Coding Portal: A Smart Online Coding Platform for Educational Institutions

Dr. A. MANJULA[1], M.SURAJ[2], G.VAISHNAVI[3] , V.RAMANJALI[4], N.VAMSHI CHARY[5]

[1]Associate Professor, Department Of CSE, Jyothishmathi Institute Of Technology And Science, Karimnagar, Telangana,India.

[2]UG Student, Department Of CSE(AI&ML), Jyothishmathi Institute Of Technology And Science, Karimnagar, Telangana,India.

[3]UG Student, Department Of CSE(AI&ML), Jyothishmathi Institute Of Technology And Science, Karimnagar, Telangana,India.

[4]UG Student, Department Of CSE(AI&ML), Jyothishmathi Institute Of Technology And Science, Karimnagar, Telangana,India.

[5]UG Student, Department Of CSE(AI&ML), Jyothishmathi Institute Of Technology And Science, Karimnagar, Telangana,India.

---***---

Abstract - The increasing requirement of programming skills in today’s educational system emphasizes the need for an efficient tool that facilitates coding practice and assessment within an academic setting. The traditional practice of conducting coding competitions is based on manual evaluation of the coding challenges. This is a time-consuming and labor-intensive process. In addition, there is a need for a centralized platform to manage the coding competitions. This paper proposes a web-based system, Coding Portal: A Smart Online Coding Platform for JITS Campus, which is an automated coding competition and programming assessment system. The proposed system is an integrated platform where students can participate in coding competitions and submit their coding challenges through a web-based interface. The proposed system is built on React, Node.js, Express.js, and MongoDB technologies. The proposed system is efficient in managing coding competitions and evaluating the coding challenges. The proposed system is a reliable solution for promoting competitive programming and improving student learning outcomes within an academic setting.

Key Words: Online Coding Platform, Automated Code Evaluation, Competitive Programming, Web-Based Learning System, Programming Assessment, Educational Technology.

1.INTRODUCTION

The skill of programming is an essential requirement in contemporary education, especially in fields such as computer science, programming, and artificial intelligence. Educational institutions are encouraging students to develop their programming skills by engaging in programming practice and coding contests. However, traditional approaches to conducting coding contests and assessments are often less effective, as they are cumbersome, lack centralized control, and result in delayed generation of results. These problems necessitate a more organized approach for managing coding activities in academic environments.

The objective of the research is to develop an automated web-based coding platform for programming practice and managing coding contests. The proposed system, named "Coding Portal," is a web-based platform where a student can practice programming, and a contest can be conducted with centralized control over the entire process. The system allows a student to write, compile, and execute the program, and the results can be obtained by evaluating the program with a set of test cases.

The significance of this study lies in its ability to improve the efficiency of coding competition management, reduce manual workload, and provide instant feedback to learners. By promoting an interactive and competitive programming environment, the proposed platform enhances student engagement and supports the development of essential programming skills required in the modern technological landscape.

2. Body of Paper

This section presents the design, implementation, and key functionalities of the proposed Coding Portal system. The system is developed as a web-based platform to automate coding competitions and programming assessments within an academic environment.

2.1 Literature Survey

[1] studies the acceptance of Online Judge systems in programming education and highlights that such platforms enable automatic evaluation of coding submissions using predefined test cases. The study shows that factors such as motivation, self-efficacy, and social influence positively impact the usage of these systems, while reducing manual effort for instructors.

[2] proposes an advanced automated code assessment model that evaluates not only program correctness but also syntax, structure, and originality. The system integrates multiple evaluation techniques to improve grading accuracy and provide detailed feedback, thereby enhancing student learning outcomes and reducing evaluation time.

[3] discusses the architecture and operational aspects of Online Judge systems, emphasizing the importance of scalability, reliability, and performance. The study presents a modular system design involving execution engines, servers, and databases to efficiently handle large volumes of code submissions.

[4] presents an online coding competition platform that allows administrators to organize contests and participants to solve problems using an integrated coding interface. The system includes features such as secure authentication, event management, and real-time evaluation, improving the efficiency of conducting coding competitions.

[5] describes an interactive coding platform designed to enhance programming skills among students. The platform supports continuous practice, automated evaluation, and performance tracking, encouraging students to improve their problem-solving abilities through regular coding activities.

However, most existing systems are developed for general use and do not focus specifically on institutional requirements. The proposed Coding Portal addresses this limitation by providing a centralized and scalable solution tailored for managing coding competitions within an academic campus environment.

2.2 System Overview

The idea behind the Coding Portal is to create a platform that enables students to participate in coding events, solve coding problems, and submit their codes using the coding interface provided in the portal. The portal is capable of handling multiple users simultaneously, allowing the interaction between the frontend, backend, and database components of the portal to be seamless. The architecture of the portal is based on the modular approach.

2.3 System Architecture

The proposed smart coding platform is based on a layered and modular system architecture that ensures scalability, security, and efficient academic management. The system architecture is composed of four major components, namely Frontend, Backend, Code Execution Engine, and Database.

