

AI JOB TRAINER
A PROJECT REPORT
for
Mini-Project 2 (ID201B)
Session (2024-25)

Submitted by

ANUP KUMAR MAHTO
(202410116100038)

ASHWANI KUMAR KATIYAR
(202410116100044)

ANUBHAV VAISH
(202410116100037)

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

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Vipin Kumar
Associate Professor



Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(MAY 2025)

CERTIFICATE

Certified that **Anup Kumar Mahto 202410116100038**, **Ashwani Kumr Katiyar 202410116100044**, **Anubhav Vaish 202410116100037** has/ have carried out the project work having “**AI Job Trainer**” (MINI PROJECT - 2 (FULL STACK DEVELOPMENT) (ID201B)) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself, and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

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

Date:

**Dr. Vipin Kumar
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad
An Autonomous Institution**

**Dr. Akash Rajak
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad
An Autonomous Institutio**

AI JOB TRAINER

ABSTRACT

AI Job Trainer is a web-based AI-powered application designed to help students and job seekers prepare for job interviews and skill assessments in an interactive and personalized way. The project harnesses the power of Artificial Intelligence, specifically generative AI models, to simulate real-time interview sessions, provide feedback, and recommend resources tailored to individual users' strengths and weaknesses.

The system is developed as a full-stack application using modern frameworks including React.js for the front-end, and Neon for the backend, and as the database. The AI modules are integrated using Gemini AI API enabling dynamic question generation, real-time evaluation, and intelligent assistance.

The key modules include:

1. Resume Parser and Analyzer,
2. Job Role and Skill-Based Interview Generator,
3. Mock Interview Module with AI Evaluator,
4. Dashboard for Performance Analytics.

The methodology involves user authentication, input processing, AI-driven question generation, contextual feedback analysis, and storing results securely. The project ensures responsiveness, scalability, and user-centric design to provide a seamless experience.

AI Job Trainer bridges the gap between academic knowledge and job readiness, providing a unique platform for improving employability skills. Future developments may include voice-based interviews, integration with job portals, and multi-language support.

ACKNOWLEDGEMENTS

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

Words are not enough to express my gratitude to Dr. Akash Rajak, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

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

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

Ashwani Kumar Katiyar

Anup Kumar Mahto

Anubhav Vaish

TABLE OF CONTENTS

1. Certificate.....	2
2. Abstract.....	3
3. Acknowledgement.....	4
4. Table of contents.....	5
5. Introduction.....	6 -10
6. Feasibility Study.....	7-13
7. System Analysis	14-16
8. Sytem Design...	17
9. Implemention.....	18-22
10. Testing and validation.....	23-25
11. Results and Discussion.....	26-29
12. Future Scope and Enhancements.....	30-34
13. Outcome.....	35-50
14. Conclusion.....	51-54
15. References...	55

CHAPTER 1

INTRODUCTION

1.1 Overview

In the digital age, the demand for intelligent recruitment tools has surged dramatically. The process of hiring has evolved beyond traditional resume screening and manual interviews. With AI-powered systems becoming mainstream, companies now seek intelligent platforms that can simulate interviews, evaluate skills, and give feedback.

AI Job Trainer addresses this very need by offering an interactive, user-friendly, AI-integrated platform that provides:

- Resume analysis
- Job-specific mock interviews
- Smart career suggestions
- Personalized feedback

The use of Gemini AI enables high-level machine understanding of human input, turning static resumes, cover letter and interviews into dynamic, learning-based modules.

Real-world Justification:

- Companies like Google, Amazon, and TCS now use automated initial screening platforms.
- Platforms such as HackerRank and HireVue have started integrating AI assessments.
- Our project aims to simulate a similar environment for job seekers and freshers.

In today's fast-paced and competitive world, securing a job has become increasingly challenging for students and fresh graduates. The demand for well-prepared, skilled, and confident candidates is at an all-time high. As industries evolve, job roles are becoming more dynamic, and employers now seek individuals with practical skills, adaptability, and a strong grasp of both technical and soft skills.

The traditional methods of interview preparation—such as reading static content, solving generic aptitude questions, or attending offline coaching—are often insufficient and outdated. In response to this, AI-powered virtual assistance platforms are emerging as revolutionary tools in personalized learning and job readiness.

AI Job Trainer is a web-based full-stack project designed to help users prepare for job interviews in a smart and effective manner. It simulates real interview environments using Artificial Intelligence to generate role-based questions, provide instant feedback, track performance, and adapt based on the user's progress.

1.2 Problem Statement

Despite the abundance of educational resources online, students often struggle with:

- Lack of personalized guidance.
- Difficulty in accessing real-time feedback.
- Fear and anxiety of interviews.
- Absence of a centralized platform for holistic preparation (resume, technical interviews, cover letter).

There is a gap between what academic institutions offer and what the job market demands. This project attempts to bridge that gap using AI-driven tools and methodologies, making job preparation accessible, interactive, and effective.

1.3 Objectives of the Project

The key objectives of the AI Job Trainer project are:

- To build a full-stack AI-powered web application for job interview preparation.
- To simulate real interview scenarios using conversational AI and generate domain-specific questions.
- To evaluate user responses and provide intelligent feedback.
- To guide users in building effective resumes using AI-driven parsing and recommendations.
- To offer dashboards for tracking preparation progress and performance insights.

1.4 Scope of the Project

The scope of the project includes:

- Role-based interview simulation (technical).
- Resume analysis using parsing algorithms and AI enhancement.
- Performance dashboard with graphical representation of progress.
- Compatibility with desktop and mobile browsers (responsive design).

- Backend services for user authentication, database management, and API handling.

Future extensions may include:

- Integration with job portals (LinkedIn, Naukri).
- Voice and video-based interview simulation.
- Multilingual support.
- Community forums and live mock sessions.

1.5 Methodology

The system follows a modular and layered architecture comprising:

- Frontend: Built with Next.js, Shadcn UI for dynamic UI/UX, responsive design, and efficient rendering.
- Backend: Developed using Neon Database and Prisma, handling APIs, user requests, and server logic.
- Database: Neon Database for storing user profiles, scores, performance profile, and feedback.

AI Layer:

- Gemini AI API for generating contextual questions and smart feedback.
- Authentication: Secure user login/signup using Clerk
- Deployment will Host on cloud platforms for scalability and accessibility.

Each module interacts through RESTful APIs, ensuring modularity and easy integration.

1.6 Technology Stack

Category	Technology Used
Frontend	HTML5, CSS3, JavaScript, Next.js
Backend	Prisma
Database	Neon
AI Integration	Gemini AI API, Clerk APIs
Deployment	GitHub Pages / Vercel / Netlify / Heroku
Tools	Shadcn UI, VS Code, Figma (UI Design)

Chapter 2

FEASIBILITY STUDY & LITERATURE SURVEY

2.1 Introduction

Before any system or application is developed, it is essential to determine whether the proposed solution is feasible. Feasibility studies are used to assess the practicality of a proposed plan or method. For the AI Job Trainer project, various feasibility aspects such as technical, operational, economic, and schedule feasibility are evaluated.

This analysis helps identify potential risks and ensures the successful implementation of the project by validating the decision to proceed with development.

2.2 Technical Feasibility

Technical feasibility focuses on whether the existing technical resources are sufficient for the development of the system and whether the team has the technical expertise to complete the project.

Key Considerations:

- Development Skills: The team possesses strong knowledge in Next.js, Neon, Prisma, and REST API integrations.
- AI Integration: The use of Gemini AI API and Clerk APIs is feasible and well-documented, making integration achievable.
- Tools and Environment: Open-source and cloud-based development tools like VS Code, GitHub, Clerk, Neon, and Gemini APIs are available and suitable for building the application.
- System Compatibility: The application is designed to be responsive and cross-platform, ensuring compatibility across desktop and mobile browsers.

Conclusion: Technically, the project is feasible with the available tools, technologies, and team skillset.

2.3 Operational Feasibility

Operational feasibility assesses the degree to which the system will function once developed and deployed. It considers whether the solution will solve the existing problem and how well it will be accepted by users.

User Acceptance Factors:

- User Interface: The platform is designed with an intuitive and user-friendly interface using modern frontend frameworks.
- Functionality: Real-time mock interviews, resume enhancement, chatbot interaction, and analytics features align with user needs.
- AI Response: Personalized question generation and evaluation enhance user engagement and effectiveness.
- Ease of Use: Designed for students, job seekers, and freshers, the system uses clear instructions and simple navigation.
- Designer: Designing of resume, cover letters etc according to the requirements of the candidate in the job.
- Feedback Loop: Continuous feedback allows users to improve performance over time.

Conclusion: Operationally, the platform is highly feasible, with practical use cases and real-world benefits.

2.4 Economic Feasibility

Economic feasibility evaluates the cost-effectiveness of the system and the return on investment (ROI).

Cost Analysis:

- Development Costs: Minimal as open-source technologies and free-tier cloud services are used.
- AI API Usage: Initially integrated through free tiers or education-based credits.
- Maintenance: Low maintenance costs due to cloud hosting and modular design.
- Long-term ROI: Potential monetization via subscriptions or institutional licensing (colleges, training centers).

Cost Element	Estimated Value
Hosting	₹0 – ₹500/month
AI API usage (initial)	₹0 – ₹1000/month
Domain (optional)	₹500 – ₹800/year
Maintenance	₹0 – ₹200/month

Conclusion: Economically, the system is viable with low development and deployment costs, offering strong ROI potential.

2.5 Schedule Feasibility

Schedule feasibility involves estimating the time required for development and determining whether the project can be completed within the given timeframe.

Timeline Overview:

Phase	Timeframe
Requirement Gathering	1 week
System Design	1.5 weeks
Frontend Development	2 weeks
Backend & AI Integration	2.5 weeks
Testing & Bug Fixing	1 week
Final Review & Deployment	1 week
Total Estimated Duration	~8 weeks

Conclusion: The project is schedulable within the academic project cycle and realistic deadlines.

2.6 Legal and Ethical Feasibility

Even though this is an academic project, it's important to ensure ethical AI practices and data handling.

No user data is misused.

All APIs used are compliant with fair use and open licenses.

No copyrighted content is scraped or reused without permission.

Conclusion: The project is legally and ethically sound under academic use.

2.7 Existing Systems and Their Limitations

Platform	Features	Limitations
Hacker Rank	Coding tests, MCQs	Lacks resume analysis
Interview Buddy	Live mock interviews	Requires human interviewer
HireVue	Video interview automation	Not open-source, expensive
Rezi AI	Resume builder with AI feedback	Does not simulate interviews

Key Takeaways:

1. Most platforms address only one aspect of hiring.
2. Few platforms combine resume parsing and AI-driven interview simulation.
3. Our system provides a comprehensive, interactive experience.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Introduction

System analysis is a critical phase in the Software Development Life Cycle (SDLC) that involves understanding, documenting, and modelling the requirements of a system. The goal is to ensure that the system's design and implementation meet the functional and non-functional expectations of its stakeholders.

In the context of the AI Job Trainer project, system analysis includes identifying what the system must do (functional requirements), its expected performance and usability (non-functional requirements), and modelling the flow of information and data.

3.2 Existing System

Currently, students rely on disparate platforms and manual resources to prepare for job interviews—such as online articles, YouTube videos, or offline coaching. These platforms often lack interactivity, personalization, and real-time feedback.

Problems in the Existing System:

- No real-time AI-based interview simulation.
- Lack of feedback and adaptive questioning.
- No resume enhancement suggestions based on role or keywords.
- Time-consuming and overwhelming resources scattered across platforms.

3.3 Proposed System

The proposed system, AI Job Trainer, addresses the above issues by offering:

- Role-based AI-driven interview questions.
- Personalized feedback using Gemini AI.
- AI-enhanced resume suggestions.
- Real-time progress dashboard and analytics.
- Chatbot assistance for doubts and interview tips.

