



CorePrep: Smart Interview Preparation

Neetu Narwal¹, Kavya Lamba², Sahiba Kangra³

¹ Associate Professor, Department of Computer Applications, Maharaja Surajmal Institute, New Delhi, India

^{2,3} Student, Department of Computer Applications, Maharaja Surajmal Institute, New Delhi, India

How to Cite this Article:

Kangra, K. L. S. (2026). CorePrep: Smart Interview Preparation. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(7), 1-9.
<https://doi.org/10.55041/ijcope.v2i7.122>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i7.122>

Abstract

Technical interviews play a significant role in the hiring process for computer science students. However, many students find it hard to prepare effectively because they lack structured guidance, realistic practice, and ongoing feedback. Most existing preparation methods depend on static question banks or self-study, which do not reflect real interview settings or help students monitor their progress over time. This paper introduces CorePrep: Smart Interview Preparation, a web-based platform designed to help students develop their technical interview skills through structured practice, mock interview simulations, and performance analysis.

Introduction

CorePrep lets users practice questions on specific topics from core computer science subjects, such as Operating Systems, Database Management Systems, Computer Networks, and Data Structures. Each attempt is evaluated, and users can see their accuracy, points, and consistency through a personalized dashboard. To mimic real interview experiences, the platform includes a mock interview module that uses speech-to-text technology to record spoken answers and generate transcripts. These responses are analyzed to provide feedback

based on technical relevance, clarity of explanation, and the use of filler words.

To keep users engaged and motivated, CorePrep includes gamification features like leaderboards, streak tracking, and a buddy system. The buddy system lets users connect with peers, see recent activities, and encourage regular practice through shared progress. Users can also customize their profiles by selecting target roles, preferred difficulty levels, and interview duration, allowing the platform to tailor preparation to their individual career goals. Additionally, CorePrep generates performance reports that highlight strengths, pinpoint areas for improvement, and offer practical suggestions.

Built with modern web technologies like Node.js, Express.js, and PostgreSQL, CorePrep aims to close the gap between theoretical knowledge and actual interview performance. The platform provides an interactive and data-driven environment that helps students practice regularly, gain confidence, and enhance their readiness for technical interviews.

Problem Statement

Technical interviews are an essential part of the recruitment process for computer science students. They assess not just knowledge of the subject but also problem-solving skills, communication abilities, and confidence. However, many students struggle to prepare effectively. This is due to a lack of structured learning platforms



that combine practice, evaluation, and realistic interview simulations. Most resources focus on individual areas like theoretical questions or coding problems. They do not provide a complete and guided preparation experience.

Current interview preparation methods often use static question banks, self-study materials, or informal peer discussions. These methods do not offer personalized feedback, measurable tracking of performance, or realistic mock interview settings. Because of this, students find it hard to accurately assess their strengths and weaknesses or gauge their readiness for real interviews. Limited peer interaction and motivation can also lead to inconsistent practice and less engagement over time.

Another big issue is the lack of tools that evaluate verbal communication and interview behavior. Many students have trouble expressing their technical knowledge clearly during interviews. Yet, most platforms do not provide speech-based mock interviews or feedback on clarity, filler words, and relevance to the role. This creates a gap between theoretical preparation and actual interview performance.

There is a need for a web-based system that offers structured practice with technical questions, realistic mock interview simulations, performance tracking, and peer motivation features all in one place. Such a system should help students practice consistently, track their progress, receive useful feedback, and prepare confidently for technical interviews that match their career goals. CorePrep aims to address these challenges by offering an integrated, interactive, and data-driven solution for interview preparation.

System Architecture

The system architecture of CorePrep: Smart Interview Preparation uses a modular and scalable web-based structure that separates the frontend, backend, and database layers. This design allows for smooth interaction between users and the system while keeping data secure, ensuring good performance, and making future improvements easier.

