

In [5] This research involves Liander, a transport utility. The organization faces the difficulty of integrating destabilizing renewable energy sources, while simultaneously delivering high quality power. The main challenge is to deploy their current assets as efficient as possible, considering the future grid. However, it is increasingly challenging to link performance outcomes to investment decisions as still a large amount of the planning is completed manually. Therefore, Liander has the desire to turn their asset management process into a more data-driven one, using information technology (IT) such as AI enabled DSS. Ultimately Liander has the desire to fully automate decision making within its asset management process. Liander has taken steps towards more data-driven decision-making. Liander created the Insights and Analytics (I&A) department, which is responsible for the transformation of the asset management process into a more data-driven one.

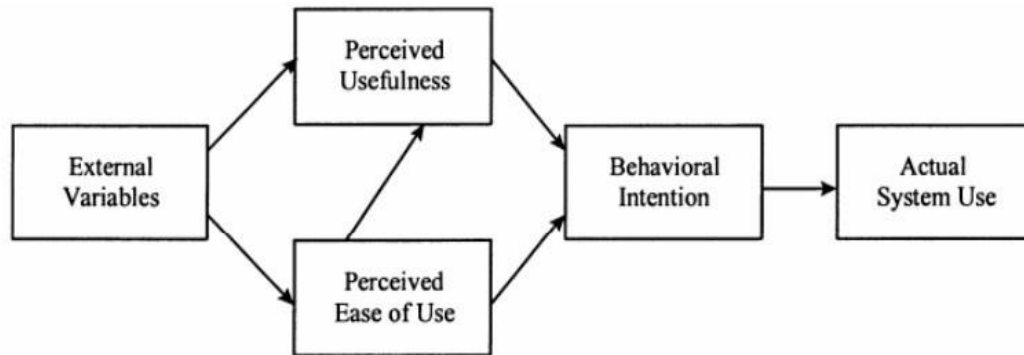


Fig. 2.2 Technology Acceptance Model

However, progress in developing, testing and getting the systems accepted among targeted users is still slow. The manager of the I&A department indicated that currently there seems little internal support for the experimentation and execution of data-driven projects. This lack of support seems to be mostly on the targeted user level. From initial conversations with the manager of the I&A department, several observations were made. First, it seems that the industry and firm's culture leave little room for error, transport utilities always have to assure high quality power to satisfy their customers. Experimenting with new ways of decision making, in the form of AI enabled DSS, seems risky to targeted users, as it could lower their performance. As a result, there is little space for experimenting with new ways of working and targeted users tend to stick to their current working processes.

CHAPTER 3

FEASIBILITY STUDY

After doing the project, study and analysing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible-given unlimited resources and in finite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements. There are three parts in feasibility study

- a) Technical Feasibility
- b) Economic Feasibility
- c) Operational Feasibility
- d) Scheduling Feasibility
- e) Security and Privacy Feasibility
- f) User Engagement and Feedback

3.1 TECHNICAL FEASIBILITY

The technical feasibility of P2P is rooted in its ability to implement and maintain a sophisticated recipe search engine. The utilization of dynamic lists, real-time updates, and user authentication systems suggests a sound technical foundation. The study evaluates the availability of required technologies, the expertise of the development team, and the feasibility of incorporating innovative features.

The term "technical feasibility" refers to the assessment of the practicality and technical requirements of a proposed project or system. This analysis is crucial during the early stages of project planning to determine whether the project is viable from a technological standpoint. Here's a breakdown of the key aspects involved in evaluating technical feasibility:

1. Technology Assessment

- **Current Technology:** Evaluate whether the current technology is capable of meeting the project's requirements. This involves assessing existing hardware, software, and network capabilities.
- **Emerging Technology:** Consider the potential of new and emerging technologies to fulfill the project needs. This includes evaluating the maturity, stability, and adoption rates of these technologies.

2. Technical Skills and Expertise

- **Availability of Expertise:** Assess whether the organization has the necessary technical skills and expertise to implement and maintain the project. If not, determine whether these skills can be acquired through training or hiring.
- **Experience:** Evaluate the team's past experience with similar technologies or projects to ensure they can handle potential challenges.

3. System Requirements

- **Hardware Requirements:** Identify the hardware necessary to support the project, including servers, workstations, and other equipment.
- **Software Requirements:** Determine the software needs, including operating systems, applications, databases, and development tools.
- **Integration Requirements:** Assess how the new system will integrate with existing systems and whether any modifications are needed.

4. Scalability and Flexibility

- **Scalability:** Evaluate whether the proposed solution can scale to meet future demands. This includes considering the potential for increased data volumes, user numbers, and transaction rates.
- **Flexibility:** Assess the system's ability to adapt to changing requirements and technologies over time.

5. Security and Compliance

- **Security Requirements:** Identify the security measures necessary to protect the system from threats and vulnerabilities. This includes data encryption, access controls, and secure coding practices.
- **Regulatory Compliance:** Ensure the system complies with relevant laws, regulations, and industry standards. This may involve data privacy regulations, industry-specific standards, and organizational policies.

6. Cost and Time Estimates

- **Cost Estimation:** Estimate the costs associated with the technology, including hardware, software, development, and ongoing maintenance.
- **Time Estimation:** Assess the time required to develop, implement, and deploy the system. This includes considering potential delays and risks that may impact the timeline.

7. Risk Assessment

- **Identification of Risks:** Identify potential technical risks that could impact the project's success. This includes risks related to technology obsolescence, integration challenges, and dependency on third-party vendors.

- **Mitigation Strategies:** Develop strategies to mitigate identified risks, such as contingency planning, alternative solutions, and phased implementation.

Steps to Conduct a Technical Feasibility Study

1. **Define Objectives:** Clearly outline the objectives and scope of the project.
2. **Gather Requirements:** Collect detailed requirements from stakeholders.
3. **Analyze Existing Systems:** Review the current systems and technology in use.
4. **Identify Technology Options:** Explore various technology solutions that could meet the project requirements.
5. **Evaluate Options:** Assess the feasibility of each option based on the criteria mentioned above.
6. **Make Recommendations:** Provide a recommendation on the most feasible technology solution.
7. **Document Findings:** Compile the findings and recommendations into a comprehensive report.

3.2 ECONOMIC FEASIBILITY

Evaluating the economic viability of P2P involves assessing the costs associated with development, maintenance, and marketing against the anticipated benefits. The feasibility study considers budgetary constraints, potential revenue streams (such as advertisements or premium features), and a projected return on investment. This analysis ensures that P2P is economically sustainable and aligns with market expectations.

Economic feasibility is a crucial aspect of project evaluation that assesses the financial aspects of a proposed project to determine whether it is economically viable. This analysis helps stakeholders understand the cost-benefit ratio, potential financial risks, and overall economic impact of the project. Here's an in-depth look at the components and steps involved in conducting an economic feasibility study:

Key Components of Economic Feasibility

1. Cost Analysis

- **Initial Costs:** Estimate the upfront costs required to start the project. This includes costs for research and development, procurement of technology, infrastructure setup, and initial marketing.
- **Operational Costs:** Calculate the ongoing costs associated with running the project. This includes labor, maintenance, utilities, supplies, and other recurring expenses.

- **Indirect Costs:** Identify indirect costs such as administrative overhead, training, and potential opportunity costs.

2. Revenue Projections

- **Revenue Streams:** Identify potential sources of revenue. This could include product sales, service fees, subscriptions, licensing, or other revenue-generating activities.
- **Pricing Strategy:** Determine the pricing model and strategy for the product or service. This includes considering competitive pricing, value-based pricing, and market demand.
- **Sales Forecast:** Project the expected sales volume over time. This involves market analysis, customer segmentation, and historical data if available.

3. Benefit Analysis

- **Tangible Benefits:** Calculate the direct financial benefits, such as increased sales, cost savings, and improved productivity.
- **Intangible Benefits:** Consider non-monetary benefits like improved customer satisfaction, brand value, market positioning, and competitive advantage.

4. Break-Even Analysis

- **Break-Even Point:** Determine the point at which total revenues equal total costs, indicating when the project will start generating profit.
- **Payback Period:** Calculate the time required to recoup the initial investment from net cash flows.

5. Net Present Value (NPV) and Internal Rate of Return (IRR)

- **NPV:** Calculate the present value of net cash flows generated by the project, discounted at the project's cost of capital. A positive NPV indicates that the project is expected to generate more value than its cost.
- **IRR:** Determine the discount rate at which the NPV of the project is zero. A higher IRR compared to the cost of capital indicates a financially attractive project.

6. Sensitivity Analysis

- **Variable Impact:** Assess how changes in key variables (e.g., costs, revenues, market conditions) impact the project's financial outcomes.
- **Risk Assessment:** Identify potential financial risks and develop strategies to mitigate them.

Steps to Conduct an Economic Feasibility Study

1. Define Objectives and Scope

- Clearly outline the financial goals and scope of the project. Define the parameters for the economic feasibility study.

2. Gather Data

- Collect data on costs, revenue projections, market trends, and other relevant financial information.

3. Estimate Costs

- Perform a detailed cost analysis, including initial, operational, and indirect costs.

4. Forecast Revenues

- Project future revenues based on market analysis, historical data, and potential revenue streams.

5. Perform Benefit Analysis

- Calculate both tangible and intangible benefits associated with the project.

6. Conduct Break-Even Analysis

- Determine the break-even point and payback period for the project.

7. Calculate NPV and IRR

- Use financial models to calculate the NPV and IRR, ensuring the project's potential profitability.

8. Conduct Sensitivity Analysis

- Evaluate the impact of varying key assumptions and identify potential financial risks.

9. Prepare the Report

- Compile the findings into a comprehensive report, including financial projections, risk assessments, and recommendations.

3.3 OPERATIONAL FEASIBILITY

The operational feasibility of P2P hinges on its seamless integration into users' daily cooking routines. User-friendliness, efficient recipe categorization, and a responsive interface are critical factors. The study explores how well P2P aligns with user behaviours and whether it can be easily adopted and embraced by the target audience. Operational feasibility also considers the

scalability of the platform to accommodate a growing user base.

Operational feasibility is an essential aspect of project evaluation that assesses how well a proposed project or system will function within the existing operational environment of an organization. It examines the compatibility of the new system with the current operations, processes, personnel, and other systems to ensure smooth implementation and sustainable performance. Here's a detailed look at the components and steps involved in assessing operational feasibility:

CHAPTER 4

DESIGN

4.1 Data Flow Diagram

Data Flow Diagram (DFD) represents the flow of data within information systems. Data Flow Diagrams (DFD) provide a graphical representation of the data flow of a system that can be understood by both technical and non-technical users. The models enable software engineers, customers, and users to work together effectively during the analysis and specification of requirements.

DFD is the abbreviation for **Data Flow Diagram**. The flow of data in a system or process is represented by a Data Flow Diagram (DFD). It also gives insight into the inputs and outputs of each entity and the process itself. Data Flow Diagram (DFD) does not have a control flow and no loops or decision rules are present. Specific operations, depending on the type of data, can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. It is useful for analyzing existing as well as proposed systems

It provides an overview of

- What data is system processes.
- What transformation are performed.
- What data are stored.
- What results are produced , etc.

Data Flow Diagram can be represented in several ways. The Data Flow Diagram (DFD) belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

Characteristics of Data Flow Diagram (DFD)

Below are some characteristics of Data Flow Diagram (DFD):

- **Graphical Representation:** Data Flow Diagram (DFD) use different symbols and notation to represent data flow within system. That simplify the complex model.
- **Problem Analysis:** Data Flow Diagram (DFDs) are very useful in understanding a system and can be effectively used during analysis. Data Flow Diagram (DFDs) are quite general and are not limited to problem analysis for software requirements specification.
- **Abstraction:** Data Flow Diagram (DFD) provides a abstraction to complex model i.e. DFD hides unnecessary implementation details and show only the flow of data and processes within information system.

- **Hierarchy:** Data Flow Diagram (DFD) provides a hierarchy of a system. High-level diagram i.e. 0-level diagram provides an overview of entire system while lower-level diagram like 1-level DFD and beyond provides a detailed data flow of individual process.
- **Data Flow:** The primary objective of Data Flow Diagram (DFD) is to visualize the data flow between external entity, processes and data store. Data Flow is represented by an arrow Symbol.
- **Ease of Understanding:** Data Flow Diagram (DFD) can be easily understand by both technical and non-technical stakeholders.
- **Modularity:** Modularity can be achieved using Data Flow Diagram (DFD) as it breaks the complex system into smaller module or processes. This provides easily analysis and design of a system.

4.1.1 Level 0 Data Flow Diagram

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

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

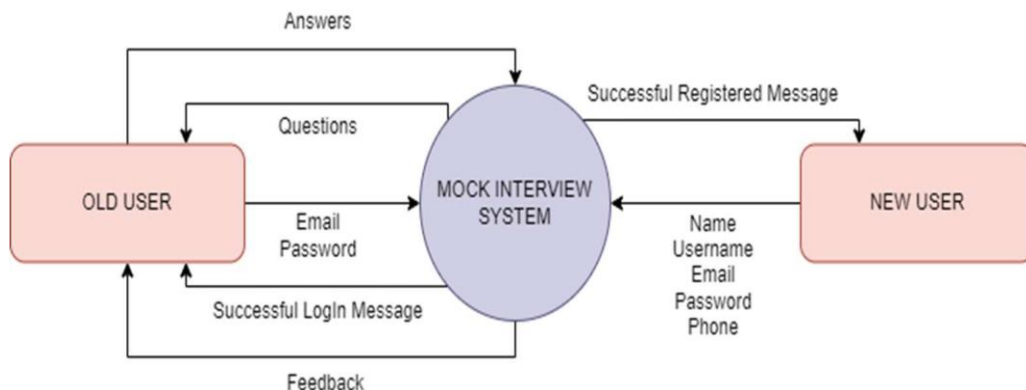


Fig. 4.1 Level 0 DFD of Mock Interview System

Fig. 4.1 elaborates the interaction between user and the system. If the user is new then user will first register to the system by providing name, username, email, password, phone. Once successfully registered a message will be display to the user of successfully registered. If the user is old, then they can directly login to the system. Once successfully logged into the system, it will provide a message to the user. Then the user will provide the domain and type of interview, based on that information system will provide you set of questions that user need to answer. System will also provide the feedback simultaneously.