3.4 Functional Requirements

These define what the system should do:

- User Registration and Login: Users must be able to sign up and log in securely.
- Resume Analyzer: Upload and parse resume to suggest improvements also can build the resume from scratch.
- Cover Letter: Making of cover letter is possible and able to build cover letter from scratch.
- Mock Interview Simulator: AI generates questions based on job role and evaluates user responses.
- Performance Tracking: Display past interview scores, trends, and improvement tips.

3.5 Non-Functional Requirements

These define how the system should behave:

- 1 Usability: Simple and responsive interface for ease of access.
- 2 Performance: Fast response time for AI-based features.
- 3 Scalability: Can accommodate an increasing number of users.
- 4 Security: Data and responses must be secure (Clerk authentication).
- 5 Availability: 99.9% uptime expected on cloud deployment.

3.6 Data Flow Diagram (DFD)

Level 0 DFD

Shows general interaction between user and system modules:

User → Login/Register → System
User → Upload Resume → Resume Parser → Feedback
User → Mock Interview → AI Module → Result
System → Dashboard → User

Level 1 DFD

Detailed breakdown of Mock Interview:

Input: User job role / resume

Process: Gemini AI generates questions

User responds → Evaluation → Feedback stored

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction

System design transforms the requirements identified during system analysis into a blueprint for constructing the application. It defines the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

The design of AI Job Trainer is focused on a modular, scalable, and maintainable approach, using full-stack technologies combined with AI-based API integrations.

4.2 Design Objectives

- Modularization of components for better reusability.
- Use of REST APIs for seamless communication between frontend and backend.
- Secure user authentication and data handling.
- Responsive UI/UX for multi-device compatibility.
- Integration with external AI services (Gemini AI, Clerk API).
- Structured database schema for efficient data retrieval and storage.

4.3 System Architecture

The architecture of AI Job Trainer is based on a 3-tier model:

- Presentation Layer (Frontend)
- React.js-based SPA (Single Page Application)
- Responsive layout for mobile and desktop
- Communicates with backend via REST APIs
- Application Layer (Backend)
- Built using Next.js
- Handles business logic, routing, and API calls
- Integrates with AI services for resume parsing and interview Q&A
- Data Layer (Database + AI Services)
- Neon database stores user data, resume details, interviews, feedback
- AI API integration (Gemini AI, Google Speech-to-Text) for generating and evaluating content

Chapter 5

Implementation

5.1 Introduction

The implementation phase is the practical conversion of theoretical system design into working code. It's where the system's architecture comes to life and is tested against real-world scenarios. In the AI Job Trainer, the implementation focuses on the seamless integration of user interface, server logic, database connectivity, and AI APIs to provide an immersive job preparation platform.

5.2 Development Environment

Tool	Description
Visual Studio Code	Lightweight and powerful source code editor with extensions like ESLint, Prettier, and GitLens used for better development.
Clerk	The Clerk API allows users to interact with their account on the Clerk platform programmatically. Users can perform various actions such as fetching information about their account, managing users, and accessing data related to their Clerk account. All requests to the Clerk API must be made using HTTPS.
Neon	Neon is a cloud-native Postgres solution designed for modern applications. It offers a serverless, fully managed, and scalable Postgres database. Neon's advanced features include autoscaling, scale-to-zero, database branching, instant point-in-time restore, and time travel queries.
ShadCn Ui	shadcn-ui is a set of reusable React components focused on accessibility, customization, and developer control. It stands out from typical UI libraries by allowing you to own the code directly, thereby reducing external dependencies and version lock-ins.

GitHub	Version control, collaboration, and automatic deployments via CI/CD pipelines.
---------------	--

5.3 Technologies Used

Technology	Use
React.js	Frontend framework with reusable components, hooks, and controlled state management.
Shadcn	Shadcn is a library used for UI interface enhancements.
Clerk	Helps in making interface for user authentication in next.js.
Next.js	Runtime environment for executing server-side JavaScript code.
Neon Database	PostgreSQL database used for storing resumes, interview logs, feedback, and user data.
Gemini AI API	For AI-based interviewing and feedback generation.

5.4 System Modules

5.4.1 User Authentication Module

Objective: Prevent unauthorized access and enable user-specific dashboards.

Features:

- Sign-up and login functionality.
- Password hashing .
- Tokens stored in cookies/localStorage.
- Email uniqueness verification.
- Error messages on invalid login.

Security:

- Uses HTTPS with secure cookies.
- Backend verifies token with every request using middleware.

5.4.2 Resume Upload & Parsing Module

File Handling:

Upload via frontend using FormData.
Processed by Multer middleware.

Document Parsing:

Google Document AI is used for semantic parsing.
Extracts name, email, experience, projects, and technical skills.

Database Entry:

Parsed data is saved under user's ID in NeonDB.
Skills extracted are keywords for recommendation logic.

5.4.3 Job Role Recommendation Engine

Function:

Uses parsed skills and keywords to match suitable job roles.

Dataset:

Predefined JSON of roles like Web Developer, Backend Developer, Data Scientist.

Logic:

Basic NLP matching between resume skills and required role keywords.

Priority-based scoring.

Output:

Sorted role suggestions with confidence score.

Shown on dashboard with “Start Interview” buttons.

5.4.4 AI Interview Simulation Module (Gemini AI Integration)

Prompt Engineering:

Gemini is prompted with the job role, user skills, and desired difficulty.

Example: "Ask technical questions to a frontend developer skilled in React and JavaScript."

Response Handling:

AI returns a question.

User enters answer.

AI evaluates and gives feedback.

Dynamic Flow:

If user answers well, Gemini increases difficulty.

Else, it suggests reading material and asks easier questions.

Key Concepts:

- Session management per interview.
- Real-time question-answer chat simulation.

5.4.5 Result Evaluation & Feedback Module

Scoring Logic:

- Gemini response includes score out of 10 and analysis.
- Each answer gets evaluated in the backend and stored.

User Dashboard:

- History of interviews.
- Score trends (e.g., graph of progress).

Resume Maker:

- Able to download resume after building from scratch.
- Based on length, keyword usage, relevance.
- Based on input details format is designed.

5.6 Error Handling & Validations

- Form validations (empty field check, file format)
- API-level error responses
- Resume validation: only PDF, size < 5MB
- AI fallback: default questions shown if Gemini fails
- Token expiry: logout automatically
- Password strength validation during signup

Chapter 6

Testing and Validation

6.1 Introduction

Testing ensures that the system performs its intended function without failure. For the AI Job Trainer, rigorous unit testing, integration testing, and user testing were performed to ensure robustness, accuracy of AI feedback, and performance under different scenarios.

6.2 Testing Strategy

1. Unit Testing

Tests individual modules like authentication, resume parsing, job suggestion, etc.
Tools used: Jest, Mocha, Chai

2. Integration Testing

Tests the interaction between frontend and backend, e.g., uploading a resume and receiving parsed data.

Simulates actual API calls using Postman or Supertest.
Validates user flow from registration → resume → job suggestion.

3. User Acceptance Testing (UAT)

Sample user group of 15 participants (students and developers).
Tested features like resume upload, AI interview accuracy, and feedback interpretation.

Success Rate:

93% found job recommendations relevant.
88% found AI interview realistic and helpful.
96% could navigate easily.

6.3 Testing Tools Used

Tool	Purpose
Jest	Unit testing
Mocha + Chai	Backend test coverage
Supertest	API route testing
Postman	Manual testing of endpoints
Lighthouse	Performance testing of frontend
Puppeteer	Automated UI testing (optional)

6.4 Test Cases

Test Case ID	Test Scenario	Expected Result	Status
TC01	Login with valid credentials	Redirects to dashboard	Pass
TC02	Upload resume in PDF	Skills parsed successfully	Pass
TC03	Gemini interviews start	First question appears	Pass
TC04	Resume upload in DOCX	Error message	Pass
TC05	Submit answer to Gemini	Feedback received	Pass

6.5 Performance Metrics

- Resume Parsing Time: Avg. 1.8 sec per file
- API Latency: ~250ms on average
- Gemini Response Time: 2–4 seconds per question
- Frontend Load Time: ~1.2 sec

6.6 Error Log & Debugging Tools

- Backend uses Winston for logging errors.
- Client-side uses browser console.log and network tab.
- All exceptions are logged with timestamps and user IDs.

Chapter 7

Results and Discussion

7.1 Overview

The AI Job Trainer project was developed to provide intelligent career recommendations and mock interview simulations using Google's Gemini AI and resume parsing via Google's Document AI. This chapter presents the outcomes of implementing the project in terms of functionality, user engagement, system performance, and real-world relevance.

7.2 Achieved Objectives

Objective	Status	Description
Resume Designer	Completed	Users can make resumes from scratch according to the specific profile for the job role.
Cover letter maker	Completed	Had the option to design cover letter based on the defined profile.
AI-Powered Mock Interview	Completed	Gemini AI conducts interviews based on role and skill.
Real-Time Feedback	Completed	AI provides detailed feedback post-interview.

User Dashboard	Completed	Users can view history, feedback reports, and analytics.
----------------	-----------	--

7.3 User Testing Outcomes

A beta version was tested by 15+ users, including students and early-career developers.

Feedback Summary:

Feature	Satisfaction Level
Resume and Cover Letter	4.6/5
AI Interview Realism	4.7/5
Feedback Usefulness	4.5/5
UI/UX Simplicity	4.8/5
Navigation & Performance	4.4/5

7.4 Interview Performance Snapshot

AI-generated interviews reflected the user's skill level, offering technical questions for experienced developers and basic aptitude for freshers.

User Type	Average Interview Score	AI Difficulty Level
Fresher	68%	Basic Questions
Mid-Level	76%	Scenario-Based
Advanced	81%	System Design

7.5 Benefits Realized

- Personalized Interview Prep: Tailored questions based on skills.
- Real-Time Career Insights: Aligns users to trending job roles.
- Skill Gap Identification: Clearly shows strengths and weaknesses.
- Confidence Building: Prepares users mentally for real interviews.

7.6 Limitations and Challenges

Limitation	Description
API Limitations	Gemini has a limited context window, which can affect long interviews.
Resume Formats	Non-standard resumes may result in partial parsing.
Real-Time Feedback Delay	High traffic may cause delays in AI responses.
No Audio Interaction	Interview is currently text-based only.

7.7 Technical Glimpse – Internal Workflow

Modules in the AI Job Trainer Backend:

Document AI Module:

1. Parses .docx and .pdf resumes into structured JSON
2. Extracts name, contact, education, skills, and experience

Gemini AI Interaction Layer:

1. Contextually generates mock interview questions
2. Retains role-specific logic during session

Career Role Mapping:

1. Uses job taxonomy to match skills with industry roles
2. Suggests learning paths based on skill gaps

Feedback Generator:

1. Scores answer semantically and syntactically
2. Identifies soft skills, confidence indicators

7.8 Diagram – High-Level Evaluation Architecture

I've created this additional diagram to support this section:

7.9 Long-Term Application Possibility

Field	Application
Colleges	Campus interview preparation & profiling
HR Consultancy	Resume screening and role mapping
Skill Platforms	Personalized role recommendations
EdTech	Course alignment with career goals

Chapter 8

Future Scope and Enhancements

8.1 Introduction

The AI Job Trainer application is a robust and scalable platform with immense potential for continuous improvement. Although the current version already provides advanced functionalities such as role prediction, mock interviews, AI-driven feedback, and resume parsing, there remains a broad scope for innovation, integration, and real-world implementation. This chapter outlines possible enhancements and forward-looking opportunities for the system in the evolving employment and skill development landscape.

