

# Skillatics: An AI-Driven Adaptive Learning and Assessment Platform for Personalized Skill Mastery and Placement Readiness

**Sourabh Bandgar, Shreyas Paranjape, Abhay Ugale**

Department of CSE (Data Science)  
D. Y. Patil College of Engineering and Technology  
Kolhapur, India  
sourabhbandgar385@gmail.com

**Mrunal Wakarekar, Suraj Patil**

Assistant Professors  
Department of CSE (Data Science)  
D. Y. Patil College of Engineering and Technology  
Kolhapur, India

**Abstract**—In the current competitive placement landscape, traditional learning platforms fail to provide personalized, adaptive, and reliable assessment mechanisms, leading to inefficient preparation and poor skill evaluation. This paper presents *Skillatics*, an AI-driven adaptive learning and assessment platform designed to enhance student placement readiness through a unified and data-driven approach. The system integrates an adaptive test engine that dynamically adjusts question difficulty based on real-time student performance, ensuring personalized learning paths. It also incorporates a secure, sandboxed coding environment using containerized execution for technical assessments, along with a lightweight proctoring system to maintain academic integrity. Furthermore, an AI-powered analytics module identifies topic-wise skill gaps and provides actionable feedback, enabling targeted improvement. To enhance engagement, the platform includes gamification features such as experience points, leaderboards, and interactive learning modes. The proposed system is implemented as a scalable web-based application using modern technologies including React, Flask, MongoDB, and external APIs such as Piston and Google Gemini. The results demonstrate that integrating adaptive learning, secure assessment, and analytics into a single platform significantly improves learning efficiency, engagement, and overall placement preparedness.

**Index Terms**—Adaptive Learning Systems, AI-Based Learning Analytics, Skill Gap Analysis, Coding Assessment Platforms, Online Proctoring Systems, Gamification in Education, Placement Preparation Systems

## I. INTRODUCTION

In today's highly competitive placement environment, students are required to possess strong aptitude, reasoning, and technical skills to secure employment opportunities. However, most students rely on traditional preparation methods such as textbooks, static online resources, and generic mock tests, which fail to adapt to individual learning needs. These approaches often lead to inefficient preparation, lack of targeted improvement, and reduced confidence among learners.

Existing online platforms provide large repositories of questions and tutorials, but they primarily follow a one-size-fits-all model. Such systems lack real-time adaptability, do not analyze student performance in depth, and fail to provide personalized feedback. Moreover, conventional assessment platforms do not ensure test integrity, as they lack secure proctoring

mechanisms. Additionally, the absence of integration between aptitude, technical practice, analytics, and engagement features results in a fragmented learning experience.

To address these challenges, this paper proposes *Skillatics*, an AI-driven adaptive learning and assessment platform that integrates multiple components into a unified system. The platform features a real-time adaptive test engine that dynamically adjusts question difficulty based on student performance, enabling personalized learning paths. It also includes a secure coding environment using containerized execution for evaluating technical skills, along with a proctoring system to ensure credible assessments.

Furthermore, the system incorporates an AI-based analytics module that identifies topic-wise skill gaps and provides actionable recommendations for improvement. To enhance user engagement, gamification elements such as experience points, leaderboards, and interactive learning modes are integrated into the platform. The system is designed with a scalable, multi-user architecture supporting students, faculty, and administrators, enabling both individual learning and institutional monitoring.

The main contributions of this paper are as follows:

- Design of an adaptive learning engine for real-time difficulty adjustment.
- Integration of a secure, sandboxed coding environment with proctoring capabilities.
- Development of an AI-driven analytics module for skill gap identification and personalized feedback.
- Implementation of a gamified learning framework to improve student engagement.
- A unified platform architecture that combines learning, assessment, analytics, and monitoring.

The remainder of this paper is organized as follows: Section II discusses the related work, Section III presents the proposed system, Section IV describes the methodology, Section V covers the implementation details, Section VI provides results and analysis, and Section VII concludes the paper with future directions.