4.1.2 Level 1 Data Flow Diagram

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

This level provides a more detailed view of the system by breaking down the major processes identified in the level 0 DFD into sub-processes. Each sub-process is depicted as a separate process on the level 1 DFD. The data flows and data stores associated with each sub-process are also shown. In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses

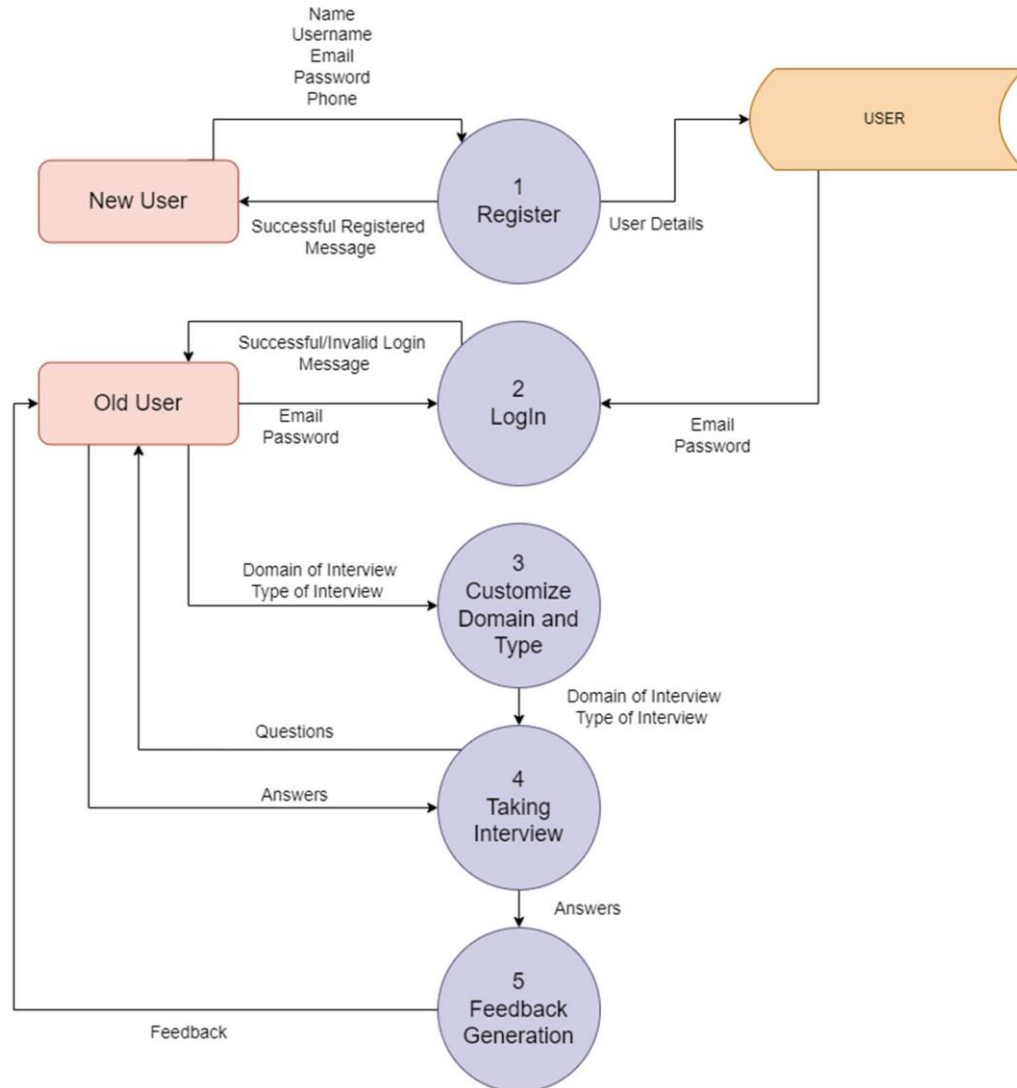


Fig. 4.2 Level 1 DFD of Mock Interview System

Fig. 4.2 explains the entire flow of user and system with all processes involved in the system. If the user is new to the system, then register to the system by providing the details to it. And all the details of the user will be stored in the database. If the user is old, then user will log into the system by email and password which will be validated from the database. Then the user will provide the customization of domain and type of the interview. After the customization system will take the interview of the user then feedback is generated and given to the user.

4.2 Data Dictionary

A Data Dictionary comprises two words i.e. **Data** which simply means information being collected through some sources and **Dictionary** means where this information is available.

A **Data Dictionary** can be defined as a collection of information on all data elements or contents of databases such as data types, and text descriptions of the system. It makes it easier for users and analysts to use data as well as understand and have common knowledge about inputs, outputs, components of a database, and intermediate calculations.

Why Data Dictionary is Essential?

There is less information and details provided by data models. So, a data dictionary is essential and needed to have proper knowledge and usage of contents. Data Dictionary provides all information about names that are used in system models. Data Dictionary also provides information about [entities, relationships, and attributes](#) that are present in the system model. As a part of the structured analysis and design tool, the implementation of a data dictionary is done.

Types of Data Dictionary

Data Dictionary is basically of two types. We will discuss each of them.

- Integrated Data Dictionary
- Stand Alone Data Dictionary

1. Integrated Data Dictionary

Integrated Data Dictionary can be seen as a catalog that can be maintained by the relational database. In previous databases, there is not any functionality of Integrated Data Dictionary, so they use Stand Alone Data Dictionary.

There are two types of **Integrated Data Dictionary**.

- **Active Data Dictionary:** [Active Database Dictionary](#) is a type of database that is updated automatically in case any changes are to be done to the database. These are self-updating.
- **Passive Data Dictionary:** Passive Databases are the databases that have to be maintained or updated manually in case of any changes have been made to the system.

Legal character: [a to z| A to Z] Digits: [0-9]

Special character: [@, \$, #, +, -, /]

Table 1: Data Dictionary

1	Name	Legal Characters
2	Domain	Legal Characters
3	Email	Legal Characters+ Digits+Special Character
4	College Name	Legal Characters
5	University	Legal Characters
6	Phone No.	Digits
7	Preferred Language	Legal Characters
8	Date of Birth	Digits+Digits+Digits

4.3 Sequence Diagram

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

To understand what a sequence diagram is, it's important to know the role of the [Unified Modeling Language](#), better known as UML. UML is a modeling toolkit that guides the creation and notation of many types of diagrams, including behavior diagrams, interaction diagrams, and structure diagrams.

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Note that there are two types of sequence diagrams: UML diagrams and code-based diagrams. The latter is sourced from programming code and will not be covered in this guide. Lucidchart's [UML diagramming software](#) is equipped with all the shapes and features you will need to model both.

Use cases for sequence diagrams

The following scenarios are ideal for using a sequence diagram:

- Usage scenario:

A usage scenario is a diagram of how your system could potentially be used. It's a great way to make sure that you have worked through the logic of every usage scenario for the system.

- Method logic:

Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to explore the logic of any function, procedure, or complex process.

- Service logic:

If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.

- Sequence diagram Visio

- Any sequence diagram that you create with Visio can also be uploaded into Lucidchart. Lucidchart supports .vsd and .vdx file import and is a great Microsoft Visio alternative. Almost all of the images you see in the UML section of this site were generated using Lucidchart.

Sequence Diagram of Registration Process

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

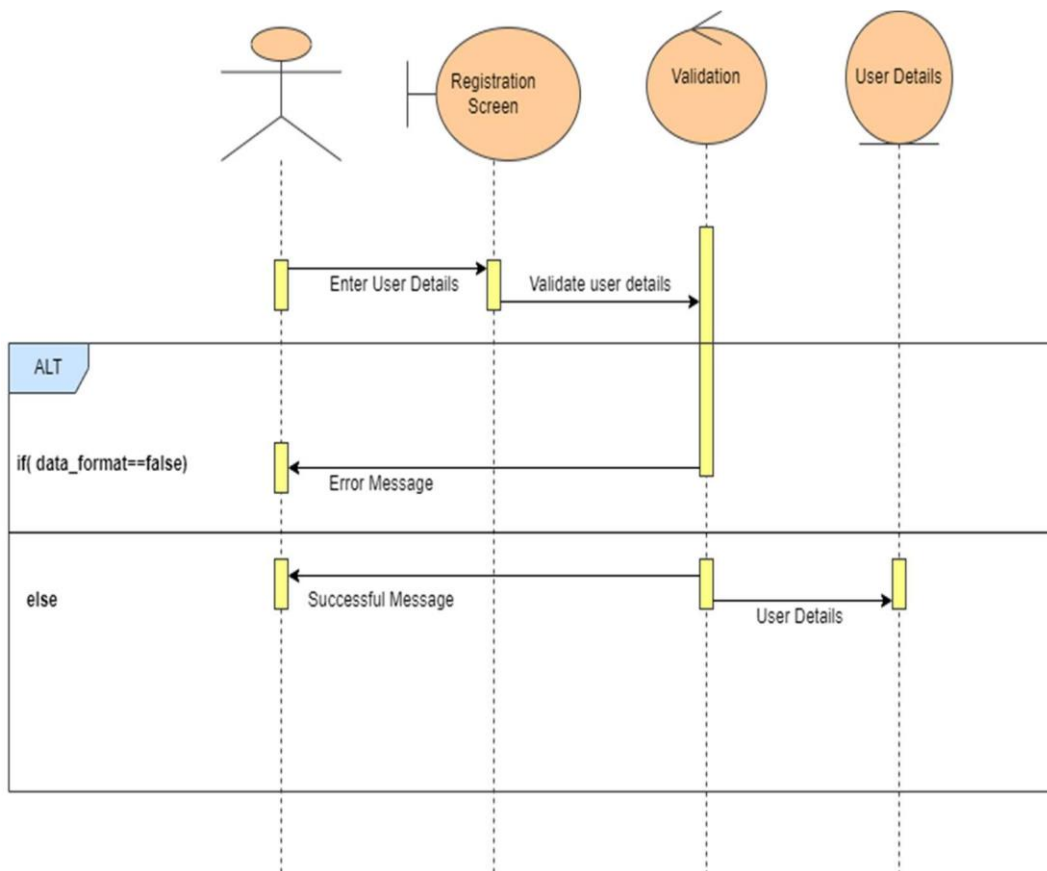


Fig. 4.3 Registration sequence diagram

Fig 4.3 explains about the process of registration where user send the details to the screen then validate those details. If details are not in correct format, then an error message is displayed. If details are in correct format, then successful message is displayed. Then details are stores in user database.

4.3.1 Sequence Diagram of Login Process

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

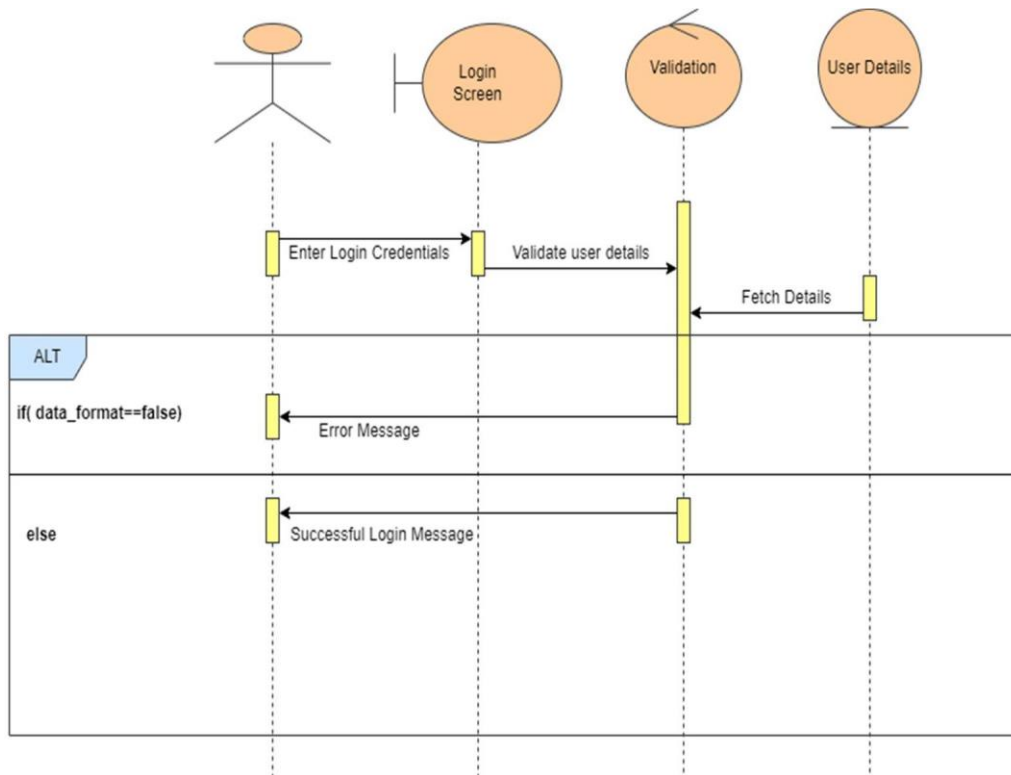


Fig. 4.4 Login sequence diagram

Fig 4.4 explains about the process of login where user send the details to the screen then validate those details. If details are not correct from fetched data from database, then an error message is displayed. If details are correct from fetched data from database, then successful message is displayed.

4.3.2 Sequence Diagram of Domain Customization Process

In this Diagram of Customization Process, it has 3 objects one actor, one boundary object, one store object.

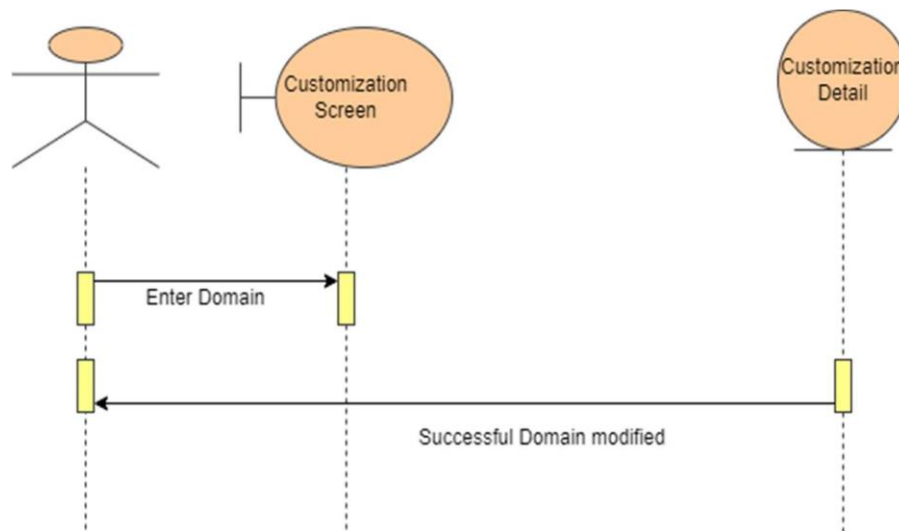


Fig. 4.5 Domain Customization Sequence Diagram

Fig 4.5 explains about the process of domain customization where user send the domain and type of interview details to the screen, then details are stores in Customization details and send the successful domain modified message.

4.3.3 Sequence Diagram of Interview and Feedback Process

In this diagram of interview and feedback Process, it has 3 objects one actor, one boundary object, one control object.

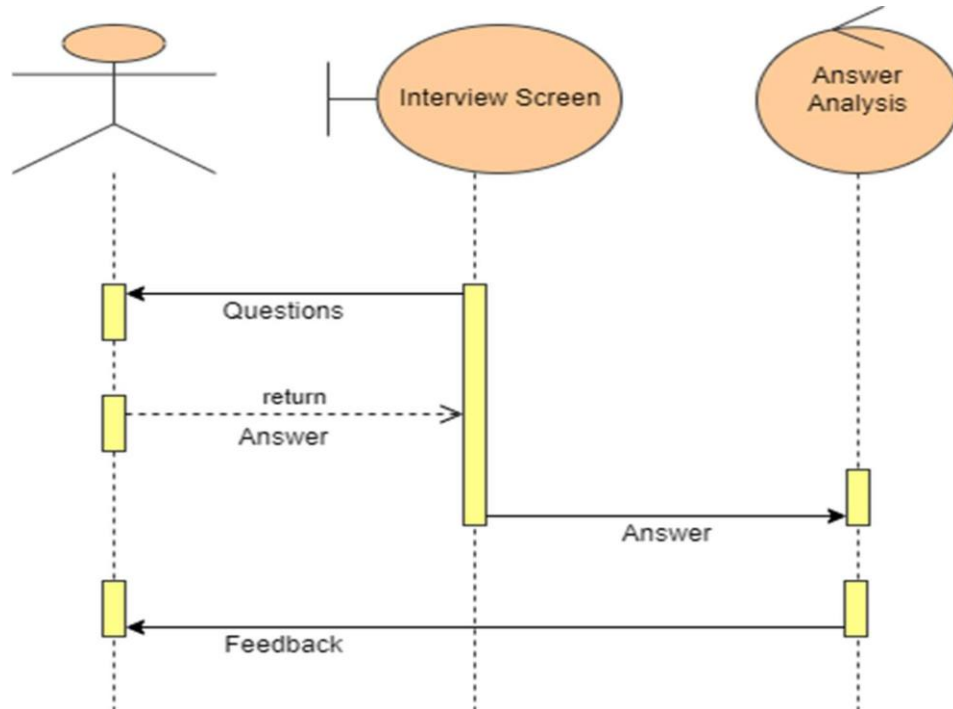


Fig. 4.6 Interview and Feedback Sequence Diagram

Fig 4.6 explains about how user will take the interview, analyze the answers, and then provide the feedback to them. Initially the Interview screen will show the questions to the user and wait for the response of the user. Now, the user will return the answers to the questions that are asked on the interview screen. Then those answers are now analyzed. Once the answers are analyzed, then those feedback are given to the user.