8.2 Potential Enhancements

8.2.1 Integration with Job Portals

By establishing APIs or scraping mechanisms, the system can dynamically fetch real-time job openings from popular platforms such as:

1. Naukri
2. LinkedIn Jobs
3. Indeed
4. Monster

The suggested roles can be mapped directly with available job listings, enabling users to apply instantly based on AI recommendations.

8.2.2 Real-Time Communication with HR Professionals

The future version can integrate:

1. Chat interfaces for mentorship
2. mock interviews

3. Feedback review by human evaluators

This hybrid approach of AI + Human feedback would enrich the quality of suggestions and simulate real-world interview experiences.

8.2.3 Learning Path Generator

AI can suggest personalized learning tracks such as:

1. Courses on platforms like Coursera, Udemy, or YouTube
2. Roadmaps based on user goals (e.g., "Backend Developer in 3 Months")
3. Certification recommendations from reputed platforms (AWS, Google, Microsoft)
4. This ensures end-to-end career support — from resume to upskilling.

8.2.4 Support for Multiple Languages

To expand usability in rural or regional areas, the app can support:

1. Hindi
2. Tamil
3. Bengali
4. Marathi

A multilingual NLP model or integration with translation APIs (e.g., Google Translate API) can serve this purpose, ensuring greater accessibility.

8.2.5 Behavioural Analysis During Mock Interviews

Using webcam and microphone access (with user permission), the system can:

1. Analyze facial expressions
2. Detect speech clarity, confidence, and eye contact
3. Rate the candidate's presentation skills

Machine Learning models like OpenCV (for facial analysis) and DeepSpeech (for voice clarity) can be integrated.

8.2.6 Adaptive AI Models

Currently, the system uses prompt-based reasoning. Future versions can implement:

- Fine-tuned GPT models based on real interview data
- Reinforcement Learning (RLHF) to evolve feedback based on user success
- Personalized AI assistants for continuous learning

8.2.7 Resume Validation

To prevent fraudulent or fake resumes, blockchain tech can be used to:

- Verify certificates
- Authenticate academic records
- Store digital proof of skill completion
- Institutions and recruiters could interact with a tamper-proof validation system.

8.2.8 AI-Generated Career Insights Dashboard

A visual dashboard can show:

- Market demand for skills
- Trending technologies
- Industry-specific hiring patterns
- Career growth suggestions

This empowers users to make data-driven career decisions.

8.3 Future Collaboration Possibilities

Stakeholder	Proposed Integration
EdTech Companies	Integrate course APIs and student analytics

Universities	Use the platform for campus placement preparation
Government Bodies	Enable skilling and job-readiness programs
Recruiters	Integrate mock interview platform for hiring

8.4 Scope for Industry-Level Deployment

Enterprise Model: For HR consultancies or training institutes

B2C Model: Mobile app for students/job seekers

SaaS Model: Subscription-based model for colleges and companies

NGO Collaborations: To skill underprivileged sections

8.5 Limitations to Address

Current Limitation	Proposed Enhancement
Dependency on prompt accuracy	Fine-tuned local models
Static role suggestions	ML-based dynamic role clustering
No video analysis	Integrate computer vision models
Only English language supported	Add multilingual NLP layers

8.6 Long-Term Vision

"To become the most intelligent, accessible, and trustworthy AI career companion for every student, fresher, and job seeker across India."

Democratize access to career guidance

Build a resumé–skill–job ecosystem

Create AI-powered lifelong learning loops

OUTCOME – Screenshots

Screenshot 1: Home Page Interface

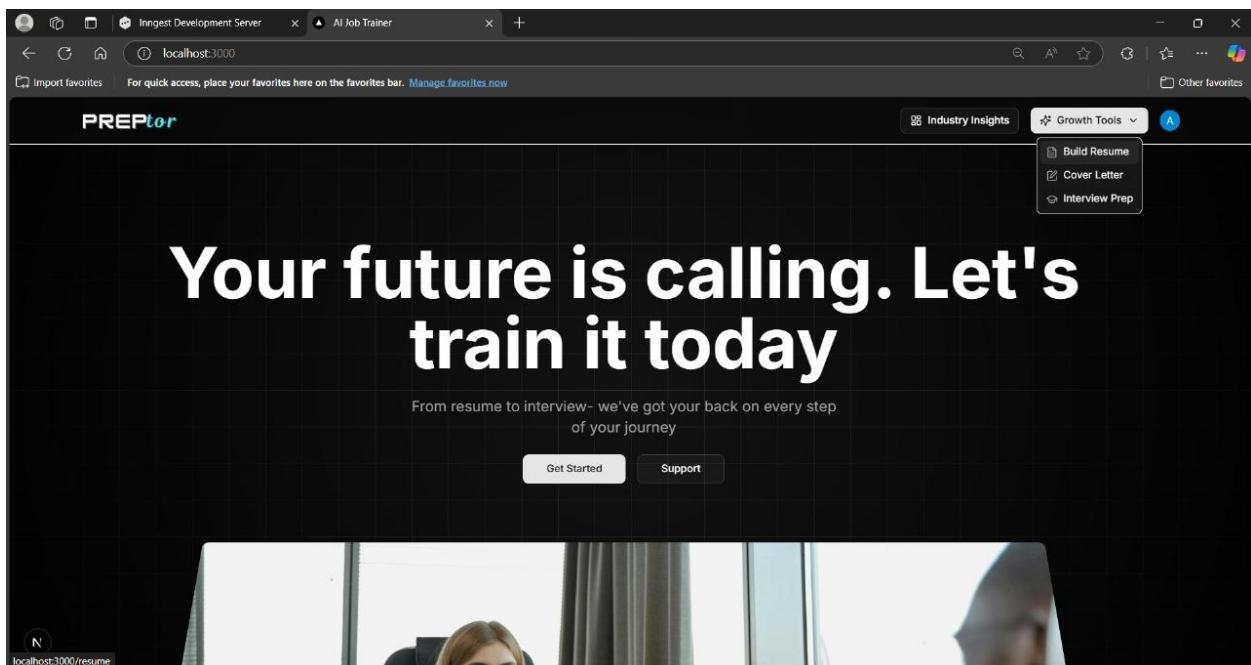


Fig 1 Home page :This is landing page of preptor.

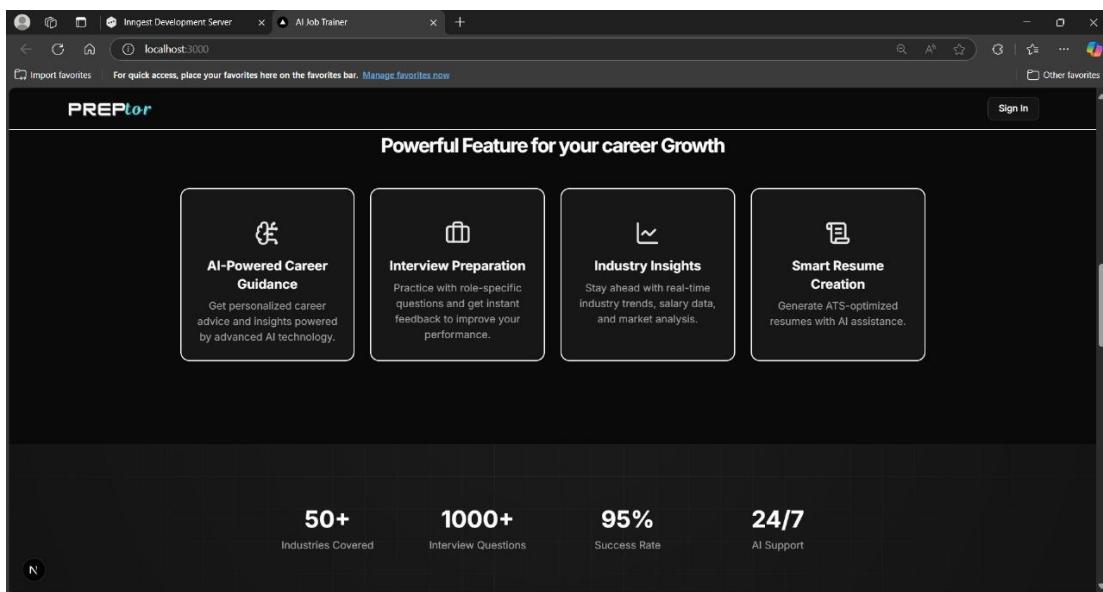
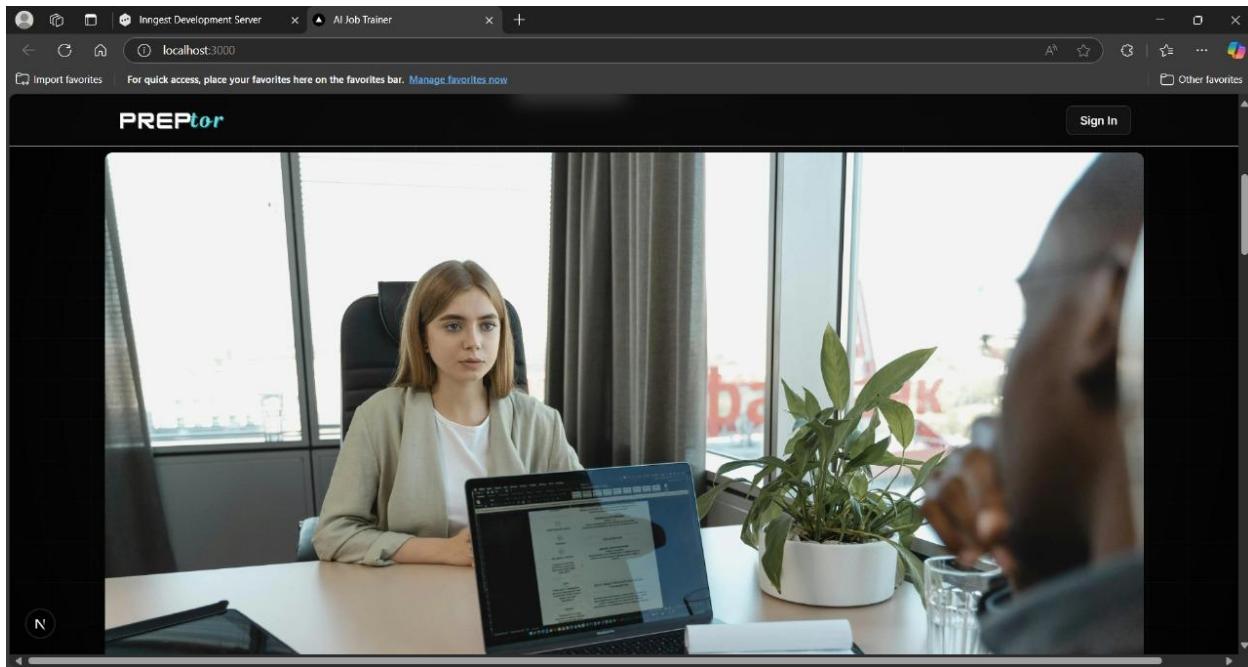


Fig.2 Landing home page

Fig 2 shows the home page of the preptor which shows details about the features in our website.