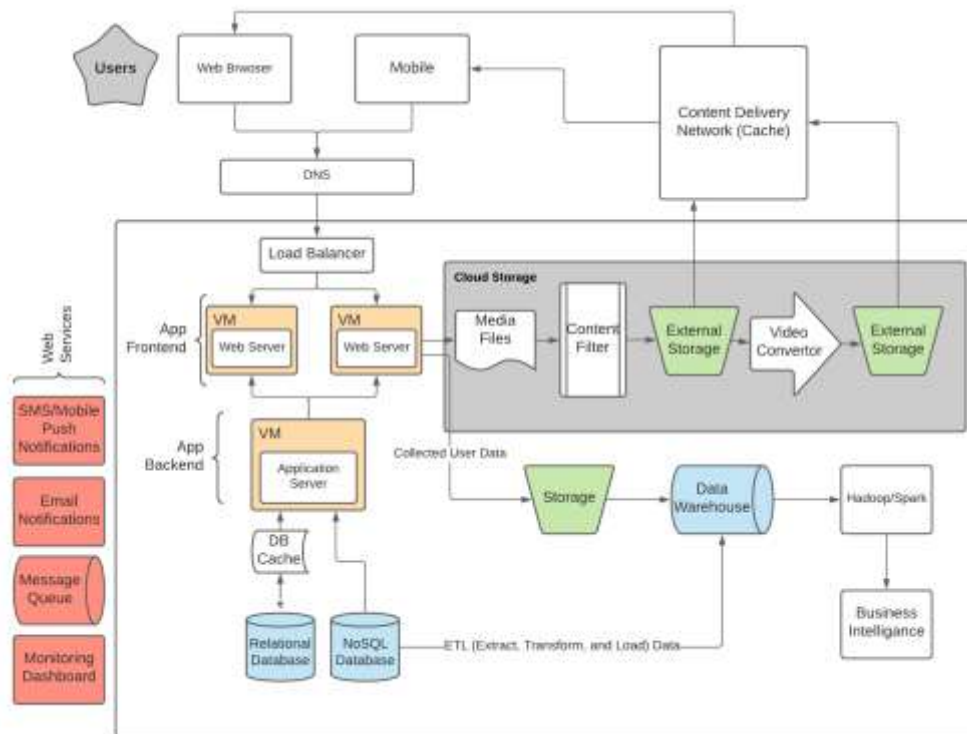


Figure 1: System Architecture of CorePrep

At the frontend layer, users interact with the application through a web interface built with EJS, HTML, CSS, and Tailwind styling. This layer renders dynamic pages like login/signup, dashboards, question practice screens, mock interview interfaces, leaderboards, and user profiles. JavaScript DOM scripting handles user interactions, form submissions, and real-time UI updates.

The backend layer is built with Node.js and Express.js, which manage all application logic and communication between the frontend and the database. This layer handles user authentication through session-based login. It processes API requests, evaluates question attempts, calculates scores, manages buddy system interactions, and generates performance reports. REST APIs ensure secure and structured data exchange between the frontend and backend.

The mock interview module integrates into the backend logic, processing speech input from users and converting it into text using speech-to-text functionality. The resulting transcript is analyzed for technical relevance, clarity, and the use of filler words. The processed results are stored and reflected in the user's performance reports and dashboard analytics.

The database layer uses PostgreSQL to store all permanent data, including user credentials, session details, question banks, attempt history, scores, buddy relationships, activity feeds, and generated reports. Unique identifiers (UUIDs) help maintain data integrity and uniquely identify users and records within the system.

Overall, the architecture of CorePrep follows a client-server model that promotes clear separation of responsibilities, efficient data handling, and scalability. This organized design allows the platform to support multiple users at once, ensuring reliable performance tracking and a smooth interview preparation experience.



Methodology

The methodology of CorePrep: Smart Interview Preparation focuses on creating and implementing a structured, user-friendly web platform. It supports technical interview preparation through practice, simulation, evaluation, and tracking progress. The system uses a modular approach, where each feature addresses a specific problem students face during interview preparation. The overall methodology includes analyzing needs, designing the system, implementing core modules, managing data, and evaluating performance.

