WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

TALENT TESTER

Mr. Subramanian E[#1], Shamruthi Durga Devi P R[#2]

[#1]Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India. Email: esubramaniancse@siet.ac.in

[#2]UG Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India. Email: shamruthidurai@gmail.com

ABSTRACT-

Talent Tester is a web-based platform designed to simplify coding practice and remote technical interviews. It provides a unified environment where users can solve programming problems independently or participate in one-to-one coding interview sessions. The platform integrates a problem catalogue, in-browser code editor, external code execution, video communication and chat facilities. It aims to make technical interviews more convenient, interactive and organized. The system is developed using modern web technologies, REST APIs, database management and real-time communication services. It serves as a functional prototype with scope for future enhancements such as collaborative editing, automated evaluation and detailed feedback reports.

Keywords - Talent Tester, Coding Practice, Technical Interviews, Collaborative Coding, Code Execution, Video Communication, Real-Time Chat, Problem Catalogue, User Authentication, Session Management.

INTRODUCTION

The technical interview process plays an essential role in evaluating programming skills and problem-solving abilities, yet conducting coding interviews remotely can sometimes be challenging. Candidates and interviewers often face difficulties in sharing code, discussing programming problems, communicating effectively, and managing interview sessions using multiple applications.

With the rapid advancement of web technologies and real-time communication services, there is a growing need for integrated platforms that simplify coding practice and remote technical interviews. This project introduces **Talent Tester**, a web-based collaborative coding platform designed to help users practice programming problems and conduct technical interviews efficiently. The platform integrates a structured problem catalogue, in-browser code editor, external code execution, interview session management, video communication, and real-time chat facilities.

1. **The Growing Need for Collaborative Coding Platforms:**

In today's technology-driven environment, programming skills and problem-solving abilities play an important role in recruitment and career development. Traditional interview processes often require separate tools for coding, video meetings, and communication, resulting in interruptions and coordination difficulties. Talent Tester simplifies these activities by providing a unified environment for coding practice and technical interviews.

2. **Enhancing Coding Practice and Interview Collaboration:**

The system bridges the gap between candidates and interviewers by providing a shared platform for solving programming problems and discussing technical concepts. Users can independently practice coding problems or create and join one-to-one interview sessions. The integrated code editor, video communication, and chat facilities enable smoother interaction throughout the interview process.

3. **Improving Accessibility, Interaction, and Efficiency:**

Talent Tester focuses on making coding practice and remote interviews more convenient, interactive, and organized. By integrating modern frontend technologies, REST APIs, database management, external code execution services, and real-time communication, the platform provides essential functionalities within a single application. The system also offers opportunities for future enhancements such as collaborative code editing, automated evaluation, and detailed interview feedback.

LITERATURE REVIEW

Collaborative Coding Platforms and Online Technical Interview Systems

Web-Based Collaborative Coding and Programming Practice:

Recent developments in web technologies have increased the adoption of online platforms for programming practice and technical interview preparation. Research highlights how browser-based coding environments simplify the process of solving programming problems without requiring users to install additional software. These platforms provide structured problem collections, integrated code editors, and programming resources that

help users improve their coding skills. Such systems make programming practice more accessible, flexible, and convenient for students and job applicants.

Online Coding Environments and Real-Time Communication Techniques

In-Browser Code Editors and External Code Execution:
Modern coding platforms integrate browser-based code editors to enable users to write, modify, and execute programs within a single environment. External code execution services support multiple programming languages and provide execution results, errors, and output. These technologies eliminate the need for separate development environments and simplify the coding process. They also provide a convenient approach for evaluating programming solutions and practising technical problems.

Real-Time Communication and Video Conferencing:

Real-time communication plays an important role in conducting remote technical interviews. WebRTC-based technologies enable audio and video communication directly through web browsers without requiring additional meeting applications. Integrated chat facilities further support communication by allowing participants to exchange messages and discuss programming problems. These capabilities improve interaction between candidates and interviewers and make remote interview sessions more effective.