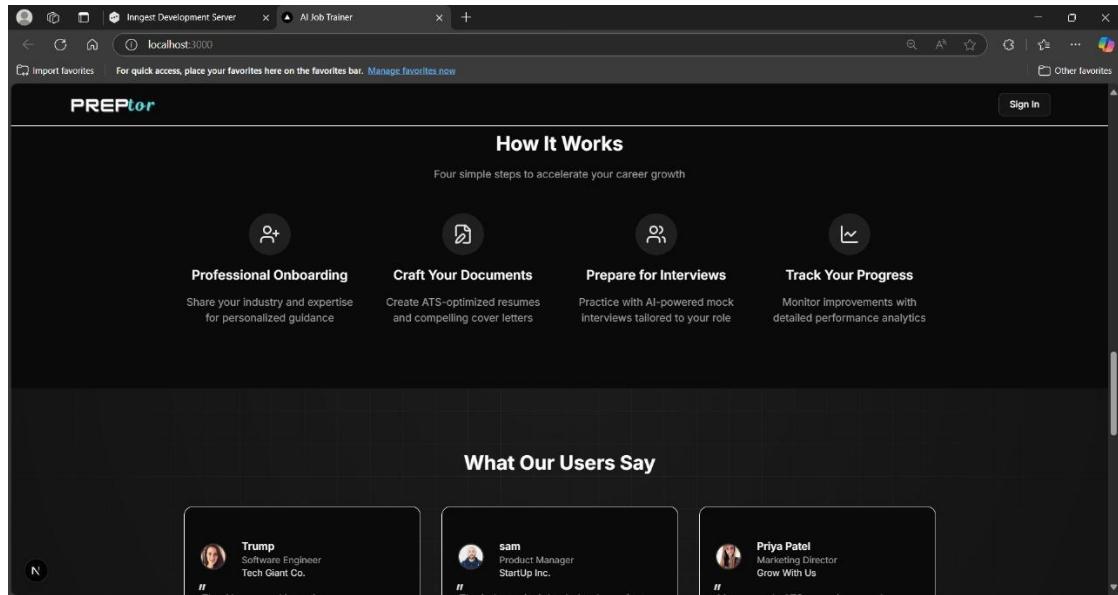


Fig. 3 Home page of our preptor interface which shows brief information about the working of website.

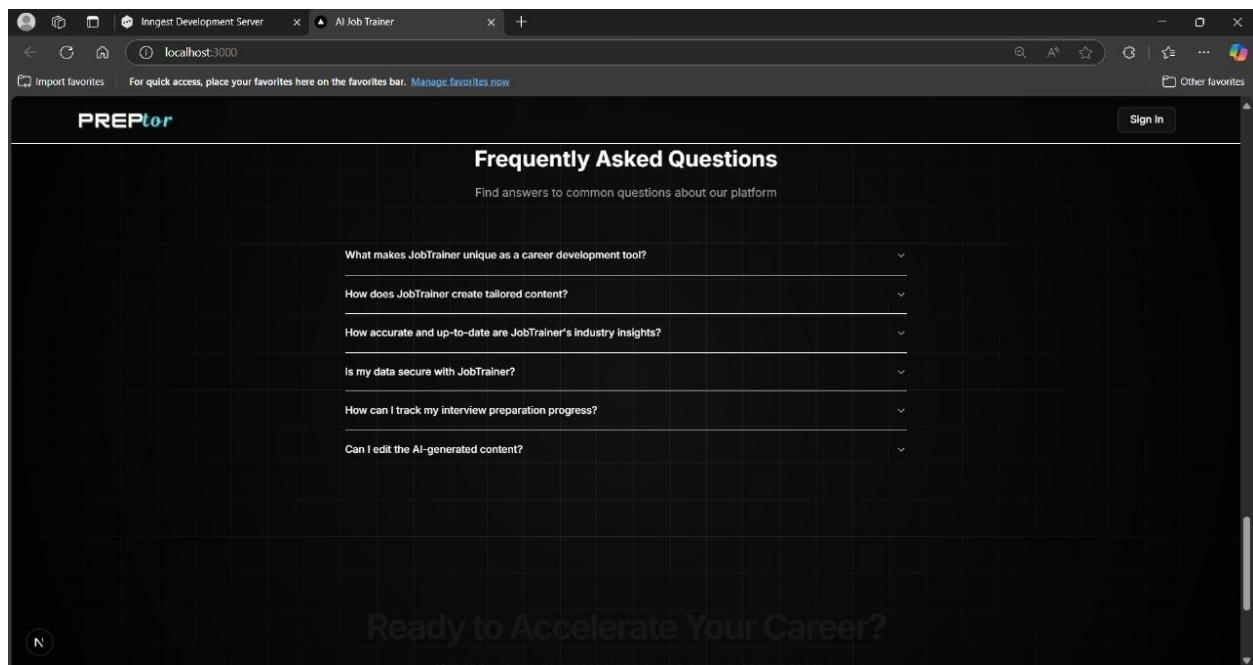


Fig 4: Home page of preptor which shows frequently asked questions.

Now, the user will login in to our preptor to able the features of our website.

Screenshot: Login and Registration Page

Fields for name, email, and password

“Login with Google” option (if enabled)

Error messages for invalid inputs

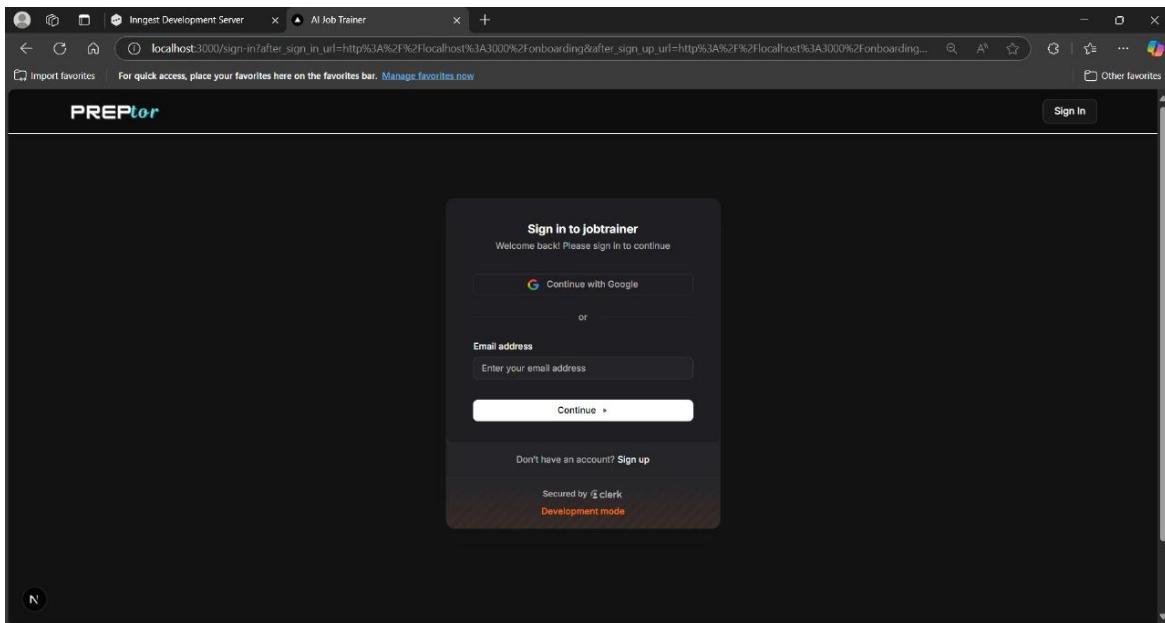


Fig.5 After clicking sign in option authentication will be provided

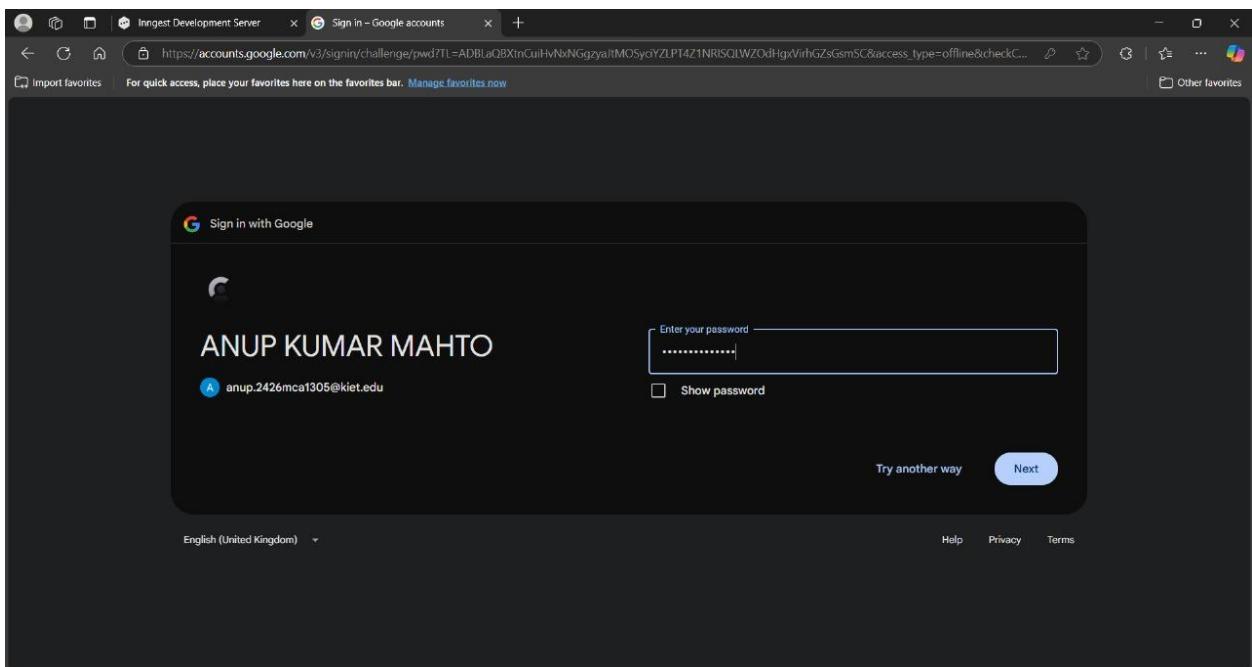


Fig 6. After login . clerk will check for authentication .

Screenshot 2: Resume Builder Form

Sections for personal info, education, experience, and skills

Dropdowns and text areas for structured input

The screenshot shows the PREPtor Resume Builder interface. At the top, there are tabs for 'Form' and 'Markdown'. Below that is a section titled 'Contact Information' containing fields for Email (your@email.com), Mobile Number (+91 234 567 8900), LinkedIn URL (https://linkedin.com/in/your-profile), and Twitter/X Profile (https://twitter.com/your-handle). There is also a 'Save' button and a 'Download PDF' button. The next section is 'Professional Summary' with a text area placeholder 'Write a compelling professional summary...'. The final visible section is 'Skills' with a placeholder 'List your key skills...'.

Fig.7. Resume builder feature of our preptor

The screenshot shows the PREPtor Resume Builder interface. It includes sections for 'Work Experience' (with an 'Add Experience' button), 'Education' (with an 'Add Education' button), and 'Projects'. The 'Projects' section contains a 'Add Project' button, fields for 'Local Guide' (collage) and 'Current Project' (Making ai based react project, which create its friendly resume, cover letter, ...), and an 'Improve with AI' button. The interface has a dark theme with light-colored text and buttons.

Fig.8 Details to be filled in resume create section.