Use Case Diagram

In Use Case Diagram we elaborate about the purpose, actor, pre-condition, post- condition, basic flow, and alternate flow of all the use cases. In our system there are two actors, one is an old user and other is the new user who interacts with the use cases of the mock interview system. It explains the details and conditions of the system to be fulfilled in order to successfully complete each use case.

A Use Case Diagram is a vital tool in system design, it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user's perspective, aiding in the communication between stakeholders and guiding the development process.

What is a Use Case Diagram in UML?

A Use Case Diagram is a type of Unified Modeling Language (UML) diagram that represents the interaction between actors (users or external systems) and a system under consideration to accomplish specific goals. It provides a high-level view of the system's functionality by illustrating the various ways users can interact with it.

Use Case Diagram Notations

UML notations provide a visual language that enables software developers, designers, and other stakeholders to communicate and document system designs, architectures, and behaviors in a consistent and understandable manner.

Actors

Actors are external entities that interact with the system. These can include users, other systems, or hardware devices. In the context of a Use Case Diagram, actors initiate use cases and receive the outcomes. Proper identification and understanding of actors are crucial for accurately modeling system behavior.

Use Cases

Use cases are like scenes in the play. They represent specific things your system can do. In the online shopping system, examples of use cases could be "Place Order," "Track Delivery," or "Update Product Information". Use cases are represented by ovals.

System Boundary

The system boundary is a visual representation of the scope or limits of the system you are modeling. It defines what is inside the system and what is outside. The boundary helps to establish a clear distinction between the elements that are part of the system and those that

are external to it. The system boundary is typically represented by a rectangular box that surrounds all the use cases of the system.

Purpose of System Boundary:

- **Scope Definition:** It clearly outlines the boundaries of the system, indicating which components are internal to the system and which are external actors or entities interacting with the system.
- **Focus on Relevance:** By delineating the system's scope, the diagram can focus on illustrating the essential functionalities provided by the system without unnecessary details about.

Use Case Diagram Relationships

In a Use Case Diagram, relationships play a crucial role in depicting the interactions between actors and use cases. These relationships provide a comprehensive view of the system's functionality and its various scenarios. Let's delve into the key types of relationships and explore examples to illustrate their usage.

Association Relationship

The Association Relationship represents a communication or interaction between an actor and a use case. It is depicted by a line connecting the actor to the use case. This relationship signifies that the actor is involved in the functionality described by the use case.

Include Relationship

The Include Relationship indicates that a use case includes the functionality of another use case. It is denoted by a dashed arrow pointing from the including use case to the included use case. This relationship promotes modular and reusable design.

Extend Relationship

The Extend Relationship illustrates that a use case can be extended by another use case under specific conditions. It is represented by a dashed arrow with the keyword "extend." This relationship is useful for handling optional or exceptional behavior.

Generalization Relationship

The Generalization Relationship establishes an "is-a" connection between two use cases, indicating that one use case is a specialized version of another. It is represented by an arrow pointing from the specialized use case to the general use case.

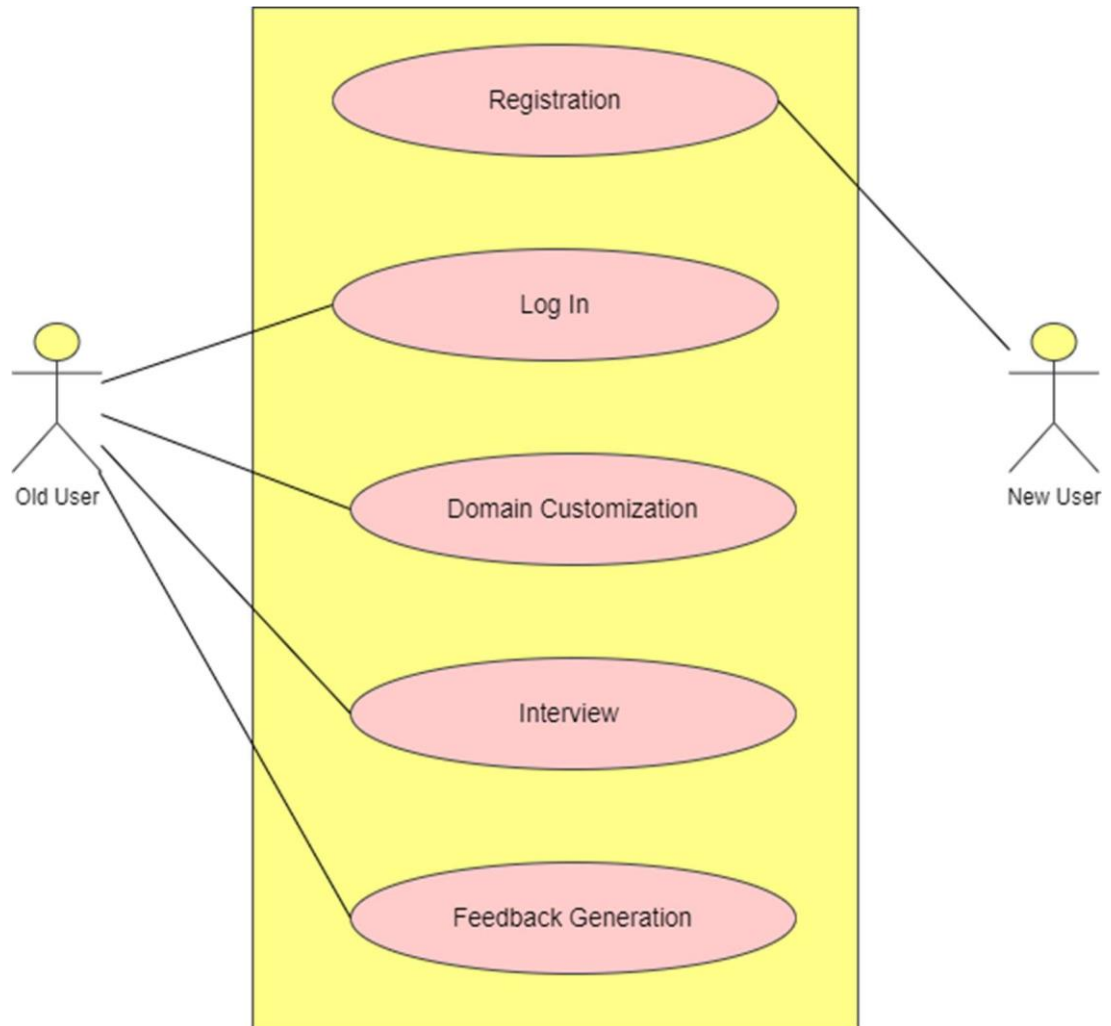


Fig. 4.7 Use Case Diagram of Mock Interview System

4.3.4 Use Case Description of Registration

Purpose:

The purpose is to register the user into the system who is new to the system.

Actor:

New User

Pre-condition:

None

Post-condition:

The user will successfully register to the system.

Basic Flow:

- The user will enter the name, email, phone, college details of themselves.
- The System will check the format of the details.
- If the details of the user are correct, then it will successfully register the user.

Alternate Flow:

If the details entered by the user are not in the right format, then it will return the error message to the user.

4.3.5 Use Case Description of Login

Purpose:

The purpose is to Login the user into the system.

Actor:

Old User

Pre-condition:

The user must be registered into the system.

Post-condition:

The user will successfully Logged in to the system.

Basic Flow:

- The user will enter the email and password.
- The System will check the email and password from the database.
- If the details of the user are correct, then it will successfully Logged in the user.

Alternate Flow:

If the details entered by the user are not correct, then it will return the error message to the user.

4.3.6 Use Case Description of Domain Customization

Purpose:

The purpose is to specify the domain and type of interview.

Actor:

Old User

Pre-condition:

The user must be logged in to the system.

Post-condition:

The domain and type of the interview are specified.

Basic Flow:

- The user will enter the type of interview i.e. HR or technical.
- The user will enter the domain of the interview.

Alternate Flow:

If the details entered by the user are not present then error message displayed.

4.3.7 Use Case Description of Interview

Purpose:

The purpose is to take the interview of the user.

Actor:

Old User

Pre-condition:

Domain and type of the interview should be specified.

Post-condition:

The interview will be successfully completed.

Basic Flow:

- The user will copy and paste the instruction to the inbox.
- The system will give questions to the user.
- Then the user will answer the questions.

Alternate Flow:

If there is no internet connection, then the interview cannot be successfully completed.

4.3.8 Use Case Description of Feedback Generation

Purpose:

The purpose is to generate feedback on the answers given by the users.

Actor:

Old User

Pre-condition:

Interview answers must be given by the user.

Post-condition:

The user will successfully get the feedback of the answers.

Basic Flow:

- The user will complete the interview.
- Then the system will give feedback to the user.

Alternate Flow:

If internet is not available, then feedback cannot be generated.

CHAPTER 5

PROPOSED WORK

5.1 Dataset Description

A dataset comprises similar sets of information that are made up of distinct elements but can be modified by a computer. In our dataset model, there are 1 main table i.e. of the user details.

Users Table: It stores the details of the user such as name, phone, email, password, college.

Table 2: Users Table Description.

Field	Type	Null	Key	Default
Name	Varchar(30)	No		NULL
Username	Varchar(10)	No	Primary Key	NULL
Phone	Number(10)	No		NULL
Email	Varchar(50)	No	Unique Key	NULL
Password	Varchar(20)	No		NULL
College	Varchar(30)	Yes		NULL

5.2 Technology Description

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

5.3 Approach Used

In the architecture of the AI-Enabled Mock Interview System, MongoDB serves as the database for its adaptability with a dynamic schema and scalability, allowing the system to efficiently manage evolving user data. The selection of React for the frontend is substantiated by its declarative UI, fostering a straightforward development of interactive interfaces through its component-based structure and the performance optimization achieved with its virtual DOM. Complementing this, Node.js is employed for the backend, providing a seamless JavaScript development environment across both frontend and backend, facilitating a unified language throughout the project. Node.js' non-blocking I/O model proves beneficial for real-time interactions in the mock interview system, ensuring responsive handling of simultaneous user actions. Additionally, Node.js leverages its extensive ecosystem via npm, contributing to the project's overall efficiency and flexibility. Together, the MongoDB, React, and Node.js creates cohesive and scalable foundation, streamlining the development process and enabling seamless interaction between the frontend and backend components in the AI-Enabled Mock Interview System.

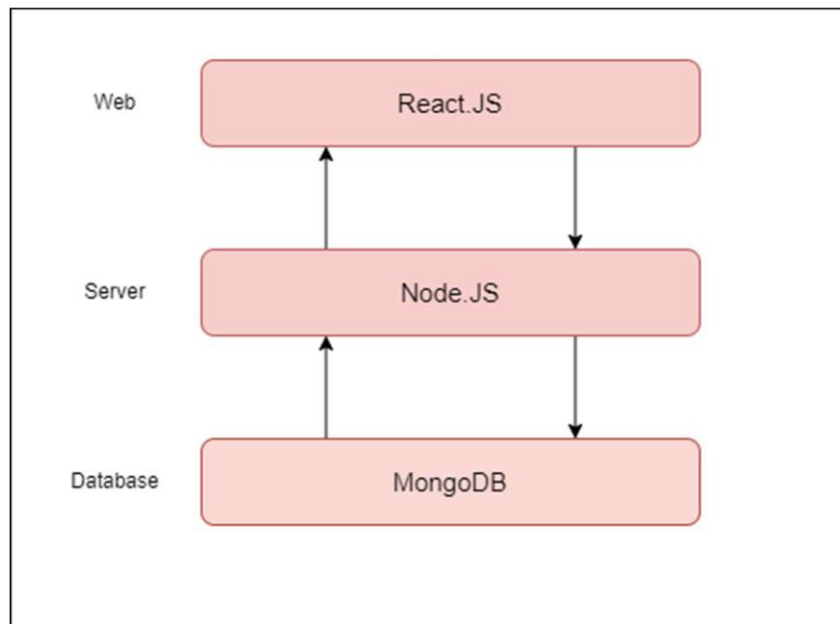


Fig. 5.1 Interconnection of Technologies

5.4 Dependencies Required

Axios: In this project version 1.4.0 of axios is used to simplify the process of sending asynchronous requests to a server and handling the responses. Axios is commonly employed

React is its simplicity and conciseness. It provides a clean and straightforward syntax for making HTTP requests compared to traditional methods.

- 5.4.1 **Chart.js:** Version 4.3.0 is used in this project. It is a popular JavaScript library for creating interactive and visually appealing charts in web applications. In React, integrating Chart.js involves installing the library, creating a React component, and utilizing the library to render charts.
- 5.4.2 **Concurrently:** Version 8.2.2 is used in this project. It is a Node.js package commonly used in the context of development workflows, especially with projects that involve both a frontend (e.g., React) and a backend (e.g., Node.js, Express). This package enables running multiple commands concurrently in a single terminal window. This can be particularly useful during development when you need to start both the frontend and backend servers simultaneously. It simplifies the process of managing different parts of your application that run on separate ports or even in different directories.
- 5.4.3 **Farmet-motion:** Version 10.12.16 is used in this project. Framer Motion is a popular animation library for React. It allows you to add smooth and declarative animations to your React components, enhancing the user experience.
- 5.4.4 **Mdb-react-ui-kit:** Version 6.1.0 is used in this project. MDB React UI Kit provides a set of ready-to-use Bootstrap components for React applications. It simplifies the process of building responsive and visually appealing user interfaces.
- 5.4.5 **React:** Version 18.2.0 is used in this project. React is a JavaScript library for building user interfaces. It enables the creation of reusable UI components and facilitates the development of single-page applications with efficient updates through its virtual DOM.
- 5.4.6 **React-chartjs-2:** Version 5.2.0 is used in this project. React Chartjs is a wrapper for the Chart.js library, making it easier to integrate dynamic and interactive charts into React applications. It provides React components that correspond to different Chart.js chart types.
- 5.4.7 **React-dom:** Version 18.2.0 is used in this project. React DOM is a package that provides methods for interacting with the DOM (Document Object Model). It's a necessary part of React to render components into the DOM, handle events, and update the user interface.
- 5.4.8 **React-icons:** Version 4.10.1 is used in this project. React Icons is a library that offers a collection of popular icon libraries as React components. It simplifies the process of adding icons to your React

application without the need for additional dependencies.

- 5.4.9 **React-router-dom:** Version 6.13.0 is used in this project. React Router DOM is a library for handling navigation in React applications. It enables the creation of dynamic and responsive Single Page Applications (SPAs) by allowing the mapping of components to different routes.
- 5.4.10 **React-scripts:** Version 5.0.1 is used in this project. React Scripts is a set of scripts provided by Create React App (CRA) to facilitate the development and build processes. It abstracts away configuration complexities, allowing developers to focus on building React applications.
- 5.4.11 **React-speech-recognition:** Version 3.10.0 is used in this project. React Speech Recognition is a library that allows integration of speech recognition capabilities into React applications. It provides components and hooks to easily capture and process speech input.
- 5.4.12 **Redux: Version 4.2.1** is used in this project. Redux is a state management library for React applications. It helps manage the state of the application in a predictable and centralized way, making it easier to handle complex state logic.
- 5.4.13 **Redux-thunk:** Version 2.4.2 is used in this project. Redux Thunk is a middleware for Redux that enables the handling of asynchronous actions. It allows you to dispatch functions as actions, providing flexibility when dealing with side effects like API calls.
- 5.4.14 **Twind:** Version 0.16.19 is used in this project. wind is a utility-first CSS framework for React. It allows you to write styles in your JavaScript/JSS files, providing a highly efficient way to manage styles in React applications.
- 5.4.15 **Web-vitals:** Version 2.1.4 is used in this project. Web Vitals is a set of metrics that help measure the performance and user experience of a web application. It includes tools and libraries for capturing and reporting essential performance metrics.

