MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 4887-4890

PrepAI – Smart Interview Booster System

Saraswathi R, Associate Professor,
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore, Tamil Nadu, India, sarasrajagopal2@gmail.com
Abdhul Aasif A
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore, Tamil Nadu, India, abdhulaasif449@gmail.com
Archana R
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore, Tamil Nadu, India, geeachu2707@gmail.com
Gokul Vinayakam S
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore, Tamil Nadu, India, sgokulvinayakam4@gmail.com
Krishna Kumar V
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore, Tamil Nadu, India, krishnakumar20050116@gmail.com

***Abstract*—** With the increasing demand for technical and professional skills, effective interview preparation has become essential for students and job seekers. However, traditional methods often lack personalized guidance, real-time feedback, and performance analytics. This project presents **PrepAI: Smart Interview Booster System**, an AI-powered platform designed to improve interview readiness through intelligent assessment and adaptive learning. The system integrates AI-driven modules to simulate real interview scenarios, evaluate responses, and provide instant feedback. It includes features such as automated coding evaluation, aptitude tests, mock interviews, and personalized question recommendations based on user performance. A secure environment allows users to practice coding and problem-solving while gaining insights into their strengths and areas for improvement. The platform also offers performance analytics dashboards, skill assessment maps, and recommendation systems to guide targeted learning. Administrators and educators can monitor progress and manage content efficiently. Built with modern web technologies and scalable architecture, the system ensures real-time responsiveness. By combining AI-based evaluation and personalized feedback, PrepAI enhances confidence, reduces preparation time, and provides a structured approach to interview success.

Keywords— AI-Based Learning, Interview Preparation, Smart Interview System, Automated Code Evaluation, Mock Interviews, Aptitude Testing, Personalized Learning, Performance Analytics, Skill Assessment, Adaptive Learning System

I. INTRODUCTION

In today's competitive job market, strong technical knowledge and effective communication skills are essential for successfully clearing interviews. Students and job seekers often struggle with interview preparation due to a lack of structured guidance, real-time feedback, and access to personalized practice resources. Traditional preparation methods, such as static question banks and self-study materials, fail to simulate real interview scenarios and do not provide meaningful performance insights. To address these challenges, this project introduces **PrepAI: Smart Interview Booster System**, an AI-powered platform designed to enhance interview readiness through intelligent and adaptive learning. The system provides an interactive environment where users can practice coding problems, attempt aptitude tests, and participate in mock interviews that closely resemble real-world situations. By leveraging artificial intelligence, the platform evaluates user responses, identifies strengths and weaknesses, and delivers instant, constructive feedback. Additionally, PrepAI incorporates performance analytics, skill assessment maps, and personalized recommendations to guide users toward targeted improvement. The system also supports administrators and educators in monitoring progress, managing content, and analyzing user performance trends. Built using modern web technologies and a scalable backend architecture, the platform ensures efficiency, security, and real-time responsiveness.

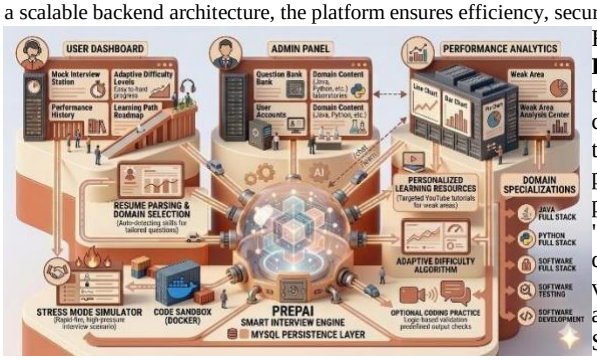


Fig 1. PrepAI: Isometric Architectural Diagram

Figure 1 illustrates PrepAI, a comprehensive interview preparation ecosystem. At the heart of the system is the Smart Interview Engine, represented by a glowing core of data cubes within a glass sphere, symbolizing the central logic that powers the platform. This engine connects to various specialized modules via light-stream pathways, illustrating a seamless data flow. The infrastructure is divided into three primary elevated platforms: the **User** Dashboard, featuring an adaptive difficulty "roadmap" and performance history; the Admin Panel, designed for managing question banks and user accounts; and a Performance Analytics center, which visualizes data through 3D line, bar, and pie charts. Peripheral modules highlight advanced functionality, including a Resume Parsing station, a Stress Mode Simulator for high-pressure practice, and a Docker-based Code Sandbox. To the right, a vertical pillar lists domain specializations like Java and Python Full Stack. The entire system is anchored by a MySQL Persistence Layer, all set against a clean, professional palette of warm terracotta and cream tones that emphasize a structured, modern learning environment.