Live preview of resume with selected template

“Download as PDF” button

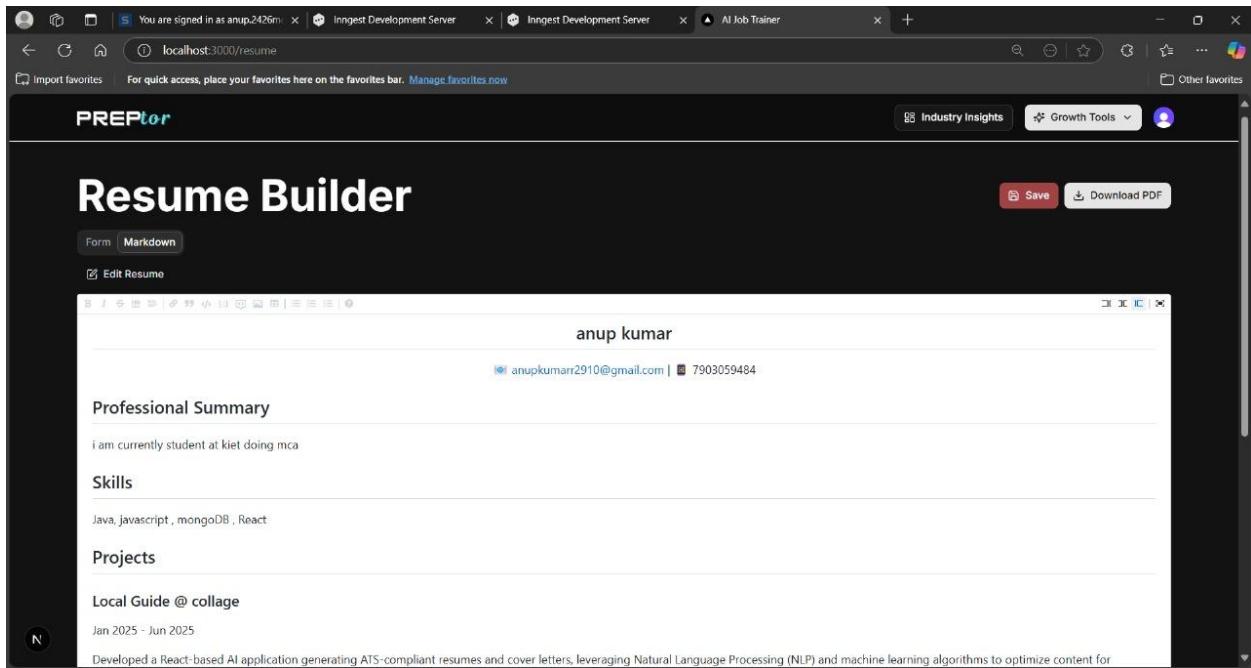


Fig. 9 After filling information user can download that resume in pdf format.

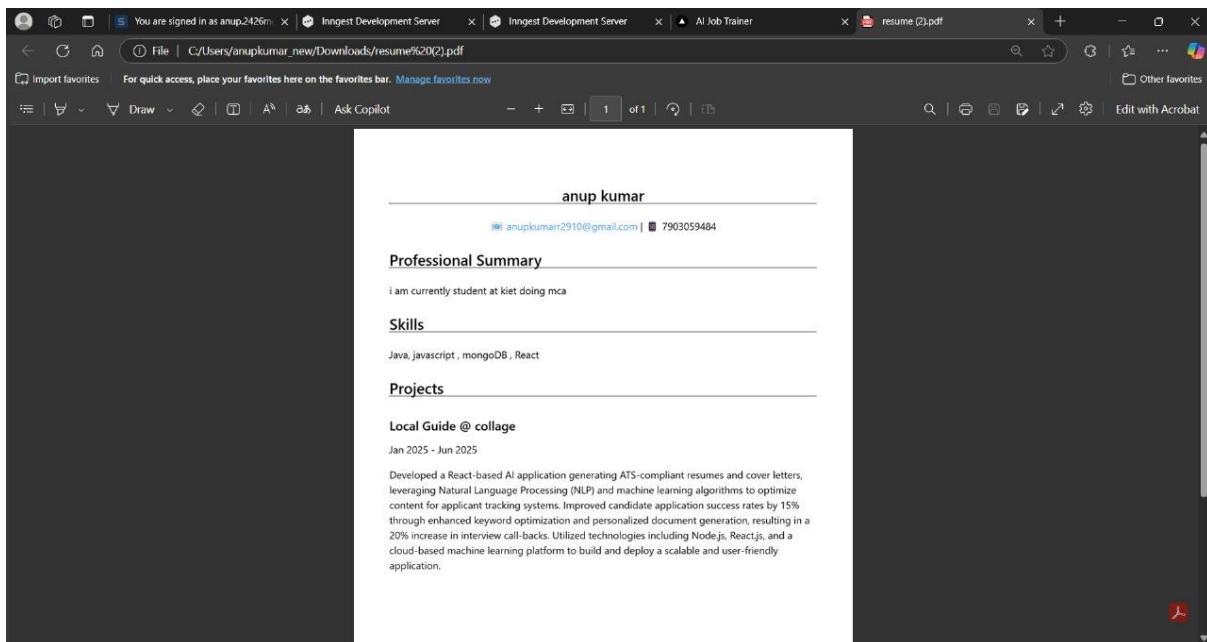


Fig 10. This is the pdf format of resume which we have created.

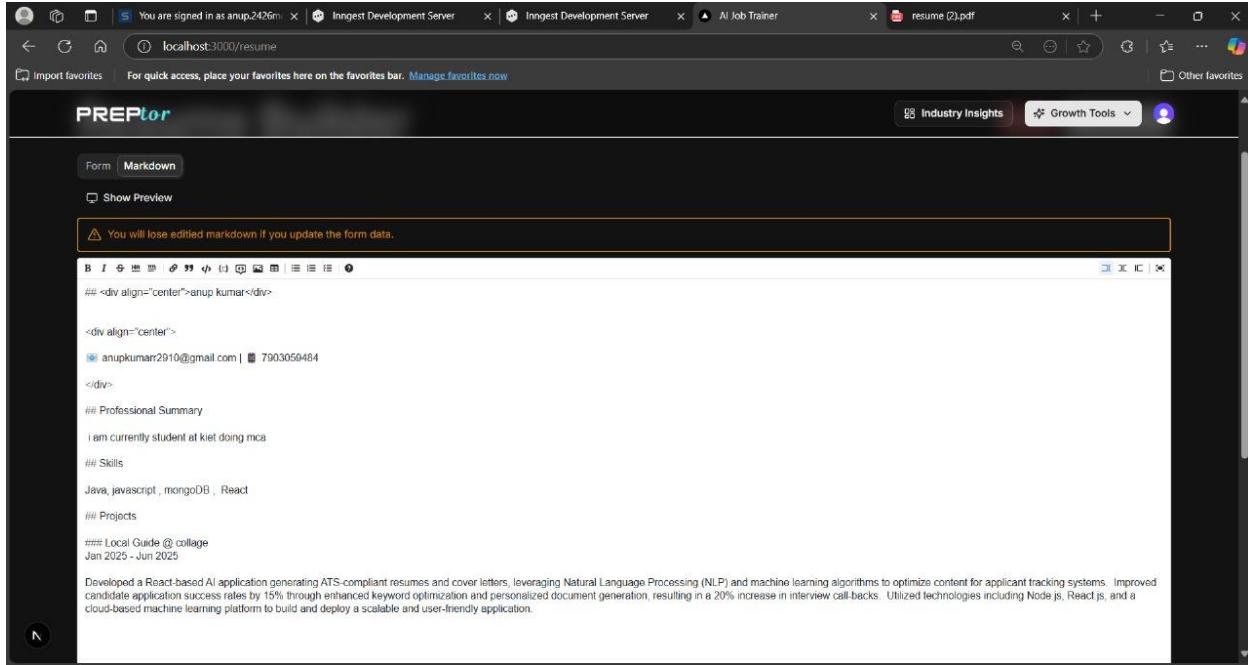


Fig 11. Markdown option in resume to edit the resume

Screenshot : Cover Letter

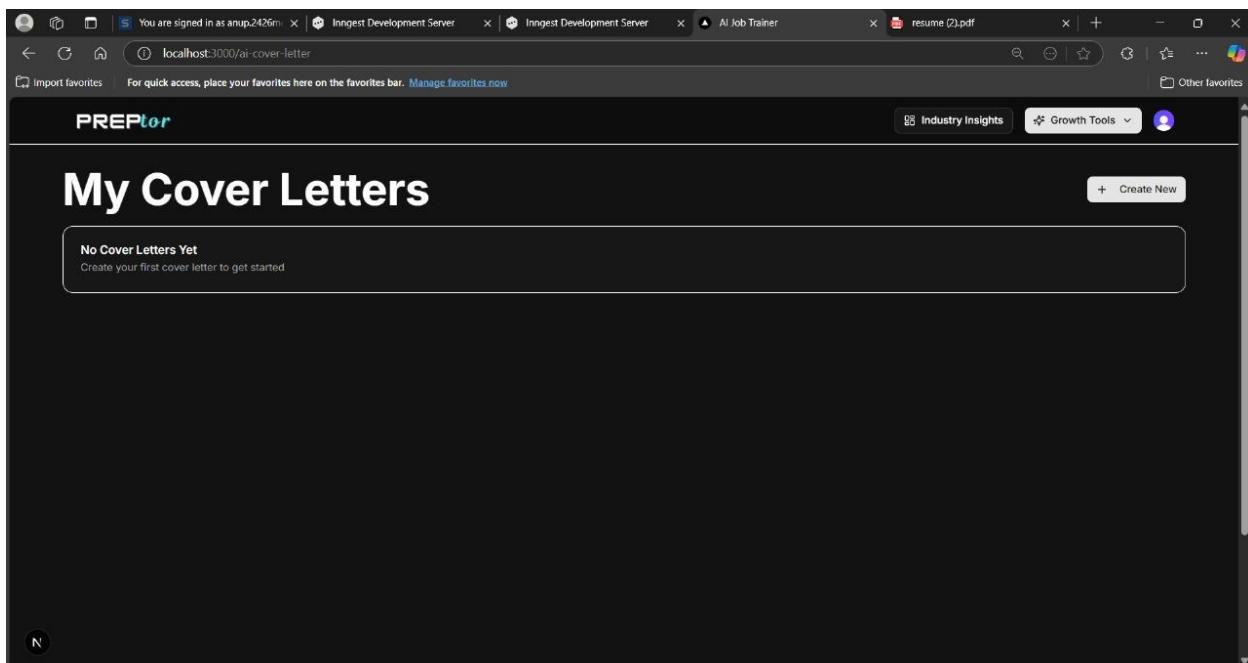


Fig 12. Cover letter feature

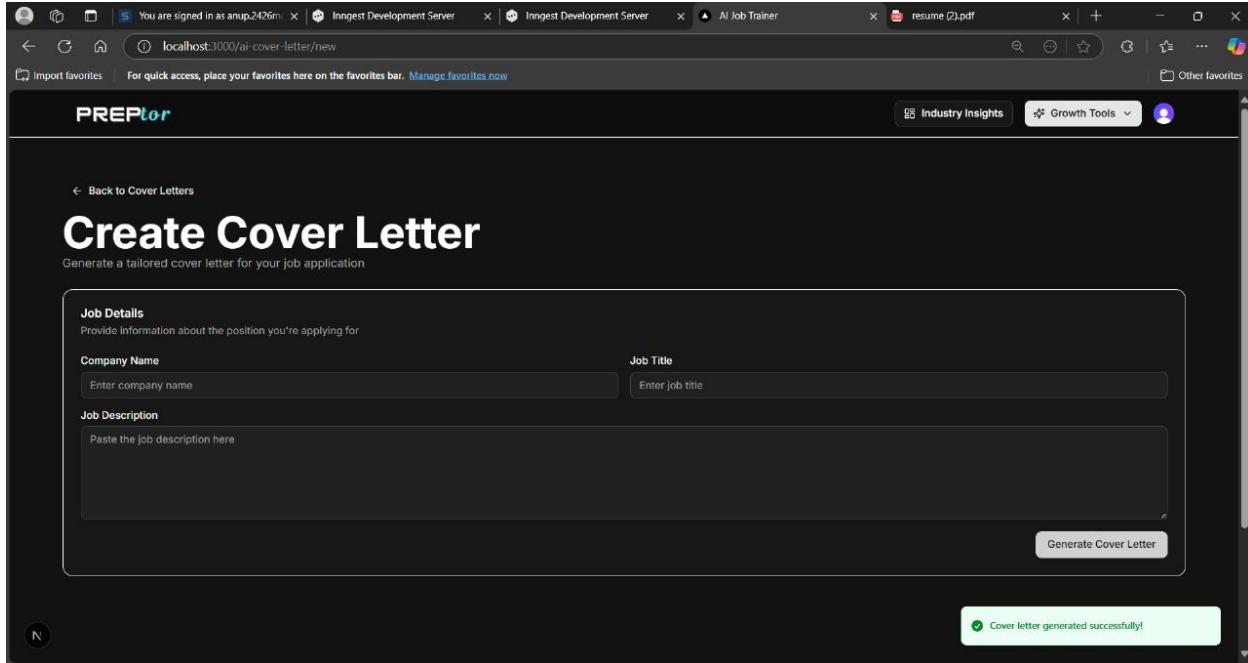


Fig. 13 Option to be filled to create cover letter.