## II. LITERATURE REVIEW

The development of adaptive learning systems has been widely explored in recent research, emphasizing the importance of personalized education. A systematic mapping study by Kabudi et al. [1] analyzed numerous AI-enabled adaptive learning systems and highlighted their effectiveness in tailoring content based on individual learner performance. However, a key limitation identified is the difficulty of achieving accurate personalization during the initial learning stages, commonly referred to as the cold-start problem.

To address this, Schu"tt et al. [2] proposed a lightweight adaptive mechanism capable of dynamically adjusting difficulty levels even with limited interaction data. Their approach demonstrates that effective adaptation can be achieved without requiring extensive historical data, which is particularly relevant for real-time learning systems.

In the domain of learning analytics, Piech et al. [3] introduced the concept of Deep Knowledge Tracing (DKT), which utilizes recurrent neural networks to model a student's evolving knowledge state. This approach enables deeper insights into learning patterns beyond simple performance metrics. Building upon this, Ahadi et al. [4] proposed a skill analytics framework that maps student performance to job-relevant competencies, enabling targeted skill gap identification and personalized recommendations.

Secure evaluation of programming skills has also been an area of significant research. Peveler et al. [5] conducted a comparative study between containerized and sandbox-based code execution environments, concluding that containerization provides a balanced trade-off between security and performance. Similarly, Dos"ilovic' and Mekterovic' [6] presented a modular architecture for Docker-based online code execution systems, validating the effectiveness of containerized solutions for safely executing untrusted code.

Gamification has been recognized as a powerful tool for enhancing student engagement. Park and Kim [7] proposed design principles for leaderboard systems, emphasizing the importance of balanced competition to prevent user demotivation. Additionally, Li et al. [8] conducted a systematic review highlighting that gamification significantly improves motivation and participation when implemented with meaningful reward mechanisms.

Despite these advancements, existing platforms such as GeeksforGeeks [9] and PrepInsta [10] primarily function as static content repositories. They lack real-time adaptability, integrated analytics, secure coding environments, and comprehensive engagement mechanisms. This results in a fragmented learning experience where students must rely on multiple tools for preparation.

The reviewed literature establishes that while individual components such as adaptive learning, analytics, secure code execution, and gamification have been explored independently, there is a lack of a unified system that integrates all these features. The proposed system addresses this gap by combining adaptive assessment, AI-driven analytics, secure coding

evaluation, and gamification into a single scalable platform, thereby providing a comprehensive and effective solution for placement preparation.

## III. PROPOSED SYSTEM

This paper proposes *Skillatics*, an AI-driven adaptive learning and assessment platform designed to provide a unified, personalized, and secure environment for placement preparation. The system integrates multiple components, including adaptive testing, coding assessment, AI-based analytics, proctoring, and gamification, into a single scalable web-based platform.

### A. System Architecture

The proposed system follows a client-server architecture consisting of three primary layers: the presentation layer, application layer, and data layer. The frontend is developed using React, providing an interactive user interface for students, faculty, and administrators. The backend is implemented using Flask, which handles business logic, authentication, and API communication. MongoDB is used as the database for storing user data, test sessions, questions, and performance metrics. Additionally, external services such as Piston API for code execution, Google Gemini for AI-based question generation, and Brevo for OTP-based authentication are integrated into the system.

### B. Adaptive Learning Engine

The adaptive learning engine dynamically adjusts the difficulty level of questions based on student performance. The test begins at a moderate difficulty level, and subsequent questions are selected based on the correctness of previous responses. This ensures that students are continuously challenged at an appropriate level, enabling efficient learning and accurate assessment of their abilities.

### C. Coding Assessment Environment

The platform incorporates a secure coding environment using the Monaco Editor for code writing and Piston API for execution. Student code is executed in a containerized sandbox to prevent unauthorized system access. Each coding problem includes multiple test cases, and the system evaluates submissions based on correctness, runtime, and memory usage.

### D. Proctoring System

To ensure assessment integrity, the system includes a lightweight proctoring mechanism. It monitors user activity through browser events such as tab switching and integrates webcam-based monitoring during test sessions. All violations are logged with timestamps and can be reviewed by faculty members to identify suspicious behavior.