- 5.4.16 **React-scripts:** Version 5.0.1 is used in this project. React Scripts is a set of scripts provided by Create React App (CRA) to facilitate the development and build processes. It abstracts away configuration complexities, allowing developers to focus on building React applications.
- 5.4.17 **React-speech-recognition:** Version 3.10.0 is used in this project. React Speech Recognition is a library that allows integration of speech recognition capabilities into React applications. It provides components and hooks to easily capture and process speech input.
- 5.4.18 **Redux:** Version 4.2.1 is used in this project. Redux is a state management library for React applications. It helps manage the state of the application in a predictable and centralized way, making it easier to handle complex state logic.
- 5.4.19 **Redux-thunk:** Version 2.4.2 is used in this project. Redux Thunk is a middleware for Redux that enables the handling of asynchronous actions. It allows you to dispatch functions as actions, providing flexibility when dealing with side effects like API calls.
- 5.4.20 **Twind:** Version 0.16.19 is used in this project. wind is a utility-first CSS framework for React. It allows you to write styles in your JavaScript/JSX files, providing a highly efficient way to manage styles in React applications.
- 5.4.21 **Web-vitals:** Version 2.1.4 is used in this project. Web Vitals is a set of metrics that help measure the performance and user experience of a web application. It includes tools and libraries for capturing and reporting essential performance metrics.

5.5 Algorithms and Flowcharts

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

5.5.1 Sign up():

This function will make the user to sign up to the system.

Algorithm

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

Flowchart

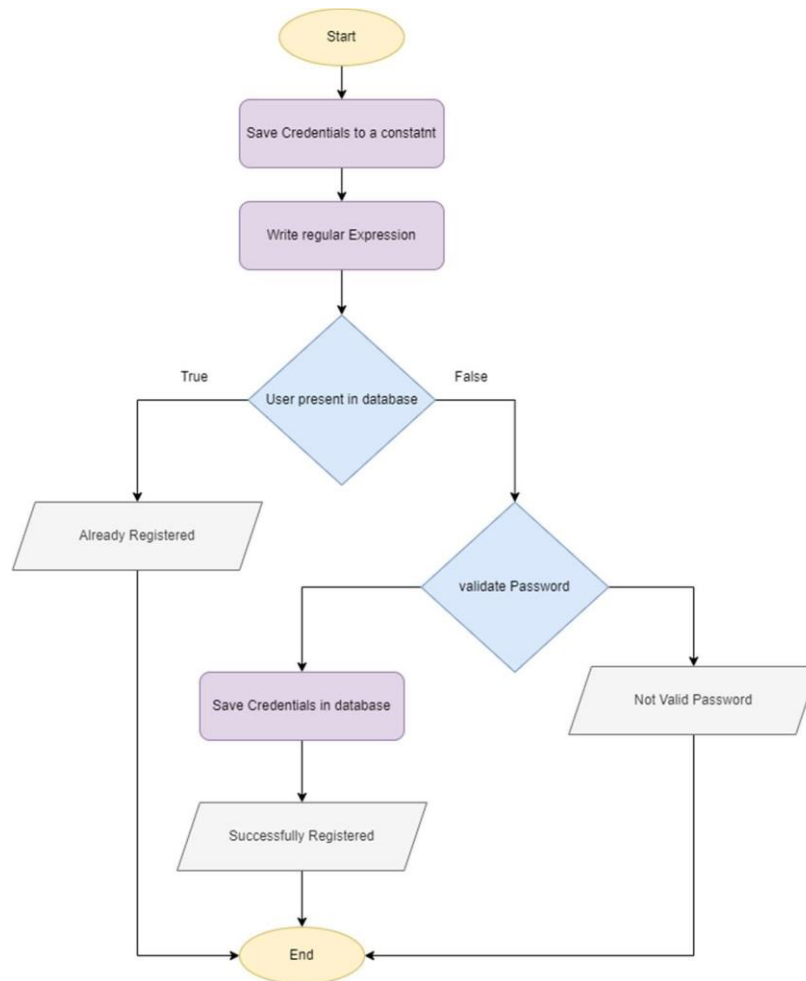


Fig. 5.2 Flowchart of signup process

5.5.2 Login

Log In():

This function will make the user to logged in to the system.

Algorithm

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

Flowchart

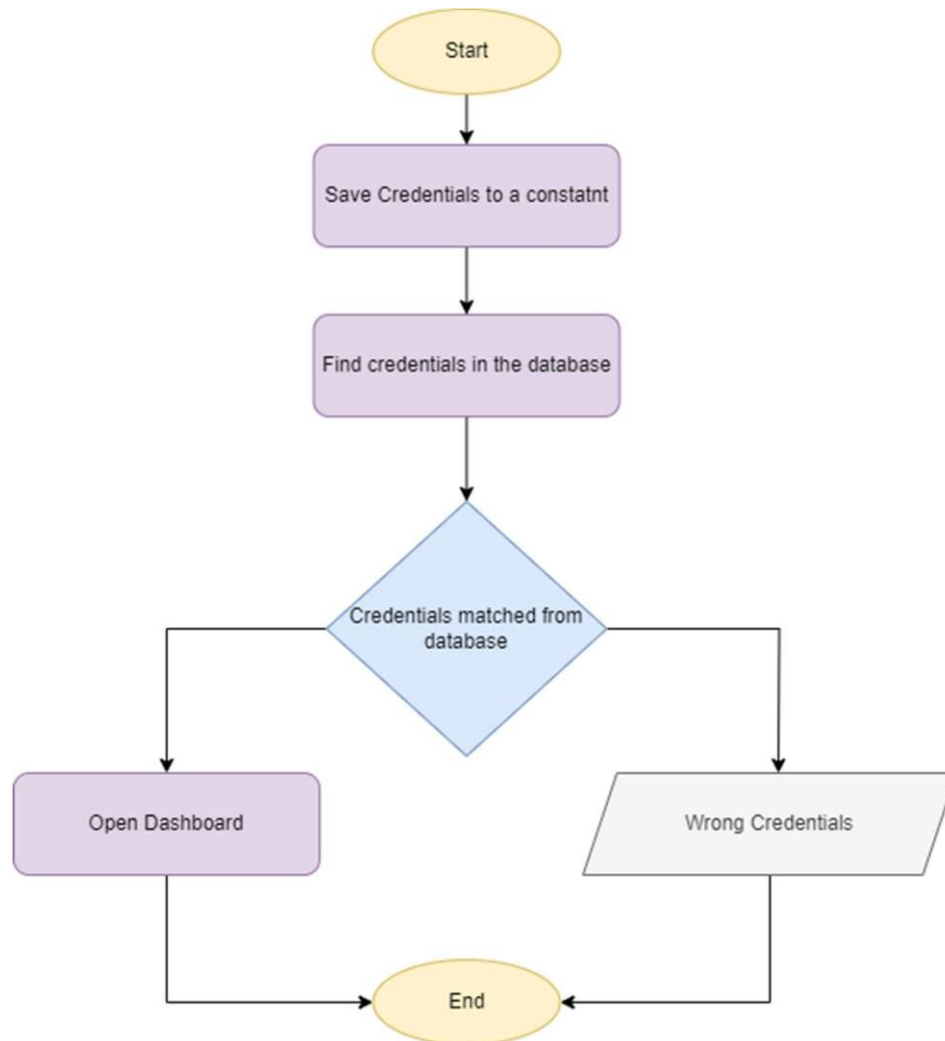


Fig 5.3 Flowchart of Login

5.5.3 Database Connectivity ():

This function will connect the server to the database.

Algorithm

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

Flowchart

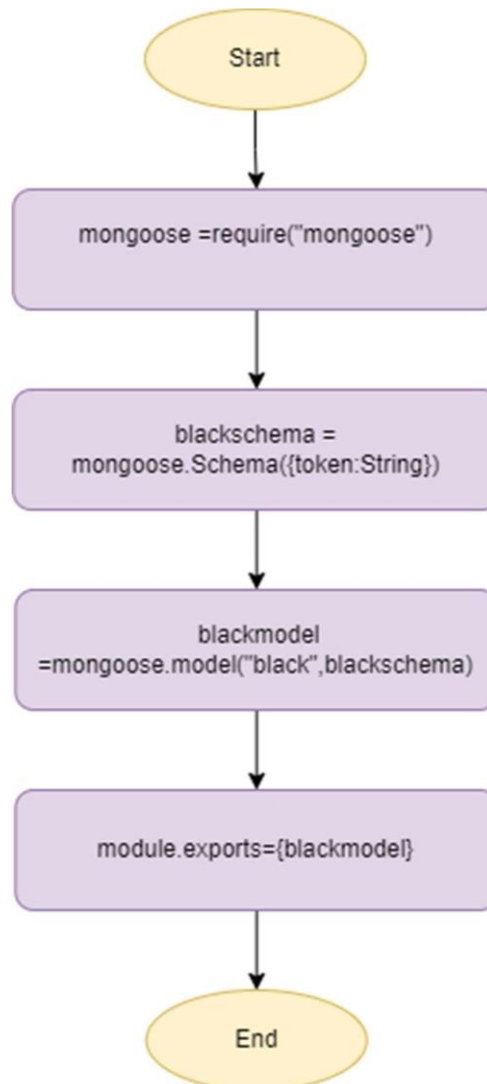


Fig 5.4 Flowchart of Database Connectivity

5.5.4 API Connectivity ():

This function will connect our website to the openAPI.

Algorithm

- 1 save the url of mongoDB to the variable named as mongourl
- 2 save the port number
- 3 add token key and refresh token key
- 4 add the open api key

Flowchart

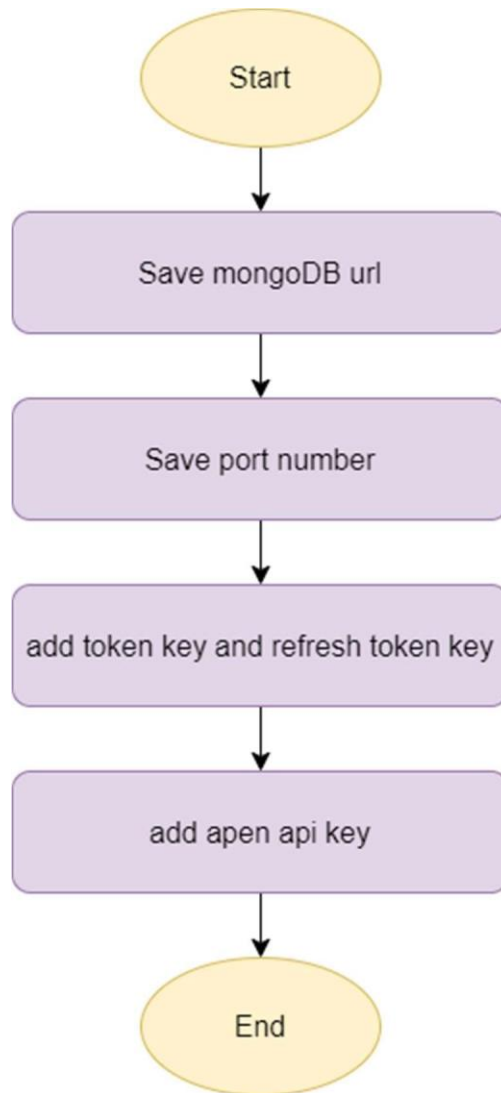


Fig 5.5 Flowchart of API Connectivity

5.5.5 Feedback Generation ():

This function will generate the feedback.

Algorithm

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

Flowchart

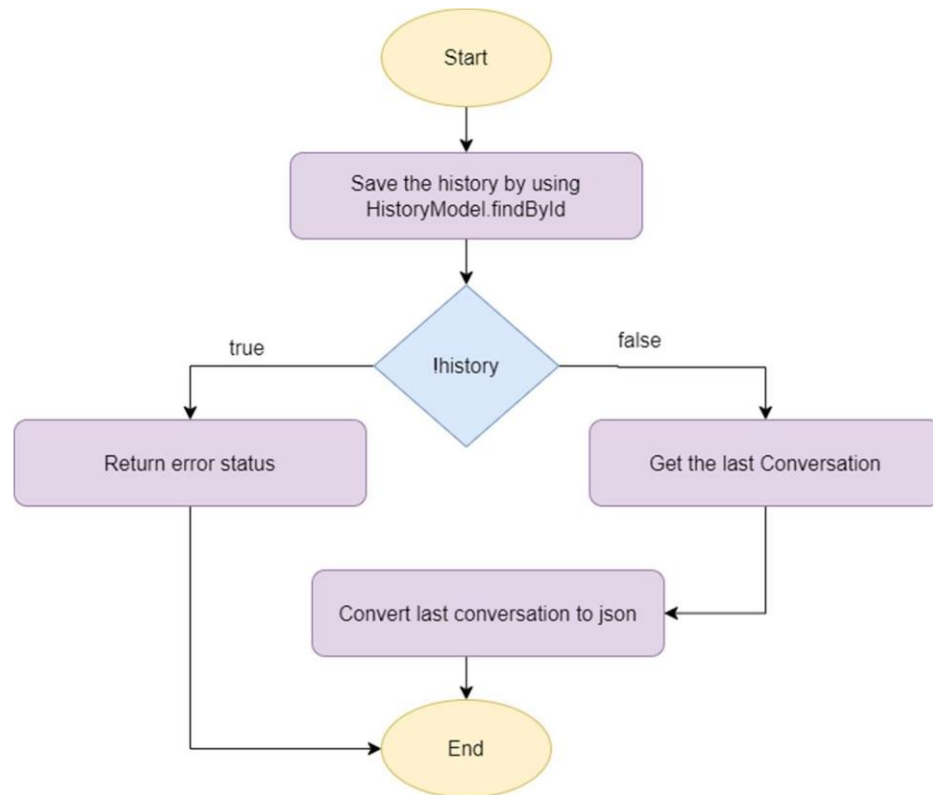


Fig. 5.6 Flowchart of Feedback Generation

CHAPTER 6

RESULTS

6.1 Screens and Explanations

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

Screen 1: Home Page

Screen 1 is the home page of the website which displays the basic information about the MockMaster such as why interview preparation is important and what our system will provide. From this home page you can log in or register to the system to start your mock interview. A home screen, often referred to as the homepage, is the main entry point of a website. It serves as the front door, welcoming visitors and guiding them to the information they need.

Key Elements of a Homepage

Header and Navigationm Logo: Positioned typically in the top left corner, serving as a clickable link back to the homepage.

Navigation Menu: Provides links to important sections of the website such as About, Services, Products, Contact, etc. It should be intuitive and easy to use.

1. Content Sections:

Testimonials or Social Proof: Quotes, reviews, or case studies from satisfied customers to build credibility.

Latest News or Blog Posts: Showcasing recent updates or articles to keep content fresh and engaging.

2. Footer:

- **Contact Information:** Address, phone number, email, and possibly a contact form.
- **Social Media Links:** Icons linking to the site's social media profiles.
- **Additional Links:** Links to privacy policy, terms of service, FAQs, and other relevant information.