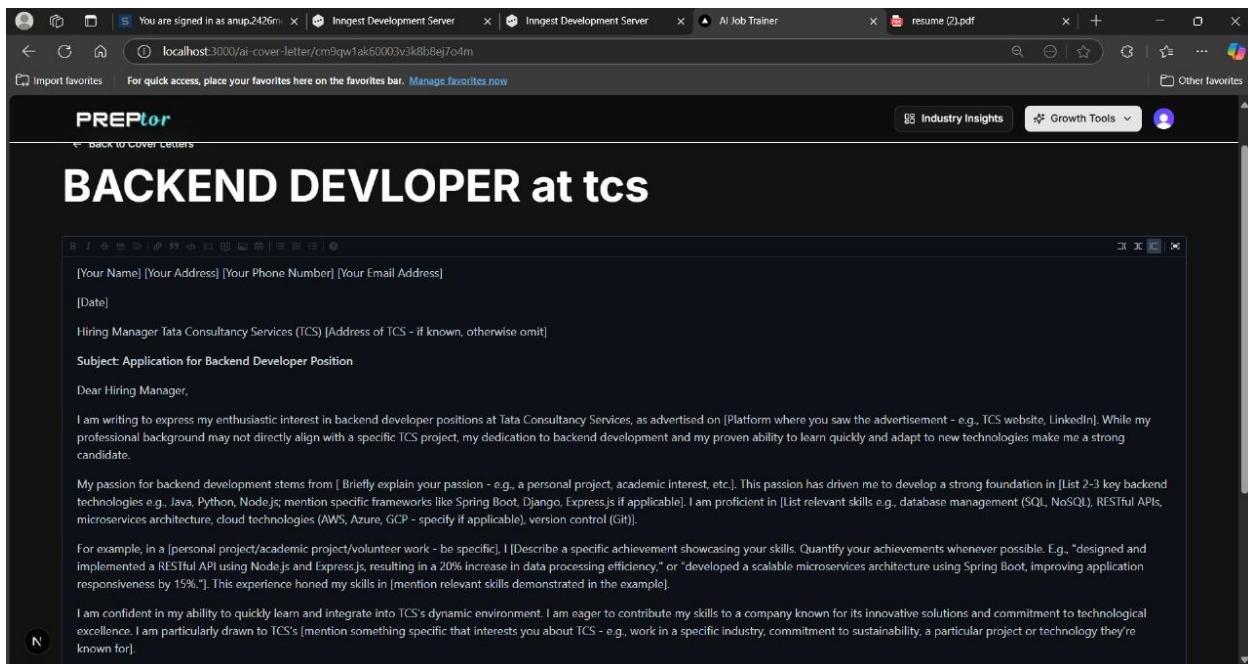


Fig 14. This is the newly created cover letter according to the details filled by the user

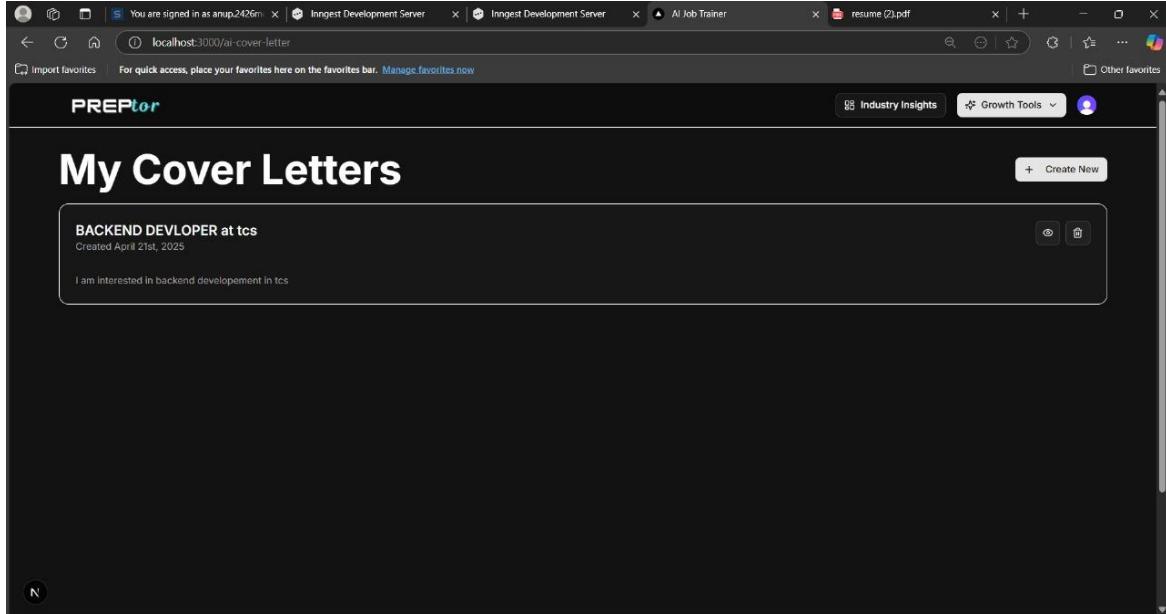


Fig 15. Newly created cover letter got saved for that user.

Screenshot : Mock Interview Interface

AI-generated questions displayed

Input area for user answers

Feedback shown after submission

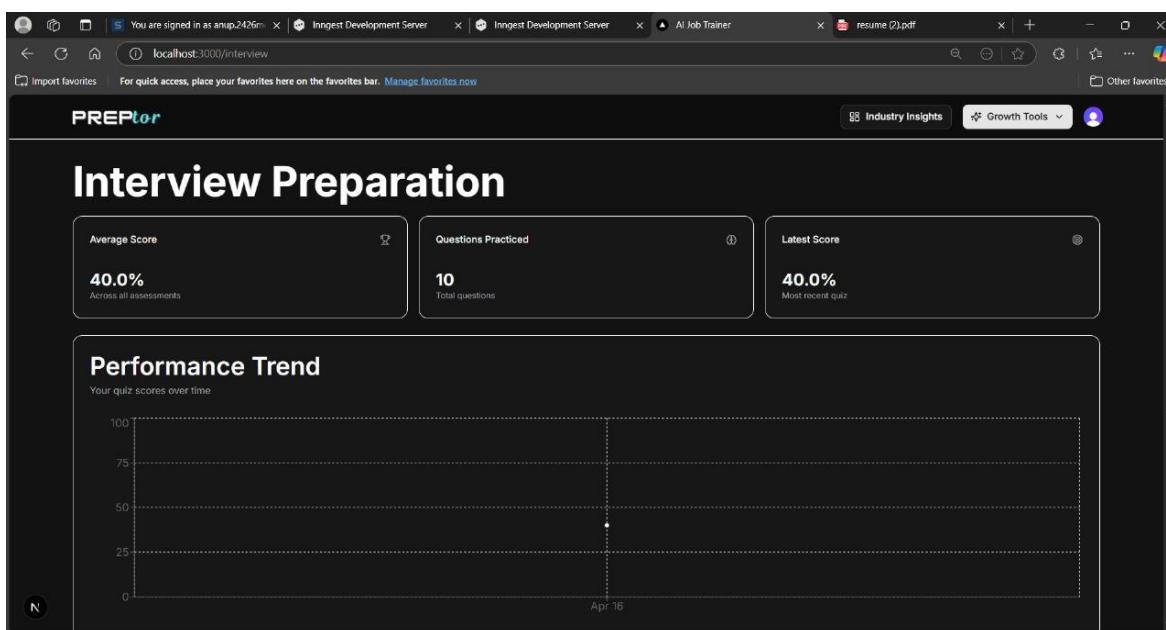


Fig 16: Interview preparation feature of our website.

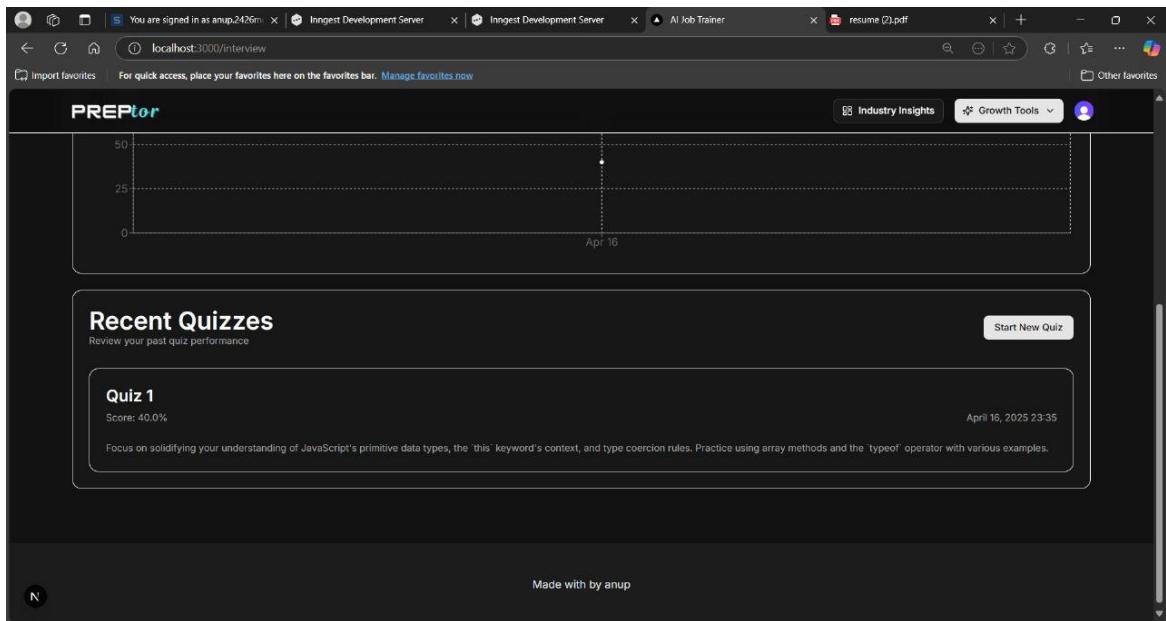


Fig 17. To start a new quiz . Option to start new quiz is provided.