### E. AI-Based Analytics and Feedback

The analytics module processes student performance data to identify topic-wise strengths and weaknesses. It generates skill gap reports and provides personalized recommendations for improvement. Visualizations such as radar charts, line graphs, and bar charts are used to present insights in an intuitive manner.

#### F. Gamification Framework

To enhance engagement, the platform integrates gamification elements including experience points (XP), levels, badges, and leaderboards. Students earn rewards based on their performance, encouraging consistent participation and motivation. Interactive game-based learning modules further reinforce concepts through practice.

#### G. Multi-User Role Management

The system supports multiple user roles, including students, faculty, TPOs, and administrators. Each role is provided with a dedicated dashboard and access control mechanisms to ensure secure and role-specific functionality. Faculty and administrators can monitor student performance and manage system data effectively.

The proposed system aims to provide a comprehensive solution by combining adaptive learning, secure assessment, data-driven insights, and engagement mechanisms into a single platform, thereby improving the overall effectiveness of placement preparation.

### IV. METHODOLOGY

The proposed system follows a data-driven approach combining adaptive learning, performance analytics, and engagement mechanisms. This section describes the core methodologies used for adaptive testing, skill gap analysis, and gamification.

#### A. Adaptive Learning Algorithm

The adaptive learning engine dynamically adjusts the difficulty level of questions based on student responses. Let  $d_i$  represent the difficulty level of the current question, where  $d_i \in [1, 5]$ .

The next difficulty level  $d_{i+1}$  is determined as follows:

$$d_{i+1} = \begin{cases} \min(d_i + 1, 5), & \text{if answer is correct} \\ \max(d_i - 1, 1), & \text{if answer is incorrect} \end{cases} \quad (1)$$

This approach ensures that the system adapts in real time, increasing difficulty for high-performing students while reducing it for struggling learners, thereby maintaining an optimal challenge level.

#### B. Test Session Workflow

Each test session consists of a predefined number of questions selected from a pool categorized by topic and difficulty. The workflow is as follows:

- Initialize test with a default difficulty level.
- Fetch questions dynamically based on current difficulty and test type.
- Update difficulty after each response using the adaptive algorithm.
- Record responses, difficulty progression, and timestamps.
- Compute final score and performance metrics at the end of the session.

#### C. Skill Gap Analysis

The system identifies skill gaps by analyzing student performance across different topics. Let  $S_t$  represent the score obtained in topic  $t$ , and  $N_t$  be the total number of questions attempted in that topic.

$$P_t = \frac{S_t}{N_t} \times 100 \quad (2)$$

where  $P_t$  represents the proficiency percentage for topic  $t$ . Topics with lower  $P_t$  values are identified as weak areas, and targeted recommendations are generated accordingly.

#### D. Gamification Model

The gamification system is based on an experience point (XP) model that rewards performance and engagement. The total XP earned is calculated as:

$$XP = (Score \times 10) + B_p + B_s \quad (3)$$

where:

- $Score$  represents the percentage score achieved in the test,
- $B_p$  is the bonus for perfect performance,
- $B_s$  is the speed bonus for completing the test within a threshold time.

Additionally, a difficulty multiplier is applied for higher-level questions to encourage tackling challenging problems.

#### E. Coding Evaluation Method

For coding assessments, each submission is evaluated against multiple test cases. Let  $T$  be the total number of test cases and  $T_p$  be the number of passed test cases.

$$Accuracy = \frac{T_p}{T} \times 100 \quad (4)$$

The system also considers execution time and memory usage to provide comprehensive feedback on code performance.

#### F. Violation Detection Mechanism

The proctoring system monitors user behavior during assessments. Events such as tab switching are detected using browser event listeners. Each violation is recorded and aggregated to generate a violation score:

$$V = \sum_{i=1}^n v_i \quad (5)$$

where  $v_i$  represents individual violation events. Higher values of  $V$  indicate suspicious activity and are flagged for review.

The combination of these methodologies enables the system to provide personalized learning, secure evaluation, and actionable insights, making it an effective solution for placement preparation.