Requirement Analysis

The first step involved understanding the common challenges students encounter while preparing for technical interviews. These challenges include a lack of organized practice, no realistic mock interview environments, difficulty tracking progress, and reduced motivation due to isolated preparation. Based on these observations, platform requirements were identified and categorized into functional and non-functional needs. Functional requirements included user login, topic-based question practice, mock interview simulation, automated evaluation, peer interaction through a buddy system, leaderboard rankings, and report generation. Non-functional requirements focused on usability, scalability, data security, and responsiveness.

System Design Approach

After identifying the requirements, a client-server architecture was adopted. This clearly separates presentation, logic, and data storage. This approach improves maintainability and allows individual modules to be updated without impacting the entire system. The design emphasizes modularity, where each feature, such as authentication, practice modules, mock interviews, and analytics, operates independently but communicates through well-defined interfaces. User workflows were carefully designed to ensure a smooth experience. After logging in, users are directed to a dashboard that gives an overview of their progress. From there, they can choose to practice questions, attempt mock interviews, view reports, or connect with buddies. This flow simplifies navigation and encourages regular use.

User Authentication and Session Management

The platform uses a secure login and signup system to provide each user with a personalized experience. During registration, user information is securely stored in the database. Session-based authentication maintains the user's status across different pages. This allows users to navigate the platform without having to log in repeatedly while keeping sensitive data safe. Sessions are validated with every request, and unauthorized access to protected areas like dashboards, reports, and mock interviews is restricted.

Topic-Wise Question Practice Module

The question practice module supports organized learning across multiple computer science subjects. Questions are categorized by subject and difficulty level. When a user selects a topic, relevant questions are retrieved from the database and shown through the frontend interface. Each attempt is evaluated for correctness, and scores are assigned accordingly. The system records attempt history, accuracy, and time spent, which are later used for analytics and report generation. This method allows users to pinpoint weak areas and focus on specific improvements instead of random practice.



Mock Interview Simulation

A core component of CorePrep is the mock interview module, designed to replicate real interview conditions. Users can set the interview duration and select target roles before beginning a session. During the mock interview, users answer technical questions verbally.

Speech input is captured and converted to text using speech-to-text processing. The resulting transcript is analyzed for technical keywords, relevance to the chosen role, clarity of explanation, and filler word usage. The methodology promotes learning through reflection, enabling users to review their transcripts and feedback to enhance their responses in actual interview settings.

Performance Evaluation and Analytics

Performance evaluation uses a rule-based scoring method that combines multiple metrics such as accuracy, consistency, interview participation, and role relevance. These metrics are visually displayed on the user dashboard to offer clear insights into progress over time.

Analytics update in real-time after each activity, ensuring immediate feedback. This data-driven method helps users measure improvement objectively rather than relying on self-assessment alone.

Buddy System and Peer Motivation

To tackle the motivation issues in solo preparation, CorePrep includes a buddy system that connects users with peers. Users can search for others using email addresses, send requests, and accept or decline them. Once connected, buddies can see each other's recent activities, such as solved questions or completed mock interviews.

Leaderboard and Gamification

Gamification is implemented through points, streaks, and leaderboards. Users earn points for completing activities, maintaining streaks, and performing well in mock interviews. Leaderboards rank users based on accumulated points, fostering healthy competition.

This feature aims to boost engagement while keeping the focus on learning rather than pure competition.

Report Generation

The report generation module compiles data from practice sessions, mock interviews, and analytics to create structured feedback reports. These reports highlight strengths, identify weak areas, and suggest ways to improve. The methodology ensures that reports are easy to understand and actionable, helping users plan their next steps effectively.

Database Management

All user data, including credentials, activity history, buddy connections, and performance metrics, is stored in a relational database. Proper indexing and unique identifiers maintain data integrity and allow for efficient retrieval. This organized storage approach supports scalability and long-term use.



Methodology of CorePrep: Smart Interview Preparation



Results

The use of CorePrep: Smart Interview Preparation effectively illustrates the value of a methodical, web-based approach to technical interview preparation. The platform offers users a comprehensive preparation environment by combining several features like topic-specific practice, simulated interviews, performance analytics, and peer interaction into a single system.