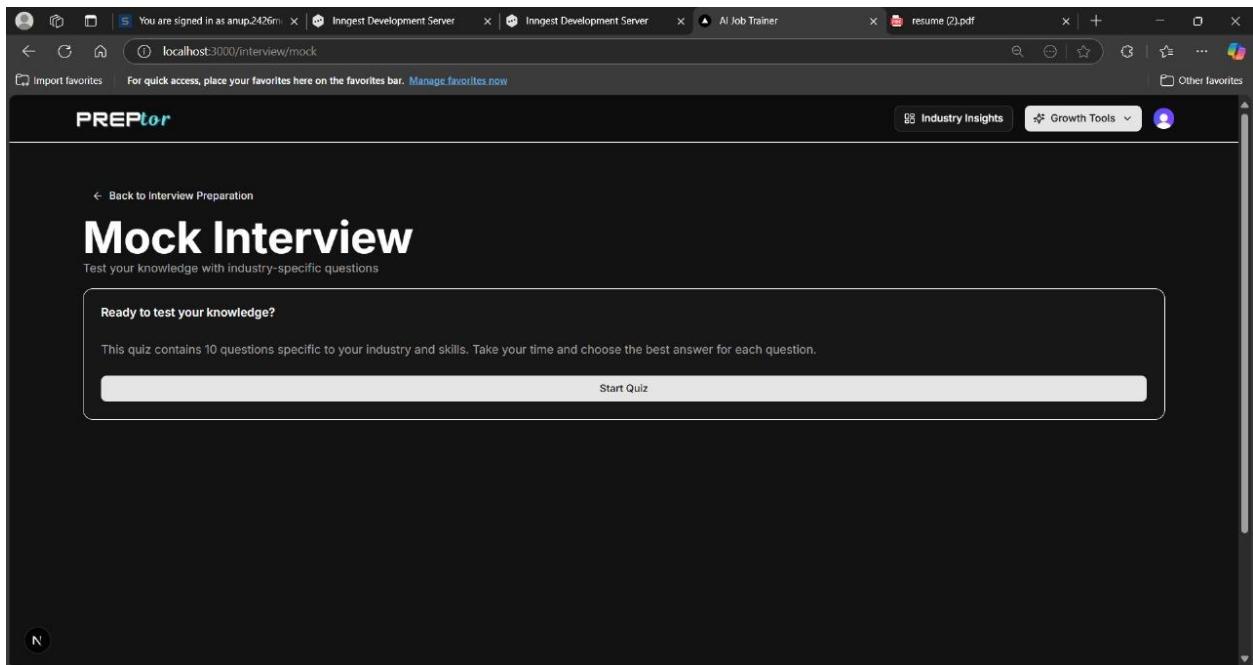


Fig. 18 Mock interview start quiz section.

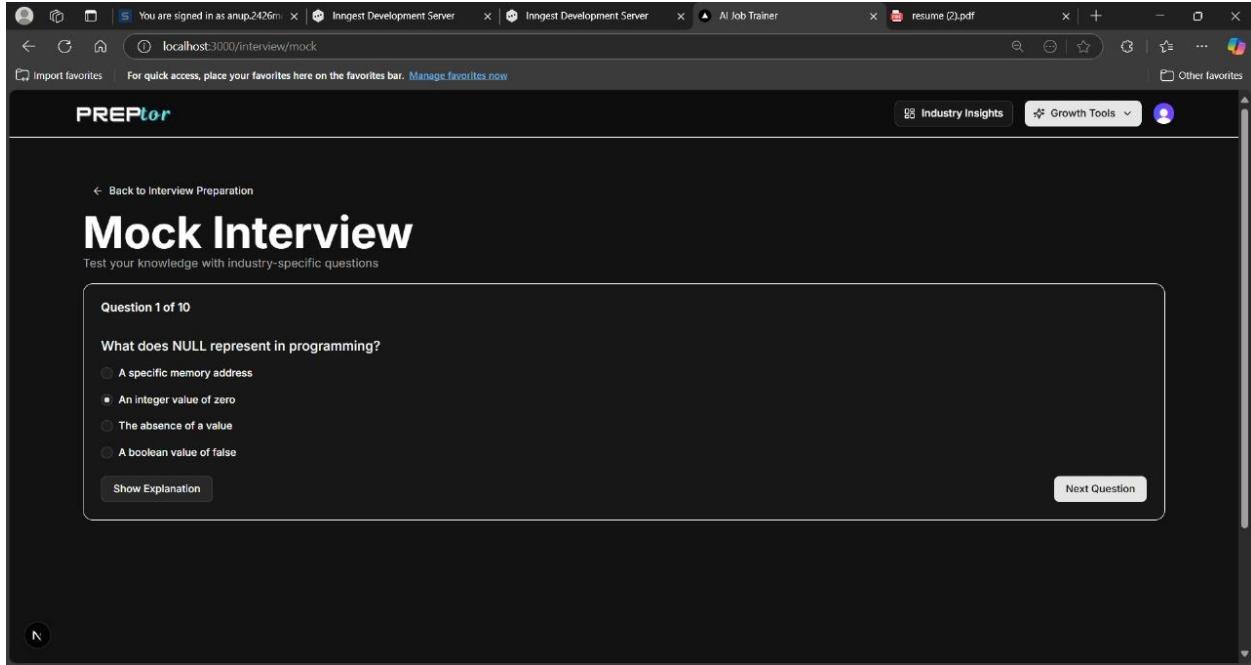


Fig 19. After starting new quiz, a set of 10 question will come.

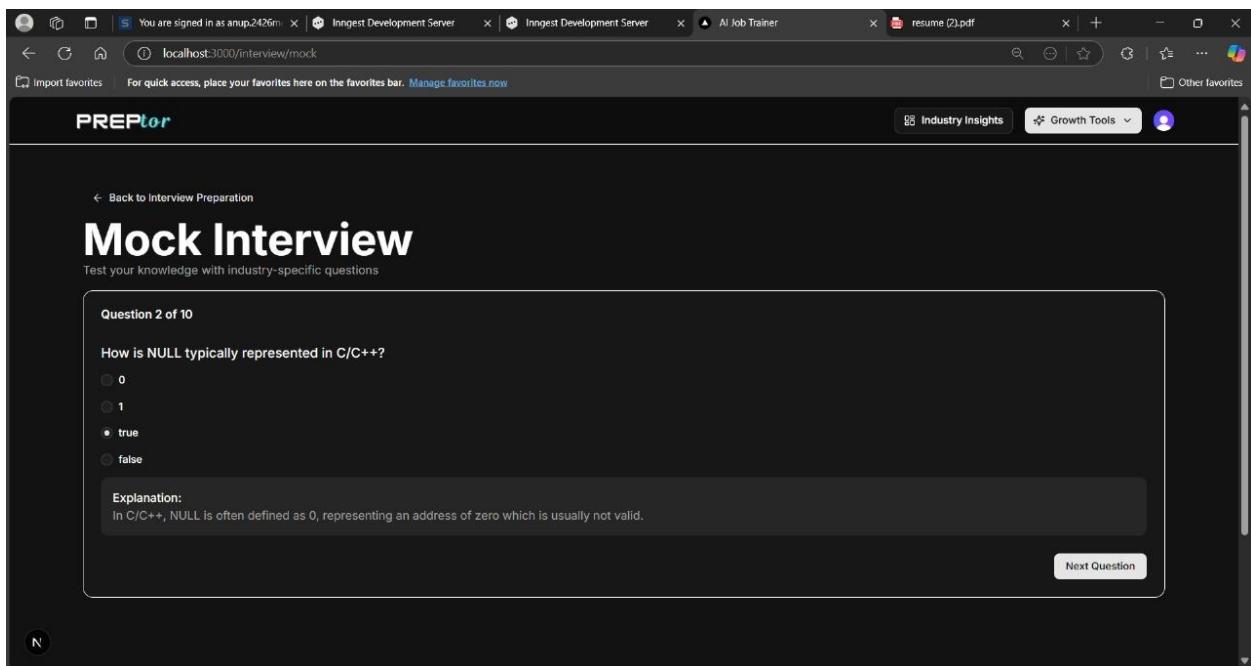


Fig 20 Second question of quiz.

You are signed in as anup.2426m | localhost:3000/interview/mock | Ingest Development Server | Ingest Development Server | AI Job Trainer | resume (2).pdf

← Back to Interview Preparation

Mock Interview

Test your knowledge with industry-specific questions

Quiz Results

30.0%

Improvement Tip:
Focus on solidifying your understanding of NULL's representation and handling in various programming languages and databases, paying special attention to pointer safety and the implications of NULL in data storage and querying. Practice writing code that correctly handles NULL values to build confidence.

Question Review

What does NULL represent in programming?
Your answer: An Integer value of zero
Correct answer: The absence of a value
Explanation: NULL typically signifies that a variable or pointer does not currently hold a valid object or value.

How is NULL typically represented in C/C++?

Quiz completed!

Fig. 21 After quiz submission . result will be provided according to the given test.

You are signed in as anup.2426m | localhost:3000/interview/mock | Ingest Development Server | Ingest Development Server | AI Job Trainer | resume (2).pdf

← Back to Interview Preparation

Mock Interview

Test your knowledge with industry-specific questions

Quiz Results

30.0%

How does comparing NULL values work in SQL?
Your answer: NULL = NULL is false
Correct answer: NULL comparisons always return NULL
Explanation: Comparing NULL values directly using = or <> will always return NULL, requiring specific functions like IS NULL or IS NOT NULL.

How can you handle NULL values gracefully in JavaScript?
Your answer: Use the Nullish coalescing operator (??)
Explanation: The ?? operator provides a concise way to handle null or undefined values providing a default value.

In databases, what is the potential impact of a large number of NULL values?
Your answer: Improved query performance
Correct answer: Increased storage space
Explanation: Null values still require storage space, potentially impacting the database's size and performance if excessively used.

What is a common strategy to mitigate problems associated with NULL values in database design?
Your answer: Use default values appropriately
Explanation: Assigning default values whenever possible reduces the reliance on NULL and simplifies database interactions.

Start New Quiz

Fig .22 All quiz question result will be provided along with explanation.

You are signed in as anup.2426n | localhost:3000/interview/mock | Ingest Development Server | Ingest Development Server | AI Job Trainer | resume (2).pdf

Import favorites For quick access, place your favorites here on the favorites bar. Manage favorites now

PREPtor

Industry Insights Growth Tools

How does comparing NULL values work in SQL?
Your answer: NULL = NULL is false
Correct answer: NULL comparisons always return NULL
Explanation:
Comparing NULL values directly using = or <> will always return NULL, requiring specific functions like IS NULL or IS NOT NULL.

How can you handle NULL values gracefully in JavaScript?
Your answer: Use the Nullish coalescing operator (??)
Correct answer: The ?? operator provides a concise way to handle null or undefined values providing a default value.
Explanation:

In databases, what is the potential impact of a large number of NULL values?
Your answer: Improved query performance
Correct answer: Increased storage space
Explanation:
Null values still require storage space, potentially impacting the database's size and performance if excessively used.

What is a common strategy to mitigate problems associated with NULL values in database design?
Your answer: Use default values appropriately
Correct answer: Assigning default values whenever possible reduces the reliance on NULL and simplifies database interactions.
Explanation:

Start New Quiz

Fig 23. Quiz question result.

You are signed in as anup.2426n | localhost:3000/interview | Ingest Development Server | Ingest Development Server | AI Job Trainer | resume (2).pdf

Import favorites For quick access, place your favorites here on the favorites bar. Manage favorites now

PREPtor

Industry Insights Growth Tools

Recent Quizzes
Review your past quiz performance

Quiz 1
Score: 40.0%
Focus on solidifying your understanding of JavaScript's primitive data types, the 'this' keyword's context, and type coercion rules. Practice using array methods and the 'typeof' operator with various examples.

Quiz 2
Score: 30.0%
Focus on solidifying your understanding of NULL's representation and handling in various programming languages and databases, paying special attention to pointer safety and the implications of NULL in data storage and querying. Practice writing code that correctly handles NULL values to build confidence.

Made with by anup

Fig.24 All attempted quiz of current user.

