

Naukri Sahayak - AI Mock Interview Portal

*A project report submitted to
Dr. Babasaheb Ambedkar Technological University, Lonere
in partial fulfillment of the requirements for the award of the degree*

Bachelor of Technology
in
Computer Engineering

Project Phase – I
(BTCOS708)



by
Mr. Aditya Kamalakar Kashikar (PRN: 2146491245010)
Mr. Prajwal Rameshwar Gurnule (PRN: 2146491245011)
Mr. Piyush Manoj Raut (PRN: 2146491245102)
Mr. Ritesh Ananta Nagtode (PRN: 2146491245019)
(VII Semester)

under the guidance of
Mr. Akshay Saraf
(Assistant Professor)

Department of Computer Engineering
Shiksha Mandal's
BAJAJ INSTITUTE OF TECHNOLOGY, WARDHA
Pipri, Arvi Road, Wardha - 442001.
(2024-25)

**Dr. Babasaheb Ambedkar Technological University, Lonere
Bajaj Institute of Technology, Wardha**
Pipri, Arvi Road, Wardha - 442001.

DEPARTMENT OF COMPUTER ENGINEERING



Certificate

This is to certify that Project Phase -I titled

Naukri Sahayak - AI Mock Interview Portal

has been completed by

Mr. Aditya Kamalakar Kashikar (PRN: 2146491245010)

Mr. Prajwal Rameshwar Gurnule (PRN: 2146491245011)

Mr. Piyush Manoj Raut (PRN: 2146491245102)

Mr. Ritesh Ananta Nagtode (PRN: 2146491245019)

of VII Semester, Computer Engineering of academic year 2024-25 in partial fulfillment of Project Phase -I (BTCOS707) course as prescribed by the Dr. Babasaheb Ambedkar Technological University, Lonere.

Mr. Akshay Saraf
(Department of Computer Engineering)

Prof. Sheetal Kale
(Head of the Department)

Dr.N.M.Kanhe
Principal
Bajaj Institute of Technology Wardha

Place: BIT, Wardha
Date: November 22, 2024

Declaration

We hereby declare that the Mini Project report titled “**Naukri Sahayak - AI Mock Interview Portal**” submitted by us to the Bajaj Institute of Technology, Wardha, in partial fulfilment of the requirement for the award of Degree of B. Tech in Computer Engineering is a record of bonafide seminar work carried out by us under the guidance of Mr. Akshay Saraf.

We, further declare that the work reported in this Mega Project report have not been submitted either in-part or in-full for the award of any other degree in any other Institute or University.

Report Title: Naukri Sahayak - AI Mock Interview Portal

SN	Student Name	PRN	Signature
1	Aditya Kashikar	2146491245010	
2	Prajwal Gurnule	2146491245011	
3	Piyush Raut	2146491245102	
4	Ritesh Nagtode	2146491245019	

Date: November 22, 2024

Place: BIT, Wardha

Acknowledgment

We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report. Special thanks is due to our Guide Mr. Akshay Saraf, Assistant Professor of Computer Engineering whose help, stimulating suggestions and encouragement helped us in all time of process and in writing this report.

We extend our heartfelt thanks to Prof. Sheetal Kale, Head of the Department Computer Engineering Bajaj Institute of Technology, Wardha for her insightful guidance, assistance and invaluable suggestions that significantly enriched the quality of these work. We also sincerely thanks for the time spent profreading and correcting our mistakes.

Our deepest thanks to Department of Computer Engineering And Dr. Narendra Kanhe, Principal Bajaj of Institute of Technology, Wardha. We would also like to acknowledge with much appreciation the crucial role of the staff in Computer Lab, who gave us a permission to use the Computers in the laboratory. Many thanks go to the all lecturers and go to all our classmates, especially to our friends for spending their time in helping and giving support whenever we need it in our Mega Project Topic.

Abstract

The process of preparing for job interviews is a critical yet challenging task for job seekers, traditionally limited to self-practice or feedback from peers, which often lacks precision and personalization. To address these limitations, our project introduces an AI-Powered Mock Interview Platform, designed to revolutionize interview preparation by leveraging artificial intelligence and modern computational techniques. This project focuses on two primary objectives: generating personalized interview questions based on user-specific parameters and providing a detailed evaluation of candidates' performance through AI-driven feedback mechanisms.

The first objective involves the dynamic generation of tailored interview questions using the Gemini API, which considers factors such as job descriptions, years of experience, and technology stacks. This ensures the questions are highly relevant to the user's target roles. Responses are evaluated using cosine similarity, comparing user answers to ideal AI-generated responses for precise scoring. Additionally, a question bank encompassing multiple domains and difficulty levels provides a comprehensive practice resource.

The second objective focuses on performance evaluation, integrating video streaming and confidence-level analysis to assess non-verbal cues such as tone, facial expressions, and voice modulation. This holistic approach provides candidates with detailed insights into their strengths and areas for improvement. The platform generates an overall grade for each session, offering a clear and actionable summary of the candidate's readiness.

Built using Next.js, Tailwind CSS, PostgreSQL, and Drizzle ORM, and deployed on Neon Serverless, the platform is scalable, efficient, and user-friendly. By combining advanced AI methodologies with robust development practices, this project enhances the efficiency and realism of interview preparation. This platform bridges the gap between traditional preparation methods and modern job market requirements by integrating AI-driven question generation, detailed feedback, and confidence analysis. It represents a significant advancement in interview preparation, empowering job seekers to pursue their career aspirations confidently.

Keywords- *Confidence Analysis, Cosine Similarity, Gemini API, Interview Preparation, Video Streaming*

Abbreviations

<i>AI</i>	Artificial Intelligence
<i>ORM</i>	Object Relation Mapping
<i>API</i>	Application Programming Interface
<i>SVM</i>	Support Vector Machine
<i>tf</i>	Tensorflow
<i>BERT</i>	Bidirectional encoder representations from transformers
<i>GPT</i>	Generative Pre-trained Transformer
<i>NLP</i>	Natural language processing
<i>XGBoost</i>	EXtreme Gradient Boosting
<i>MSE</i>	Mean square error
<i>MAE</i>	Mean absolute error
<i>RF</i>	Random Forest

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Benefits of the computational drug discovery	2
1.3	Challenges Faced by the Project	3
1.4	Problem statement	3
1.5	Objectives	4
1.6	Scope of work	5
1.7	Organization of the Report	5
1.8	AI Mock Interview Portal Work Flow	6
2	Literature Survey	7
3	Methodology	11
3.1	Proposed Solution	11
3.1.1	AI-driven Interview	11
3.1.2	User Experience Level Questions	12
3.1.3	Detailed Feedback and Insights	12
3.2	Models and Algorithms	13
3.2.1	Gemini API Model	13
3.2.2	Cosine Similarity	13
3.2.3	BlazeFace	14
3.2.4	Tenserflow	14
3.2.5	Face Landmarks Detection	15
3.2.6	MediaPipe	15
3.3	System Architecture	16
4	Implementation	18
4.1	User Authentication	18
4.2	Backend Setup(Drizzle ORM + PostgreSQL)	20
4.3	Generate Mock Interview Question with AI	21
4.4	Interview Landing Page	22
4.5	Show Interview Question	24
4.6	Record User Answer With AI Feedback	25
4.7	Interview Feedback	26
5	Result & Discussion	27
5.1	Question Generation	27
5.2	Confidence Model	28
5.3	Feedback Model	29

5.4	Deployment	31
6	Conclusion & Future Scope	32
6.1	Conclusion	32
6.2	Future scope	33
REFERENCES		34
APPENDICES		35
A	Project Requirement	36
A.1	Database Schema	36
A.2	Packages	38

List of Figures

1.1	AI Mock Interview Portal Workflow	6
3.1	System Architecture for the Naukri Sahayak	16

Chapter 1

Introduction

In today's fast-paced and competitive job market, excelling in interviews is critical for securing desired positions. However, traditional interview preparation methods often fall short in providing candidates with a realistic and adaptive environment to practice and refine their skills. To address this gap, we have developed an AI-Powered Mock Interview Platform, an innovative solution that leverages the latest advancements in artificial intelligence and modern web technologies.

This platform is designed to simulate real-world interview scenarios, offering personalized and interactive experiences tailored to individual job roles, years of experience, and specific technology stacks. By integrating the Gemini API for generating role-specific questions and evaluating responses using cosine similarity, the platform provides accurate ratings of candidates' performance. Additionally, it incorporates video streaming and analysis to assess confidence levels based on non-verbal cues, making the feedback comprehensive and actionable.

With cutting-edge technologies like Next.js, Tailwind CSS, PostgreSQL, and Drizzle ORM, the platform ensures seamless performance and scalability. It empowers users to practice technical, behavioral, and domain-specific interviews while receiving detailed feedback, insights, and an overall grade to identify strengths and areas for improvement. This project aims to revolutionize interview preparation by providing job seekers with a robust, AI-driven tool that enhances their skills and boosts their confidence, ultimately increasing their chances of securing their dream jobs.

1.1 Motivation

The motivation behind the AI-powered Mock Interview Platform stems from the growing challenges faced by job seekers in preparing for interviews in an increasingly competitive job market. Traditional methods of interview preparation, such as self-practice or informal peer feedback, often fail to provide comprehensive, realistic, and tailored experiences. These limitations can lead to inadequate preparation, decreased confidence, and missed career opportunities.

By leveraging artificial intelligence and advanced computational techniques, this platform aims to provide a highly efficient and personalized solution for interview

preparation. Through dynamic question generation based on job-specific parameters and detailed performance evaluations, candidates receive targeted insights into their strengths and areas for improvement. The integration of confidence-level analysis further addresses the often-overlooked non-verbal aspects of interview performance, helping candidates project a stronger presence.

This approach not only saves time and effort compared to traditional methods but also significantly enhances candidates' chances of success by offering realistic simulations and actionable feedback. Ultimately, the platform seeks to empower job seekers, helping them confidently navigate the interview process and secure their desired roles, thereby addressing a critical gap in career development tools.

1.2 Benefits of the computational drug discovery

- **Increased Efficiency:** The platform automates the generation of personalized interview questions and provides instant feedback, saving candidates significant time compared to traditional preparation methods. By focusing on relevant job roles and experience levels, users can quickly identify their strengths and areas for improvement, making the preparation process highly targeted and efficient.
- **Higher Success Rates:** With AI-driven question generation and performance evaluation, the platform equips candidates with realistic interview scenarios. This enables them to refine their responses, improve their non-verbal communication, and build confidence. The comprehensive feedback ensures candidates are better prepared for real interviews, increasing their chances of securing job offers.
- **Comprehensive Feedback and Insights:** The platform combines textual evaluation with confidence-level analysis through video streaming, offering a holistic assessment of a candidate's interview performance. This dual approach addresses both technical and behavioral aspects, providing actionable tips for improvement in both areas.
- **Customizability and Relevance:** By tailoring interview sessions to specific job roles, industries, and experience levels, the platform ensures that users practice with questions and scenarios that are directly aligned with their career goals. This personalization increases the relevance of the preparation process and enhances its effectiveness.
- **Innovation in Interview Preparation:** By integrating AI technologies such as the Gemini API and confidence analysis, the platform represents a significant advancement in interview preparation tools. It empowers job seekers with cutting-edge resources that bridge the gap between traditional preparation methods and modern interview requirements.

1.3 Challenges Faced by the Project

In these section, we elaborate on the challenges faced in the AI Mock Interview Portal. These challenges include:

- **Question Relevance:** Ensuring that the generated interview questions matched the job description and user profile with precision was a complex task. Fine-tuning the integration with the Gemini API and validating the generated questions for correctness and applicability was an iterative process.
- **Performance Metrics:** Evaluating answers using cosine similarity and non-verbal communication required developing robust models that could handle variations in user responses, accents, and expression styles while maintaining fairness and accuracy.
- **Confidence Assessment:** Developing machine learning models to accurately analyze non-verbal cues like tone, pace, and clarity, and translating them into confidence scores, presented challenges in both design and validation.

1.4 Problem statement

The problem statement for the Naukri Sahayak project involves two primary objectives:

Generating Job-Specific Interview Questions: This involves creating a system capable of dynamically generating interview questions tailored to a candidate's job description, technical stack, and experience level. The goal is to leverage advanced AI technologies, such as the Gemini API, to:

- Develop relevant and meaningful questions that simulate real-world technical interviews.
- Ensure that the questions reflect the latest trends and requirements in specific industries and domains.
- Adapt the question generation process to various experience levels, from entry-level to senior professionals.
- Achieving this requires precise integration with external APIs and effective utilization of large-scale datasets to provide a diverse and accurate question bank.

Evaluating Candidate Responses: This objective focuses on creating a robust framework for evaluating candidates' responses in real time. The system analyzes both verbal and non-verbal aspects of the response, aiming to:

- Assess the correctness and relevance of the answer by comparing it with ideal responses generated using Bard AI and cosine similarity techniques.
- Measure the confidence level of the candidate based on voice modulation, tone, and clarity using machine learning models.
- Identify areas for improvement and provide personalized feedback for skill enhancement.
- Developing this evaluation framework requires advanced AI models capable of processing audio, video, and text inputs while ensuring fairness and accuracy.

1.5 Objectives

The objectives of the drug discovery project divided into the three main categories:

1. Tailored Interview Preparation Based on Job Role, Experience, and Technology Stack

Dynamic Question Generation:

- Utilize the Gemini API to generate a diverse set of interview questions aligned with the user's job role, years of experience, and technology stack.
- Ensure that the questions cover both technical and behavioral aspects relevant to the role and industry.

Personalized Interview Sessions:

- Customize the difficulty level of questions to match the user's experience level (e.g., entry-level, mid-level, senior-level).
- Include industry-specific scenarios to simulate real-world challenges and decision-making processes.

Question Bank Development:

- Build a comprehensive repository of interview questions across various domains and industries.
- Continuously update the repository to include the latest trends and frequently asked questions in different fields.