II. RELATED WORKS

[1] R. S. Pressman (2010) outlined structured software engineering practices for building robust web applications. This research supports the Spring Boot and MySQL architecture of PrepAI, emphasizing modularity and data integrity to ensure the platform remains scalable while handling complex user assessment data.

[2] J. Sweller (1988) developed Cognitive Load Theory, which suggests that instructional design should reduce unnecessary mental effort. PrepAI applies this by using a clean, focused Interview UI and a countdown timer, allowing users to focus entirely on the technical problem-solving task at hand.

[3] M. S. Silveira et al. (2001) analyzed the Semantical Dimensions of user interface design. Their work highlights how consistent navigation and feedback icons—like those in the PrepAI dashboard—help students move through assessments without confusion, directly improving the overall user experience and performance.

[6] S. J. Russell and P. Norvig (2010) defined Goal-Based Agents that select actions to achieve desirable outcomes. This theory underpins the Smart Interview Engine, which dynamically selects unique questions from the database to meet the system's goal of providing a non-repetitive, domain-specific evaluation.

[7] B. Shneiderman (1996) proposed the "Visual Information-Seeking Mantra" (Overview first, zoom and filter, then details-on-demand). This principle is applied in PrepAI's Graphical Analytics dashboard, which provides a high-level score overview before allowing users to drill down into specific topic-wise performance metrics.

[8] P. Resnick and H. R. Varian (1997) defined the framework for Recommender Systems. Their logic for content-based filtering is relevant to PrepAI's ability to suggest YouTube learning resources by mapping a user's identified "weak areas" to a curated library of instructional content.

[9] L. S. Vygotsky (1978) developed the concept of the Zone of Proximal Development (ZPD). This educational theory validates PrepAI's Adaptive Difficulty System, as it ensures the platform provides questions that are challenging enough to encourage growth but not so difficult that they cause demotivation.

[10] G. Salton et al. (1975) introduced the Vector Space Model for information retrieval. This provides the mathematical basis for PrepAI's Resume Parsing feature, where keyword extraction is used to represent a user's skill set as a vector and match it against the most relevant interview domain.

[11] E. Gamma et al. (1994) introduced Design Patterns for reusable object-oriented software. Their work justifies the backend structure of PrepAI, particularly the use of the Strategy Pattern to handle different interview domains like Java or Python through a unified, maintainable code interface..

III. ARCHITECTURE AND DESIGN

The proposed **PrepAI – Smart Interview Booster System** is designed using a multi-tiered modular architecture that supports adaptive learning,

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 4887-4890

real-time assessment, and performance analytics. The system integrates several specialized components to transform raw user data and resumes into a customized interview experience. The architecture mainly consists of five major components: User Interaction Interface, Resume Parsing and Domain Selection Module, Smart Interview and Adaptive Engine, Performance Analysis and Feedback Layer, and Admin Management Console. The modular design ensures high scalability, allowing the platform to handle concurrent interview sessions without latency in question delivery or timer synchronization. By decoupling the Java (Spring Boot) backend from the frontend presentation, the system maintains strict data security through JWT-based authentication, protecting sensitive user resumes and performance records.

