CHITKARA UNIVERSITY

PROJECT SYNOPSIS REPORT

ON

PROPREP

SUBMITTED

TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FOR

INTEGRATED PROJECT(CS203)

Submitted By:

Nishtha, Priya Gupta, Pragti Gupta

2210991992, 2210992096, 2210992056

Semester: 6th

Session: 2022-2026

Index

Sr. no	Topic	Page No
1	Problem Statement	3
2	Title of project	3
3	Objective & Key Learning’s	3
4	Options available to execute the project	4 - 5
5	Advantages/ Disadvantages	5 - 7
6	References	7

Problem Statement

Job seekers often face significant challenges in crafting optimized resumes and preparing effectively for interviews. These hurdles are primarily due to the fragmented nature of existing tools, a lack of personalized guidance, and limited access to real-time feedback. This project addresses these pain points by integrating two AI-powered solutions: an AI Resume Builder App that helps users create professional, tailored resumes with intelligent suggestions, and an AI Mock Interview App that provides seamless, real-time interview simulations and feedback. By leveraging AI, the platform streamlines the job application journey, enhances career readiness, and ultimately increases the user's chances of success in the hiring process.

Title of project:

ProPrep :

"Shaping the Future of Career Success with AI- Interview Support"

Objective & Key Learnings:

- **Develop an Integrated AI-Powered Career Toolkit:** Gained hands-on experience in building a seamless platform that combines AI-driven mock interviews and intelligent resume generation, streamlining the career preparation process for job seekers.
- **Leverage AI for Personalization and Real-Time Feedback:** Applied advanced AI technologies (e.g., Gemini AI) to deliver personalized resume suggestions and dynamic, real-time interview feedback, significantly improving the relevance and effectiveness of the user experience.

- **Design a User-Centric Interface:** Strengthened UI/UX design skills by crafting a cohesive and intuitive interface that supports smooth navigation, accessibility, and engagement across both the resume builder and mock interview tools.
- **Build and Integrate Full-Stack Technologies:** Acquired practical expertise in full-stack development using technologies like React, Vite, Tailwind CSS, Next.js, Drizzle ORM, and Clerk, enabling the creation of a robust, scalable, and secure platform.

Options available to execute the project:

Mock Interview Functionality (AI-Driven Feedback)

- **Gemini AI for Adaptive Feedback**: Utilized to provide personalized, context-aware feedback on user responses during mock interviews, helping users improve their communication and technical skills.

- **WebSocket/Socket.io for Real-Time Interaction**: Enables live interaction between users and the AI system, simulating real interview scenarios with dynamic question delivery and instant feedback.

Resume Builder Functionality (AI Powered Suggestions)

- **Gemini AI for Smart Resume Suggestions:** Integrated AI to generate context-relevant suggestions, keywords, and formatting tips to optimize resumes based on job roles and user input.
- **PDF Generation Libraries:** Tools like react-pdf or html2pdf used to export resumes in clean, professional formats.

User Authentication & Management

- **Clerk for Authentication**: Secure user authentication and management using Clerk to handle sign-ups, logins, and profiles.

Backend Development (API & Data Management)

- **Next.js API Routes**: For handling backend logic, AI model integration, and interactions with the database.

- **Drizzle ORM**: For handling database interactions and ensuring smooth data management.

Data Storage and Management

- **Cloudinary or S3 for File Storage**: For uploading, storing, and retrieving user uploaded files (interview recordings).

Advantages:

- **AI-Driven Personalized Experience:** The integration of AI in both mock interviews and resume generation delivers highly personalized guidance, enabling users to receive targeted feedback and suggestions aligned with their specific career goals and job profiles.

- **Scalable Architecture:** Built using Next.js for both frontend and backend development, and deployed via platforms like Vercel or Heroku, the application is designed to scale efficiently with increasing user demand.

- **Real-Time Feedback and Interaction:** The mock interview module utilizes Gemini AI along with WebSocket/Socket.io to simulate live interview environments, offering users instant, adaptive feedback that enhances learning and confidence.

- **Secure and Seamless User Authentication:** Clerk integration ensures robust user authentication and session management across the platform, safeguarding sensitive user information and supporting smooth access across both tools.

- **Cloud-Based File and Data Management:** Using services like Cloudinary and S3 allows for secure and scalable storage of resumes, interview recordings, and profile data—making them easily accessible and well-organized.

Disadvantages:

- **Reliance on AI Accuracy:** AI-generated feedback—whether for resume optimization or interview evaluation—may occasionally produce inaccurate or generic suggestions, which can impact user trust or effectiveness.

- **Data Privacy and Security Concerns:** Despite secure authentication via Clerk, storing resumes, interview data, and personal details in the cloud poses potential privacy concerns and requires continuous monitoring to avoid breaches.

- **Dependency on Cloud Services:** The platform's reliance on third-party services such as Cloudinary, Clerk, and MongoDB Atlas introduces risks related to service downtime, API limits, or pricing model changes.

- **High Computational Demands**: Real-time AI processing (especially during interviews) and resume generation can consume substantial server resources, leading to higher operational costs and the need for efficient scaling solutions.

- **Real-Time System Optimization Challenges:** Implementing real-time interactions using WebSocket or Socket.io can introduce performance bottlenecks, particularly when managing multiple concurrent sessions, requiring careful backend optimization and load handling strategies.

References:

1. https://gemini.google.com
2. https://nextjs.org/
3. https://socket.io/
4. https://strapi.io/