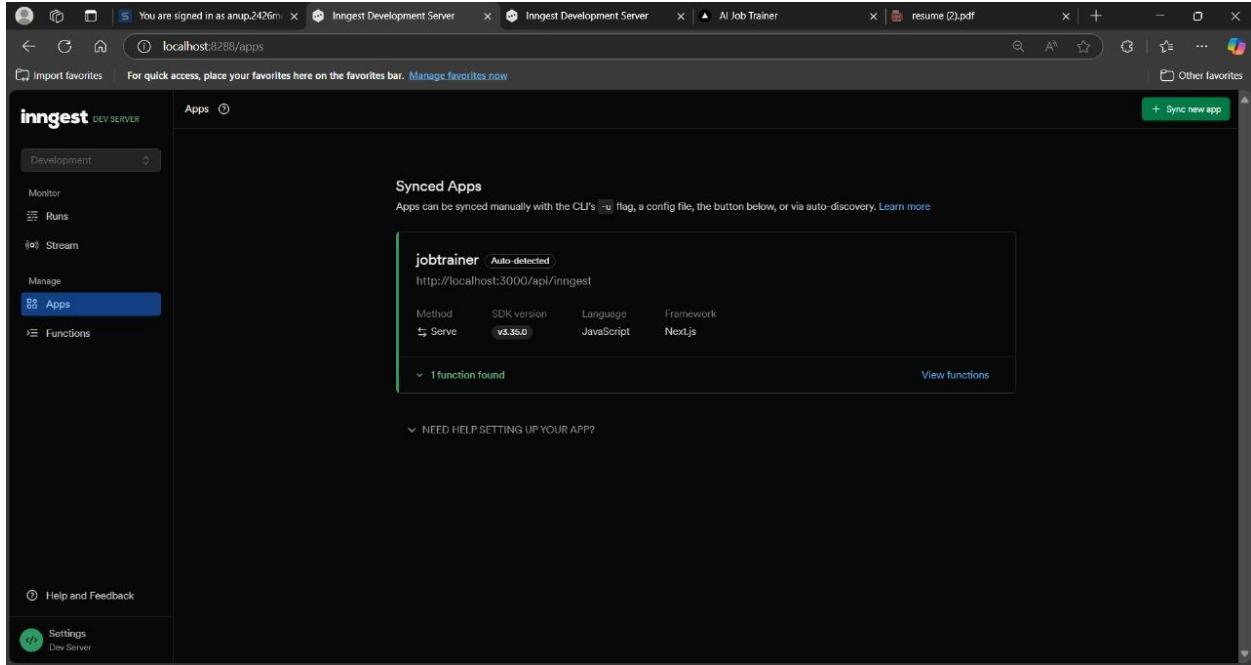


Fig 25 . Inngest to start the server.

	id	text	userId	quizScore	questions	category
1	cm9k7zgc200053uc67lf...	066952e1-4de8-42b4-b18...	30	[{"answer": "true", "que...	Technical	
2	cm9k8skvy000530g47fcct...	ef86e2d7-c4a4-475e-9b4...	40	[{"answer": "11", "quest...	Technical	
3	cm9k87kv000dv30gd96a...	e9ab7905-75a9-46b0-974...	40	[{"answer": "To represe...	Technical	
4	cm9n4hhu00001v3fcgynn9...	066952e1-4de8-42b4-b18...	60	[{"answer": "true", "que...	Technical	
5	cm9ppewfj0001v3qgnczow...	066952e1-4de8-42b4-b18...	50	[{"answer": "true", "que...	Technical	
6	cm9suae40001v39wnkifv...	066952e1-4de8-42b4-b18...	40	[{"answer": "The absenc...	Technical	
7	cm9qw8iis0005v3k8gijf...	ef86e2d7-c4a4-475e-9b4...	30	[{"answer": "The absenc...	Technical	

Fig. 26 Neon database which shows all the structure of our preptor.
(Assessment details of all user)

The screenshot shows the Neon Console interface with the URL console.neon.tech/app/projects/jolly-bar-86935261/branches/br-bold-waterfall-a87ajm3u/tables?database=jobtrainer. The left sidebar shows the project structure with 'Tables' selected. The main area displays the 'CoverLetter' table with the following data:

	id text	userId text	content text	JobDescription text	companyName text
	cm9prhry60003v3ik61j17...	e9eb7905-75a9-46b0-874...	[Your Name] [Your Addr...	I am interest in sales...	puma
	cm9sqw1ek60003v3k8b8ej7...	ef86e2d7-c4a4-475e-9b4...	[Your Name] [Your Addr...	I am interested in bac...	tcs

Fig 27 Cover letter Data of all created cover letters generated by the user.

The screenshot shows the Neon Console interface with the URL console.neon.tech/app/projects/jolly-bar-86935261/branches/br-bold-waterfall-a87ajm3u/tables?database=jobtrainer. The left sidebar shows the project structure with 'Tables' selected. The main area displays the 'Resume' table with the following data:

	id text	userId text	content text	atsScore double precision	feedback text
	cm9prmb5u0001v3ik3g499...	e9eb7905-75a9-46b0-874...	## <div align="center">...	NULL	NULL
	cm9sqserll0001v3rswnib...	006952e1-4de8-42b4-b18...	## <div align="center">...	NULL	NULL
	cm9qvw6670001v3k87xuh2...	ef86e2d7-c4a4-475e-9b4...	## <div align="center">...	NULL	NULL

Fig 28 Resume Data of all created Resume generated by the user.

The screenshot shows the Neon Console interface with the URL console.neon.tech/app/projects/jolly-bar-86935261/branches/br-bold-waterfall-a87ajm3u/tables?database=jobtrainer. The left sidebar includes sections for PROJECT (Dashboard, Branches, SQL Editor, Restore, Monitoring, Integrations, Auth, Settings), BRANCH (Overview, Tables), and Feedback. The main area is titled 'Tables' and displays the 'User' table. The table has columns: id, clerkUserId, email, name, and imageUrl. There are three rows of data:

	id	clerkUserId	email	name	imageUrl
1	066952e1-4de8-42b4-b18...	user_2vWxsxeuMqguzeSXF...	anup.2426mca1305@kiet...	ANUP KUMAR MAHTO	https://img.clerk.com/...
2	e9eb7905-75a9-46b0-874...	user_2vp1o8ndRutYfralC...	anupkumarr10290@gmail.c...	Anup Kumar	https://img.clerk.com/...
3	ef86e2d7-c4a4-475e-9b4...	user_2vp00kmpBN2aKgNPn...	anupkumarr2910@gmail.c...	anup kumar	https://img.clerk.com/...

Fig 29 List of all the user who had logged-in to our website.

Chapter 9

Conclusion

Overview

In a world where Artificial Intelligence is revolutionizing almost every domain, from healthcare to finance, its potential in education and employment preparation is particularly significant. The AI Job Trainer project has been an effort to bring this technological advancement directly to students, fresh graduates, and job seekers. It helps them confidently prepare for interviews, discover relevant career paths, and improve their professional outlook through intelligent feedback systems.

This project has demonstrated how AI can be leveraged not only for automation but also for intellectual augmentation — offering aspirants a digital mentor that is available 24/7, adaptable, and scalable.

9.1 Recap of Problem and Solution

Problem Identified

1. Lack of personal mentorship and interview guidance
2. Inaccessibility of career coaching for students in tier 2 and tier 3 cities
3. Difficulty in understanding and matching one's resume with job market expectations
4. Absence of real-time, intelligent interview preparation platforms
5. Proposed AI-Based Solution
6. Use of AI models to generate custom interview questions
7. Resume parsing with Natural Language Processing (NLP)
8. Role recommendations using Machine Learning classification
9. Feedback analysis for continuous learning and improvement
10. Deployment of a web application accessible on all devices

9.4 System Reliability and User Feedback Considerations

The system has been built with robust error-handling mechanisms, ensuring:

- Safe failure for invalid inputs or file formats
- Auto-detection of incomplete resumes and guidance for improvement
- Real-time feedback loops to improve model suggestions

For future versions, user feedback will be crucial. Integration of user satisfaction surveys and performance tracking can help the AI model learn from actual candidate outcomes and refine its recommendations.

9.5 Societal & Educational Impact

The broader societal influence of the AI Job Trainer can be measured through:

- Improved interview readiness across diverse demographics
- Reduction in unemployment rates through better-prepared applicants
- Empowering educational institutions with AI tools for student grooming
- Rural digital upliftment by offering access to intelligent career guidance

It supports the United Nations Sustainable Development Goals (SDGs), particularly:

1. SDG 4 – Quality Education
2. SDG 8 – Decent Work and Economic Growth
3. SDG 10 – Reduced Inequalities

9.6 Technical Enhancement Possibilities

The system can be improved with the following:

Feature	Future Enhancement
Resume Analyzer	Use of advanced models like BERT and GPT-4 for better semantic matching
Interview	Integration of speech recognition for voice-based Q&A

Learning Path Recommender	Connecting with platforms like Coursera, Udemy, or NPTEL
Adaptive Training	Dynamic difficulty adjustment based on user performance

9.7 Technical Reflection & Knowledge Acquired

This project has been a journey of full-stack integration and cross-disciplinary learning. Some of the deeper insights gained include:

1. Understanding the tokenization, embedding, and semantic similarity involved in NLP
2. Designing REST APIs with Flask to interface AI predictions with user inputs
3. Deploying secure, scalable applications using cloud services
4. Debugging data input pipelines and managing user session histories

The development team also learned to use tools such as:

1. Postman (API testing)
2. Git/GitHub (Version control)
3. Canva/Dia (for designing system diagrams)
4. SQLite/MySQL (for lightweight data storage)
5. Gemini AI / ML APIs (for NLP and ML model services)

9.8 Entrepreneurial & Business Potential

This application could evolve into a commercial SaaS product for educational institutes and HR consultancy firms. With proper marketing, it could be monetized through:

1. Premium subscriptions
2. Personalized mentoring add-ons
3. Campus licensing models
4. White-labelled versions for institutions

9.9 Closing Perspective

The AI Job Trainer isn't merely a final-year project—it's a prototype of the future. A future where:

- Career coaching is no longer a privilege
- Interview prep is personalized and intelligent

- AI doesn't replace but amplifies human potential

The successful implementation of this system underscores a vital truth: when technology meets empathy, great things happen.

Final Summary:

“The AI Job Trainer equips users not just with interview practice but also allows them to build modern, visually appealing resumes using an intuitive resume builder module — a feature especially helpful for first-time job seekers.”

Chapter 10

References

10.1 Books and Academic Resources

“Artificial Intelligence: A Modern Approach” by Stuart Russell and Peter Norvig

Publisher: Pearson

Used For: Understanding AI architecture, agent-based systems.

“Web Technologies: HTML, JavaScript, PHP, Java, JSP, ASP.NET, XML and AJAX, Black Book”

Author: Kogent Learning Solutions Inc.

Used For: Front-end and back-end web development references.

10.3 Websites and Online Resources

<https://reactjs.org/>

Used For: React.js component design, hooks, and functional programming model.

<https://jsfiddle.net/>

Used For: Prototyping components and real-time testing.

<https://html2pdf.app/>

Used For: Resume-to-PDF conversion using HTML templates.

<https://jspdf.dev/>

Used For: PDF generation methods in the Resume Builder.

<https://gemini.google.com/>

Used For: Gemini AI usage and documentation reference.

<https://developer.mozilla.org/>

Used For: JavaScript functions, syntax, event handling.

<https://tailwindcss.com/>

Used For: UI styling and responsive layout development.

10.5 AI and API Services Documentation

Gemini API Documentation – <https://ai.google.dev>

Google OAuth / Identity Services – <https://developers.google.com/identity>

OpenAI Prompt Engineering Guide (for alternative examples) –
<https://platform.openai.com/docs/guides/prompt-engineering>