Interview Session Management and Collaborative Learning

Online Interview Creation and Session Management:

Efficient technical interview management requires a structured process for creating, joining, and managing interview sessions. Web-based session management techniques allow users to create interview rooms and invite participants through unique session identifiers or joining links. Authentication mechanisms help ensure that only authorized users can access the platform. These features provide an organized environment for conducting one-to-one coding interviews and collaborative programming activities.

Integrated Coding Interviews and Collaborative Learning:

Combining coding environments with communication facilities provides an effective approach to collaborative programming and technical assessment. Integrated platforms allow participants to discuss problem statements, write code, execute programs, and communicate during interview sessions. Such systems reduce the need to switch between multiple applications and improve the overall interview experience. Future developments in this area include synchronized collaborative code editing, automated evaluation, hidden test cases, interview feedback, and detailed performance reports.

EXISTING SYSTEM

In traditional technical interview environments, coding practice and remote interview processes are generally conducted using separate applications and conventional development tools. Candidates use different platforms to practise programming problems, write and execute code, attend video meetings, and communicate with interviewers. Similarly, interviewers rely on external meeting applications, code-sharing tools, and messaging platforms to conduct technical assessments and evaluate candidates.

The traditional interview process has several inherent limitations. First, conducting remote coding interviews can be challenging because participants need to switch between multiple applications to access problem statements, write code, execute programs, and communicate. Managing these activities separately can interrupt the interview process, create coordination difficulties, and reduce the overall efficiency of technical assessments.

In many cases, existing coding platforms focus primarily on programming practice and problem-solving activities. While these platforms provide the coding challenges and execution environments, they offer limited support for integrated one-to-one technical interview sessions. As a result, candidates and interviewers may need additional applications to communicate and discuss programming solutions during interviews.

Another limitation of the existing system is the lack of integrated real-time communication facilities. Video conferencing and chat services are generally provided through separate applications. Participants must switch between coding environments and communication tools to discuss problems, clarify requirements, and share feedback. This can result in communication gaps and difficulties in maintaining a smooth interview experience.

Furthermore, traditional online coding environments may provide limited interview session management capabilities. Creating interview rooms, inviting participants, managing access, and coordinating interview activities often require additional tools or manual procedures. Existing platforms may also lack a unified environment that combines coding, execution, video communication, and chat facilities.

From a broader perspective, conventional coding practice and interview systems have limited integration of programming environments and real-time collaboration features. They may not provide a centralized platform for independent coding practice and interactive technical interviews. As a result, students, job applicants, mentors, and interviewers face challenges in conducting convenient and organized programming sessions.

Overall, the existing approach depends on multiple independent applications, involves unnecessary coordination, and provides limited integration of coding and communication facilities. These challenges highlight the need for an integrated web-based platform that can simplify programming practice, support one-to-one coding interviews, and improve remote technical interview efficiency.

PROPOSED SYSTEM

The proposed Talent Tester system is a web-based collaborative coding platform designed to simplify programming practice and remote technical interviews by integrating essential coding and communication facilities into a single application. The system provides an organized environment where authenticated users can independently solve programming problems or participate in one-to-one coding interview sessions. It eliminates the need to switch between multiple applications for coding, communication, and interview management.

The proposed system provides a structured problem catalogue containing programming challenges that users can access for practice and interview preparation. An integrated in-browser code editor enables users to write and modify programs using supported programming languages. By integrating an external code execution service, the platform allows users to execute their code and view the corresponding outputs and errors without requiring a separate development environment.

One of the major features of the proposed system is its one-to-one coding interview functionality. Authenticated users can create interview sessions and invite other participants to join through session details. The system provides session management capabilities to organize interview activities and control participant access. This enables candidates and interviewers to conduct technical discussions within a shared platform.

The proposed platform also integrates real-time video communication and chat facilities to improve interaction between participants. Users can communicate through video calls and exchange messages while discussing programming problems. These features reduce dependency on external communication applications and provide a more convenient remote interview experience.

Furthermore, the system is designed with a modern frontend framework, REST-based backend services, database persistence, and user authentication mechanisms. These technologies support efficient application functionality, secure access, and organized data management. The platform is intended for students, job applicants, mentors, peer learners, and
interviewers who require an integrated environment for programming practice and technical discussions.