2. Comprehensive Candidate Performance Analysis:

Confidence Level Assessment:

- Use integrated video streaming and audio analysis to evaluate the candidate's confidence and clarity during responses.
- Develop machine learning models to assess voice modulation, facial expressions, and non-verbal cues.

Cosine Similarity for Answer Evaluation:

- Compare candidate responses with ideal answers generated by the Gemini API using cosine similarity to provide a numerical rating.
- Offer insights into the alignment of candidate answers with industry expectations and best practices.

Feedback and Insights:

- Provide detailed feedback on candidate performance, including strengths, areas for improvement, and actionable suggestions for enhancement.
- Summarize results into an overall performance grade for quick assessment and goal setting.

1.6 Scope of work

The AI-Powered Mock Interview Platform aims to create an advanced and user-centric solution to enhance the interview preparation process for job seekers. The project encompasses the following core components:

1. Generating Tailored Interview Questions

- Integrate the Gemini API to generate role-specific, domain-specific, and experience-level-appropriate interview questions.
- Ensure the generated questions encompass both technical and behavioral aspects to provide a well-rounded interview simulation.

2. Performance Evaluation and Feedback System:

- Use cosine similarity to compare candidate responses with ideal responses generated by the Gemini API, providing accurate ratings for content quality.
- Implement video and audio analysis to evaluate non-verbal communication skills, such as confidence, clarity, and tone.
- Provide granular feedback, including strengths, areas for improvement, and actionable suggestions for enhancing interview performance.

3. Interactive User Experience:

- Design an intuitive and responsive interface using Next.js and Tailwind CSS to ensure seamless user interaction.
- Incorporate features like recent interview history, overall grading, and easy access to feedback for continuous improvement.

4. Backend and Database Implementation:

- Employ PostgreSQL for data storage and management, ensuring scalability and reliability.
- Use Drizzle ORM for efficient database interactions and handling complex data relationships.

1.7 Organization of the Report

In these section, The organization of our report is elaborated. this is organized into six chapters. which are explained below:

- Chapter - 1 describes the introduction of our project, the motivation, objectives, problem statement, scope of the work, and the work flow diagram of our Naukri Sahayak Project.
- Chapter - 2 describes the Literature Survey of the project. Initially the literature review which include the related work of the project. furthermore, the identification of gaps in literature and brief summaries of the research paper are elaborated.

- Chapter - 3 describes the methodology of the project which include the sections as proposed solution, methods and algorithms and the system architecture.
- Chapter - 4 provides the implementation of our project which include User Authentication, Backend Setup, Generate Mock Interview Question, Interview Landing Page, Show Interview Question, Record User Answer with AI Feedback and Interview Feedback which are carried out.
- Chapter - 5 shows the results of our drug discovery project which consist of the Question Generation, Confidence model, Feedback Model and the deployment.
- Chapter - 6 describes the Conclusion and Future Scope of our Naukri Sahayak Project.

1.8 AI Mock Interview Portal Work Flow

In these chapter, we elaborate on the work flow diagram of our AI Mock Interview Portal.

The workflow begins with profile creation, which includes an authentication process offering options such as Google, email, and Facebook for user convenience and security. Once the profile is successfully created, the candidate can select a mock interview tailored to a specific role by filling in details such as the technology stack, job role, and years of experience. Based on these inputs, the system generates customized interview questions using the Gemini API.

During the interview, candidates respond to the generated questions through video recording, allowing the system to analyze their confidence levels based on voice modulation and facial expressions. The candidate's submitted response is then compared with an ideal response generated by the Gemini API using techniques like cosine similarity. Finally, detailed feedback is provided to the candidate, highlighting their performance, strengths, and areas for improvement, thus enabling them to refine their skills and prepare effectively for real-world interviews.

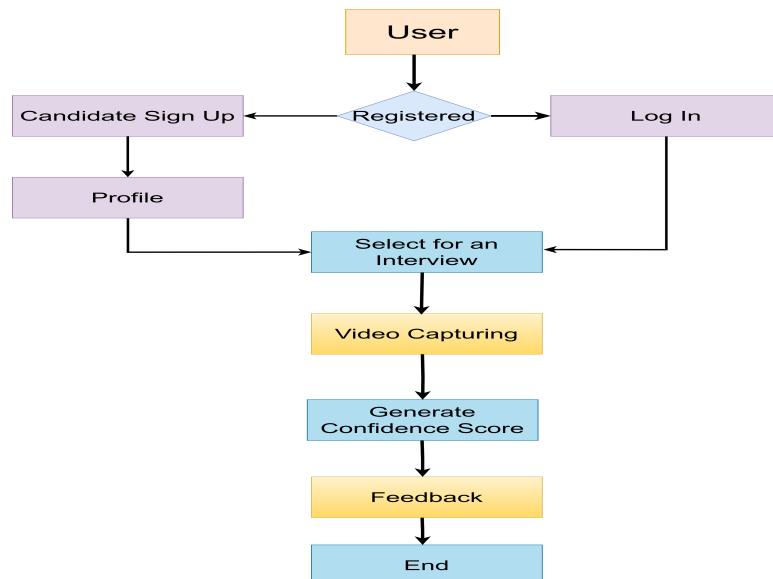


Figure 1.1: AI Mock Interview Portal Workflow

Chapter 2

Literature Survey

This chapter outline the literature Survey for our project AI Mock Interview Portal,encompassing the theory and research carried for understanding the concepts. The chapter begins with the literature review which provide the related work, followed by the gap identified in the research paper and the brief summaries of papers.

Paper 1:

Title and Description: Skillup Bot: An AI Driven Mock Interview Platform

Work Reported:

In the contemporary job market, preparation for interviews is a pivotal determinant of success in securing desired employment opportunities. To address this critical need, we introduce a comprehensive mock interview platform aimed at empowering candidates in their preparation journey. Our platform incorporates an array of advanced functionalities harnessing state-of-the-art artificial intelligence (AI) technologies to deliver an immersive and effective preparation experience. Central to our platform is the utilization of AI-driven video and audio analysis, providing candidates with real-time feedback on their interview performance. Through sophisticated machine learning (ML) algorithms, we offer nuanced assessments of various elements including communication skills, body language, and tone of voice. This enables candidates to gain valuable insights into their strengths and areas for improvement, enhancing their overall interview readiness.

Moreover, our platform features a dynamic resume builder tool, enabling candidates to craft personalized resumes tailored to specific job opportunities. This empowers candidates to present themselves in the best possible light, aligning their skills and experiences with the requirements of prospective employers. Additionally, we integrate a programming quiz platform designed to assess candidates' technical proficiency, particularly relevant for roles in the IT and software development sectors. By offering a diverse range of evaluation tools, our platform caters to the multifaceted nature of modern job interviews, addressing both technical and soft skills requirements. Following completion of interview simulations and technical assessments, candidates receive comprehensive performance reports based on the analysis conducted by our ML algorithms. These reports serve as valuable insights, guiding candidates in their ongoing preparation efforts and facilitating continuous improvement.

Gaps Identified:

The understanding and management of electronic waste (E-waste) within international material cycles and waste management. Firstly, there is a lack of comprehensive data on the quantities of waste materials moving between countries, making effective tracking and management difficult. Additionally, the emissions from hazardous substances during the recycling process are insufficiently studied, especially in developing countries where inappropriate technologies are often employed. Existing regulations, such as those under the Basel Convention, are not entirely effective in controlling the international trade of secondhand electronics and E-waste due to unclear definitions, leading to inadequate regulation. The enforcement of producer responsibility for E-waste, especially in the context of international trade, is weak and needs strengthening to ensure proper collection and recycling of end-of-life products. The paper also notes the lack of a balanced evaluation of the positive and negative aspects of international trade in E-waste and calls for comprehensive strategies to maximize benefits while minimizing drawbacks. Inconsistent terminology and classification of E-waste versus secondhand goods create confusion and hinder effective management. Furthermore, there is a need for better recycling technologies and processes, as well as the development of more environmentally friendly materials and products that facilitate easier recycling.

Conclusion:

In conclusion, the proposed system effectively addressed the crucial need for interview preparation in today's job market. Our comprehensive mock interview platform, leveraging advanced AI technologies and diverse evaluation tools, empowered candidates in their preparation journey. The integration of AI-driven video and audio analysis, machine learning algorithms, and dynamic resume building tools provided immersive and effective preparation experiences. Incorporating ATS Resume analysis and programming quizzes addressed modern interview demands. Future research could refine AI algorithms for nuanced assessments and include user-provided answer analysis for feedback. Adding a smart coding platform could further enhance technical proficiency evaluation, laying a foundation for advancing interview preparation methodologies and benefiting candidates in their career pursuits.

Paper 2:

Title and Description: IntelliView: An AI-Mock Interview Platform

Work Reported:

IntelliView is a groundbreaking platform that aids novice job seekers by harnessing the power of advanced AI and NLP technologies. It combines HTML, CSS, JavaScript, and the Deep Face method to offer a comprehensive real-time interview practice experience. Users can engage in text and video-based interviews and receive detailed feedback on their verbal and non-verbal communication skills. Additionally, the platform provides an audio analysis module that utilizes CNN LSTM to evaluate emotional states during interviews, offering valuable insights for improvement. Users can also take advantage of IntelliView's resume builder to create personalized, professional resumes that cater to specific industry standards.

The platform's seamless integration of AI and NLP technologies helps guide first-time job seekers in refining their communication techniques and understanding their emotional responses. By offering these transformative tools, IntelliView empowers users to approach the competitive job market with confidence and competence.

Gaps Identified:

IntelliView by offering a broader scope of features, more advanced adaptability, and a deeper integration of cutting-edge technologies. While IntelliView provides a real-time interview practice platform that combines AI, NLP, and technologies like CNN-LSTM for emotional analysis, it primarily targets verbal and non-verbal communication skills. In contrast, the second paper emphasizes creating real-world interview scenarios tailored to job roles, experience levels, and technology stacks, thereby offering a more adaptive and personalized experience. The integration of the Gemini API for generating role-specific questions and cosine similarity for performance evaluation sets the second paper apart by providing a context-sensitive assessment not explicitly addressed in IntelliView.

Additionally, IntelliView uses the Deep Face method for non-verbal analysis, which focuses on emotional insights, whereas the second paper includes video-based confidence analysis that extends beyond emotions to assess overall non-verbal performance comprehensively. Moreover, the second paper's incorporation of modern web technologies like Next.js, Tailwind CSS, and PostgreSQL ensures scalability and seamless performance, which are not highlighted in IntelliView. Lastly, while both platforms include resume-building tools, the second paper enhances the feedback process with detailed, actionable insights and structured grading systems, making it more effective for guiding continuous improvement. These distinctions underline the second paper's contribution to a more immersive, adaptive, and technologically advanced interview preparation platform.

Conclusion:

In conclusion, the proposed system effectively addressed the crucial need for interview preparation. To sum up, the IntelliView project is revolutionizing the process of preparing job applications by utilizing cutting edge technologies to enable job seekers at the entry level. Through a comprehensive framework integrating AI, NLP, and the innovative Deep Face method, IntelliView redefines interview analysis with real-time assessments, constructive feedback, and dynamic insights into non-verbal communication. The CNN LSTM is used to effectively improve your tone and confidence and provides a way to speak properly at the time of a real interview. The multifaceted modules collectively signify a paradigm shift, providing a nuanced approach to the competitive job market. As technology and human interaction converge, IntelliView equips individuals with essential competencies, offering a glimpse into the future of job application preparation. This research underscores the project's technological underpinnings, methodological rigor, and its broader implications for reshaping the landscape of employment preparation.

Paper 3:

Title and Description: AI-BASED MOCK INTERVIEW EVALUATOR: AN EMOTION AND CONFIDENCE CLASSIFIER MODEL

Work Reported:

The "AI-Based Mock Interview Evaluator" is an innovative project aiming to enhance interview preparation by incorporating an emotion and confidence classifier model. Traditional mock interviews often lack nuanced feedback on non-verbal cues such as

emotion and confidence, critical aspects in professional interactions. The proposed system employs artificial intelligence to analyze facial expressions, tone, and language patterns during a mock interview, providing comprehensive insights into the interviewee's emotional state and confidence levels. This intelligent evaluator is designed to mimic real-world interview scenarios, facilitating a more authentic practice environment.

Gaps Identified:

The second paper addresses key gaps in the AI-Based Mock Interview Evaluator by offering a broader and more versatile platform for interview preparation. While the AI-Based Mock Interview Evaluator focuses on analyzing emotion and confidence through facial expressions, tone, and language patterns, it primarily emphasizes non-verbal cues and emotional feedback. The second paper expands beyond this by creating a comprehensive, adaptive platform tailored to job roles, experience levels, and technology stacks, integrating role-specific question generation using the Gemini API and cosine similarity for response evaluation.

Moreover, while the AI-Based Mock Interview Evaluator aims to mimic real-world scenarios, it does not detail scalability or the use of advanced web technologies. The second paper addresses this gap by incorporating modern technologies like Next.js, Tailwind CSS, PostgreSQL, and Drizzle ORM, ensuring a seamless, scalable, and high-performing platform. Additionally, the second paper extends its scope by combining technical interview preparation, domain-specific assessments, and behavioral evaluations, offering detailed grades and actionable insights, which are not explicitly covered by the AI-Based Mock Interview Evaluator. These distinctions highlight the second paper's focus on a more holistic, personalized, and technology-driven approach, addressing both technical and soft skills comprehensively.

Conclusion:

The "AI-Based Mock Interview Evaluator" project concludes with the successful development of a sophisticated system designed to provide candidates with personalized feedback on their emotional expressions and confidence levels during mock interviews. Through the integration of advanced artificial intelligence techniques, including computer vision and natural language processing, the system accurately analyzes candidates' facial expressions and speech patterns. Key highlights include emotion recognition, confidence assessment, a user-friendly interface, and performance optimization. The project demonstrates its effectiveness in empowering candidates to enhance their interview skills and offers invaluable support to educators and interviewers. In conclusion, the "AI-Based Mock Interview Evaluator" represents a groundbreaking advancement in interview preparation, serving as a valuable tool for professional development and confidence building.