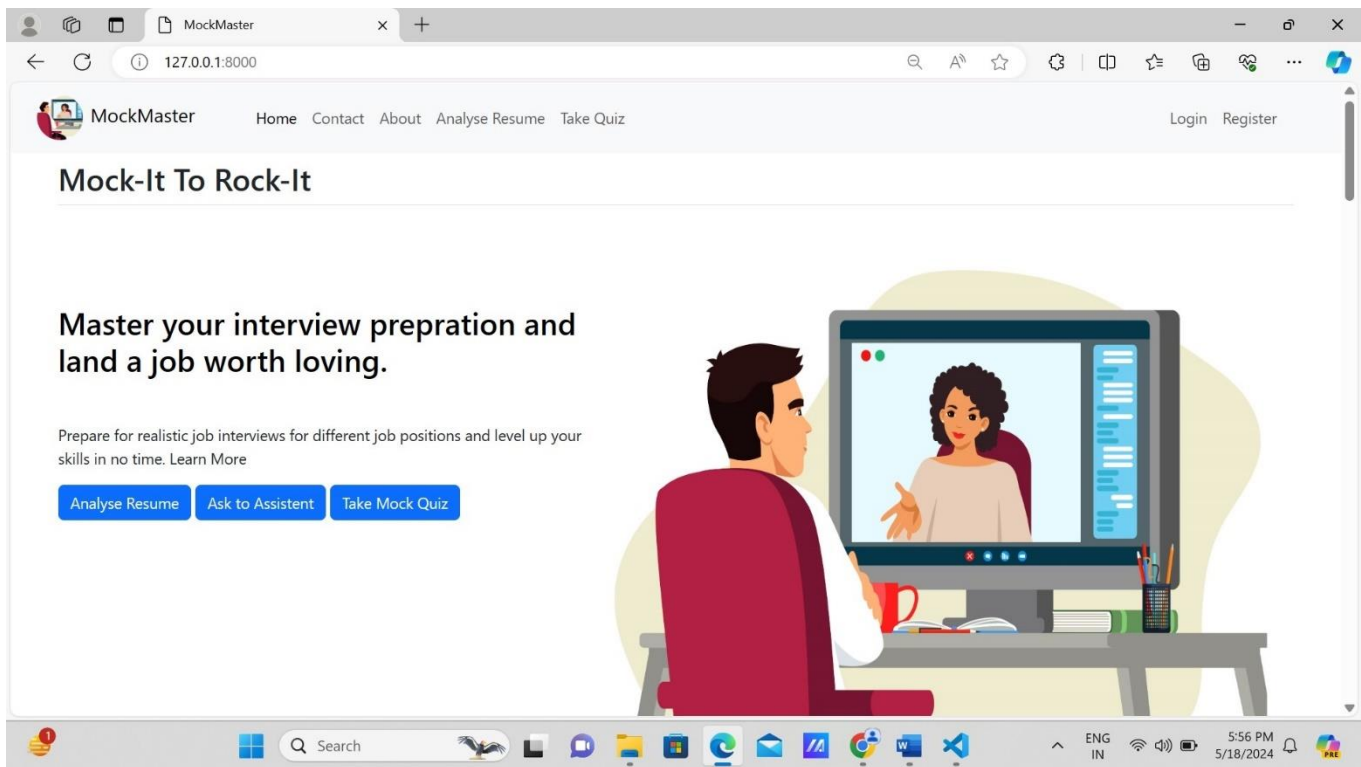


Fig. 6.1 Home Page

Screen 2: Login and Registration Screen

Screen 2 is the log in and the registration page. Where if the user is new to the system, then he or she can register themselves to the system by providing the name, email and password. Password validation is also done at the time of registration. If the user is not new or already registered to the system, then he or she can directly log in to the system by providing some credentials such as email and password. The user can toggle between the login and the registration page.

Key Elements of a Registration Page

1. Form Fields:

- **Full Name:** Field for entering the user's full name.
- **Username:** Unique identifier for the user (optional on some sites).
- **Email Address:** Used for communication and login.
- **Password:** Secure field for entering a password, often with strength indicators.
- **Confirm Password:** Additional field to ensure the password is entered correctly.

2. Action Buttons:

- **Register/Sign Up:** Button to submit the registration form and create a new account.

3. Additional Information:

- **Terms and Conditions:** Checkbox to agree to the site's terms of service and privacy policy.
- **Optional Fields:** Fields for additional information such as phone number, address, etc.

Key Elements of a Login Page

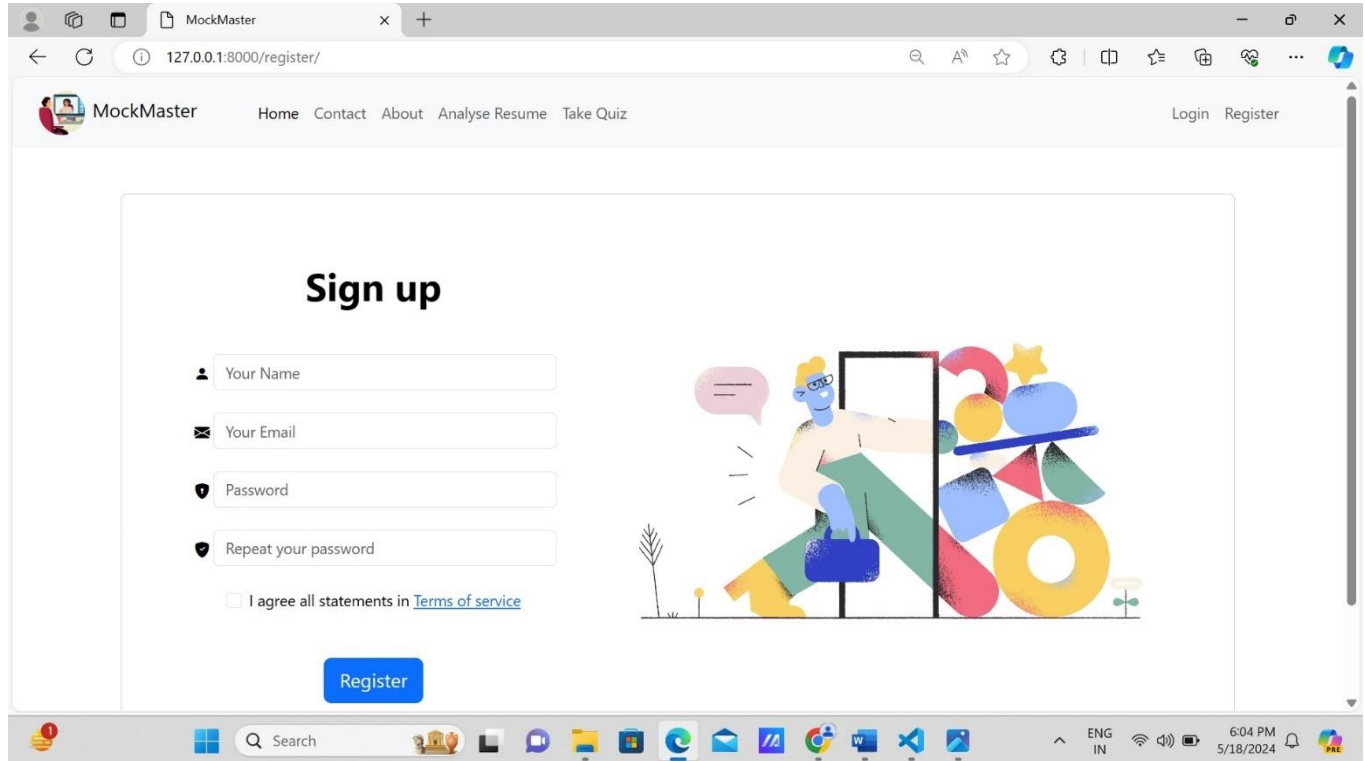
1. Form Fields:

- a. **Username/Email:** Field where users input their registered email address or username.
- b. **Password:** Secure field for entering their password. Often includes an option to show/hide the password.

2. Action Buttons:

- a. **Login:** Button to submit the form and authenticate the user.
- b. **Remember Me:** Checkbox allowing users to stay logged in for future sessions

- c. **Forgot Password?:** Link to a page where users can recover or reset their password if they've forgotten it.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/register/". The page title is "MockMaster". The navigation bar includes links for "Home", "Contact", "About", "Analyse Resume", and "Take Quiz", along with "Login" and "Register" buttons. The main content area is titled "Sign up" and contains a registration form with the following fields: "Your Name", "Your Email", "Password", and "Repeat your password". Below these fields is a checkbox labeled "I agree all statements in [Terms of service](#)". A blue "Register" button is positioned at the bottom of the form. To the right of the form is a colorful illustration of a person with a speech bubble, surrounded by various geometric shapes and symbols.

Fig. 6.2 Registration Screen

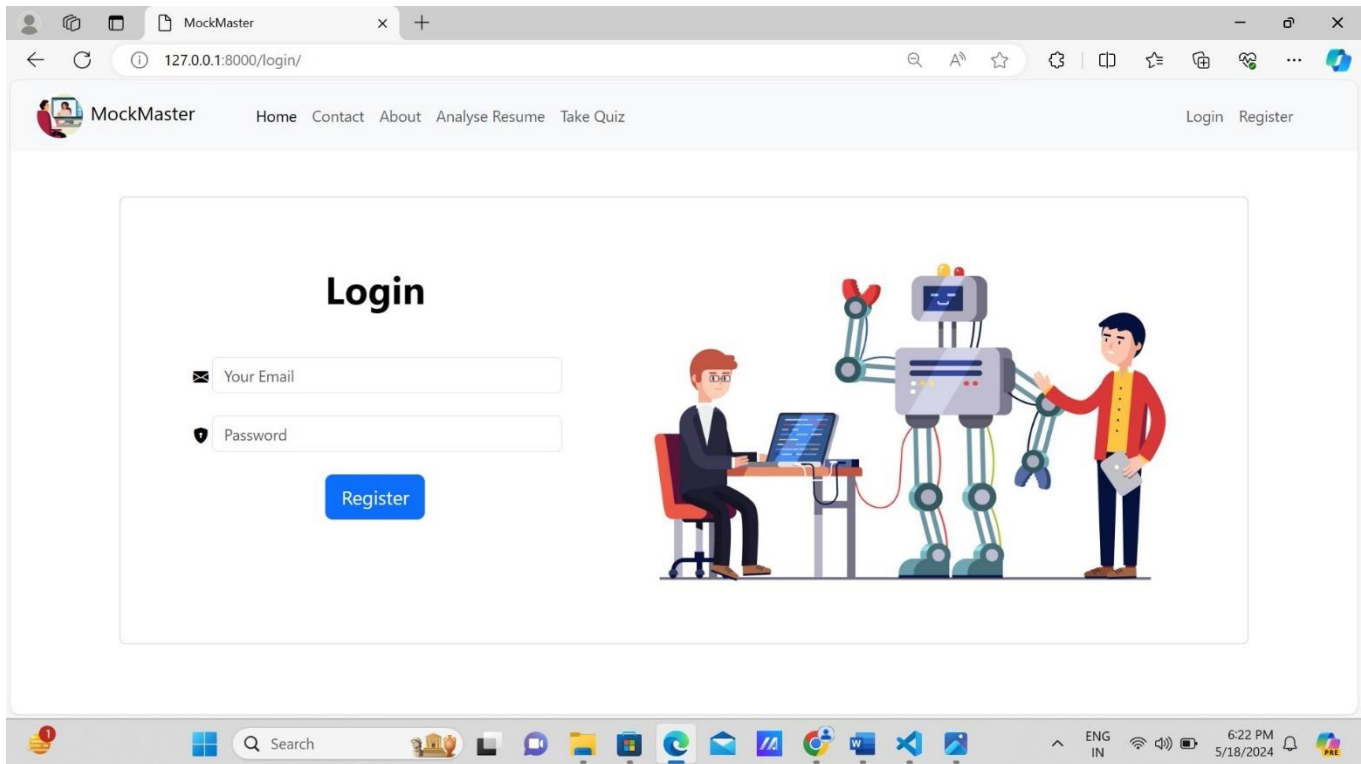


Fig 6.3 Login Page

Screen 3: Analyze Resume

Screen 3 Analyzing resumes through a website typically involves using a combination of automated tools and manual processes to evaluate the qualifications, experience, and skills of job applicants. Here's an overview of how resume analysis works, key elements involved and best practices. Here we upload our resume and it analyzes our resume and give score according to our resume . And it also recommend the best suited job for us according to our resume . Our project analuze resume and give score to us.

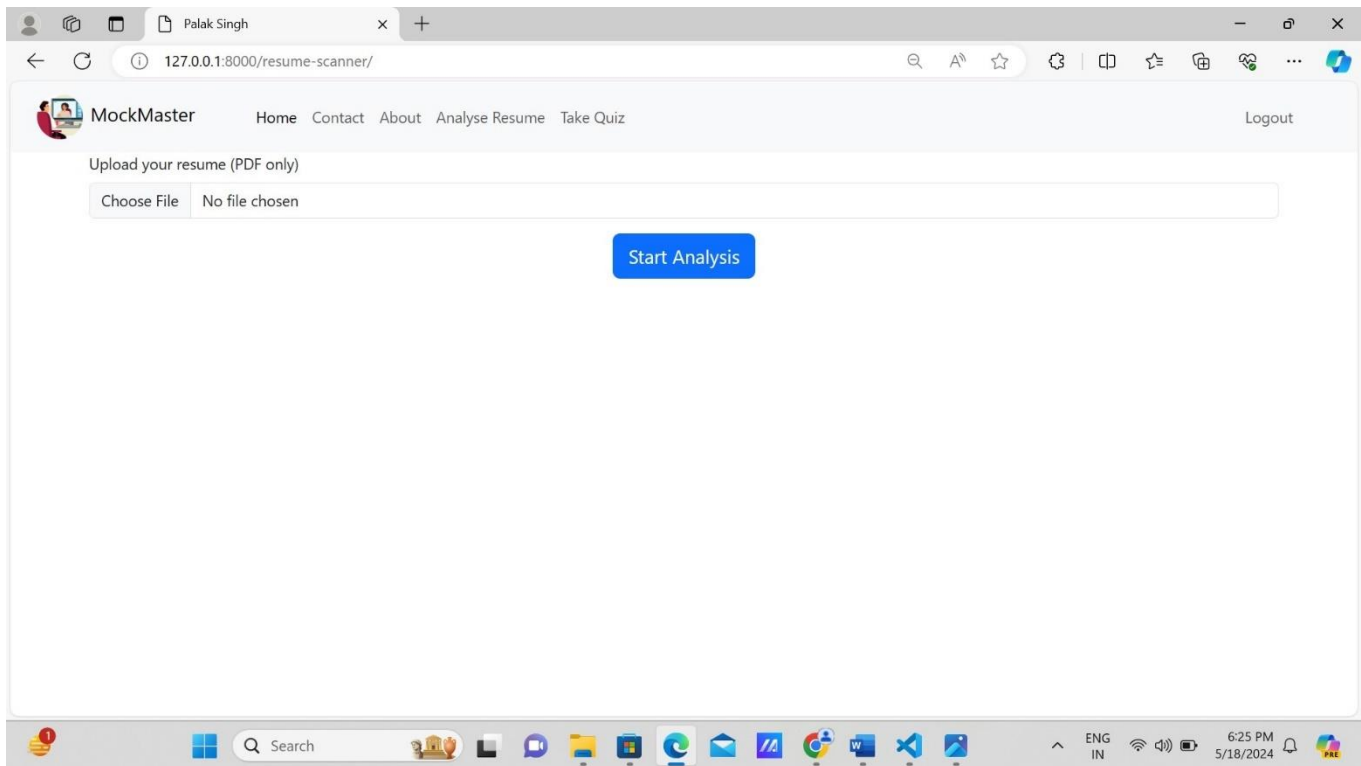


Fig. 6.4 Resume Analyze Screen

Screen 4: Detailed Resume Analysis

In today's competitive job market, the importance of a well-crafted resume cannot be overstated. With the increasing use of automated systems to screen resumes, it is crucial for candidates to ensure their resumes are optimized and tailored to specific job descriptions. The Detailed Resume Analysis Screen (DRAS) is a tool designed to assist both job seekers and recruiters by providing in-depth analysis and feedback on resumes. This project report introduces the DRAS, outlining its purpose, features, and the benefits it offers.

Purpose

The primary goal of the Detailed Resume Analysis Screen is to improve the effectiveness and precision of resume evaluations. This tool is intended to:

- Assist job seekers in enhancing their resumes to better match job requirements.
- Aid recruiters in quickly identifying the most suitable candidates.

- Streamline the resume screening process by automating the initial evaluation phase.

Features of the Detailed Resume Analysis Screen

1. Keyword Matching:

- **Function:** Compares the resume content with the job description to identify relevant keywords and phrases.
- **Benefit:** Ensures that the resume is tailored to the specific job, increasing the chances of passing through Applicant Tracking Systems (ATS).