Overall, the proposed Talent Tester system aims to provide a convenient, interactive, and organized solution for coding practice and remote technical interviews. Future enhancements may include synchronized collaborative code editing, hidden test cases, automated evaluation, detailed performance reports, interviewer feedback, and interview scheduling.

METHODOLOGY

The proposed Talent Tester platform follows an integrated methodology that combines web development technologies, external code execution services, user authentication, session management, and real-time communication to simplify coding practice and remote technical interviews. The system is developed through a series of stages, including user registration, problem management, code editing, program execution, interview session creation, real-time communication, and system testing.

1. **User Registration and Authentication:**

The system provides a secure registration and login mechanism for users to access the platform. Users can create accounts using their credentials and authenticate themselves before accessing the available features. User information is maintained in a database to support account management, session access, and personalized activities. Authentication ensures that platform functionalities are accessible to authorized users.

2. **Programming Problem Catalogue and Practice:**

The programming practice module provides a structured catalogue of coding problems for users. Each problem contains relevant details such as problem descriptions, input requirements, output expectations, and constraints. Users can browse the available problems, select a challenge, and attempt to solve it using the integrated coding environment. This module supports independent programming practice and technical interview preparation.

3. **In-Browser Code Editing and Execution:**

The code editor module allows users to write, modify, and execute programs directly through the web application. The platform integrates an external code execution service to support program execution in supported programming languages. Users can submit their code for execution and view the generated output or error messages. This eliminates the need for separate development environments and provides a convenient coding experience.

4. **One-to-One Coding Interview Session Management:**

The interview module enables authenticated users to create and join one-to-one coding interview sessions. A user can initiate an interview session and share the corresponding session details with another participant. The system manages session information and participant access to ensure organized interview activities. Both participants can access the coding environment and discuss the selected programming problems during the session.

5. **Real-Time Video Communication and Chat:**

The communication module integrates video conferencing and chat facilities to support interaction between interview participants. Real-time video communication enables candidates and interviewers to discuss programming problems directly through the platform. The chat facility allows

participants to exchange messages and clarify technical concepts during the interview. These features reduce dependency on separate communication applications.