## V. IMPLEMENTATION

The proposed system is implemented as a full-stack web application using modern technologies to ensure scalability, performance, and maintainability. The implementation consists of frontend, backend, database, and external service integrations.

### A. Frontend Implementation

The frontend is developed using React 18 with Vite as the build tool, providing a fast and responsive user interface. Tailwind CSS is used for styling, enabling a clean and consistent design across the platform. The Monaco Editor is integrated to provide a professional coding interface for programming assessments. Chart.js is utilized to visualize performance data through line charts, bar charts, and radar charts. Axios is used for handling HTTP requests and communication with the backend APIs.

### B. Backend Implementation

The backend is built using the Flask framework, which provides a lightweight and modular architecture for developing RESTful APIs. Authentication is implemented using JSON Web Tokens (JWT) with the Flask-JWT-Extended library, ensuring secure and stateless user sessions. Passwords are securely stored using bcrypt hashing through the passlib library. Flask-CORS is used to handle cross-origin requests.

The backend is structured into modular components, including authentication, adaptive testing, coding execution, gamification, analytics, and administrative management. Each module is exposed through dedicated API endpoints, enabling efficient communication with the frontend.

### C. Database Design

MongoDB Atlas is used as the primary database due to its flexible schema and scalability. The system maintains collections such as users, test sessions, questions, coding problems, and gamification profiles. PyMongo and Flask-PyMongo are used for database interaction.

### D. Code Execution Integration

The platform integrates the Piston API as the primary code execution engine. User-submitted code is executed in an isolated environment, ensuring secure and reliable execution without direct access to the host system. The system evaluates code against multiple test cases and returns outputs such as standard output, error messages, execution time, and memory usage.

### E. AI Integration

The system utilizes the Google Gemini API for generating multiple-choice questions dynamically. This enables scalable content generation and reduces dependency on manually curated question banks.

### F. Email and Authentication Services

The Brevo API is used for OTP-based email verification during authentication. OTPs are time-bound and securely validated to ensure safe access control.

### G. Deployment and Infrastructure

The entire application, including both frontend and backend, is deployed on the Render cloud platform. The frontend and backend services are hosted as separate deployments, enabling seamless integration and scalability.

MongoDB Atlas is used as the cloud database service, providing high availability and data persistence. Environment variables are used to securely store API keys and configuration parameters. HTTPS is enforced to ensure secure communication between system components.

This deployment architecture ensures a scalable, secure, and production-ready system capable of handling real-time user interactions.

## VI. RESULTS AND ANALYSIS

The performance of the proposed system was evaluated using real-time data generated from student interactions across adaptive tests, coding assessments, analytics dashboards, and gamified activities. The results demonstrate the effectiveness of the system in improving learning outcomes, engagement, and assessment credibility.

### A. Performance Improvement Analysis

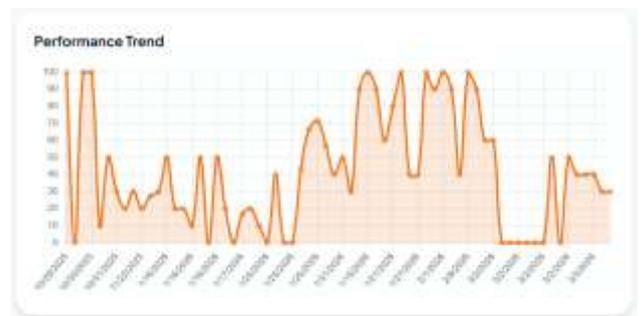


Fig. 1. Student Performance Trend Over Time

As shown in Fig. 1, the student performance trend exhibits fluctuations initially, followed by gradual improvement in later stages. This behavior reflects the impact of the adaptive learning engine, where the system dynamically adjusts difficulty levels based on performance, enabling progressive learning and skill enhancement.

### B. Skill Gap Analysis

As shown in Fig. 2 the radar chart representing topic-wise proficiency levels. It is observed that certain areas such as ratios and averages have lower scores, indicating weak topics. The system effectively identifies these gaps and enables targeted recommendations for improvement.