Chapter 3

Methodology

This chapter discuss the proposed solution for our Naukri Sahayak - AI Mock Interview Portal. This section serves as a roadmap, outlining the steps, and techniques used to collect and analyze the ML model. It also Outline the Models and Algorithms and followed by the System Architecture of our project .

3.1 Proposed Solution

In these section, we elaborate on the proposed solution of our Naukri Sahayak project. these section divide the proposed solution as depending on the task to perform.

3.1.1 AI-driven Interview

The AI-Driven Interview feature is the heart of the Naukri Sahayak platform, offering an intelligent and adaptive mock interview experience tailored to help users refine their skills in a realistic environment. Leveraging advanced AI technologies and a robust backend, it dynamically generates contextually relevant and role-specific questions using the Gemini API. The system adapts to the user's job role, industry requirements, and experience level, ensuring that every session is personalized and relevant. The AI interviewer simulates real-time interactions, comprehending user responses through natural language processing (NLP) and providing a highly engaging experience with options for text and voice-based input.

Following each session, users receive detailed feedback that highlights strengths, pinpoints weaknesses, and offers actionable suggestions for improvement. This includes behavioral insights, such as clarity and confidence, along with technical or domain-specific feedback. The platform also calculates a performance grade using predefined metrics and benchmarks the user's performance against industry standards or peer averages, helping them understand their competitive positioning.

The backend integrates seamlessly with PostgreSQL, managed via Drizzle ORM for efficient data handling, while the frontend, built with Next.js and Tailwind CSS, offers an intuitive and responsive interface. Users can easily start interview sessions, review feedback, and access a rich question bank for additional practice. The AI models continuously learn and evolve, improving question relevance and feedback quality based on user interactions. This comprehensive system enables users to practice anytime,

receive insightful evaluations, and make targeted improvements, empowering them to excel in their career pursuits.

3.1.2 User Experience Level Questions

The User Experience Level Questions feature is designed to ensure that interview sessions are tailored to the user's professional background and expertise. This adaptability helps users engage with questions that are neither too basic nor overly complex, making the mock interview process more effective and relevant to their needs.

When setting up a mock interview, users can specify their experience level, such as Beginner, Intermediate, or Advanced. Based on this selection, the platform dynamically adjusts the complexity, depth, and focus of the questions. For instance, a beginner might receive foundational questions to assess basic knowledge and understanding, while an advanced user could face challenging, scenario-based questions that test critical thinking, problem-solving, and expertise in their field.

The feature leverages AI to analyze the user's responses during the interview, refining subsequent questions in real-time to maintain an appropriate difficulty level. Additionally, the platform incorporates feedback loops to ensure that questions align with the latest industry standards and role-specific expectations. By tailoring questions to the user's experience level, the platform enhances engagement, builds confidence, and ensures that every session contributes meaningfully to the user's growth and preparation for real-world interviews.

3.1.3 Detailed Feedback and Insights

The Detailed Feedback and Insights feature is a cornerstone of the Naukri Sahayak platform, designed to provide users with an in-depth understanding of their mock interview performance. Beyond surface-level evaluations, this feature delivers actionable and personalized feedback that enables users to identify areas of strength, recognize opportunities for improvement, and develop strategies for success.

After completing a mock interview session, users receive a comprehensive feedback report. This report includes a performance breakdown, highlighting critical areas such as technical proficiency, problem-solving abilities, communication skills, and behavioral attributes like confidence and clarity. The system uses AI-driven analysis to evaluate each response in terms of relevance, depth, and alignment with the expectations of the chosen job role and industry. Behavioral interviews are assessed for soft skills, including articulation, body language (if integrated with video), and emotional intelligence.

To ensure actionable improvement, the feedback is supplemented with specific suggestions tailored to the user's weaknesses. For example, a user struggling with concise communication might receive tips on structuring answers effectively, while someone lacking domain knowledge might be directed to relevant resources or questions from the platform's question bank.

Additionally, the feedback includes strengths analysis, celebrating the user's achievements and areas where they excel, boosting their confidence. The feature also offers an overall performance grade and benchmarking data, comparing the user's performance against industry standards or similar professionals. This benchmarking helps users gauge their readiness and competitiveness in the job market.

The insights are presented in an easy-to-understand format, with visual aids such as charts and graphs to track progress across multiple sessions. By providing such granular, actionable, and visually engaging feedback, the platform ensures users can continuously refine their skills and approach interviews with greater confidence and preparedness.

3.2 Models and Algorithms

In these section, we elaborate on the Model and Algorithms that are applicable in our Naukri Sahayak Project.

3.2.1 Gemini API Model

The Gemini API Model serves as the backbone for the AI-driven functionalities of the Naukri Sahayak platform, providing powerful tools for generating, analyzing, and delivering dynamic interview experiences. This API integrates advanced machine learning and natural language processing (NLP) models to simulate realistic interview scenarios, assess user responses, and provide comprehensive feedback.

At its core, the Gemini API leverages state-of-the-art NLP techniques to create personalized and context-aware interview questions. These questions are dynamically tailored to the user's job role, industry, and experience level, ensuring relevance and alignment with professional expectations. During mock interviews, the Gemini API processes user responses—whether text or voice—through advanced linguistic and semantic analysis to evaluate their accuracy, relevance, and clarity.

The API also plays a critical role in the feedback process. It helps generate insights into a user's strengths, weaknesses, and areas for improvement by comparing responses to predefined benchmarks or ideal answers. It supports additional features like sentiment analysis, tone detection, and contextual understanding, enhancing the behavioral assessment of users.

By integrating the Gemini API, the Naukri Sahayak platform ensures a seamless, real-time, and highly adaptive interview experience. Its robust capabilities enable scalable, accurate, and intelligent interactions, empowering users to prepare effectively for their career opportunities.

3.2.2 Cosine Similarity

Cosine similarity is a crucial mathematical technique used in the feedback model of the Naukri Sahayak platform to analyze and evaluate user responses during mock interviews. It measures the similarity between two vectors, which, in the context of this platform, represent the user's response and the ideal or reference response for a given interview question.

Both the user's response and the ideal response are transformed into numerical vector representations using natural language processing (NLP) techniques such as word embeddings (e.g., Word2Vec, GloVe). Cosine similarity computes the cosine of the angle between these two vectors. A score close to 1 indicates high similarity, meaning the user's response is closely aligned with the ideal answer. A score near 0 suggests little to no similarity. The similarity score is incorporated into the feedback model to assign grades to user responses, ensuring a fair and standardized evaluation process.

By leveraging cosine similarity, the Naukri Sahayak platform ensures a scientifically robust and transparent method for analyzing user responses. This integration enhances the accuracy of the feedback model, helping users make meaningful progress in their interview preparation journey.

3.2.3 BlazeFace

BlazeFace is a lightweight and efficient face detection model designed for real-time applications, particularly on mobile and web platforms. In the context of the Naukri Sahayak platform, BlazeFace can be integrated into the feedback model to enhance the evaluation of behavioral and non-verbal communication skills during video-based mock interviews. By detecting and analyzing facial expressions, BlazeFace contributes to a more comprehensive assessment of the user's performance.

BlazeFace operates by identifying facial landmarks and tracking expressions such as eye contact, smiles, and other emotional cues, all in real-time. This information is used to evaluate key non-verbal traits like confidence, attentiveness, and emotional control—factors that play a critical role in interviews. Its lightweight design ensures minimal latency, making it suitable for seamless integration without compromising the user experience. By analyzing facial expressions alongside verbal responses, the platform provides richer, multi-dimensional feedback, helping users improve both technical and behavioral aspects of their performance. This integration ensures users are well-prepared for real-world interviews where non-verbal cues significantly impact impressions.

3.2.4 Tensorflow

TensorFlow, an open-source machine learning framework, is a critical component in the feedback model of the Naukri Sahayak platform. It provides the foundation for implementing advanced AI capabilities such as natural language processing (NLP), speech recognition, and facial expression analysis, enabling the platform to deliver highly accurate and personalized feedback.

In the platform, TensorFlow powers models that evaluate user responses in real-time. For text-based answers, TensorFlow's NLP capabilities help generate embeddings and compute metrics like cosine similarity to compare the user's response with ideal answers. In video-based interviews, TensorFlow facilitates facial expression analysis and speech processing, leveraging its deep learning libraries to detect non-verbal cues like eye contact, tone, and confidence. TensorFlow's flexibility allows the integration of pretrained models and the creation of custom ones, ensuring high adaptability to job-specific requirements.

By using TensorFlow, the platform ensures efficient computation and scalability, enabling real-time feedback delivery without compromising performance. This robust framework enhances the accuracy and depth of the feedback, helping users refine both technical and behavioral skills for their interview preparation.

3.2.5 Face Landmarks Detection

Face landmarks detection is a computer vision technique used to identify and track key points or features on a face, such as the eyes, nose, mouth, and jawline. This technology plays a significant role in enhancing the behavioral feedback capabilities of the Naukri Sahayak platform, especially during video-based mock interviews.

Using advanced face landmarks detection models, the platform can analyze users' non-verbal cues in real time. These cues include eye contact, facial expressions, head movements, and emotional states, which are essential components of effective communication during interviews. For example, consistent eye contact might indicate confidence and attentiveness, while excessive fidgeting or a lack of expression might suggest nervousness or disengagement.

Face landmarks detection works by identifying precise points on the user's face and tracking their movement throughout the interview. These points are processed to measure metrics like gaze direction, smile frequency, and overall facial symmetry during interactions. The lightweight and efficient algorithms used in this detection ensure smooth real-time performance without lag, creating a seamless experience for users.

3.2.6 MediaPipe

MediaPipe, an open-source framework by Google, is a powerful tool for building cross-platform machine learning solutions with a focus on real-time perception. Within the Naukri Sahayak platform, MediaPipe plays a pivotal role in enhancing behavioral analysis through features like facial landmarks detection, gesture recognition, and body pose estimation during mock interviews.

MediaPipe's Face Detection and Face Mesh modules are used to identify and track facial landmarks such as eyes, nose, mouth, and jawline with high precision. These landmarks allow the platform to analyze non-verbal communication cues like eye contact, facial expressions, and emotional states. For instance, it can assess whether a user maintains steady eye contact or displays confidence through their expressions. MediaPipe's real-time capabilities ensure these analyses occur seamlessly during live video interviews.

Additionally, Hand Tracking and Pose Estimation modules can be integrated to evaluate body language and gestures, such as hand movements or posture. These behavioral aspects are critical in interviews, where positive gestures and confident posture contribute significantly to creating a good impression.

MediaPipe's framework is lightweight and optimized for real-time processing, making it suitable for web and mobile applications. Its cross-platform compatibility ensures that users experience consistent functionality regardless of the device they use. By leveraging MediaPipe, the Naukri Sahayak platform delivers a richer, more nuanced feedback experience, enabling users to refine their non-verbal communication skills and excel in real-world interview scenarios.

3.3 System Architecture

In this section, we will elaborate on the key design ideas and concepts regarding the architecture of the proposed Naukri Sahayak. As depicted in Figure 3.1

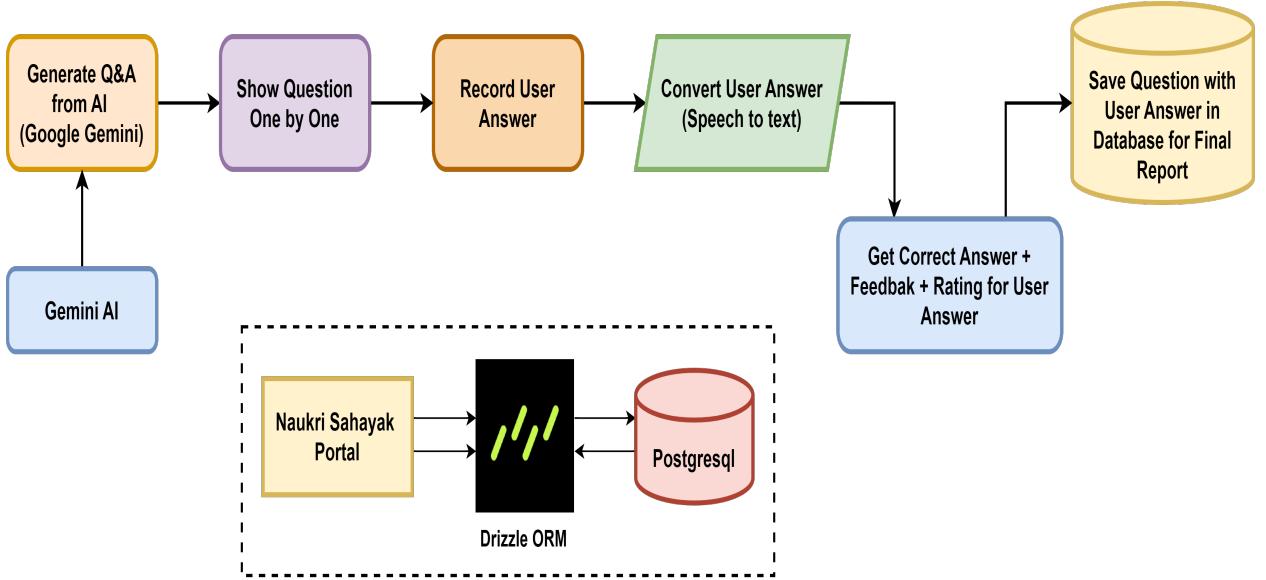


Figure 3.1: System Architecture for the Naukri Sahayak

This system architecture represents the workflow of the Naukri Sahayak AI-Powered Mock Interview Platform, designed to deliver personalized mock interviews and detailed feedback to users. Below is a descriptive breakdown of the entire process:

1. Generate Q&A from AI (Google Gemini)

The process starts with the integration of Google Gemini AI, an advanced AI model capable of generating context-specific interview questions and corresponding ideal answers. These questions are dynamically tailored to match the user's job role, industry, and experience level, ensuring relevance and a realistic interview experience. For example, a senior software developer might receive technical algorithm questions, while an entry-level candidate may get foundational queries.