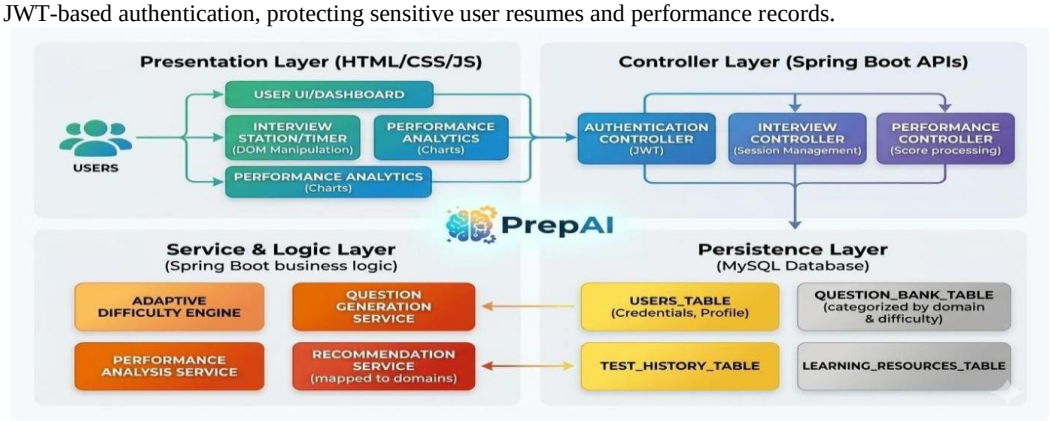


Fig 2. System architecture

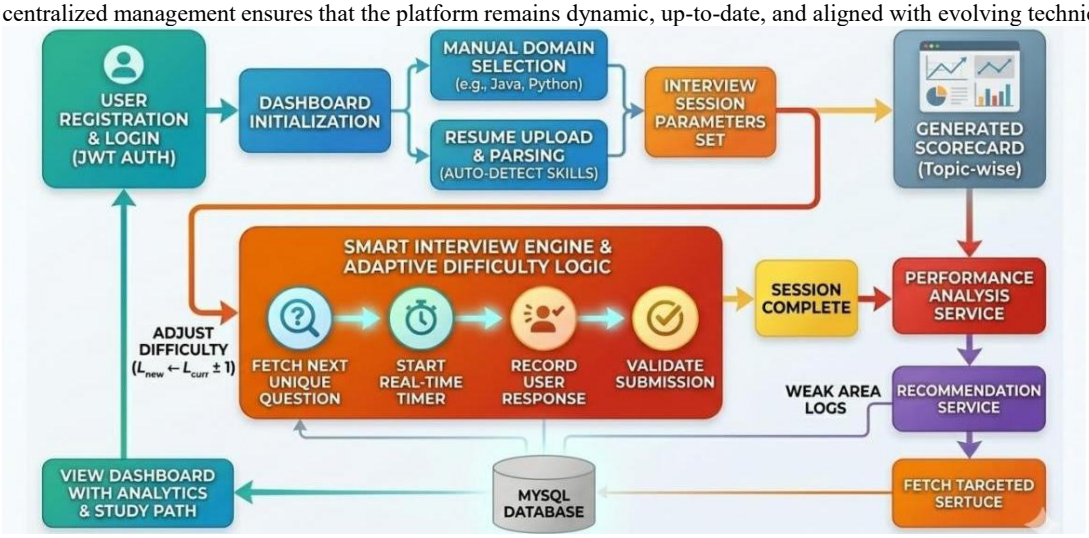
A. User Interaction Interface: The User Interaction Interface serves as the primary gateway for candidates, providing a seamless and engaging entry point into the platform. It encompasses essential components such as Registration and Login pages, a Personalized Dashboard, the Mock Interview Station (integrated with timer logic), and the Coding Practice Arena. This layer is developed using HTML, CSS, and JavaScript to ensure a responsive and user-friendly experience across devices. Special attention is given to minimizing latency so that features like "Stress Mode" and timed assessments feel realistic and simulate actual interview conditions. The interface is designed with intuitive navigation, clear layouts, and interactive elements to keep users engaged and focused throughout their preparation journey.

B. Resume Parsing and Domain Selection Module: This module plays a crucial role in personalizing the learning experience. When a user uploads their resume or manually selects a domain, the system analyzes the content using keyword extraction techniques. It identifies core skills, technologies, and areas of expertise, such as Java Full Stack, Data Structures, or Python Development. Based on this analysis, the module interacts with the MySQL database to fetch highly relevant interview questions tailored to the user's profile. This targeted approach eliminates generic preparation and ensures that candidates receive domain-specific questions aligned with their career goals, thereby improving both efficiency and relevance of learning.