This analysis is based on aggregated performance data across multiple test sessions, allowing the system to detect



Fig. 2. Topic-wise Skill Gap Analysis

consistent patterns of underperformance rather than isolated mistakes. By quantifying topic-wise proficiency, the platform ensures that learning interventions are data-driven and personalized. This targeted approach helps students focus on critical weak areas, thereby improving overall learning efficiency and conceptual clarity.

C. Personalized Recommendation System



Fig. 3. AI-Based Personalized Recommendations

As shown in Fig. 3 the AI-driven recommendation system, which provides actionable feedback based on performance trends. The system suggests focusing on weak areas and highlights performance issues such as low scores, thereby guiding students toward structured improvement.

D. User Engagement and Gamification

As shown in Fig. 4, the leaderboard system ranks students based on XP, levels, and achievements. This gamification framework introduces a competitive environment that encourages continuous participation and performance improvement. The integration of XP-based progression and achievement badges provides immediate feedback and rewards, reinforcing positive learning behavior. Students are incentivized to complete more tests, maintain streaks, and achieve higher ranks, which leads to increased platform engagement.

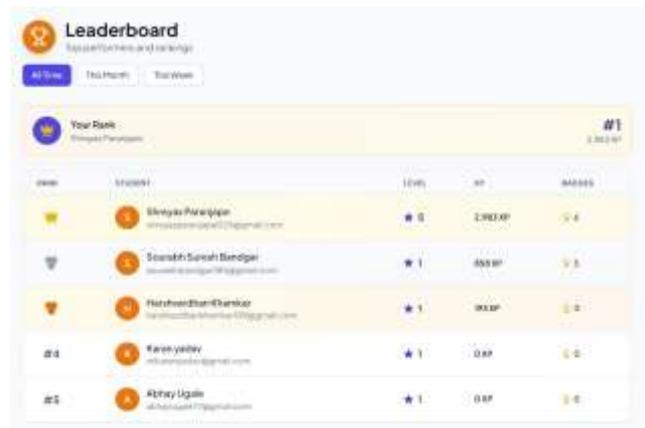


Fig. 4. Leaderboard and Gamification System

Furthermore, the leaderboard mechanism promotes peer comparison, which acts as a motivational factor for students to improve their performance relative to others. This competitive aspect not only enhances engagement but also contributes to consistent learning habits over time.

Overall, the gamification model effectively transforms traditional learning into an interactive and rewarding experience, significantly improving student motivation and retention.

E. Coding Assessment Evaluation

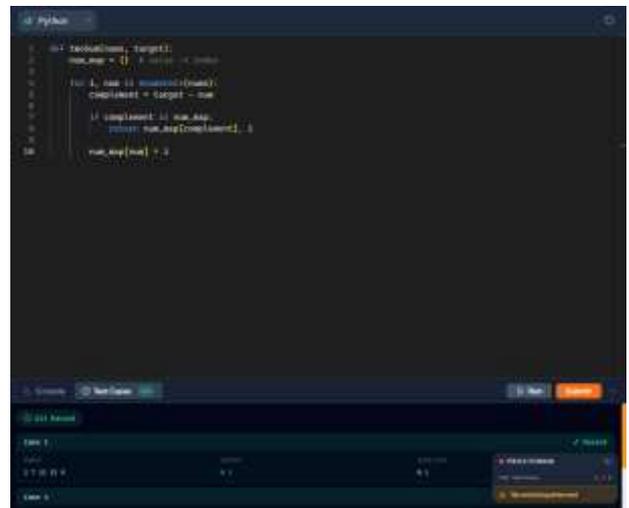


Fig. 5. Coding Environment and Test Case Execution

As shown in Fig. 5 the coding environment where student submissions are executed using the Piston API. The system evaluates code against multiple test cases and provides detailed outputs, including correctness validation and execution feedback, ensuring reliable assessment of programming skills.