2. Show Questions One by One

The system then displays the questions on the Naukri Sahayak Portal interface, presenting them one at a time to maintain focus and simulate an authentic interview setting. This design prevents cognitive overload and enables users to respond thoughtfully to each question.

3. Record User Answer

Users provide answers to the displayed questions:

They can type their responses if it's a text-based interview. Alternatively, for spoken interviews, their answers are captured as audio or video recordings. This feature supports flexibility, mimicking different interview formats commonly encountered in real life.

4. Convert User Answer (Speech-to-Text)

If the user provides an audio answer, the system employs a speech-to-text module to transcribe the spoken response into text. This ensures uniformity for further analysis. For example, the user's verbal response, "I implemented a dynamic programming approach to optimize the algorithm," is converted into text for easier comparison and

feedback generation.

5. Get Correct Answer + Feedback + Rating for User Answer

The system evaluates the user's answer by comparing it with the AI-generated ideal answer. Advanced techniques such as cosine similarity (to measure textual alignment) and semantic analysis are employed to assess the accuracy, completeness, and relevance of the response. During this step:

- Feedback is generated to highlight areas of improvement, such as content depth, terminology, or structure.
- Ratings are assigned based on performance metrics, such as relevance to the question, technical accuracy, and the clarity of the response.
- Additionally, if the answer is video-based, the system may analyze non-verbal cues like tone, facial expressions, and confidence using tools like MediaPipe.

6. Save Question with User Answer in Database for Final Report

Once the analysis is complete, the question, user response, feedback, and ratings are stored in a PostgreSQL database. This centralized storage ensures that users can revisit their session data later. The information is also used to generate detailed progress reports that track the user's performance over multiple mock interviews.

7. Output: Comprehensive Feedback Report

The final output is a detailed report containing:

- All questions asked during the session.
- The user's answers, with corresponding evaluations and feedback.
- Correct or ideal answers for comparison.
- Numerical ratings or grades for overall performance.
- Suggestions for improvement, both technical (e.g., knowledge gaps) and behavioral (e.g., confidence, clarity).

This architecture ensures a seamless, end-to-end mock interview experience by combining AI-generated questions, real-time response processing, and actionable feedback. It equips users with the tools needed to improve their performance across both technical and behavioral aspects of interviews, ultimately increasing their chances of success in real-world scenarios.

Chapter 4

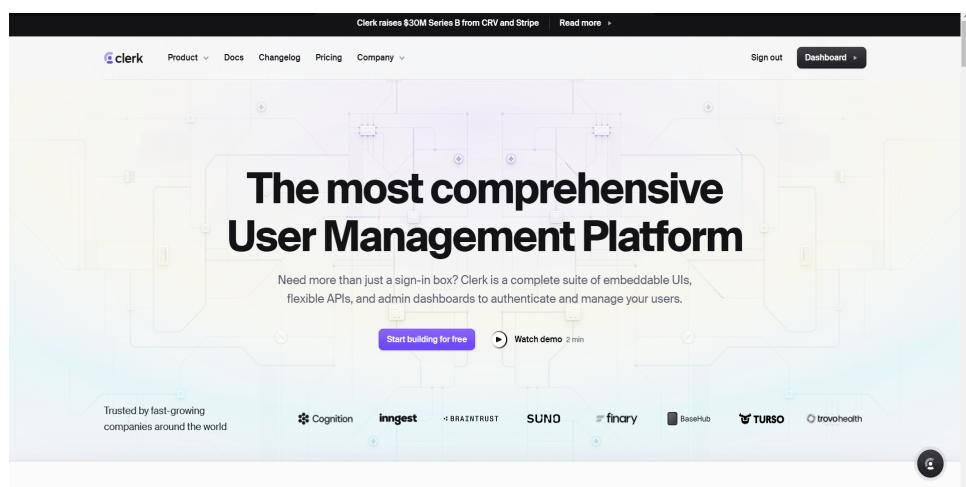
Implementation

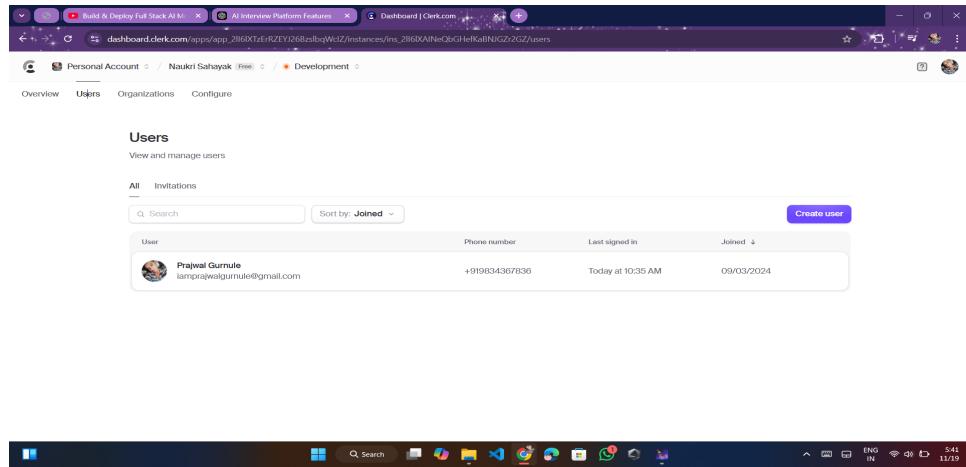
This chapter outline the implementation detail of our AI Mock Interview Portal, encompassing the technology used in React and Next.js . The chapter begins by introducing the chosen User Authentication followed by the Data pre-processing and the Feature Extraction.

This Chapter provides a comprehensive understanding of the technical aspects behind our project.

4.1 User Authentication

Clerk is a modern authentication and user management platform designed to simplify the process of building secure sign-ups, logins, and user management features in web applications. It provides developers with easy-to-implement tools for adding user authentication, identity management, and authorization capabilities to their applications. Clerk authentication plays a critical role in managing user authentication and ensuring secure access to the Naukri Sahayak platform. Clerk, a modern authentication and user management library, is designed to handle authentication workflows, providing a seamless and secure experience for both users and administrators. Here's how Clerk authentication functions within the project:

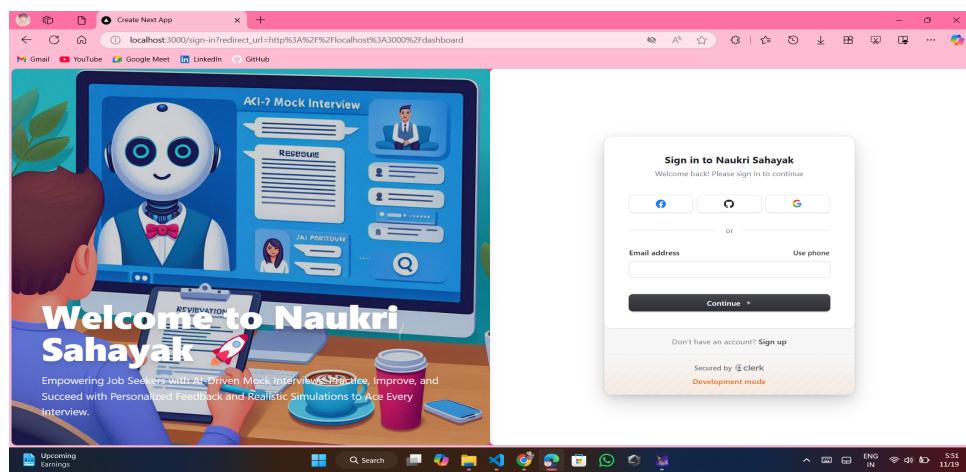




Clerk simplifies the user registration and login process by supporting various authentication methods:

- Email/Password Authentication:** Users can register and log in using their email address and a secure password.
- Social Authentication:** Clerk enables integration with third-party platforms like Google, GitHub, or LinkedIn, allowing users to authenticate via their existing accounts. This is particularly useful for professional platforms like Naukri Sahayak, where users often prefer to connect via LinkedIn.
- Passwordless Authentication:** Options like OTP (One-Time Password) sent to email or phone enhance user convenience.

This ensures that users have flexible options to access the platform while maintaining high-security standards.



Clerk authentication is seamlessly integrated into the Next.js framework used by Naukri Sahayak. It provides APIs and pre-built components that allow the platform to:

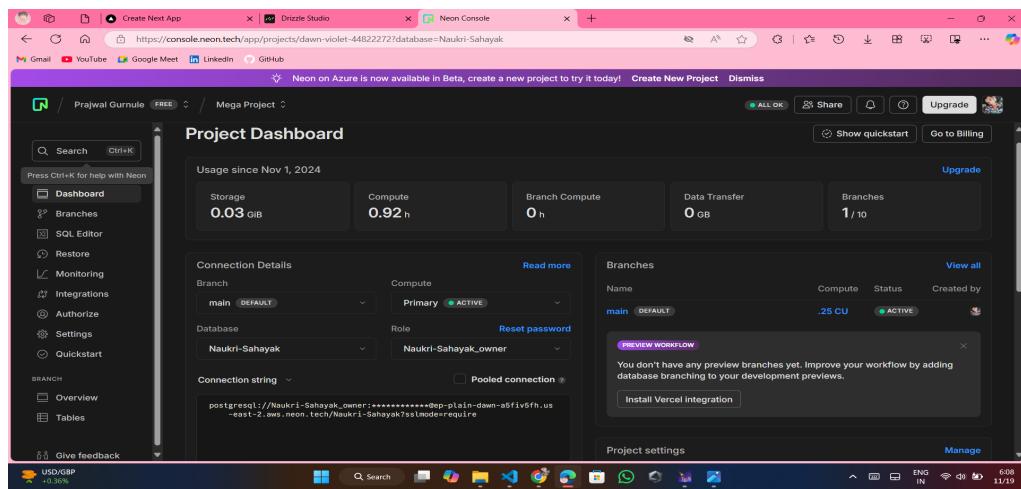
- Protect Routes:** Ensure that pages like interview sessions, feedback reports, or personal dashboards are only accessible to authenticated users.
- Manage User Profiles:** Clerk enables users to update their profiles securely, including personal details like name, email, and preferences for interview settings.

- **Handle Logout:** A secure logout mechanism ensures that users can safely terminate their sessions.

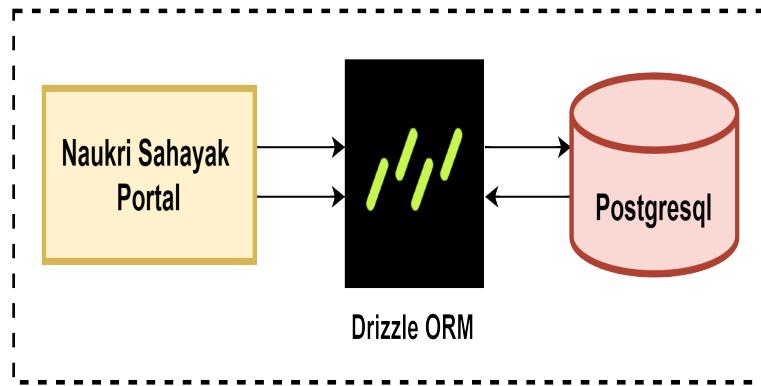
4.2 Backend Setup(Drizzle ORM + PostgreSQL)

The backend setup of the Naukri Sahayak AI-Powered Mock Interview Platform is built using Drizzle ORM and PostgreSQL, ensuring a robust, efficient, and scalable data management system. Drizzle ORM is a modern, TypeScript-first Object Relational Mapper (ORM) designed for simplicity and type safety, making it an ideal choice for handling database interactions in a Next.js environment. It provides a clean abstraction for defining, querying, and managing database schemas while leveraging the power of PostgreSQL as the underlying relational database.

PostgreSQL, an open-source, high-performance database system, is used to store and manage the platform's data, including user information, questions, answers, feedback, and detailed reports. The schema is designed to handle complex relationships, such as associating users with their interview sessions, storing AI-generated questions, and linking user answers with feedback and ratings. With Drizzle ORM, developers can write type-safe SQL queries directly in TypeScript, reducing the chances of runtime errors and improving code maintainability.



The setup also benefits from Neon serverless PostgreSQL, which offers scalability and reduced operational overhead, ensuring that the platform can handle a growing number of users seamlessly. By combining Drizzle ORM's developer-friendly APIs with PostgreSQL's robust data-handling capabilities, the backend ensures fast query execution, secure data storage, and smooth integration with the AI models and frontend. This setup provides a solid foundation for delivering a seamless and interactive experience to users while maintaining data consistency and reliability.



The database schema consists of key tables, including the `mockInterview` table, which stores data such as AI-generated questions (in JSON format), job-related details like position, description, and experience level, as well as metadata such as the user who created the session and timestamps. Drizzle ORM enables developers to define these tables and their relationships in TypeScript using a schema builder, ensuring type safety and robust query execution. For instance, the `mockInterview` table captures essential data for each interview session, with columns for storing dynamic mock questions, the job role, and session-specific identifiers. The use of JSON for questions ensures flexibility in handling structured or unstructured data, while other fields like overall provide placeholders for performance metrics. This structured backend design, combined with Drizzle's modern ORM capabilities, facilitates efficient database operations, streamlined data access, and easy scaling for the platform's growing user base.