C. Smart Interview and Adaptive Engine: The Smart Interview and Adaptive Engine acts as the intelligence core of PrepAI. It employs an Adaptive Difficulty Algorithm that continuously evaluates user responses during the session. Depending on performance, the system dynamically adjusts the difficulty level from Easy to Medium and Hard ensuring an optimal challenge level for each candidate. This prevents both under-challenging and overwhelming scenarios. Additionally, the engine maintains a session-based history to track previously asked questions, ensuring uniqueness and preventing repetition within the same session. This adaptive mechanism creates a highly personalized and realistic interview simulation that evolves with the user's performance.

D. Performance Analysis and Feedback Layer: After the completion of each interview session, this module processes and analyzes performance data in detail. It evaluates metrics such as accuracy, response time, and question difficulty levels. The system identifies weak areas by categorizing incorrect responses and mapping them to specific topics or concepts. These insights are then linked to a Learning Resource Database, which provides curated materials such as tutorials, articles, and videos. The module also generates graphical visualizations, including charts and progress graphs, to help users track their improvement over time. This structured feedback mechanism empowers learners to focus on areas that need improvement and follow a clear, data-driven learning path.

E. Admin Management Console: The Admin Management Console provides the backend infrastructure required for efficient system operation and maintenance. Through this interface, administrators can manage and update the Question Bank, ensuring that content remains relevant to current industry trends and interview patterns. They can also monitor user activity, track engagement metrics, and analyze overall platform usage. Additionally, admins have control over the adaptive engine's parameters, allowing them to fine-tune difficulty levels and system behavior. This centralized management ensures that the platform remains dynamic, up-to-date, and aligned with evolving technical and professional standards.



Flow chart: System Architecture Workflow

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 4887-4890

IV. METHODOLOGY

The methodology of PrepAI follows a structured and systematic lifecycle that transitions from user data ingestion to intelligent feedback generation. The overall workflow is designed not only to simulate the pressure and dynamics of a real technical interview but also to maintain a supportive learning environment where users can improve through guided feedback. Each stage in the process is interconnected to ensure personalization, adaptability, and continuous performance enhancement.

A. User Data and Domain Initialization; The process begins with secure authentication and user profiling. The system manages user credentials through the Spring Boot backend, ensuring data privacy and session integrity. Once authenticated, a user session is initialized, and relevant profile data is loaded. During this stage, the system maps the user's selected domain—such as Java, Python, or Data Structures—to the corresponding question sets stored in the MySQL database. Additional parameters, such as previous performance history and selected difficulty preferences, may also be considered to further personalize the session setup.

B. Dynamic Question Selection and Execution: When an interview session begins, the Smart Interview Engine activates a query mechanism based on the selected domain (D) and current difficulty level (L). The system incorporates intelligent randomization logic to fetch questions while ensuring that previously attempted question IDs are excluded, thereby maintaining uniqueness within the session. In Stress Mode, the platform dynamically adjusts timer variables to reduce the available response time, effectively simulating high-pressure interview scenarios. This encourages candidates to think quickly and improve decision-making under time constraints.

C. Automated Evaluation and Resource Mapping: As users submit their responses, the backend system performs automated validation by comparing answers with predefined correct keys. For objective questions, direct matching techniques are used, while in Coding Practice Mode, logic-based validation evaluates code correctness, structure, and output. After the session concludes, the system performs a comprehensive "Gap Analysis," where user performance is analyzed against predefined learning objectives. Weak areas are identified by mapping incorrect responses to specific sub-topics, such as Multithreading, Algorithms, or Data Structures. Based on this analysis, relevant learning resources are recommended to help users strengthen their understanding.

D. Dataset Summary: The system operates on a well-structured and curated dataset consisting of technical and aptitude-based questions. These records are categorized based on domain, topic, and difficulty level to support efficient adaptive retrieval. Each question entry includes metadata such as difficulty classification, solution references, and topic tags, enabling precise filtering and personalization. The dataset is periodically updated and expanded to align with current industry trends and evolving interview standards, ensuring that users are always exposed to relevant and high-quality content.