2. Skills Analysis:

- **Function:** Evaluates the listed skills against the job requirements, highlighting any gaps or mismatches.
- **Benefit:** Helps candidates understand which skills they need to highlight or acquire to be a better fit for the job.

3. Experience Evaluation:

- **Function:** Analyzes the candidate's work experience, focusing on relevance, duration, and achievements.
- **Benefit:** Provides insights into how well the candidate's experience aligns with the job role.

4. Formatting and Presentation:

- **Function:** Assesses the resume's format, structure, and readability.
- **Benefit:** Ensures that the resume is professionally presented and easy to read, which is crucial for making a positive first impression

5. Language and Grammar Check:

- **Function:** Detects and suggests corrections for grammatical errors, spelling mistakes, and inappropriate language usage.
- **Benefit:** Enhances the overall quality and professionalism of the resume.

6. ATS Compatibility Check:

- **Function:** Reviews the resume for common issues that might prevent it from being correctly parsed by ATS.
- **Benefit:** Increases the likelihood that the resume will be successfully processed by automated systems used by recruiters.

The Detailed Resume Analysis Screen represents a significant advancement in the recruitment process, offering comprehensive analysis and actionable insights to enhance resume quality. By leveraging advanced algorithms and machine learning techniques, the DRAS not only streamlines the hiring process but also empowers job seekers to present their qualifications in the best possible light. As the job market continues to evolve, tools like the DRAS will play an increasingly vital role in connecting the right candidates with the right opportunities.

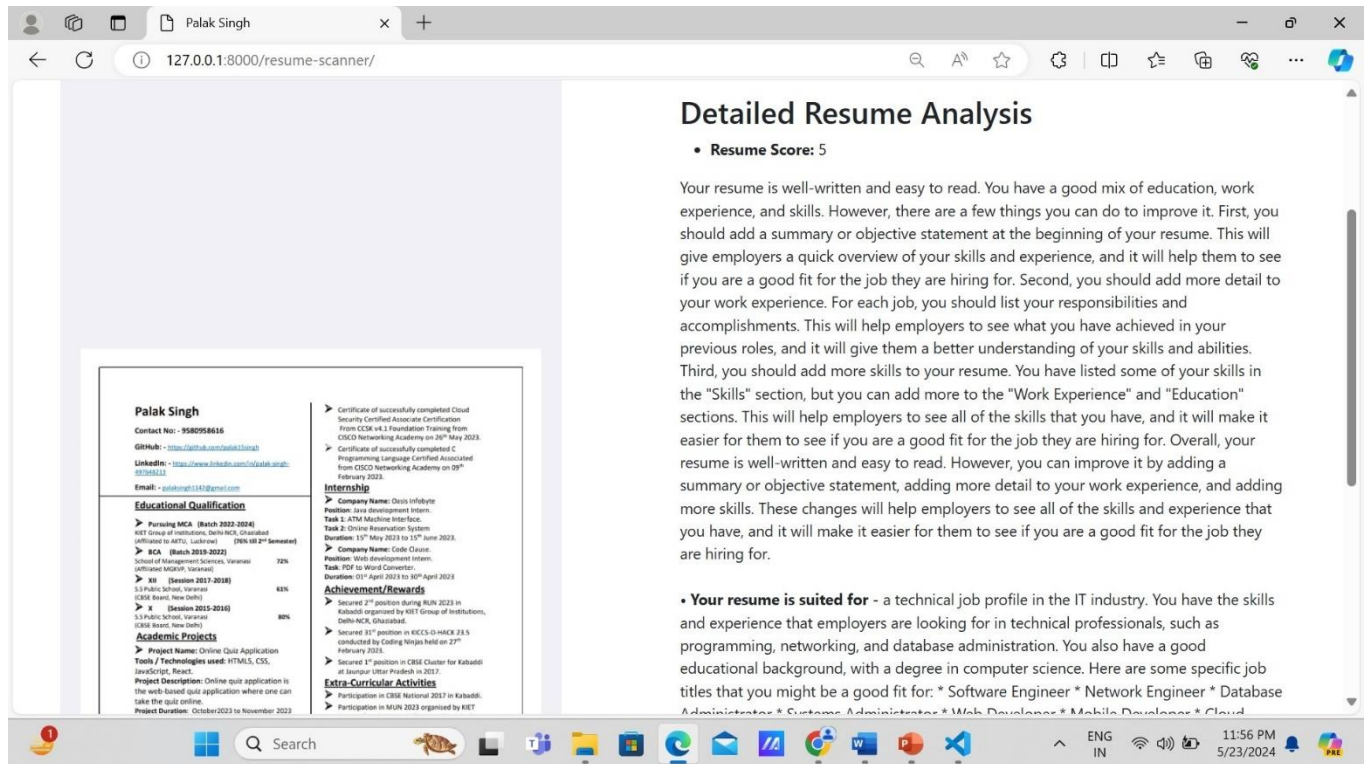


Fig. 6.5 Detailed Resume Analysis Screen

Screen 5: About Us Screen

Screen 5 The "About Us " screen is a specialized component within an application or website, designed to cater to the needs and interests of developers. This screen provides detailed information about the development team, the technology stack, and the development practices used within the organization. This project report introduces the "About Us for Developers" screen, outlining its purpose, features, and the benefits it provides to developers and the organization.

The primary goal of the "About Us " screen is to engage the developer community by offering transparency about the technical aspects of the organization.

The "About Us for Developers" screen is a crucial component for organizations looking to engage and attract the developer community. By providing detailed insights into the development team, technology stack, and development practices, this screen fosters transparency, builds trust, and encourages collaboration. As the role of developers continues to grow in importance, the "About Us for Developers" screen will remain a key tool for organizations to connect with and support the developer community, driving innovation and growth.

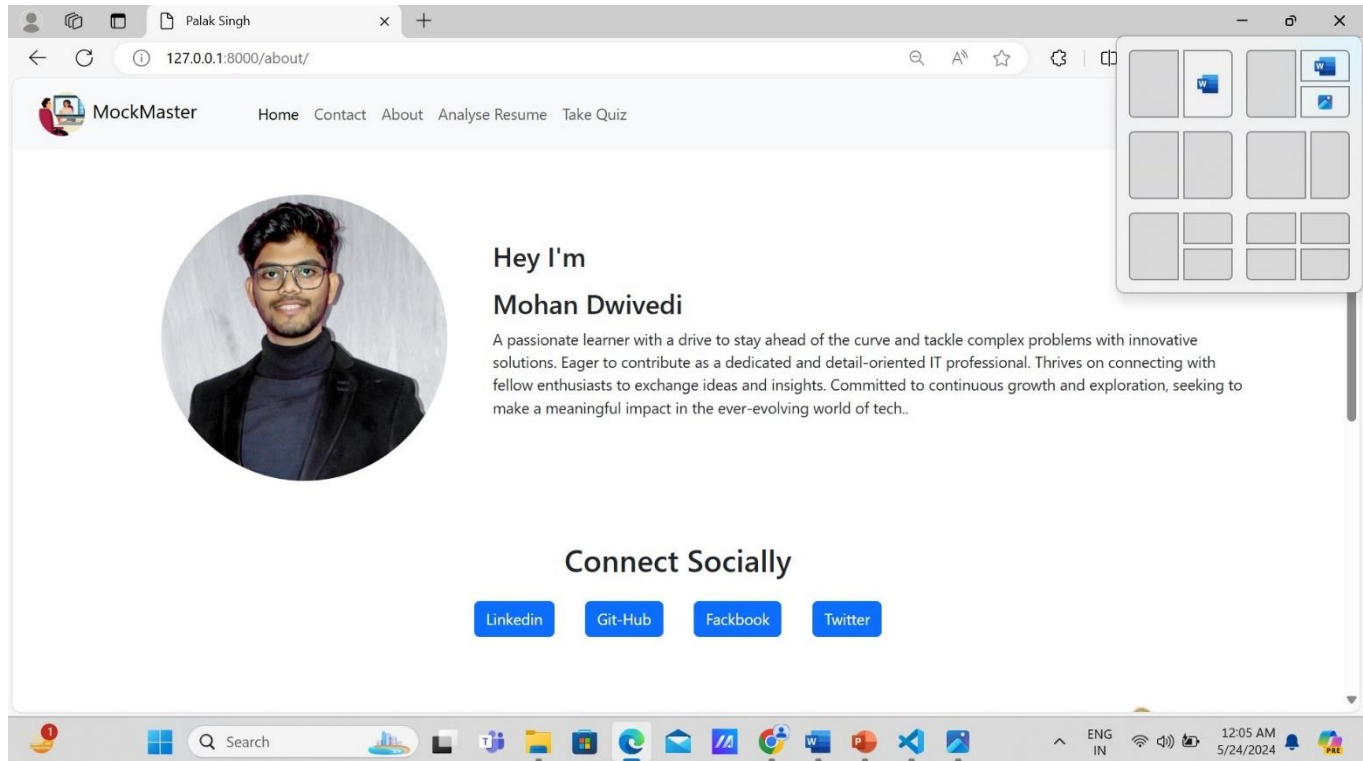


Fig. 6.6 About Us Screen

Screen 6: Interview Customization Screen

Screen 6 is post-process of the Start New Interview, once we click on the button “Start New Interview” a pop up is displayed that is our screen 4 which asks about the title of the interview that the user wants to save with. It also includes the type of interview the user wants to take such as technical interview or the HR interview. At the end it will ask for an interview track or the domain of technology in which the user wants to give the interview in.

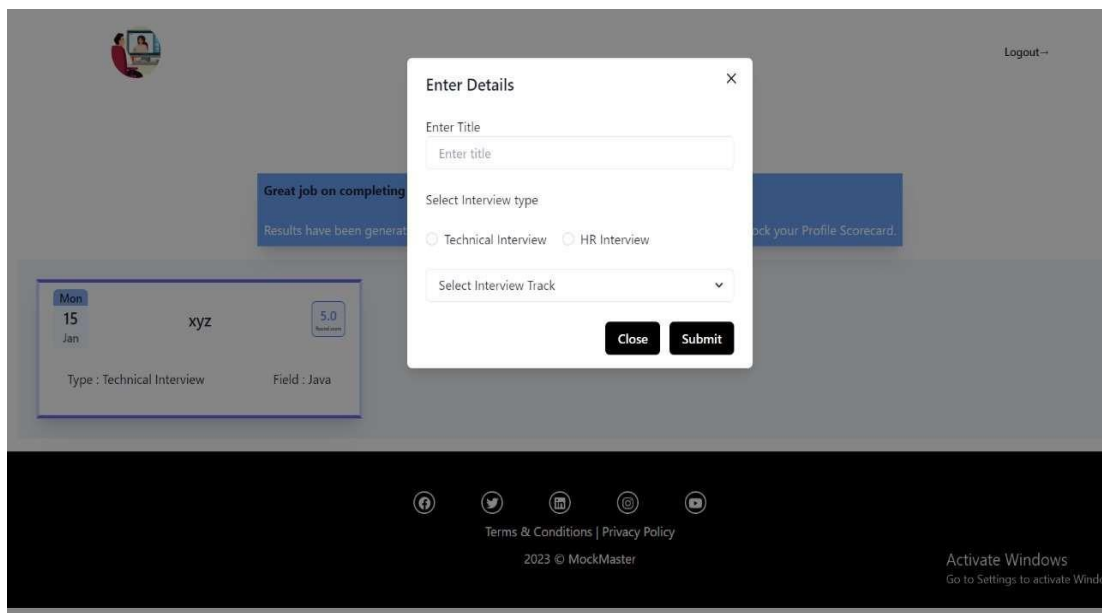


Fig. 6.7 Interview Customization Screen

Screen 7: Interview Screen

Screen 7 is the screen where the actual interview takes place. To start the interview the user, have to follow instructions given at the left-hand side and copy the text given in the instruction to be copied and paste it in the chat box. And send it to the system the system will use the open ai to ask the questions to the user on that user specified domain and type. We will also provide audio-based chat just as in WhatsApp and other chatting tools. And the user can also start a fresh interview at any point of time if he or she wants to do so.



Fig. 6.8 Interview Screen

Screen 8: Response Screen

Screen 8 will show the response from the open ai which includes the questions based on the domain given by the user. Then the user will provide the answer either by writing it or by sending the audio. Then the open ai will give the feedback to the user through which user can check if he or she gave the right answer or not. Even if the user does not know the answer the ai will provide the answer to the user so that he or she can learn from the response of the system.



Logout--

Follow the below steps to conduct the interview:-

Step 1: Copy the below prompt and send to start the interview -

You are an interviewer. Ask me 3 questions related to , one after the other. You should go to the next question only after I give an answer to the already asked question. Give me feedback at the end and give me rating out of 10

Step 2: Record or Type your answer to the question asked and send

Step 3: Once all the questions are answered, send the below prompt to get your feedback and score -

Step 4: Click on Finish Interview to save your Interview.

Step 5: Copy the feedback received and submit it along with scores in the Form shown.

Please answer the first question.

One significant difference between Abstraction and Encapsulation in Java is how they represent data. Abstraction represents only useful data, whereas encapsulation wraps data and codes for necessary information. Moreover, it helps developers to easily organize the whole code.1

Great answer! You mentioned the key differences between abstraction and encapsulation in Java. Abstraction focuses on providing a simplified view of an entity by hiding unnecessary details, while encapsulation is about bundling data and methods together to ensure data integrity and access control. Now, let's move on to the second question. Can you explain the concept of multithreading in Java?

Send Finish Interview

Fig. 6.9 Response Screen

Screen 9: Contact Screen

Screen 9 A "Contact Us" screen is a crucial element of any web application, providing users with a direct way to get in touch with the organization for support, inquiries, feedback, or other purposes. Here's a detailed overview of the essential elements, best practices, and examples of effective "Contact Us" screens:

Key Elements of a "Contact Us" Screen

Contact Form:

- **Name Field:** Input for the user's full name.
- **Email Address:** Field to capture the user's email address for correspondence.
- **Phone Number:** Field where phone number required.
- **Message:** Text area for users to write their message or inquiry.
- **Submit Button:** Button to submit the form, often labeled "Send," "Submit," or "Contact Us."