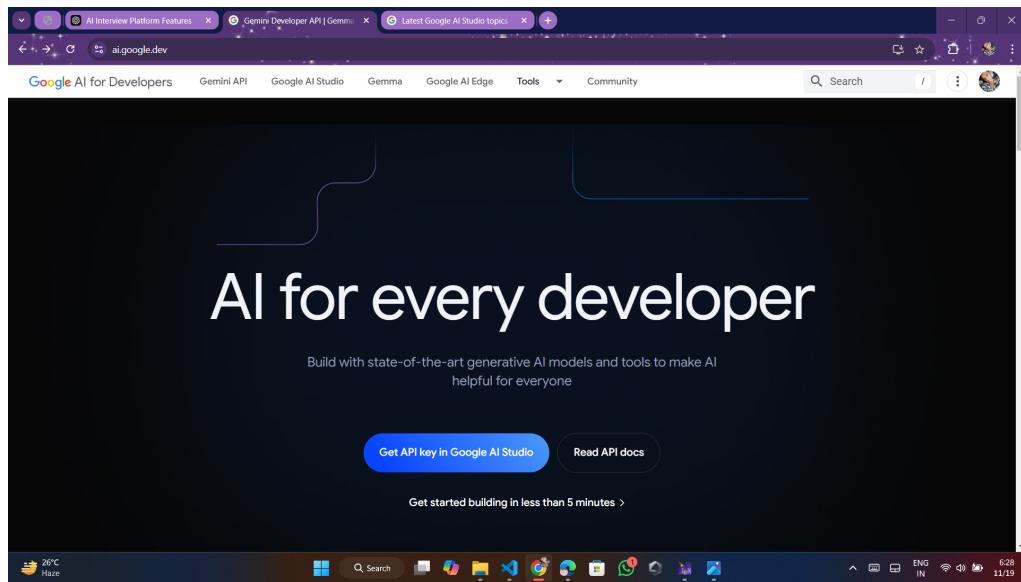
The screenshot shows a web-based interface for managing a database table named `mockInterview`. The table has the following structure:

	<code>id</code>	<code>jsonMockResp</code>	<code>jobPosition</code>	<code>jobDesc</code>	<code>jobExperience</code>	<code>createdBy</code>	<code>createdAt</code>	<code>mockId</code>	<code>overall</code>
39	[{"question": "Te...", "jobPosition": "software developer", "jobDesc": "software", "jobExperience": "1", "createdBy": "iamprajwalgurn...", "createdAt": "18-11-2024", "mockId": "4c45f937-ced...", "overall": null}								
40	[{"question": "Ex...", "jobPosition": "javascript developer", "jobDesc": "javascript", "jobExperience": "1", "createdBy": "iamprajwalgurn...", "createdAt": "18-11-2024", "mockId": "7cf15f67-9bc...", "overall": null}								
41	[{"question": "De...", "jobPosition": "Senior Mechanical Eng.", "jobDesc": "Mechanics, A...", "jobExperience": "34", "createdBy": "iamprajwalgurn...", "createdAt": "18-11-2024", "mockId": "7901263a-fcb...", "overall": 0.7}								
42	[{"interviewQuestio...", "jobPosition": "English Teacher", "jobDesc": "english Gram...", "jobExperience": "1", "createdBy": "iamprajwalgurn...", "createdAt": "19-11-2024", "mockId": "e60a8a4b-883...", "overall": null}								
43	[{"question": "De...", "jobPosition": "civil engineer", "jobDesc": "civil", "jobExperience": "1", "createdBy": "iamprajwalgurn...", "createdAt": "19-11-2024", "mockId": "950a8644-a70...", "overall": null}								

4.3 Generate Mock Interview Question with AI

In the Naukri Sahayak project, the Gemini API Model serves as the core AI-powered engine for generating mock interview questions tailored to specific job roles, industries, and experience levels. This process involves leveraging the capabilities of Gemini AI to create dynamic, contextually relevant, and domain-specific interview questions. The workflow integrates Gemini's natural language processing (NLP) capabilities to

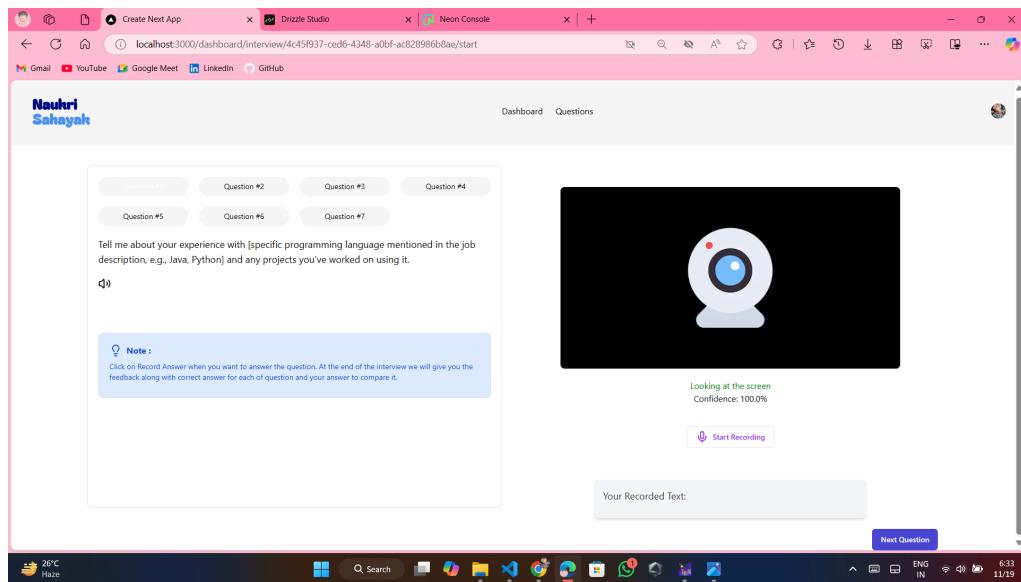
ensure the questions are comprehensive, realistic, and reflective of real-world interview scenarios.



The Gemini API Model plays a pivotal role in generating tailored mock interview questions for the Naukri Sahayak platform. Leveraging the advanced natural language processing capabilities of Gemini AI, the system creates dynamic, contextually relevant questions customized for specific job roles, industries, and experience levels. By using structured inputs such as the job title, domain description, and user expertise, the API generates realistic questions that align with actual industry standards. For example, a "Software Developer" role might receive questions on algorithms, data structures, or programming languages, while behavioral and situational questions assess soft skills and problem-solving abilities. The API allows for real-time adaptability, modifying question difficulty based on user responses to simulate a progressive interview experience. Questions are delivered in various formats, such as open-ended queries, coding challenges, and multiple-choice problems, ensuring comprehensive preparation. This seamless integration of Gemini API ensures users are equipped with a realistic and personalized interview simulation, enhancing their readiness and confidence for real-world interviews.

4.4 Interview Landing Page

The Interview Landing Page of the Naukri Sahayak platform is designed to provide a seamless and interactive experience for users engaging in mock interview sessions. Here's a breakdown of its components and functionality:



- 1. Question Display Panel:** At the top left, there is a navigation bar with numbered tabs for questions. These tabs represent the questions in the interview session, allowing users to track their progress. The current question is displayed prominently with clear text. In this case, the question prompts the user to describe their experience with a specific programming language and related projects.
- 2. Instructions and Guidance:** Below the question, there is a "Note" section providing instructions to the user. It clarifies the steps, such as clicking on "Record Answer" when ready and highlights the feedback process, which involves comparing the user's answer with the correct answer provided by the platform.
- 3. Interactive Audio Features:** Next to the question, there is an audio icon indicating that the platform can read the question aloud. This feature enhances accessibility and user engagement, especially for users who prefer auditory input.
- 4. Webcam Confidence Section:** On the right side, there is a webcam interface with a live feed and visual indicators. This feature monitors the user's face during the interview, providing real-time feedback, such as "Looking at the screen" and confidence metrics (e.g., "Confidence: 100.0%"). This helps users improve their eye contact and non-verbal communication skills.
- 5. Answer Recording Panel:** Below the webcam, there is a "Start Recording" button for users to begin recording their answers. This feature integrates speech-to-text technology, which converts the recorded audio into written text displayed in the "Your Recorded Text" box below. Users can review their responses before proceeding.
- 6. Navigation Controls:** At the bottom-right corner, there is a "Next Question" button that allows users to navigate to the next question after completing their current response. This ensures a smooth and structured flow throughout the mock interview session.

This landing page is designed to simulate a real interview environment while incorporating features like speech-to-text conversion, real-time webcam feedback, and detailed instructions to enhance the user's preparation and confidence.

4.5 Show Interview Question

The Question Generation Page in the Naukri Sahayak platform is designed to dynamically render and manage mock interview questions, providing users with an interactive and user-friendly interface. It uses React for component-based rendering and employs Tailwind CSS for responsive styling. The page generates a list of questions dynamically by mapping through the mockInterviewQuestion array, displaying each question as a tab with its corresponding index. The currently active question is prominently highlighted using conditional styling, ensuring clarity and focus for users. Additionally, the component includes a text-to-speech feature powered by the Web Speech API, allowing users to listen to the questions aloud by clicking on a volume icon. If the browser does not support this feature, a fallback alert notifies the user. Instructions or additional notes are displayed in a "Note" section, represented by a lightbulb icon, with the content fetched dynamically from an environment variable, making it easily customizable. Overall, the component combines dynamic rendering, accessibility features, and responsive design to deliver a seamless user experience for practicing mock interviews.

```

File Edit Selection View Go Run Terminal Help ← → 🔍 naukri-sahayak
EXPLORER ... feedback_model.py M schema.js GeminiAIModel.js route.js ...generate-feedback M QuestionsSection.jsx X route.js ...update-overall-rating M ...
NAUKRI-SAHAYAK
app > dashboard > interview > [interviewID] > start > _components > QuestionsSection.jsx > QuestionsSection
You, 9 hours ago | 1 author (You)
1 import { Lightbulb, Volume2 } from 'lucide-react'
2 import React from 'react'
3
4 function QuestionsSection({mockInterviewQuestion,activeQuestionIndex}) {
5   const textToSpeech=(text)=>
6     if('speechSynthesis' in window){
7       const speech=new SpeechSynthesisUtterance(text);
8       window.speechSynthesis.speak(speech)
9     }
10   else{
11     alert('Sorry, Your browser does not support text to speech')
12   }
13 }
14 You, 9 hours ago * First commit
15 return mockInterviewQuestion&&
16 <div className='p-5 border rounded-lg my-10'>
17   <div className='grid grid-cols-2 md:grid-col-3 lg:grid-cols-4 gap-5'>
18     {mockInterviewQuestion.map((question,index)=>
19       <h2 className='p-2 bg-secondary rounded-full text-xs md:text-sm text-center cursor-pointer' 
20         ${activeQuestionIndex==index&& ' animate-pulse bg-white bg-purple-600'}>Question #{index+1}</h2>
21     ))
22   </div>
23   <h2 className='my-5 text-md md:text-lg'>{mockInterviewQuestion[activeQuestionIndex].question}</h2>
24   <Volume2 className='cursor-pointer' onClick={()=>textToSpeech(mockInterviewQuestion[activeQuestionIndex].question)}>
25     <div className='border rounded-lg p-5 bg-blue-100 mt-20'>
26       <h2 className='flex gap-2 items-center text-blue-700'>
27         <Lightbulb />
28         <strong>Note : </strong>
29       </h2>
30       <h2 className='text-sm text-blue-700 my-2'>{process.env.NEXT_PUBLIC_QUESTION_NOTE}</h2>
31     </div>
32   </Volume2>
33 )
34 }
35
36 export default QuestionsSection

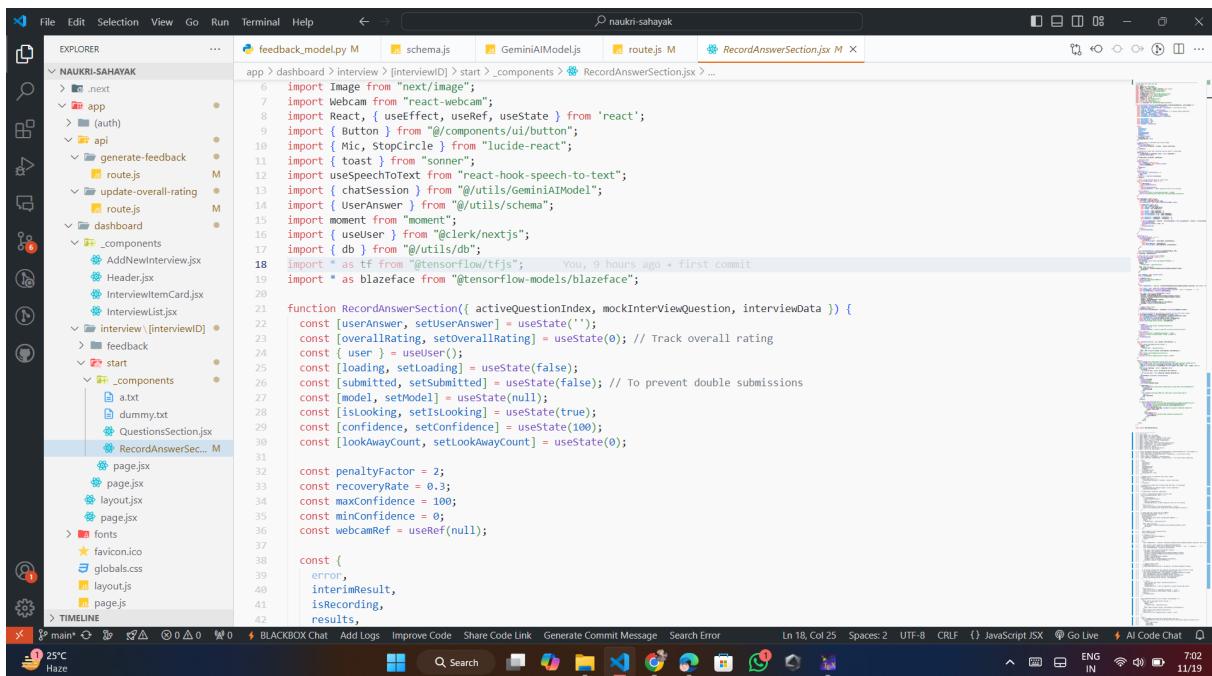
```

File main+ 🌐 ⚡️ 0 0 0 0 BLACKBOX Chat Add Logs Improve Code Share Code Link Generate Commit Message Search Error Ln 14, Col 3 Spaces: 2 UTF-8 CRLF {} JavaScript JSX Go Live AI Code Chat 25°C Haze Search 🌐 ENG IN 654 11/19

4.6 Record User Answer With AI Feedback

The RecordAnswerSection component is a key part of a mock interview system that allows users to record, transcribe, and submit their responses to predefined questions. It integrates multiple advanced functionalities, including speech-to-text conversion, face detection for attention monitoring, and AI-powered feedback generation. The component uses the react-hook-speech-to-text library for handling voice recording and transcription, enabling users to provide spoken answers, which are transcribed and displayed in real-time. Simultaneously, it employs BlazeFace, a TensorFlow.js-based face detection model, to track whether the user is looking at the screen via webcam input. The system calculates a "confidence" score that decreases if the user looks away from the screen and gradually recovers if they remain focused. A penalty factor and recovery rate dynamically adjust this score, with visual feedback (e.g., "Looking at the screen" or "Movement Detected!") displayed to inform the user about their focus.