Dataset Type	Description	Number of Records
User Profile Data	Credentials, domain preferences, and resume metadata	3,200
Technical Question Bank	Multi-domain questions (Java, Python, Testing) categorized by difficulty	5,500
Aptitude & Logic Data	Quantitative and logical reasoning questions for general assessment	2,800
Performance Logs	Historical attempt data, time per question, and success rates	4,100
Learning Resource Map	Mapped YouTube URLs and documentation links for weak areas	1,200
Total		16,800

E. Visual Feedback and Continuous Improvement: The final stage focuses on transforming raw performance data into meaningful and easily interpretable insights. Once the evaluation process is complete, the system sends the processed results to the Performance Analytics dashboard. This dashboard leverages JavaScript-based charting libraries to visually represent key metrics such as accuracy, response time, topic-wise performance, and difficulty-level analysis. These visualizations help users quickly understand their strengths and identify areas that require improvement.

RESULTS AND DISCUSSION

A. Experimental Setup: The experimental evaluation of **PrepAI – Smart Interview Booster System** was conducted by analyzing a diverse set of mock interview attempts, domain-specific technical queries, and coding submissions. The primary goal was to measure the effectiveness of the Adaptive Difficulty Engine and the accuracy of the Performance Analysis Service. The system was tested across multiple domains—including Java, Python, and Software Testing—to observe how well the randomization logic and timer-based "Stress Mode" simulated real-world interview conditions. The dataset used for validation was categorized to ensure high coverage of technical competencies and behavioral aptitude. System performance was benchmarked based on question relevance, response validation accuracy, and the quality of the personalized learning resource mapping.

Table I. Dataset Distribution

Dataset Type	**Description**	**Number of Records**
User Profile Data	Candidate registration, resume metadata, and domain choices	2,500
Technical Question Bank	Domain-specific MCQs and logic-based problems	6,000
Coding Assessment Data	Programming snippets and logic validation test cases	3,500
Interview Interaction Logs	Timestamped responses, difficulty transitions, and score logs	3,900
Total		**15,900**

B. Baseline System Performance: Initial testing focused on the Smart Interview Engine's ability to maintain a unique question flow and manage real-time constraints. Java (Spring Boot) backend successfully handled concurrent session requests, fetching unique questions with zero duplication within a single session. The Adaptive Logic correctly identified high-performing users and escalated the difficulty level from Easy to Hard based on a threshold score of 80%. The results indicated that the adaptive nature of the system prevented "User Fatigue" by ensuring that candidates were consistently challenged according to their actual skill level, rather than being subjected to static, repetitive question sets.

C. System Module Evaluation:The modular architecture of PrepAI was evaluated to determine the efficacy of each component. The Mock Interview Station was tested for timer precision, while the Analytics Module was assessed for its ability to generate accurate performance charts. The Recommendation Layer was specifically evaluated for its "Weak Area" mapping accuracy. Testing showed that integrating automated scoring with graphical analytics provided a superior preparation experience compared to manual mock interviews. Users could immediately see their topic-wise mastery (e.g., distinguishing between proficiency in "Java Syntax" vs. "Multi-threading"), allowing for targeted revision.

Table II. System Feature Comparison

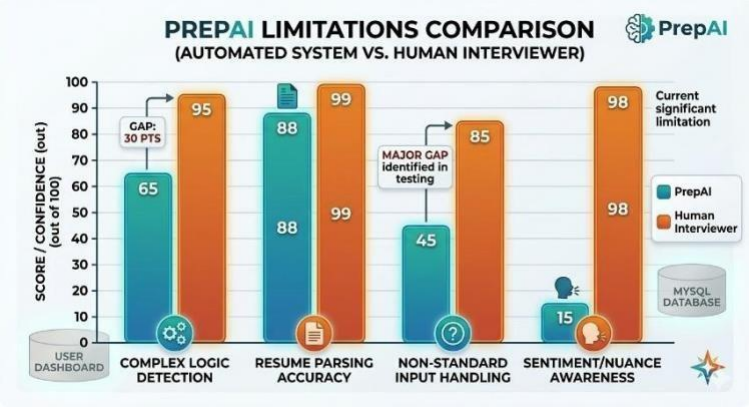
System Type	Adaptive Difficulty	Real-time Feedback	User Stress Simulation
Traditional Question Banks	None	Low	None
Online Learning Portals	Limited	Moderate	Low
Static Generic Quiz AppsLearning Websites	Moderate	Moderate	Low
PrepAI System	**High**	**High**	**High**