The system demonstrated dependable performance in managing user authentication, session management, and data storage during testing and use. Registration, secure login, and uninterrupted access to customized dashboards were all possible for users. In order to help users identify their strengths and areas for improvement, the topic-wise question practice module efficiently tracked user attempts, accuracy, and progress across various computer science subjects.

The effectiveness of the mock interview feature has been one of the most impactful elements of the system. The speech-based response system and transcript generation enabled users to effectively engage in a simulated interview setting. The automated system provided significant feedback regarding the relevance of the response, clarity of speech, and filler words, which are highly pertinent in a real-life interview setting.

The implementation of gamification elements like points, streaks, and leaderboards has significantly enhanced user engagement and motivated users to maintain a consistent level of practice. The buddy system has also been a significant factor in maintaining user engagement and accountability. The report generation feature has effectively enabled users to summarize their performance and provided actionable feedback regarding progress, making it easier for users to evaluate their progress.

Overall, CorePrep has effectively addressed the problem of bridging the gap between theoretical knowledge and interview skills. The system has effectively improved user engagement, provided significant insights regarding performance, and simulated a real-life interview setting.



Future Scope

Although the CorePrep platform is already equipped with the best tools to prepare for interviews, several improvements can be made to enhance its performance and growth in the future. One such improvement is the incorporation of advanced natural language processing techniques to ensure the best possible feedback during mock interviews.

Another improvement can be the incorporation of coding interview features into the platform. This can be done by allowing users to practice coding during mock interviews. It will help them to assess their coding skills as well. Moreover, the question bank can be improved by including several company-specific interview styles.

In the future, several improvements can be made to the CorePrep platform. Video-based mock interviews can be included to assess the personality and presentation skills of the users. Moreover, mobile application support can be included to ensure greater accessibility. Furthermore, the platform can be improved to include mentor support to ensure the best possible feedback.

In conclusion, the CorePrep platform is full of growth and improvement potential. With several improvements and feature enhancements, the platform can be transformed into one of the best interview preparation tools for students.

Conclusion

This research paper has proposed a new concept called CorePrep: Smart Interview Preparation, which is a web-based system designed to help computer science students effectively prepare for their technical interviews. The system was developed with the aim of helping students overcome common issues they face while preparing for their interviews, such as a lack of proper practice, inaccessibility of proper interview simulations, and difficulty in tracking their progress over time. By incorporating multiple preparation tools under a single platform, CorePrep offers a more organized and engaging platform for students to prepare for their interviews.

The system incorporates features such as topic-wise question practice, speech-based mock interviews, and performance analytics, along with a buddy system, which helps students engage with each other in a competitive manner.

The results show that CorePrep successfully bridges the gap between theoretical knowledge and actual interview performance. Users are able to take advantage of a structured learning process, measurable feedback, and realistic practice scenarios, which help build both technical and communication skills.

In conclusion, CorePrep is a viable solution for a technical interview preparation system. By promoting continuous learning, self-assessment, and peer motivation, this system has tremendous potential to enhance students' interview readiness and success rates. By building on this concept and with further enhancements in technology, CorePrep can become a powerful system to help students with their long-term career growth and development.

References

- [1] MDN Web Docs, "HTML, CSS, and JavaScript Documentation," 2025.
- [2] Node.js Foundation, "Node.js Documentation," 2025.



- [3] PostgreSQL Global Development Group, “PostgreSQL Manual,” 2025.
- [4] Tailwind Labs, “Tailwind CSS Framework,” 2025.
- [5] Various research papers on interview preparation systems and learning analytics.
- [6] Research articles on Speech-to-Text and Natural Language Processing techniques
- [7] W3Schools, “Web Development Tutorials,” Available: <https://www.w3schools.com>
- [8] GeeksforGeeks, “Technical Interview Preparation Resources,” Available:
<https://www.geeksforgeeks.org>