When the user stops recording, their transcribed response is automatically submitted if it meets a minimum length requirement. The component sends the answer, along with the correct answer, to an API endpoint to generate feedback and a rating. This feedback, produced by an AI model using a predefined prompt, includes constructive suggestions and a numeric rating. The responses, ratings, and feedback are saved in a database, ensuring that all data is properly recorded. After all questions have been answered, the component calculates an overall average rating for the session and updates it in the database. The UI is designed to enhance user experience, featuring a live webcam feed with a mirrored display, a confidence meter, a "Start/Stop Recording" button with state-based animations, and a display of both interim and final transcribed text. Error handling ensures that the process remains smooth, with notifications and loading indicators to guide the user. This component is a comprehensive solution for conducting interactive mock interviews, combining real-time interaction, AI feedback, and an intuitive interface.



4.7 Interview Feedback

The ‘Feedback’ component is designed to display the results and feedback of a mock interview session to the user. It retrieves detailed feedback for each question, along with an overall interview rating, and presents them in an organized and visually appealing manner. Upon mounting, the component triggers two asynchronous functions, ‘GetFeedback’ and ‘GetOverallRating’, to fetch the relevant data from the database. The ‘GetFeedback’ function queries the ‘UserAnswer’ table for records matching the specific interview ID (‘params.interviewID’), sorting the results by their IDs to maintain order. The fetched feedback includes the interview questions, the user’s answers, the correct answers, ratings, and personalized feedback for improvement. Simultaneously, the ‘GetOverallRating’ function queries the ‘MockInterview’ table to retrieve the overall rating for the mock interview session based on the same interview ID. If no rating is found, a fallback message (“No rating available”) is displayed.

The fetched data is displayed in a structured layout. If no feedback records are found, a message (“No Interview Feedback Record Found”) informs the user that no data is available. Otherwise, the component congratulates the user and presents their overall rating, formatted as a score out of 10. It then iterates over the ‘feedbackList’ array to display each question and its corresponding feedback in collapsible sections. Each section contains the question, the user’s answer, the correct answer, the rating (displayed with a star symbol), and constructive feedback for improvement. These collapsible sections, implemented using the ‘Collapsible’, ‘CollapsibleTrigger’, and ‘CollapsibleContent’ components, provide an interactive and clean way to display detailed information while maintaining a compact layout. The ‘CollapsibleTrigger’ serves as a clickable header for each question, showing the question text and a chevron icon (‘ChevronsUpDown’) for toggling visibility. The ‘CollapsibleContent’ reveals the rating, the user’s answer, the correct answer, and the feedback when expanded.

At the bottom, the component provides a ”Go Home” button that redirects the user back to the dashboard. This is achieved using Next.js’s ‘useRouter’ hook to programmatically navigate the user to the ”/dashboard” route. The button is styled with hover effects to enhance the user experience. Overall, this component provides an interactive, user-friendly interface for users to review their interview performance, understand areas of improvement, and track their progress through detailed feedback and ratings. It demonstrates seamless integration with a database and prioritizes clarity, accessibility, and user engagement in presenting the results.

```

File Edit Selection View Go Run Terminal Help < > naukri-sahayak
EXPLORER
NAUKRI-SAHAYAK
> .next
  app
    > auth
    > api
      generate-feedback
        generateFeedback.js
      update-overall-rating
        updateOverallRating.js
    dashboard
      _components
        AddNewInterview.js
        Header.jsx
        InterviewCard.jsx
        InterviewList.jsx
      interview-[interviewID]
        feedback
          dummy-feedback.txt
        page.jsx
      start
        page.jsx
        layout.jsx
        page.jsx
      fonts
        favicon.ico
        globals.css
        layout.js
        page.js
      components/ui
        alert.jsx
        button.jsx
        collapsible.jsx
    > TIMELINE
    > main.js
    > BLACKBOX Chat Add Logs Improve Code Share Code Link Generate Commit Message Search Error
    In 158 Col 25 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live AI Code Chat
    ENG IN 7:13 11/19
  
```

```

const Feedback = ({ params }) => {
  return (
    <div className="p-10">
      {feedbackList?.length === 0 ? 
        <h2>No Interview Feedback Record Found</h2>
        : 
        <h2>Congratulations!! Your interview feedback:</h2>
        <h2>Your overall interview rating: ${overallRating} / 10 : "No rating available"</strong>
      }
      {feedbackList.map(item=>(
        <Collapsible key={index}>
          <CollapsibleTrigger>
            {item.question} <chevronupdown className="h-4"/>
          </CollapsibleTrigger>
          <CollapsibleContent>
            <div className="flex flex-col g-2">
              <h2>${item.rating} / 10</h2>
              <h2>${item.userAnswer} <span>Your Answer: ${item.userANS}</span></h2>
              <h2>${item.correctAnswer} <span>Correct Answer: ${item.cor}</span></h2>
            </div>
          </CollapsibleContent>
        </Collapsible>
      ))}
    )
    <button className="bg-primary text-white hover:bg-primary/80 my-4" onClick={() => router.replace("/dashboard")}>
      Go Home <img alt="Home icon" style={{ height: 15px }} />
    </button>
  </div>
)
}
  
```

Chapter 5

Result & Discussion

This Chapter outline the Results and Discussion described in our Naukri Sahayak Project . This chapter is elaborated with the Question Generation, Confidence Model, Feedback Model, Deployment.

5.1 Question Generation

The question generation through the Gemini API operates by leveraging advanced AI algorithms to produce contextually relevant and job-role-specific interview questions. When a request is made, the Gemini API processes the input, which may include job roles, skill requirements, and associated keywords. Using its large language model capabilities, the API generates a set of questions tailored to evaluate critical competencies and domain knowledge. These questions are returned as structured data, which is then integrated into the application. The client-side component dynamically displays these questions, allowing users to interact with them seamlessly. This approach ensures the generated questions are diverse, role-specific, and aligned with the intended evaluation criteria, providing a comprehensive preparation tool for mock interviews.

The screenshot shows a modal window with a dark header bar containing the title "Tell us more about your job Interviewing" and a close button. Below the title is a subtitle "Add Details about your Job Position/Role, Job Description and years of experience". The form consists of three input fields:

- Job Role/Job Position:** A text input field containing "Javascript Developer".
- Job Description/ Tech Stack (In Short):** A text input field containing "Javascript".
- Years of experience:** A text input field containing "1".

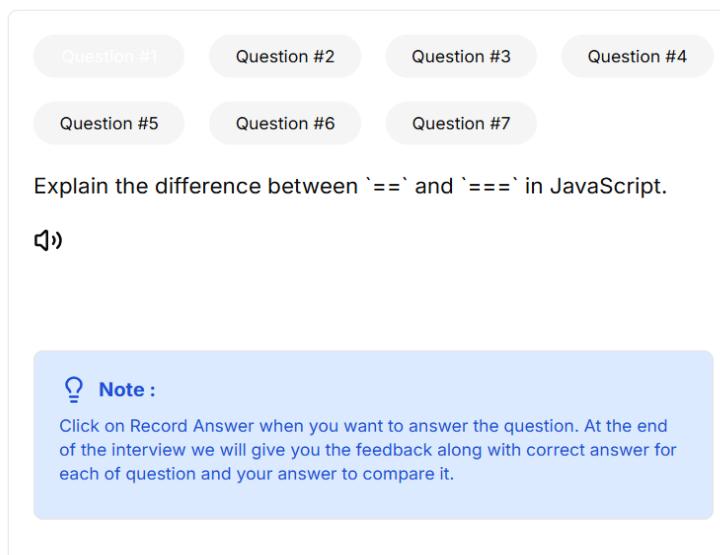
At the bottom right of the modal, there are two buttons: "Cancel" and a purple button labeled "Generating from AI" with a circular loading icon.

The QuestionsSection component effectively presents job-role-specific interview questions generated dynamically by the Gemini model. This design emphasizes a user-centric approach, enabling users to interact with questions through a visually appealing grid layout.

NEXT_PUBLIC_GEMINI_API_KEY=AIZaSyC1I6yqeqK3drFEXoatPN9Mj2TBSYHYuRM

NEXT PUBLIC INTERVIEW QUESTION COUNT=7

Each question is represented as a clickable card, where active questions are highlighted with distinctive animations and colors for clear identification.



The integration of the text-to-speech feature allows enhanced accessibility, catering to diverse user needs by vocalizing the selected question. Additionally, a prominent Note section provides crucial context or instructions, dynamically retrieved from environment variables for flexible configuration. This results in a seamless experience that combines engaging visuals, intuitive functionality, and robust AI-driven content tailored to professional scenarios. The implementation ensures scalability and adaptability for various job roles, making it an invaluable tool for mock interview preparation.

5.2 Confidence Model

The confidence model in Naukri Sahayak is an innovative feature designed to assess and enhance a user's engagement and attentiveness during mock interview sessions. This model leverages TensorFlow.js and the BlazeFace facial detection model to monitor a user's focus in real-time through their webcam. It detects the user's face and eyes, measuring their alignment with the screen to determine whether they are looking directly at it or have been distracted. The system continuously evaluates the user's gaze by calculating the position of key facial landmarks, such as the eyes, and comparing them against predefined thresholds for central alignment. If the user looks away from the screen or moves out of frame, the model identifies this as a loss of focus.



Looking at the screen
Confidence: 100.0%

The confidence metric starts at 100% and dynamically adjusts based on the user's behavior. When the model detects that the user is engaged and looking at the screen, it gradually recovers confidence at a predefined recovery rate (e.g., 0.3% per interval). However, if distractions are detected, the confidence score decreases at a faster penalty factor (e.g., 2% per interval). This dynamic scoring system mimics real-life feedback by rewarding focused behavior and penalizing inattentiveness, encouraging the user to maintain eye contact and stay engaged. The confidence score is displayed in real-time, providing immediate visual feedback to the user, such as a percentage bar or textual indicators like "Looking at the screen" or "Warning: Movement Detected!"

Additionally, the model tracks the number of times a user looks away (look-away count), offering insights into patterns of distraction. This data can be used to generate feedback on the user's overall attentiveness during the interview. By integrating this confidence model into the broader feedback mechanism of Naukri Sahayak, the platform not only evaluates the user's verbal and cognitive responses but also their non-verbal engagement, which is a critical aspect of job interviews. This holistic approach empowers users to improve both their content delivery and presentation skills, thereby boosting their overall readiness for real-world interviews.

5.3 Feedback Model

The feedback model in Naukri Sahayak is a robust system designed to provide insightful, constructive, and actionable feedback to users based on their mock interview performance. This model integrates multiple components, such as real-time speech recognition, a question-answer evaluation mechanism, and advanced natural language processing through the Gemini AI Model. Its primary goal is to help users understand their strengths and weaknesses while preparing for job interviews.

When a user answers a mock interview question, the system captures their response using speech-to-text technology. The transcribed response is then analyzed against a predefined correct answer for the question. The Gemini AI Model processes the user's response by evaluating its accuracy, coherence, and relevance to the question. A custom feedback prompt is generated, instructing the model to evaluate the response and

provide feedback in a concise and actionable manner, usually consisting of a numerical rating (out of 10) and qualitative feedback (3-5 lines). The feedback highlights areas for improvement, such as missing key points, inaccuracies, or suggestions to make the answer more impactful. The feedback system also incorporates user engagement metrics collected from the confidence model. If the confidence model detects frequent distractions or low attentiveness, this can be factored into the feedback to suggest improvements in non-verbal aspects like maintaining focus or avoiding distractions during the interview.

Congratulations !! 🎉

Here is your interview feedback

Your overall interview rating: 0.7/10

Find below interview question with correct answer, your answer and feedback for improvement ✨

Describe your experience with Computational Fluid Dynamics (CFD) and its application in aerospace design. Provide specific examples of projects where you utilized CFD.

Rating: 1 ★

Your Answer: my experience is in water mechanics

Correct Answer: I have extensive experience using CFD software such as ANSYS Fluent and OpenFOAM for aerodynamic analysis and design optimization. For instance, in my work on the [Project Name] project, I used Fluent to simulate the airflow over a new wing design, identifying areas of high drag and improving the overall efficiency by [quantifiable result, e.g., 15%]. I've also leveraged CFD in [mention other relevant projects and specific applications].

Feedback: The answer is completely irrelevant to the question. The question asks about CFD in aerospace design. If you lack experience in this area, honestly state that and highlight transferable skills from water mechanics that might be relevant. Do not provide unrelated information.

Explain your understanding of different turbulence models and when you would choose one over another.

Describe a challenging technical problem you encountered and how you overcame it.

What are your experiences with different aerospace materials and their selection criteria?

Discuss your familiarity with aerospace standards and regulations (e.g., FAA, EASA).

How do you stay current with the latest advancements in mechanical aerospace engineering?

[Go Home 🏠](#)

Additionally, the feedback model supports an overall rating mechanism. As users progress through the series of mock interview questions, their individual question ratings are aggregated to calculate an overall performance rating for the entire session. This is stored in the database and displayed alongside specific feedback for each question. If the user completes all questions, the system calculates an average rating, which is stored in the MockInterview database. This overall rating gives users a clear idea of their overall readiness for interviews and allows them to track progress over time.