EXPERIMENT AND RESULT



Fig 8.1 Home page

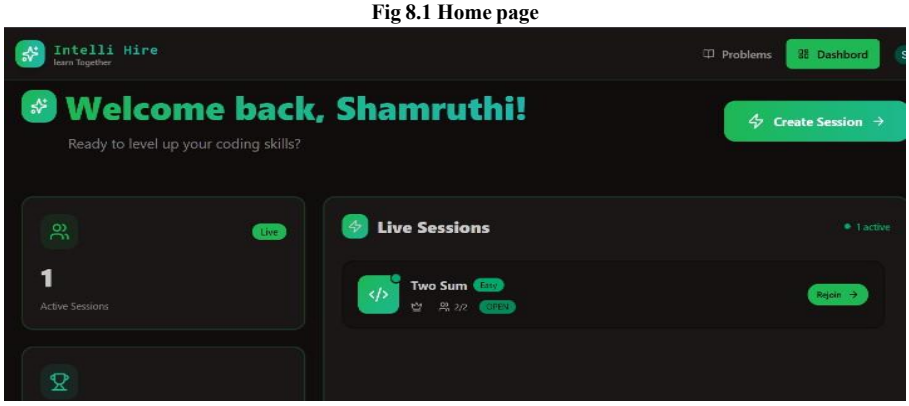


Fig 8.3. Dashboard



Fig 8.3. Practice Problems

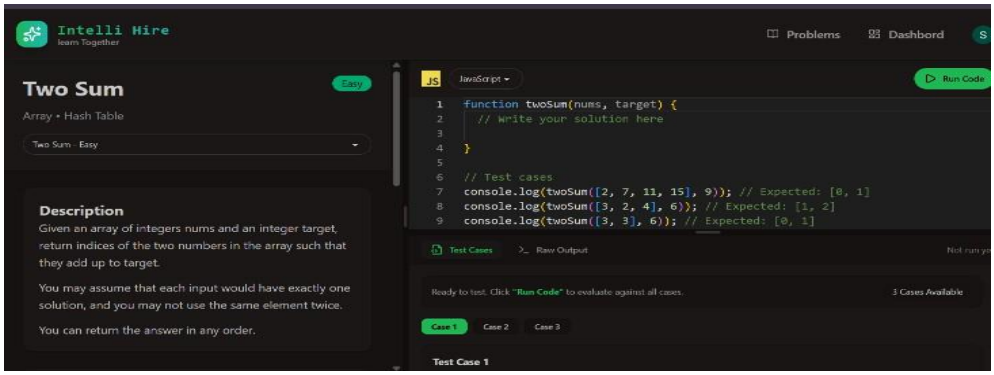


Fig 8.4. Work Space

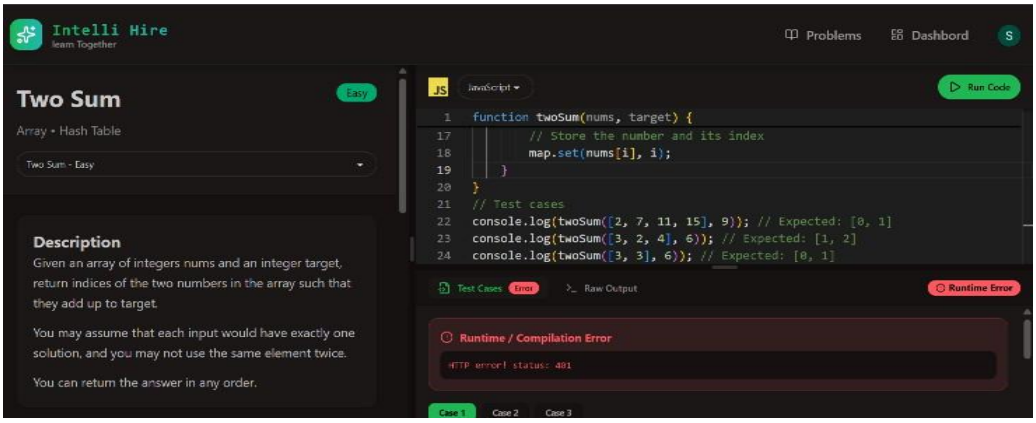


Fig 8.6. Result

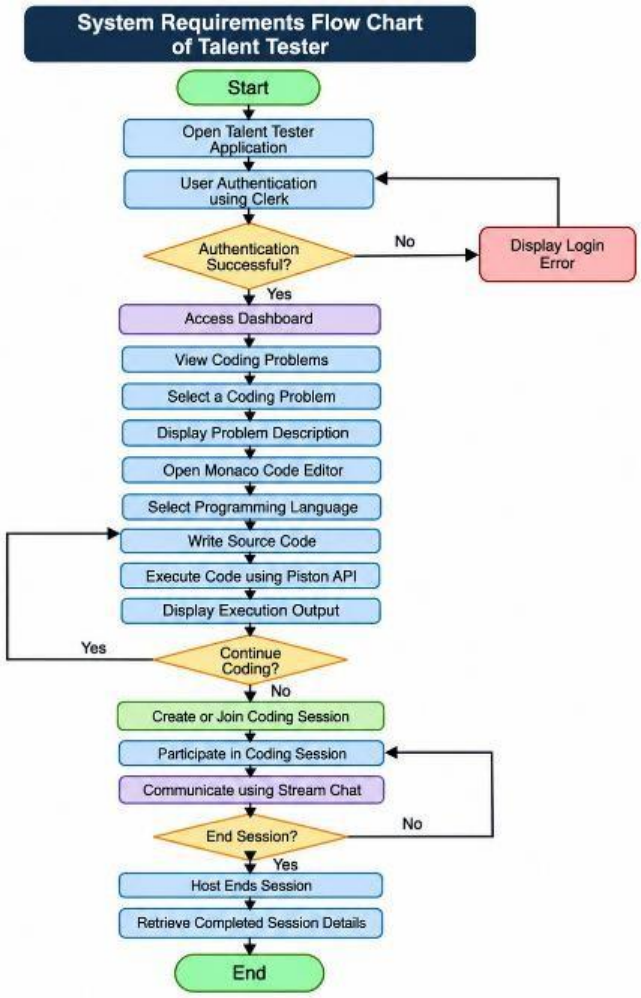


Fig 8.8. Flow chart

ADVANTAGES

- Unified Platform for Coding Practice and Technical Interviews.
- Easy Access to Structured Programming Problems.
- Integrated In-Browser Code Editor.
- Support for Multiple Programming Languages.
- Quick Code Execution and Output Generation.

- Convenient One-to-One Interview Sessions.
- Real-Time Video Communication.
- Integrated Chat for Effective Collaboration.
- Secure User Authentication and Session Management.

CONCLUSION

The **Talent Tester** platform offers an effective approach to improving traditional coding practice and remote technical interview processes through an integrated web-based environment. The system combines a structured programming problem catalogue, an in-browser code editor, and external code execution services to enable users to solve programming challenges and evaluate their solutions. It also incorporates one-to-one interview session management, real-time video communication, and chat facilities to support effective interaction between candidates and interviewers.

By combining independent coding practice with collaborative interview functionalities, the system reduces dependency on multiple applications, improves coordination, and enhances the overall interview experience. User authentication and session management further support organized access and efficient handling of interview activities.

Overall, the proposed Talent Tester platform demonstrates how modern web technologies and real-time communication services can simplify programming practice and remote technical assessments. It provides a convenient and user-friendly solution for students, job applicants, mentors, and interviewers, with future possibilities for synchronized collaborative code editing, automated code evaluation, hidden test cases, interview scheduling, and detailed performance reports.

FUTURE WORK