### F. Proctoring and Violation Detection

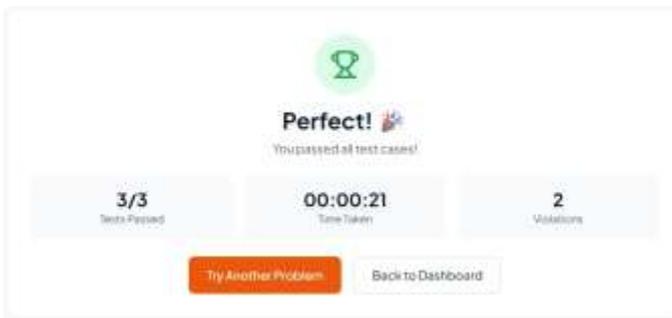


Fig. 6. Violation Detection and Monitoring

As shown in Fig. 6 the proctoring system detecting tab-switching violations during test sessions. The system logs such activities and provides real-time alerts, ensuring the credibility and integrity of online assessments.

### G. Analytics Dashboard Evaluation



Fig. 7. Student Analytics Dashboard

As shown in Fig. 7 the comprehensive analytics dashboard, which includes performance metrics, skill analysis, and historical trends. This enables students and faculty to monitor progress and make data-driven decisions.

### H. Overall Analysis

The results demonstrate that the integration of adaptive learning, AI-based analytics, coding evaluation, and gamification significantly enhances learning efficiency and engagement. The system provides a unified platform that not only improves student performance but also ensures secure and credible assessment, making it highly effective for placement preparation.

## VII. CONCLUSION

This paper presented *Skillatics*, an AI-driven adaptive learning and assessment platform designed to improve placement readiness through a unified and data-driven approach. The system integrates multiple components, including an adaptive learning engine, secure coding environment, AI-based analytics, proctoring mechanisms, and gamification features, into a single scalable platform.

The adaptive test engine dynamically adjusts question difficulty based on student performance, enabling personalized learning experiences. The integration of a secure coding environment ensures reliable evaluation of technical skills, while the proctoring system enhances the credibility of online assessments. Additionally, the AI-based analytics module provides meaningful insights into student performance by identifying skill gaps and generating actionable recommendations.

The results demonstrate that the proposed system improves learning efficiency, enhances user engagement, and provides a more reliable assessment framework compared to traditional static platforms. By combining personalization, security, and engagement, the system offers a comprehensive solution for modern placement preparation.

Overall, the proposed platform addresses key limitations of existing systems and establishes a foundation for intelligent, scalable, and effective learning environments in higher education.

This work demonstrates the potential of AI-driven systems in transforming modern education and assessment methodologies.

## VIII. FUTURE WORK

The proposed system can be further enhanced by incorporating advanced features to improve functionality and scalability. Future work includes the integration of AI-based face detection and eye tracking for enhanced proctoring, as well as screen recording for more robust monitoring of assessments.

The platform can be extended with a mobile application to improve accessibility and user engagement. Additionally, an AI-powered chatbot can be introduced to provide real-time assistance and doubt resolution for students. Integration of video-based learning content linked to identified weak areas can further enhance personalized learning.

In the long term, the system can be expanded into a multi-institution platform with support for corporate hiring integration, enabling companies to evaluate and recruit students based on platform performance. The introduction of proctored certification and structured learning paths can further transform the platform into a comprehensive ecosystem for skill development and placement preparation.

## REFERENCES

- [1] Kabudi et al., "Adaptive learning systems: A systematic mapping study," 2021.
- [2] Schu'tt et al., "Dynamic difficulty adjustment for adaptive learning," 2020.
- [3] C. Piech et al., "Deep Knowledge Tracing," NIPS, 2015.
- [4] A. Ahadi et al., "Skills analytics for programming education," 2016.
- [5] Peveler et al., "Containerized vs sandboxed execution environments," 2020.
- [6] Dos'ilovic' and Mekterovic', "Docker-based online judge system," 2018.
- [7] Park and Kim, "Leaderboard design principles in gamification," 2019.
- [8] Li et al., "Gamification in education: A systematic review," 2020.
- [9] GeeksforGeeks, "Online learning platform," Available: <https://www.geeksforgeeks.org>
- [10] PrepInsta, "Placement preparation platform," Available: <https://prepinsta.com>