The feedback model also provides a seamless interface for users to view detailed insights. It displays the question, the correct answer, the user's response, the assigned rating, and the feedback, all organized in collapsible sections for ease of access. This organization enables users to review each question-answer pair in detail, helping them understand their mistakes and areas of improvement. The model ensures that the feedback is constructive, focusing not just on what went wrong but also on how to improve, making the process highly user-centric.

By combining advanced AI, real-time evaluation, and a focus on user engagement, the Naukri Sahayak feedback model ensures a holistic and effective preparation experience for job seekers. It empowers users to enhance both their technical and soft skills, ultimately increasing their confidence and competence for real-world interviews.

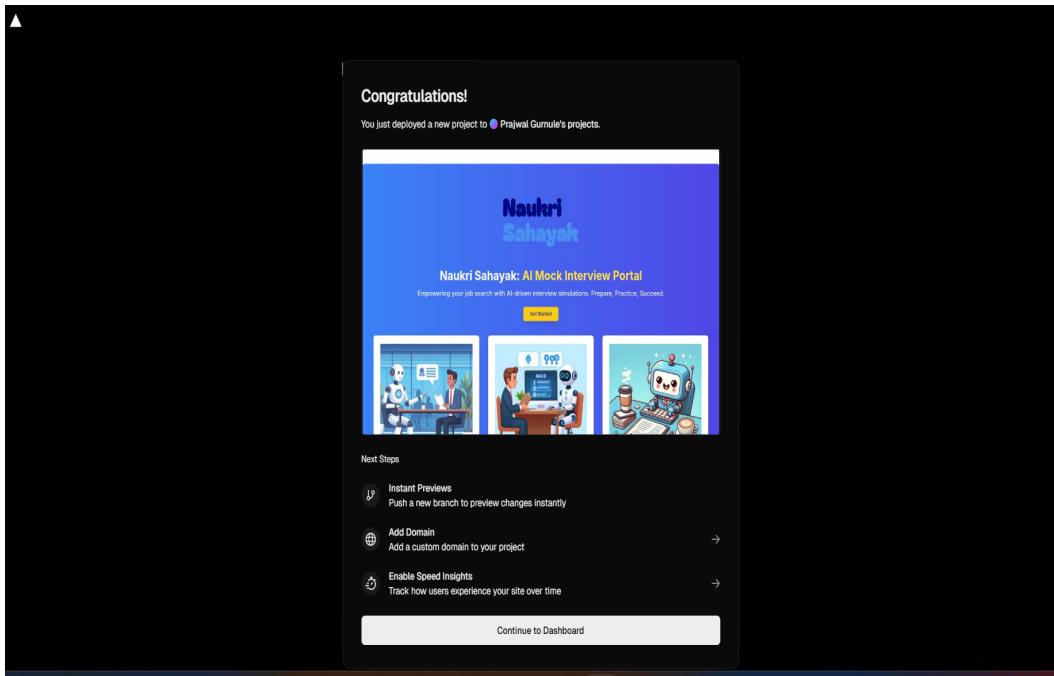
5.4 Deployment

The deployment process of the Naukri Sahayak project was successfully executed on Vercel and Render, ensuring that the application is accessible to users via the web. Deployment involves hosting the locally developed project on a cloud platform to make it publicly available. This ensures seamless delivery of the platform's AI-powered mock interview features to users, enhancing its accessibility and usability.

For deployment on Vercel, the project repository was connected directly to Vercel's platform, which automatically detected the Next.js framework used for development. Vercel's intelligent build system configured the build process, compiled the code, and generated all necessary static and server-rendered files. Once the build was successfully completed, the application was deployed to Vercel's globally distributed Content Delivery Network (CDN), ensuring fast and efficient access for users across different locations. A live deployment link was generated, which provides instant previews and allows users to interact with the application in real-time. Additionally, Vercel's dashboard offered advanced options such as adding a custom domain, monitoring performance metrics, and managing future updates.

On Render, the project was also deployed as a part of a parallel setup for backend and full-stack hosting needs. The process involved linking the project repository to Render, configuring environment variables, defining build commands, and setting run-time configurations. Render's infrastructure handled scaling and ensured continuous availability of the application to end-users.

The deployment on these platforms not only streamlined the hosting process but



also ensured optimal performance, minimal latency, and ease of updates. This setup supports the scalability of the Naukri Sahayak platform, enabling it to handle user demands efficiently while providing a reliable mock interview experience.

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

In conclusion, the proposed system effectively addressed the crucial need for interview preparation in today's job market. Our comprehensive mock interview platform, leveraging advanced AI technologies and diverse evaluation tools, empowered candidates in their preparation journey. The integration of AI-driven video and audio analysis, machine learning algorithms, tools provided immersive and effective preparation experiences.

The proposed system revolutionizes interview preparation by addressing critical challenges faced by candidates in today's highly competitive job market. Combining the power of artificial intelligence with advanced technologies like computer vision and natural language processing, the platform delivers an unparalleled mock interview experience. This innovative system provides candidates with precise, personalized feedback on key aspects such as emotional expression, confidence levels, and communication skills. By analyzing facial expressions, vocal tones, and speech patterns, it accurately assesses performance, offering actionable insights to help users refine their skills.

The integration of emotion recognition and confidence analysis ensures a deeper understanding of candidate behavior, while the intuitive, user-friendly interface makes the platform accessible to individuals with varying technical expertise. Beyond benefiting candidates, the system serves as a powerful tool for educators and interviewers, offering a structured framework to mentor and guide aspirants effectively. This holistic approach not only equips users with the confidence and skills needed for success but also exemplifies the transformative potential of AI in addressing real-world problems, setting a new benchmark in the domain of professional development.

The project development of a sophisticated system designed to provide candidates with personalized feedback on their emotional expressions and confidence levels during mock interviews. Through the integration of advanced artificial intelligence techniques, including computer vision and natural language processing, the system accurately analyzes candidates' facial expressions and speech patterns. Key highlights include emotion recognition, confidence assessment, a user-friendly interface, and performance optimization. The project demonstrates its effectiveness in empowering candidates to enhance their interview skills and offers invaluable support to educators and interviewers.

6.2 Future scope

Multi-Modal Analysis: Integrating additional modalities such as speech, gesture, and body language can provide a more comprehensive assessment of interviewee performance. Analysing speech patterns, hand gestures, and posture alongside facial expressions can offer deeper insights into the interviewee's emotional state, confidence level, and communication skills.

Real-Time Feedback: Implementing realtime feedback during mock interviews can enhance the learning experience for interviewees. Providing immediate feedback on facial expressions, tone of voice, and overall performance can help candidates identify areas for improvement and adjust their behaviour accordingly.

Personalized Recommendations: Developing a personalized recommendation system based on interview performance can guide candidates in areas for skill development and improvement. Tailored suggestions for enhancing communication skills, managing nerves, and projecting confidence can empower candidates to succeed in future interviews.

Cross-Cultural Interview Training : Developing modules tailored to cultural nuances in communication styles, etiquette, and body language can prepare candidates for interviews in diverse global contexts. This feature would be particularly beneficial for multinational job seekers.

Employer-Specific Simulations : Collaborating with organizations to create employer-specific mock interview scenarios can help candidates prepare for the unique expectations and standards of their targeted companies, enhancing job success rates.

Integration with Online Job Portals : Collaborating with online job platforms to sync candidate profiles and recommend tailored interview preparation modules based on job applications can streamline the preparation process and increase job readiness.

References

- [1] Jadhav, Aaditya & Ghodake, Rushikesh & Muralidharan, Karthik & Varma, G & Jagan, Vijaya Bharathi. (2023). INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM) AI Based Multimodal Emotion and Behavior Analysis of Interviewee. 10.55041/IJSREM19049
- [2] . J. Kaur, J. Saxena, J. Shah, Fahad and S. P. Yadav, "Facial Emotion Recognition," 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2022, pp. 528-533, doi: 10.1109/CISES54857.2022.9844366.
- [3] A. W. Qurashi, V. Holmes and A. P. Johnson, "Document Processing: Methods for Semantic Text Similarity Analysis," 2020 International Conference on Innovations in Intelligent Systems and Applications (INISTA), Novi Sad, Serbia, 2020, pp. 1-6, doi: 10.1109/INISTA49547.2020.9194665
- [4] Sahil Temgire , Akash Butte , Rohan Patil , Varun Nanekar, Shivganga Gavhane, 2021, Real Time Mock Interview using Deep Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 05 (May 2021), doi : 10.17577/IJERTV10IS050213
- [5] A. K. A, A. H, N. P. Nair, V. A and A. T, "Interview Performance Analysis using Emotion Detection," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 1424-1427, doi: 10.1109/ICIRCA54612.2022.9985667.
- [6] . S. K. Sinha, S. Yadav and B. Verma, "NLP-based Automatic Answer Evaluation," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode,India, 2022, pp. 807-811, doi: 10.1109/ICCMC53470.2022.9754052
- [7] N. C. Brintha, J. A. Narayana, G. L. V. S. Jaswanth, G. J. Chandrapal and D. Venkat, "Realtime Facial Emotion Detection Using Machine Learning," 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2022, pp. 1-5, doi: 10.1109/ICSES55317.2022.9914318
- [8] S.V. Kedar and D. S. Bormane, "Automatic Personality Assessment: A Systematic Review," in 2015 International Conference on Information Processing (ICIP), Vishwakarma Institute of Technology, Dec 16-19, 2015.

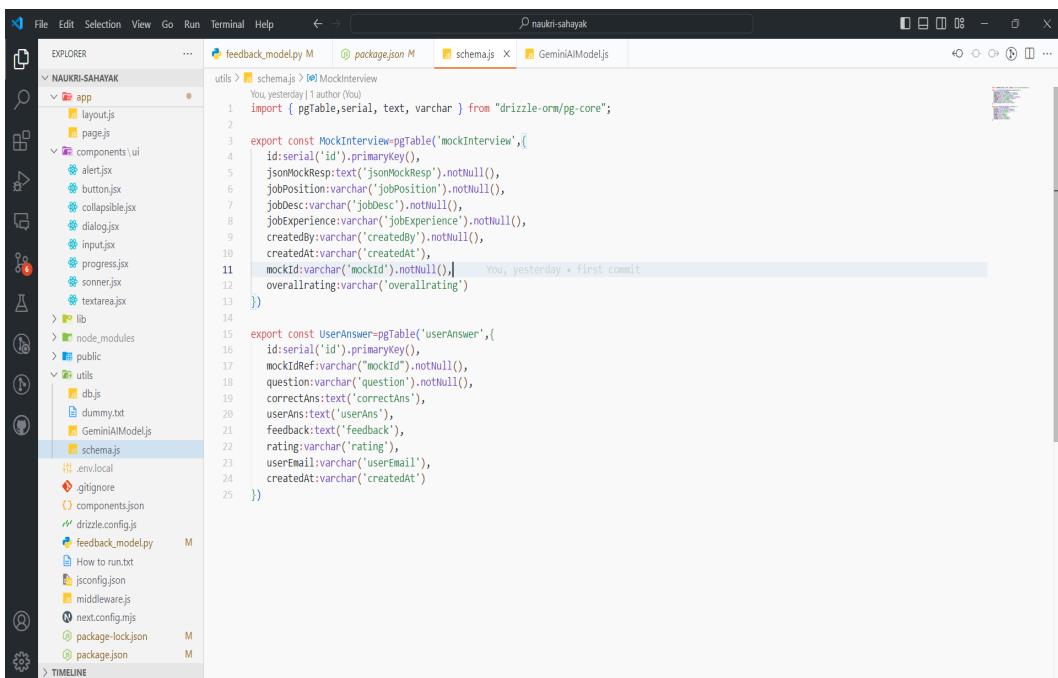
- [9] Yin, L.; Wei, X.; Sun, Y.; Wang, J.; Rosato, M.J. A 3D facial expression database for facial behavior research. In Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, Southampton, UK, 10–12 April 2006; pp. 211–216
- [10] Pathar.R,Adivarekar.A, A. Mishra and A. Deshmukh, "Human Emotion Recognition using Convolutional Neural Network in Real Time," 2019 1st International Conference on Innovations in Information and Comm
- [11] Shreya Narang , Divya Gupta," Speech Feature Extraction Techniques: A Review", International Journal of Computer Science and Mobile Computing, , IJC-SMC, Vol. 4, Issue. 3, March 2015, pg.107 – 114.
- [12] Tyagi, Rinki. (2020). Resume Builder Application. International Journal for Research in Applied Science and Engineering Technology. 8. 14-18. 10.22214/ijraset.2020.5003
- [13] L. S. Nguyen, D. Frauendorfer, M. S. Mast and D. GaticaPerez, "Hire me: Computational Inference of Hirability in Employment Interviews Based on Nonverbal Behavior," in IEEE Transactions on Multimedia, vol. 16, no. 4, pp. 1018- 1031, June 2014, doi: 10.1109/TMM.2014.2307169.
- [14] V. Narkhede, O. Surushe, S. Kulkarni, H. Solanki, T. Ekbote and D. Joshi, "AVA: A Photorealistic AI Bot for Human-like Interaction and Extended Reality," 2023 International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India
- [15] H. Suen, K. Hung, and C. Lin, "Intelligent video interview agent used to predict communication skill and perceived personality traits," Department of Technology Application and Human Resource Development, National Taiwan Normal University, Taipei City, Taiwan, and College of Science Technology, Ningbo University, Ningbo, China, 2019.
- [16] Swati Uparkar, Saurabh Hundare, Varun Gazala, Sarvesh Chaudhari, Ankush Jain. "IntelliView: An AI Based Mock Interview Platform", 2024 International Journal of Scientific Research in Engineering and Management (IJSREM), <https://www.doi.org/10.55041/IJSREM29201>.

Appendix A

Project Requirement

A.1 Database Schema

In the Database Schema section of the Naukri Shayak project, we describe the structure of the database tables and their role in storing and managing data related to mock interviews and user responses.