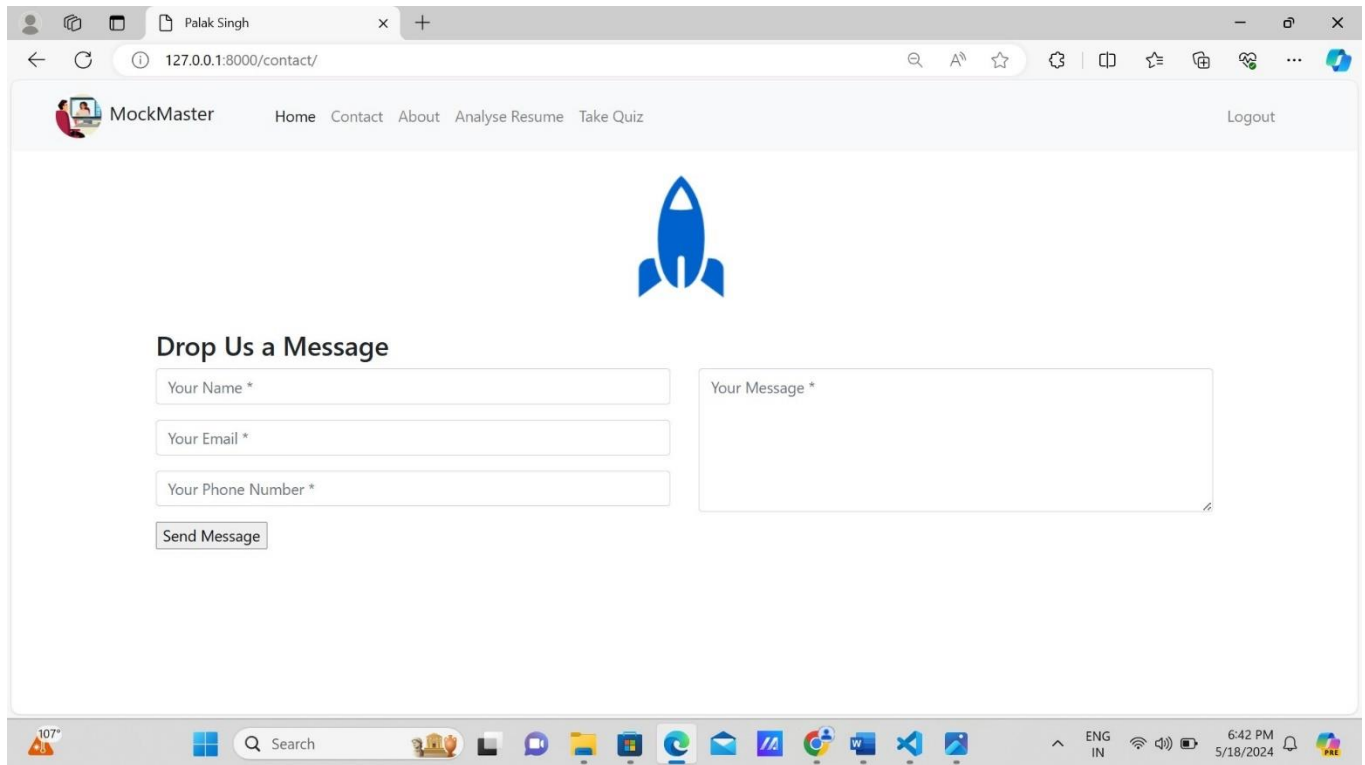


Fig. 6.10 Response Screen

Screen 10: Quiz Screen

Screen 10 A "Take Quiz" feature in a web application allows users to participate in quizzes for various purposes such as education, entertainment, assessment, or surveys. Here's a detailed overview of the essential elements, best practices, and examples of effective quiz features:

Key Elements of a "Take Quiz" Feature

1. User Interface:

- **Welcome Screen:** Introduction to the quiz, including its purpose, instructions, and estimated time to complete.
- **Question Format:** Various types of questions such as multiple-choice, true/false, short answer, and matching.
- **Navigation Controls:** Buttons for moving to the next question, previous question, and submitting the quiz.

Question Types:

1. **Multiple-Choice Questions:** Users select one or more answers from a list of options.

- **True/False Questions:** Users choose between two options to indicate the truthfulness of a statement.

- **Feedback and Scoring:**

- **Instant Feedback:** Immediate response indicating whether the answer was correct or incorrect (optional).

- **Final Score:** Displayed at the end of the quiz, showing the user's performance.

- **Explanations:** Detailed explanations for the correct answers, enhancing learning.

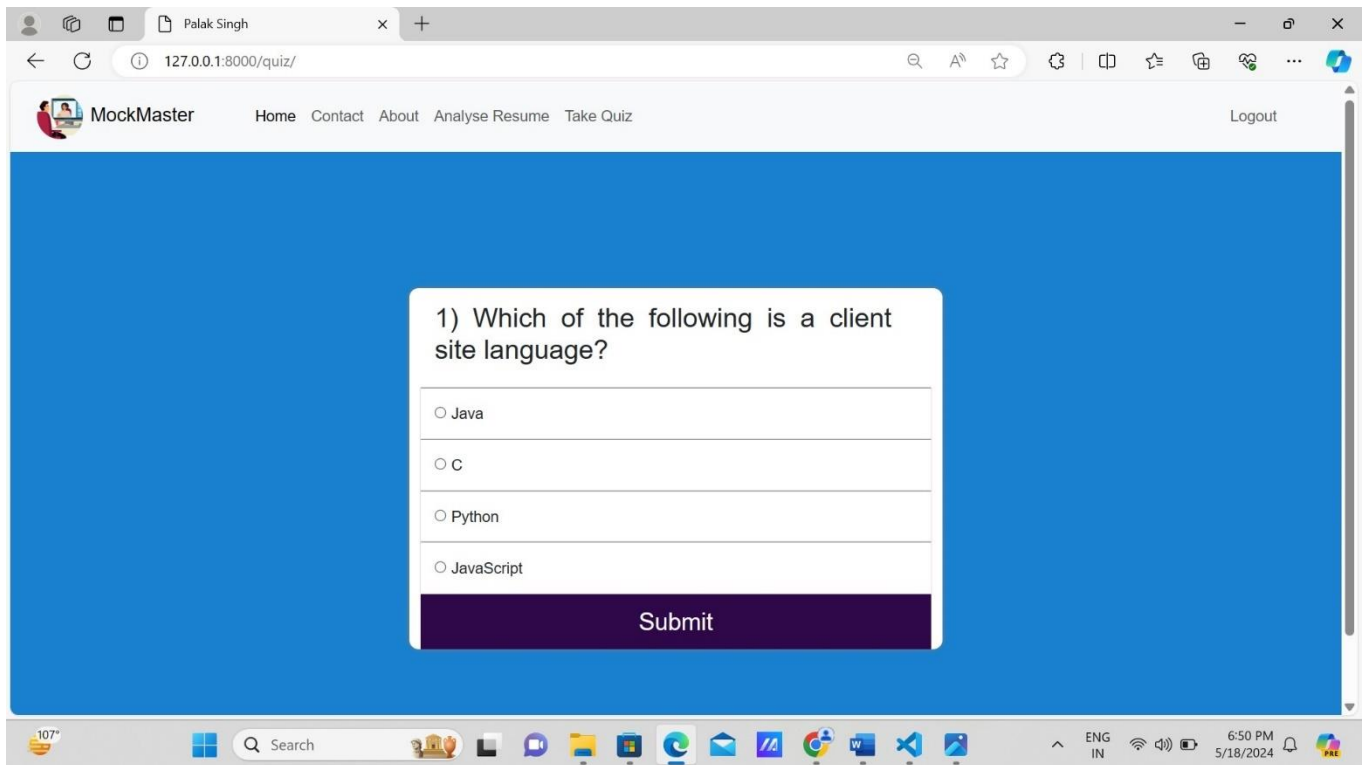


Fig 6.11 Take quiz Screen

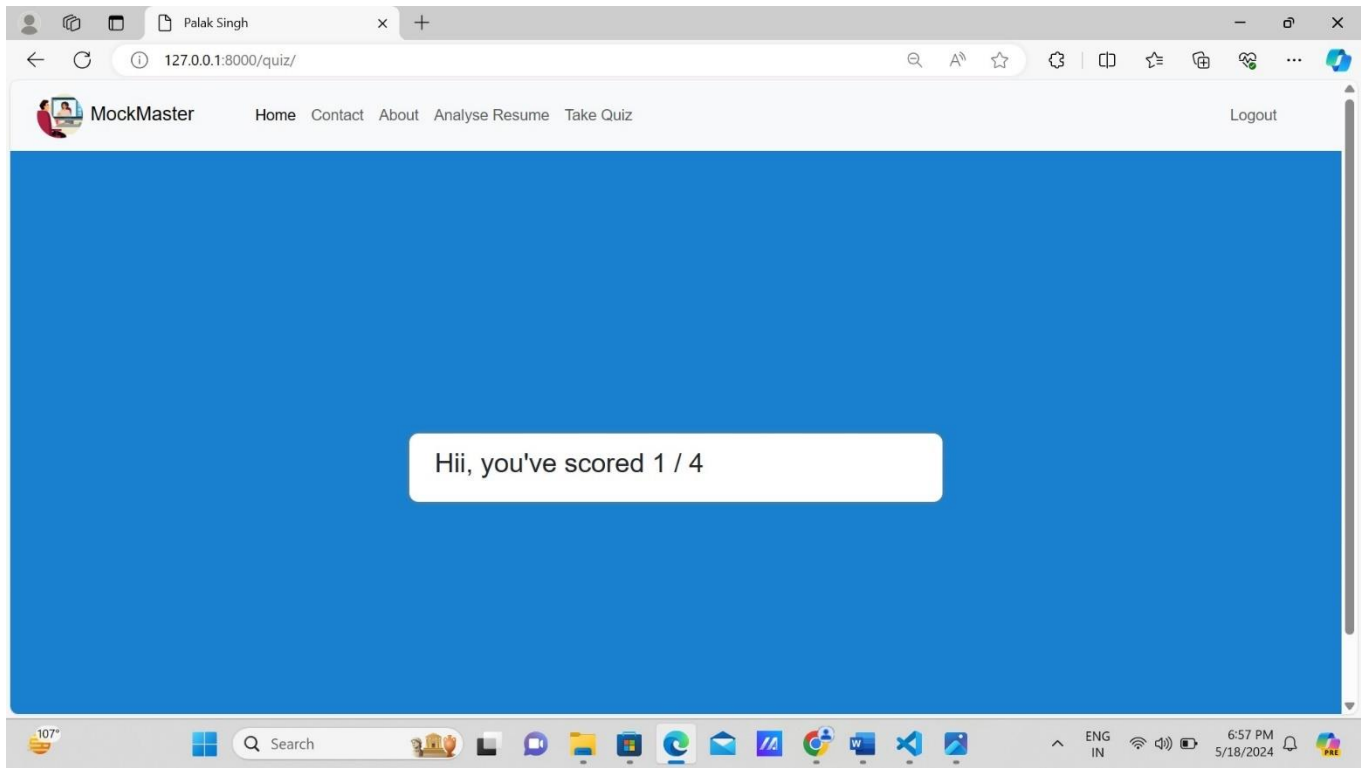


Fig 6.12 Score Screen

CHAPTER 7

DISCUSSION

The Discussions section of this report delves into crucial aspects of the AI-Enabled Mock Interview System, shedding light on the strategic choices and implications for user experience, scalability, and future developments. The technological integration within the MERN stack is highlighted, emphasizing the deliberate use of MongoDB for its flexibility and scalability, React for its declarative UI, and Node.js for its non-blocking I/O, crucial for real-time interactions.

7.1 Performance

The success of the AI-Enabled Mock Interview System is contingent upon the robustness and efficiency of the underlying AI model responsible for generating interview questions and providing feedback.

- 7.1.1 The Performance of the system based on the version of the open ai used as we had used the api for the question generation and feedback generation.
- 7.1.2 Internet connectivity of the user also affect the performance of the system.
- 7.1.3 If the user provides irrelevant information for the questions, then interview will never be the accurate and professional.
- 7.1.4 Versions of the modules used in node.js and react.js will also affect the performance based on their functionalities and optimality.

Creating an AI-enabled interview web application can significantly enhance the recruitment process by automating various aspects of candidate assessment and providing valuable insights. Here's an overview of the key features, best practices, and examples of AI-enabled interview applications:

Key Features of an AI-Enabled Interview Web Application

Automated Resume Screening:

- **Resume Parsing:** Extracts relevant information from resumes and matches it against job requirements.

- **Keyword Matching:** Identifies key skills and experiences that align with the job description.

2. Interview Scheduling:

- **Automated Scheduling:** Allows candidates to select interview slots based on availability, reducing the need for manual coordination.
- **Calendar Integration:** Syncs with calendars to ensure availability and send reminders.

3. Virtual Interview Platform:

- **Video Interviews:** Facilitates live or pre-recorded video interviews.
- **Interview Recording:** Records interviews for later review and analysis.
- **Real-Time Feedback:** Provides real-time feedback and scoring during live interviews.

7.2 Limitations of the System

The disadvantage of our system is mainly because of the dependency on the API. Since the system is completely based on API so if the API has any issue will definitely affect the system.

- 7.2.1 **Dependency on AI model accuracy:** The effectiveness of the mock interviews heavily relies on the accuracy and relevance of the underlying AI model, and limitations or biases in the model may impact user experience.
- 7.2.2 **Limited real-world unpredictability:** While the system aims to simulate real-world interviews, it may not fully capture the unpredictability and nuances encountered in actual job interviews, potentially resulting in a gap between the simulated and real experiences.
- 7.2.3 **Lack of natural human interaction:** The system, being AI-driven, may not fully replicate the nuanced and spontaneous interactions that occur in face-to-face interviews, potentially missing certain aspects of human communication and rapport-building.
- 7.2.4 **Generalization challenges:** The AI model's ability to generate questions and provide feedback may face challenges in adapting to highly specialized or niche industry domains, limiting its effectiveness in certain professional fields.

7.3 Future Research Directions

Our System is initial stage of the interview system. So, there is a lot of scope to enhance the system in system. Here are some research directions that should be considered in the future.

- 7.3.1 In future a audio and video based interview can also be implemented in future to improve the user interactivity with the system.
- 7.3.2 Explore expanded industry domains to cater to diverse professional backgrounds.
- 7.3.3 Investigate the integration of advanced AI capabilities, including NLU, sentiment analysis, and adaptive learning algorithms for nuanced interactions.
- 7.3.4 Consider implementing multilingual support to accommodate users in languages other than the primary one.
- 7.3.5 Explore features for real-time collaboration, simulating group interview scenarios or collaborative problem-solving exercises.
- 7.3.6 Investigate partnerships or integrations with job platforms and career services for tailored feedback based on industry trends.

- 7.3.7 Implement advanced user analytics to provide detailed insights into user performance over time.
- 7.3.8 Explore virtual reality (VR) integration for a more immersive and lifelike interview environment.
- 7.3.9 Collaborate with educational institutions to incorporate the system into career development programs and curricula.
- 7.3.10 Establish a continuous model refinement process using user feedback and performance data.
- 7.3.11 Incorporate features assessing and providing feedback on soft skills such as communication, empathy, and adaptability.
- 7.3.12 Conduct ongoing research on ethical AI practices, addressing biases and ensuring responsible AI algorithms.

The AI-Enabled Mock Interview System showcases a promising avenue for revolutionizing traditional interview preparation. Its current performance is measured through various metrics such as question relevance, coherence, and contextual appropriateness, ensuring a simulated interview experience that aligns with user preferences and industry domains. User feedback integration plays a pivotal role in refining the underlying AI model, fostering an iterative development process that addresses challenges like question ambiguity and coherence issues. Despite its strengths, the system faces limitations, including dependency on AI model accuracy, potential biases, and an occasional lack of natural human interaction, prompting considerations for future improvements.

Looking ahead, several future directions aim to propel the system's effectiveness and user impact. Exploring expanded industry domains and multilingual support seeks to make the system more versatile and inclusive. The integration of advanced AI capabilities, real-time collaboration features, and virtual reality (VR) enhancements aims to enrich the mock interview experience, providing users with a more immersive and adaptive platform. Collaborations with job platforms and educational institutions, coupled with continuous model refinement and advanced user analytics, are envisioned to further tailor the system to user needs and industry trends. Additionally, the system's ethical considerations are addressed through ongoing research on responsible AI practices, ensuring fairness, transparency, and unbiased performance evaluation.

In conclusion, the AI-Enabled Mock Interview System, while currently demonstrating commendable performance, acknowledges its limitations and looks toward a future marked by continuous improvement and innovation. The blend of user-centric enhancements, technological advancements, and ethical considerations positions the system as a dynamic

and evolving tool for individuals seeking to enhance their interview preparedness and professional success.

CHEPTEr 8

TESTING

Testing is the process of executing a program to find errors. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software. In this article, we will discuss first the principles of testing and then we will discuss, the different types of testing.

Principles of Testing

- All the tests should meet the customer's requirements.
- To make our software testing should be performed by a third party.
- Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- All the tests to be conducted should be planned before implementing it
- It follows the Pareto rule(80/20 rule) which states that 80% of errors come from 20% of program components.
- Start testing with small parts and extend it to large parts.
- Types of Testing

Different Types of Software Testing

1. Manual Testing

2. Automation Testing

1. Manual Testing

Manual testing is a technique to test the software that is carried out using the functions and features of an application. In manual software testing, a tester carries out tests on the software by following a set of predefined test cases. In this testing, testers make test cases for the codes, test the software, and give the final report about that software. Manual testing is time-consuming because it is done by humans, and there is a chance of human errors.

Advantages of Manual Testing:

- **Fast and accurate visual feedback:** It detects almost every bug in the software application and is used to test the dynamically changing GUI designs like layout, text, etc.