The Talent Tester platform can be further enhanced by incorporating advanced web technologies and intelligent programming features to improve coding practice, collaboration, and technical interview experiences. Future developments may focus on introducing real-time collaborative code editing, allowing multiple participants to write, modify, and review code simultaneously during interview sessions. Automated code evaluation and hidden test cases can also be implemented to assess program correctness, execution efficiency, and problem-solving abilities. The communication module can be enhanced with interview recording, screen sharing, and improved collaboration facilities to support more interactive technical discussions. Integration with AI-based code review tools can help identify programming errors, analyse code quality, and provide intelligent suggestions for improvement. Additionally, advanced interview management features can enable interview scheduling, automated reminders, and detailed candidate performance tracking.

Future enhancements may also include AI-powered question recommendations, coding difficulty analysis, personalized practice suggestions, and automated interview feedback. The platform can be extended to support mobile applications and cloud-based deployment, making coding practice and remote technical interviews more accessible and scalable.

Overall, these improvements can transform Talent Tester into a more comprehensive collaborative coding and technical assessment platform by providing advanced programming assistance, efficient interview management, improved communication, and an enhanced user experience.

REFERENCES

- React Team, “React Documentation,” Meta Platforms, Inc. Available: https://react.dev/
- Vite Team, “Vite Documentation,” Available: https://vite.dev/guide/
- Tailwind Labs, “Tailwind CSS Documentation,”Available: https://tailwindcss.com/docs
- DaisyUI, “DaisyUI Documentation,”Available: https://daisyui.com/docs/
- OpenJS Foundation, “Node.js Documentation,”Available: https://nodejs.org/docs/
- Express.js, “Express Web Framework Documentation,”Available: https://expressjs.com/
- MongoDB, Inc., “MongoDB Manual,”Available: https://www.mongodb.com/docs/
- Automatic,“MongooseDocumentation,” Available: https://mongoosejs.com/docs/
- Clerk, “Clerk Authentication Documentation,” Available: https://clerk.com/docs
- Microsoft, “Monaco Editor,” Available: https://microsoft.github.io/monaco-editor/
- Piston, “Piston API – Code Execution Engine,” GitHub Repository. Available: https://github.com/engineer-man/piston
- Stream, “Stream Video and Chat Documentation,” Available: https://getstream.io/video/docs/ and https://getstream.io/chat/docs/