The screenshot shows a code editor interface with the file 'schema.js' selected in the left sidebar. The code defines two PostgreSQL tables: 'MockInterview' and 'UserAnswer'. The 'MockInterview' table has columns for id (serial primary key), jsonLockResp (text), jobPosition (varchar), jobDesc (varchar), jobExperience (varchar), createdBy (varchar), createdAt (varchar), mockId (varchar), and overallRating (varchar). The 'UserAnswer' table has columns for id (serial primary key), mockIdRef (varchar), question (varchar), correctAns (text), userAns (text), feedback (text), rating (varchar), userEmail (varchar), and createdAt (varchar).

```
utils > schema.js > MockInterview
You, yesterday | author (You)
1 import { pgTable, serial, text, varchar } from "drizzle-orm/pg-core";
2
3 export const MockInterview = pgTable("mockInterview", {
4   id: serial("id").primaryKey(),
5   jsonLockResp: text("jsonLockResp").notNull(),
6   jobPosition: varchar("jobPosition").notNull(),
7   jobDesc: varchar("jobDesc").notNull(),
8   jobExperience: varchar("jobExperience").notNull(),
9   createdBy: varchar("createdBy").notNull(),
10  createdAt: varchar("createdAt"),
11  mockId: varchar("mockId").notNull(),
12  overallRating: varchar("overallRating")
13 })
14
15 export const UserAnswer = pgTable("userAnswer", {
16   id: serial("id").primaryKey(),
17   mockIdRef: varchar("mockId").notNull(),
18   question: varchar("question").notNull(),
19   correctAns: text("correctAns"),
20   userAns: text("userAns"),
21   feedback: text("feedback"),
22   rating: varchar("rating"),
23   userEmail: varchar("userEmail"),
24   createdAt: varchar("createdAt")
25 })
```

The database schema for Naukri Shayak is designed to efficiently store and manage the data related to mock interviews and user responses. It consists of two primary tables: MockInterview and UserAnswer.

The MockInterview table stores detailed information about each mock interview session. This includes the job position and description, the required job experience, and the overall rating for the interview. It also tracks the creator of the mock interview and the timestamp of its creation. The table uses a serial type primary key (id) to uniquely identify each mock interview session, along with a unique mockId to reference each interview instance.

This column likely contains -

	<code>id</code>	<code>jsonMockResp</code>	<code>jobPosition</code>	<code>jobDesc</code>	<code>jobExperience</code>	<code>createdBy</code>	<code>createdAt</code>	<code>mockId</code>	<code>overallRating</code>
	39	[{"question": "Explain your software development experience."}]	software developer	software	1	iamprajwalgurn...	18-11-2024	4c45f937-ced...	NULL
	40	[{"question": "Explain your javascript development experience."}]	javascript developer	javascript	1	iamprajwalgurn...	18-11-2024	7cf15f67-9bc...	NULL
	41	[{"question": "Describe your Senior Mechanical Engineering background."}]	Senior Mechanical...	Mechanics, A...	34	iamprajwalgurn...	18-11-2024	7901263a-fcb...	0.7
	42	[{"interviewQuestion": "What is your favorite programming language?"}]	English Teacher	english Gram...	1	iamprajwalgurn...	19-11-2024	e604a84b-883...	NULL
	43	[{"question": "Explain your civil engineering background."}]	civil engineer	civil	1	iamprajwalgurn...	19-11-2024	950b8644-370...	NULL
	44	[{"question": "Explain your science background."}]	science	science	1	iamprajwalgurn...	20-11-2024	3ac5249b-378...	NULL
	45	[{"interviewQuestion": "What is your favorite drawing tool?"}]	drawing	drawing	1	iamprajwalgurn...	21-11-2024	0f2c3b42-d02...	NULL
	46	[{"question": "Explain your devops engineering background."}]	devops engineer	devops	1	iamprajwalgurn...	21-11-2024	3bbb8c59-a0a...	NULL
	47	[{"question": "Explain your next.js background."}]	next.js	next.js	1	iamprajwalgurn...	21-11-2024	bdc617df-e98...	NULL
	48	[{"question": "Explain your project management background."}]	project manager	project	1	iamprajwalgurn...	21-11-2024	abb7315f-505...	NULL

id: A primary key field of type serial, which automatically generates unique identifiers for each record.

jsonMockResp: Stores the mock interview response in JSON format. This could be the full response, possibly including questions, answers, and ratings.

jobPosition: Stores the job position for the mock interview (e.g., Software Developer, Data Scientist, etc.).

jobDesc: Stores the description of the job for which the mock interview is being conducted.

jobExperience: Stores the required or preferred experience level for the job position.

createdBy: Tracks who created the mock interview entry (e.g., the recruiter, admin, or AI system).

createdAt: A timestamp for when the mock interview was created. **mockId:** A unique identifier for the specific mock interview, useful for linking related data or for tracking multiple mock interviews.

overallRating: A field to store the overall rating given to the mock interview, potentially summarizing the performance of the participant.

The UserAnswer table is used to store the responses provided by users during the mock interview sessions. It is linked to the MockInterview table through the mockIdRef, which ensures each answer is associated with the correct mock interview. This table holds the user's answer, the correct answer, feedback, and a rating for the response. Additionally, it stores the user's email address and a timestamp of when the answer was recorded. This allows the system to evaluate user performance and provide personalized feedback for improvement.

This column likely contains -

id: A primary key field for each answer record.

SQL runner									
Add record 18 rows · 4s 50 0									
	id	mockId	question	correctAns	userAns	feedback	rating	userEmail	createdAt
Drizzle runner	72	7cf15f67-9bc...	Explain the `==` performs lo...	so both are ...	The answer i...	0	iamprajwalgur...	18-11-2024	
schema: public	73	7cf15f67-9bc...	Explain the `==` performs lo...	there is no ...	The answer i...	3	iamprajwalgur...	18-11-2024	
Search tables	74	7981263a-fcb...	Describe you...	I have extensiv...	my experienc...	The answer i...	1	iamprajwalgur...	18-11-2024
mockinterview	75	7981263a-fcb...	Explain your...	I'm familiar wit...	no I don't u...	The answer d...	1	iamprajwalgur...	18-11-2024
userAnswer	76	7981263a-fcb...	Describe a c...	[Describe a spec...	chat GPTchat...	This answer ...	1	iamprajwalgur...	18-11-2024
	77	7981263a-fcb...	What are you...	I have extensiv...	I have no ex...	The answer i...	1	iamprajwalgur...	18-11-2024
	78	7981263a-fcb...	Discuss your...	I'm familiar wit...	I am not at ...	This answer ...	1	iamprajwalgur...	18-11-2024
	79	7981263a-fcb...	How do you s...	I actively parti...	I read artic...	The answer i...	3	iamprajwalgur...	18-11-2024
	80	7cf15f67-9bc...	Explain the `==` performs lo...	hello hello	No specific ...	0	iamprajwalgur...	19-11-2024	
	81	7cf15f67-9bc...	What are clos...	A closure is a f...	Java are use...	No specific ...	0	iamprajwalgur...	19-11-2024
	82	7cf15f67-9bc...	Explain the `==` performs lo...	hello hello	The answer i...	0	iamprajwalgur...	19-11-2024	
	83	3bb80859-ab9a...	Describe you...	In my previous r...	iska answer -	The answer i...	0	iamprajwalgur...	21-11-2024
	84	bdc617df-e98...	Explain your...	Next.js is a Rea...	hello hello	The answer i...	0	iamprajwalgur...	21-11-2024
	85	bdc617df-e98...	Describe you...	In Next.js, SSR ...	hello helloh...	This answer ...	0	iamprajwalgur...	21-11-2024
	86	abb7315f-505...	Tell me abou...	While my formal ...	hello hello	The response...	0	iamprajwalgur...	21-11-2024
	87	abb7315f-505...	Tell me abou...	While my formal ...	hello hello	This respons...	0	iamprajwalgur...	21-11-2024
	88	abb7315f-505...	Tell me abou...	While my formal ...	hello hello	This respons...	0	iamprajwalgur...	21-11-2024
	89	abb7315f-505...	Tell me abou...	While my formal ...	hello hello	The respons...	0	iamprajwalgur...	21-11-2024

mockIdRef: A reference to the mock interview (mockId) to link the user's answer to a specific mock interview.

question: Stores the question asked during the mock interview.

correctAns: The correct answer to the question, which may be used for evaluation purposes.

userAns: The answer given by the user during the mock interview.

feedback: The feedback provided by the system or evaluator for the user's answer. This could be helpful for performance improvement.

rating: A rating given to the user's answer, which could be used to assess the quality of their response (e.g., Excellent, Good, Needs Improvement).

userEmail: The email ID of the user participating in the mock interview.

createdAt: A timestamp for when the user's answer was recorded.

Together, these tables form the backbone of the data management system for Naukri Shayak, enabling the efficient tracking of mock interview sessions, user interactions, and evaluations. The data stored in these tables can be utilized for performance analysis, feedback generation, and continuous improvement of the job preparation process for candidates.

A.2 Packages

clerk/nextjs (5.4.1):

Role: Provides authentication and user management tools for Next.js apps. It helps with user login, registration, and session management.

In the Project: Likely used for handling user authentication and providing a secure user experience in the job-related application.

google/generative-ai (0.17.1):

Role: Google's API for accessing generative AI models, possibly for creating text, images, or code.

In the Project: Could be used for generating content like job descriptions, suggestions, or even enhancing user engagement with AI-generated responses.

neondatabase/serverless (0.9.5):

Role: Serverless database for running SQL queries in a cloud environment with automatic scaling.

In the Project: Likely used for backend database management (e.g., storing user data, job listings, and application statuses).

@radix-ui/react-collapse (1.1.0):

Role: Provides collapsible UI components for React (e.g., accordion-style menus).

In the Project: Can be used to create expandable sections like job listings or filters, improving user interface interactivity.

radix-ui/react-dialog (1.1.1):

Role: Provides a highly customizable dialog component (modals) for React apps.

In the Project: Used for modals to display job details, application forms, or alerts.

radix-ui/react-progress (1.1.0):

Role: Progress bar components to show ongoing tasks. In the Project: Could be used to show loading states during actions like uploading resumes or applying for jobs.

radix-ui/react-slot (1.1.0):

Role: Provides slot-based composition for React components, allowing greater flexibility in how components are structured.

In the Project: Used for custom layouts, enabling dynamic content placement in the UI, such as user profile cards or job listings.

tensorflow-models/blazeface (0.1.0):

Role: A pre-trained TensorFlow model for detecting faces in images.

In the Project: Can be used for verifying user identity or adding face-related features (like profile photos) to user applications.

tensorflow/tfjs (4.22.0):

Role: TensorFlow.js, a library for running machine learning models directly in the browser.

In the Project: Could be used for client-side AI/ML tasks like resume parsing, skill matching, or automatic job recommendations.

class-variance-authority (0.7.0):

Role: Utility for managing class names in React components.

In the Project: Helps in styling components dynamically based on state or props (e.g., changing button styles based on user actions).

clsx (2.1.1):

Role: A tiny utility for conditionally combining class names in JavaScript/React.

In the Project: Simplifies handling of dynamic styles, e.g., for buttons, active states, or error messages.

drizzle-orm (0.33.0):

Role: ORM (Object Relational Mapping) for SQL databases, focusing on making SQL queries simpler and more intuitive.

In the Project: Likely used for managing the database interactions, such as storing user profiles, job listings, and application statuses.

framer-motion (11.5.5):

Role: A library for animations and transitions in React applications.

In the Project: Can be used for adding smooth transitions and animations to the UI (e.g., when a job is applied to or when a user interacts with the interface).

lucide-react (0.439.0):

Role: A set of open-source, customizable icons for React.

In the Project: Used for UI icons, such as buttons for applying to jobs, filtering job results, or navigating the site.

ml-distance (4.0.1):

Role: A library for computing various distance metrics (like Euclidean distance).

In the Project: May be used for similarity matching between users and job listings, for example, based on user skills and job requirements.

moment (2.30.1):

Role: A date/time manipulation library.

In the Project: Used for handling and formatting dates, like job posting dates, application deadlines, or user registration dates.

natural (8.0.1):

Role: A natural language processing (NLP) library for tasks like tokenization, stemming, etc.

In the Project: May be used for analyzing job descriptions, resume text, or user queries to match jobs with applicants more efficiently.

next (14.2.12):

Role: The React-based framework for building server-rendered apps.

In the Project: The core framework for developing the frontend and backend, including rendering job listings, user profiles, and serving API routes.

next-themes (0.3.0):

Role: A Next.js library for adding theme support, such as dark and light modes.

In the Project: Used to provide users with theme customization options (e.g., a dark/light mode for the job portal).

react (18):

Role: The JavaScript library for building user interfaces.

In the Project: Core library used to build the interactive UI elements like forms, job listings, and user interactions.

react-dom (18):

Role: Provides DOM-specific methods for React to render components.

In the Project: Allows rendering React components in the browser, such as the UI for job search, applications, etc.

react-hook-speech-to-text (0.8.0):

Role: A React hook for converting speech into text.

In the Project: Could be used for voice input features, allowing users to search for jobs or apply via voice commands.

react-icons (5.3.0):

Role: A collection of customizable icons for React applications.

In the Project: Used for displaying icons (e.g., job search, application, or user profile actions).

react-webcam (7.2.0):

Role: A React component for integrating webcam functionality.

In the Project: Could be used for users to upload profile pictures or for webcam-based authentication during job application.

sonner (1.5.0):

Role: A React toast notification library.

In the Project: Used for displaying notifications like application success, errors, or

reminders.

tailwind-merge (2.5.2):

Role: A utility for merging Tailwind CSS class names.

In the Project: Helps streamline the use of Tailwind CSS in the UI, resolving conflicts between different class names.

tailwindcss-animate (1.0.7):

Role: Provides animation utilities for Tailwind CSS.

In the Project: Enhances the UI with animations for interactive elements, like buttons, modals, and transitions.

uuid (10.0.0):

Role: A library for generating unique IDs (UUIDs).

In the Project: Used for generating unique identifiers for job listings, user sessions, or applications.