- **Less expensive:** It is less expensive as it does not require any high-level skill or a specific type of tool.
- **No coding is required:** No programming knowledge is required while using the black box testing method. It is easy to learn for the new testers.
- **Efficient for unplanned changes:** Manual testing is suitable in case of unplanned changes to the application, as it can be adopted easily.

2. Automation Testing

Automated Testing is a technique where the Tester writes scripts on their own and uses suitable Software or Automation Tool to test the software. It is an Automation Process of a Manual Process. It allows for executing repetitive tasks without the intervention of a Manual Tester.

Advantages of Automation Testing:

- **Simplifies Test Case Execution:** Automation testing can be left virtually unattended and thus it allows monitoring of the results at the end of the process. Thus, simplifying the overall test execution and increasing the efficiency of the application.
- **Improves Reliability of Tests:** Automation testing ensures that there is equal focus on all the areas of the testing, thus ensuring the best quality end product.
- **Increases amount of test coverage:** Using automation testing, more test cases can be created and executed for the application under test. Thus, resulting in higher test coverage and the detection of more bugs. This allows for the testing of more complex applications and more features can be tested.
- **Minimizing Human Interaction:** In automation testing, everything is automated from test case creation to execution thus there are no chances for human error due to neglect. This reduces the necessity for fixing glitches in the post-release phase.

Types of Manual Testing

1. **White Box Testing**
2. **Black Box Testing**
3. **Gray Box Testing**

. White Box Testing

White box testing is a software testing technique that involves testing the internal structure and workings of a software application. The tester has access to the source code and uses this knowledge to design test cases that can verify the correctness of the software at the code level.

Advantages of Whitebox Testing:

- **Thorough Testing:** White box testing is thorough as the entire code and structures are tested.
- **Code Optimization:** It results in the optimization of code removing errors and helps in removing extra lines of code.
- **Early Detection of Defects:** It can start at an earlier stage as it doesn't require any interface as in the case of black box testing.
- **Integration with SDLC:** White box testing can be easily started in the Software Development Life Cycle.
- **Detection of Complex Defects:** Testers can identify defects that cannot be detected through other testing techniques.

2. Black Box Testing

Black-box testing is a type of software testing in which the tester is not concerned with the internal knowledge or implementation details of the software but rather focuses on validating the functionality based on the provided specifications or requirements.

Advantages of Black Box Testing:

- The tester does not need to have more functional knowledge or programming skills to implement the Black Box Testing.
- It is efficient for implementing the tests in the larger system.
- Tests are executed from the user's or client's point of view.
- Test cases are easily reproducible.
- It is used to find the ambiguity and contradictions in the functional specifications.

3. Gray Box Testing

Gray Box Testing is a software testing technique that is a combination of the Black Box Testing technique and the White Box Testing technique.

1. In the Black Box Testing technique, the tester is unaware of the internal structure of the item being tested and in White Box Testing the internal structure is known to the tester.
2. The internal structure is partially known in Gray Box Testing.
3. This includes access to internal data structures and algorithms to design the test cases.

Advantages of Gray Box Testing:

1. Clarity of goals: Users and developers have clear goals while doing testing.
2. Done from a user perspective: Gray box testing is mostly done from the user perspective.
3. High programming skills not required: Testers are not required to have high programming skills for this testing.
4. Non-intrusive: Gray box testing is non-intrusive.
5. Improved product quality: Overall quality of the product is improved.

Types of Black Box Testing

1. **Functional Testing**
2. **Non-Functional Testing**

1. Functional Testing

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on the simulation of actual system usage but does not develop any system structure assumptions. The article focuses on discussing function testing.

Benefits of Functional Testing:

- **Bug-free product:** Functional testing ensures the delivery of a bug-free and high-quality product.
- **Customer satisfaction:** It ensures that all requirements are met and ensures that the customer is satisfied.
- **Testing focussed on specifications:** Functional testing is focussed on specifications as per customer usage.

- **Proper working of application:** This ensures that the application works as expected and ensures proper working of all the functionality of the application.
- **Improves quality of the product:** Functional testing ensures the security and safety of the product and improves the quality of the product.

2. Non-Functional Testing

Non-functional Testing is a type of Software Testing that is performed to verify the non-functional requirements of the application. It verifies whether the behavior of the system is as per the requirement or not. It tests all the aspects that are not tested in functional testing. Non-functional testing is a software testing technique that checks the non-functional attributes of the system. Non-functional testing is defined as a type of software testing to check non-functional aspects of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. Non-functional testing is as important as functional testing.

Benefits of Non-functional Testing

- **Improved performance:** Non-functional testing checks the performance of the system and determines the performance bottlenecks that can affect the performance.
- **Less time-consuming:** Non-functional testing is overall less time-consuming than the other testing process.
- **Improves user experience:** Non-functional testing like Usability testing checks how easily usable and user-friendly the software is for the users. Thus, focus on improving the overall user experience for the application.
- **More secure product:** As non-functional testing specifically includes security testing that checks the security bottlenecks of the application and how secure is the application against attacks from internal and external sources.

8.1 TESTING CASE 1 (Login)

8.1.1 Functional Test Cases-

- Verify if a user will be able to login with a valid username and valid password.
- Verify if a user cannot login with a valid username and an invalid password.
- Verify the login page for both, when the field is blank and Submit button is clicked.
- Verify the messages for invalid login.

8.1.2 Non-Functional Security Test Cases-

- Verify the time out functionality of the login session.
- Verify the login page by pressing 'Back button' of the browser. It should not allow you to enter the system once you log out.
- Verify if a user should not be allowed to log in with different credentials from the same browser at the same time.

8.2 TESTING CASE 2 (Adding Recipes)

8.2.1 Functional Test Cases-

- Verify that the all required fields are filled.
- Verify that the ingredients are properly field with commas.
- Verify that the adding recipe will update to the list.
- Verify that the adding recipe will be available for the only registered user or not.
- Verify that the only registered user can edit or delete the save recipe.

8.2.2 Non-Functional Security Test Cases-

- Verify that the all fields are visible to the user.
- Verify that the "nav bar" should be visible for direct reach to the "All recipes" tab for show updated recipe.

8.3 TESTING CASE 3 (Logout)

8.3.1 Functional Test Cases-

- Verify After successful login in Pantry2Plate, click on the profile icon to check logoutbutton is visible or not.
- Verify by Clicking on the sign-out button without an internet connection and reconnecting to the internet to check if it's properly logout or not.
- Verify by clicking on the logout button, after successful logout on the login screen press the back button.
- Verify, login into more than two browser or mobiles and log out from anyone them and check all other account is properly working or all get logout.
- Verify after logout tries to re-login with the same or different account it'sallowing or not.

8.3.2 Non-Functional Security Test Cases-

- Verify the logs for the login and logout sessions.
- Verify if the logs contain multiple IPs for a single ID at the same time.

CHAPTER 9

CONCLUSION

The AI-Enabled Mock Interview System marks a pivotal advancement in the realm of interview preparation, ushering in a dynamic and technology-driven approach to refine users' skills. Performance evaluation forms the cornerstone of the project, employing metrics like question relevance and coherence to ensure a nuanced and tailored mock interview experience. Through this evaluative lens, the system demonstrates its commitment to providing a realistic platform that adapts to user preferences and industry domains. The integration of user feedback stands as a testament to the project's responsiveness, fostering an iterative development process. This iterative approach addresses challenges, such as question ambiguity and coherence issues, ensuring a continual refinement of the underlying AI model.

However, a comprehensive understanding necessitates acknowledging the inherent limitations of the project. The dependency on AI model accuracy emerges as a critical consideration, and the occasional shortfall in mimicking natural human interaction prompts contemplation for future advancements. This candid recognition of challenges underscores the project's commitment to transparency, user awareness, and a relentless pursuit of improvement. The systemic recognition of potential biases is crucial, aligning with ethical considerations that are vital in the development and deployment of AI technologies.

Looking ahead, the future trajectory of the project is envisioned through a lens of innovation and adaptability. The exploration of expanded industry domains and multilingual support aims to broaden the system's applicability, fostering inclusivity across diverse professional backgrounds. Integrating advanced AI capabilities, such as natural language understanding and sentiment analysis, seeks to enrich the mock interview experience, making it more nuanced and aligned with real-world scenarios. Features like real-time collaboration and virtual reality enhancements underscore the commitment to creating an immersive and adaptive environment, pushing the boundaries of traditional interview preparation.

Collaborations with job platforms and educational institutions are strategic initiatives to embed the system within wider career development ecosystems. This collaborative approach positions the project as an integral component of holistic professional growth, seamlessly connecting users with industry trends and educational resources. Continuous model refinement emerges as a linchpin, ensuring that the AI model evolves in tandem with user needs and technological advancements. Advanced user analytics provide a granular understanding of user performance over time, enabling users to tailor their preparation strategies.

Ethical considerations are interwoven throughout the project's fabric, with ongoing research focused on responsible AI practices. This commitment to ethical standards is pivotal in addressing biases, ensuring fairness, and establishing a system that prioritizes user trust and societal impact. The project, in its envisioned future, transcends its role as a technological tool, assuming the mantle of a catalyst in broader discussions about the ethical use of AI in professional development.

As the project matures, it not only redefines individual interview preparedness but also contributes meaningfully to discussions on the responsible use of technology. The adaptability, user-centric design, and unwavering commitment to improvement position the AI-Enabled Mock Interview System as a valuable asset for individuals navigating the competitive landscape of job interviews. Its impact extends beyond individual users, resonating with the broader narrative of technology's potential to empower individuals on their career paths.

In essence, the AI-Enabled Mock Interview System represents a convergence of technological innovation, user-centric design, and ethical considerations. The project's journey from conception to continuous refinement mirrors the evolving nature of the professional world it seeks to prepare users for. In navigating this journey, the project not only meets the immediate needs of users but contributes to a future where interview preparation is not merely a process but a reflection of the diverse and dynamic nature of the professional landscape.

AI-enabled mock interview web applications represent a significant advancement in interview preparation and career development. By providing personalized, data-driven feedback, realistic simulations, and accessible practice opportunities, these tools can dramatically improve a candidate's readiness for actual job interviews. As technology continues to evolve, these applications will become even more sophisticated, offering deeper insights and more tailored experiences.

Ultimately, AI-enabled mock interview platforms empower users to enter the job market with greater confidence and competence, making them an indispensable resource for modern career preparation.

An AI-enabled mock interview web application can be a transformative tool for job seekers and educational institutions alike. By leveraging advanced AI technologies, such an application can provide a realistic, interactive, and comprehensive interview preparation experience. Here are the key benefits and features that make AI-enabled mock interview applications invaluable:

1. **Personalized Feedback:**

- AI algorithms analyze responses, body language, and tone to provide detailed, personalized feedback.
- Users can understand their strengths and areas for improvement, leading to more focused preparation.

2. **Realistic Simulations:**

- Simulated interviews with dynamic questioning adapt to user responses, offering a realistic experience akin to actual job interviews.
- Users can practice a wide range of interview types, including technical, behavioral, and situational interviews.

3. **Skill Assessment:**

- The application can include technical skill assessments and coding challenges, with real-time feedback and scoring.
- These assessments help users gauge their technical proficiency and readiness for specific roles.

17 **Redux-thunk:** Version 2.4.2 is used in this project. Redux Thunk is a middleware for Redux that enables the handling of asynchronous actions. It allows you to dispatch functions as actions, providing flexibility when dealing with side effects like API calls.

18 **Twind:** Version 0.16.19 is used in this project. wind is a utility-first CSS framework for React. It allows you to write styles in your JavaScript/JSX files, providing a highly efficient way to manage styles in React applications.

19 **Web-vitals:** Version 2.1.4 is used in this project. Web Vitals is a set of metrics that help measure the performance and user experience of a web application. It includes tools and libraries for capturing and reporting essential performance metrics.

CHAPTER 10

REFERENCE

1. Predinger, Ishizuka, "The empathic companion: a character-based interface that addresses users' affective states", 2007.
2. Nunamaker Jr., C. Derrick, et al.: "Embodied Conversational Agent-Based Kiosk for Automated Interviewing", 2011.
3. Maras et al. "Ameliorating the disadvantage for autistic job seekers: An initial evaluation of adapted employment interview questions", 2021.
4. Robinson, Marica F., "Artificial Intelligence in Hiring: Understanding Attitudes and Perspectives of HR Practitioners", 2019.
5. Hosselet, P.F., "The acceptance of AI enabled decision support systems a project management perspective", 2018.
6. Tom Taulli, "Artificial Intelligence Basics: A non Technical Basics", 2019.
7. Crutsinger, Herrera, "Mock Interviews: Leveraging AI Resources To Enhance Professional Skills", 2022.
8. B. Powell et al., "An overview of mock interviews as a training tool for interviewers of children", 2022.
9. Chou et al., "An AI Mock-interview Platform for Interview Performance Analysis", 2022.
10. Harchar, Ed.D., "Mock Interview Strategy: An action research study of administrator and teacher candidates' preparation for interview field experience", 2020.
11. Temgire et al., "Real Time Mock Interview using Deep Learning", 2021.
12. Lee, Kim, "Development Of An Ai-Based Interview System For Remote Hiring", 2021
13. Anderson, J., & Shackleton, P. "Interview Performance and Job Offer Success: An Empirical Study.", 2016.
14. Smith, K., & Johnson, A. "The Role of Interview Preparation in Job Seeker Success: A Quantitative Analysis.", 2018.
15. Brown, M., & Davis, S. "Customized Interview Practice: An Empirical Study on its Impact on Interview Outcomes." , 2019.
16. Patel R., et al. "Tailoring Interview Practice to Individual Needs: A Review of Customization Techniques." , 2020
17. Chen L., et al. "AI-Enhanced Mock Interviews: An Overview of Current Trends

18. Wu H., Kim S. "The Role of Artificial Intelligence in Interview Simulations: A Comprehensive Analysis.", 2017.
19. Williams E., Brown S. "Enhancing Interview Performance through FeedbackRich Mock Interviews: An Experimental Study.", 2020
20. Park, J., Lee, M. "Feedback Mechanisms in Interview Practice: A Comparative Analysis.", 2018.
21. Rogers, M., White, C. "Improving Communication Skills in Mock Interviews through AI-Driven Character Interactions.", 2018.
22. Yang, Q., et al. "The Role of AI in Enhancing Communication Skills: A Case Study of Interview Preparation.", 2019.
23. Lee, S., Park, K. "AI Assessment of Competencies in Mock Interviews." Computational Intelligence and Neuroscience, 2017.
24. Kim, H., et al. "Assessing Technical Competencies in Mock Interviews: A Comparative Study of AI and Human Evaluations." ,2018.

CHAPTER 11

BIBLIOGRAPHY

11.1 Online Websites

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

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

11.2 Reference Books

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

11.2.1 "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig

This comprehensive textbook provides insights into various aspects of artificial intelligence, including machine learning and natural language processing.

11.2.2 "Building Scalable and Responsive Web Applications with React" by Adam Freeman

Ideal for those working with ReactJS, this book covers building scalable and responsive web applications, which aligns with the frontend development aspect of your project.

11.2.3 "Mastering React: Build Scalable and High-Performance Web Applications with React 17" by Tom Banks

This book delves into advanced concepts and best practices for mastering React, which can be valuable for optimizing the frontend of your application.

11.2.4 "Practical Natural Language Processing with Python: A Comprehensive Guide to Building Real-World NLP Applications" by Dipanjan Sarkar

Focused on natural language processing, this book provides practical insights and hands-on examples for building NLP applications, aligning with the language processing aspect of your project.

11.2.5 "Virtual Reality in Education: A Practical Guide for Teachers and Developers" by Charles Wankel and Patrick Blessinger

Explore the potential of virtual reality in education, which could be insightful for understanding the implications of VR integration in your mock interview system.

11.2.6 "Web Analytics: An Hour a Day" by Avinash Kaushik

For those interested in understanding web analytics, this book provides a structured approach and best practices, which can be beneficial for analyzing user interactions on your platform.