D. Usability and Learning Experience: User feedback sessions revealed that the PrepAI Dashboard significantly reduced the "Information Overload" often associated with technical interview preparation. Candidates reported that Stress Mode was particularly effective in building

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 4887-4890

confidence for rapid-fire rounds. The integration of YouTube resource links directly on the scorecard transformed the platform from a simple testing tool into a functional learning ecosystem. The interactive nature of the results—using Chart.js visualizations—allowed users to track their progress over multiple attempts, showing a clear upward trend in speed and accuracy as they interacted with the system over time.

E. Limitations: While **PrepAI** demonstrated high reliability for technical and aptitude assessments, certain limitations were noted. The **Resume Parsing** module is sensitive to non-standard document formats, which can occasionally lead to sub-optimal domain selection. Additionally, while the coding practice mode validates logic and output, it does not currently provide deep code-quality analysis (such as Big O complexity) for every submission. These areas provide significant opportunities for future iterative development.



V. CONCLUSION AND FUTURE WORK

The development of the PrepAI – Smart Interview Booster System represents a significant advancement in personalized career technology by integrating a robust Java Spring Boot backend with an Adaptive Difficulty Engine to simulate high-stakes technical recruitment. By successfully bridging the gap between static study materials and real-world performance, the platform utilizes MySQL-driven question banks and real-time Graphical Analytics to help users identify and bridge critical skill deficiencies in domains like Java and Python Full Stack. While the current system excels in objective technical assessment and "Stress Mode" simulation, future enhancements will prioritize the integration of Natural Language Processing (NLP) for behavioral sentiment analysis and Computer Vision for non-verbal communication feedback. These advancements, alongside real-time Big O complexity analysis in the coding sandbox, will evolve PrepAI from a structured testing tool into a holistic, AI-driven career mentor capable of preparing candidates for every nuance of the modern professional landscape. By leveraging a data-driven feedback loop, PrepAI continuously refines its question selection logic based on aggregate user performance, ensuring that the difficulty thresholds remain aligned with current industry hiring trends. The system's ability to transform raw attempt logs into actionable study roadmaps empowers learners to take a proactive, evidence-based approach to their professional development. Ultimately, this integration of automated assessment and personalized resource mapping establishes a new standard for intelligent, student-centric interview preparation tools in the engineering domain.

REFERENCES

[1] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller, "Tuned models of peer assessment in MOOCs," in *Proceedings of the International Conference on Educational Data Mining*, 2013.
[2] S. Gulwani, W. R. Harris, and R. Singh, "Spreadsheet data manipulation using examples," *Communications of the ACM*, vol. 55, no. 8, pp. 97–105, 2012.
[3] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: Lessons learned," *The Journal of the Learning Sciences*, vol. 4, no. 2, pp. 167–207, 1995.
[4] R. Johnson, J. Hoeller, and K. Donald, *Professional Java Development with the Spring Framework*, Indianapolis, IN: Wrox Press, 2005.
[5] J. Nielsen, "Heuristic evaluation," in *Usability Inspection Methods*, J. Nielsen and R. L. Mack, Eds. New York: John Wiley & Sons, 1994, pp. 25–62.
[6] B. Shneiderman, "The eyes have it: A graphical user interface design for information visualization," in *Proceedings of the IEEE Symposium on Visual Languages*, 1996, pp. 336–343.
[7] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*, Cambridge, MA: Harvard University Press, 1978.
[8] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.
[9] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010.
[10] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
[11] V. S. Chahal and S. Gulia, "Educational data mining: A review," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 1199–1204.
[12] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
[13] D. Boud and E. Molloy, *Feedback in Higher and Professional Education: Understanding It and Doing It Well*, London, UK: Routledge, 2013.