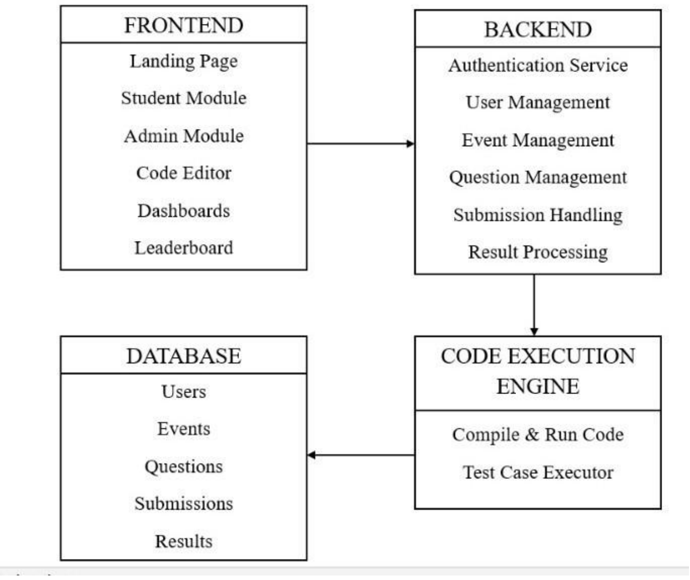


Fig -1: System Architecture of the Coding Portal

In the above Fig-1, the system architecture for the coding portal is represented. Here, the frontend is the user interface for the system. It provides separate interfaces for students and administrators. The frontend contains modules such as landing page, student module, admin module, code editor, dashboard, and leaderboard. This layer allows the student to participate in coding events and also enables them to view the performance metrics.

The Backend is the central processing unit for the system. This layer is responsible for handling user authentication, user management, event management, question management, and result processing. This layer coordinates the workflow execution among the frontend, code execution engine, and database.

The Code Execution Engine is responsible for executing the programs uploaded by the user. This engine receives program execution requests from the backend and returns execution results for further processing.

The Database provides persistent storage for system data, including user details, event information, programming ques- tions, code submissions, and evaluation results. This central- ized storage supports performance tracking, dashboard visualization, and leaderboard generation.

Overall, this modular architecture ensures smooth interac- tion between users and system services while maintaining security, scalability, and performance. By separating interface design, business logic, data management, and code execution, the proposed coding portal provides a robust foundation for institutional-level programming practice and academic assessment.

2.4 Methodology

The proposed coding event platform is developed as a web-based system with the aim of supporting secure and scalable programming competitions in academic institutions. The system is designed in a way that user interface operations, program execution, data storage, and application processing are separated for better maintainability and stability.

2.4.1. Frontend Implementation

The user interface of the Coding Portal is created using React with TypeScript, which creates a dynamic user interface that is responsive to both students and administrators. The front end is responsible for rendering all user interface components, such as dashboards, coding, and leaderboards. Tailwind CSS is used to create a clean and responsive layout, which is efficient on different devices.
An integrated Monaco Editor is used to provide an in-browser coding experience, where users can edit their code, submit their code, etc. This is done using multiple programming languages, providing a better user experience through various features, such as syntax highlighting. Interaction with backend services is done using Axios, which handles all requests such as logging in, participating in events, retrieving problems, etc.

2.4.2. Backend Implementation

The server-side logic is implemented using Node.js and Express.js. This is the main processing unit of the system. The backend takes care of the basic operations such as user authentication, events, questions, and evaluations.
User authentication is done by JSON Web Tokens (JWT). This allows for controlled access to the system's functionalities. Passwords are also encrypted using bcrypt for maximum security. The backend offers RESTful APIs for the frontend's efficient interaction with the system. Once the code is submitted, the backend will validate the request and then route it for execution.

2.4.3. Database Implementation

A cloud-based MongoDB database is used to manage all the data of the application. To simplify the handling of the database and the data stored in it, the Prisma ORM is used. The data stored in the database includes the details of the users, the events, the coding questions, and the submissions. The details of the users help in the authentication and authorization of the users. The details of the events include the parameters of the competitions, like the time and the number of users. The questions include the problem statement and the test cases. The submissions include the code and the results of the code.

2.4.4. Code Execution Engine Implementation

The evaluation of the submitted programs is done by a special execution component. When a program is received, it is executed in a safe and isolated manner to avoid misuse of the system. It supports various programming languages such as Python, C, and Java.
For each program submitted, it is checked against a set of input cases, and the output produced is compared with a set of results that are already present in the system. On the basis of this comparison, various verdicts are provided by the system, such as accepted, incorrect output, and compilation error.
The integration of these components ensures that the Coding Portal provides a consistent and efficient solution for conducting a programming competition in a fully automated manner.

2.5 Key Features

The Coding Portal system includes several important features designed to enhance the efficiency of coding competitions and improve user experience. The major features of the system are as follows:

2.5.1. User Authentication

The system provides secure registration and login functionality for both students and administrators. Authentication mechanisms ensure that only authorized users can access the platform and perform specific actions based on their roles.

2.5.2. Event Management

Administrators can create, manage, and control coding events. They can define event duration, assign programming questions, and monitor participant activities throughout the competition.

2.5.3. Coding Interface

The platform offers a browser-based coding environment where users can write, edit, and submit programs. The interface supports multiple programming languages and provides features such as syntax highlighting for better coding experience.

2.5.4. Automated Code Evaluation

Submitted programs are automatically compiled and executed against predefined test cases. The system evaluates the correctness of the solution and generates results instantly, reducing manual effort.

2.5.5. Leaderboard System

A real-time leaderboard is maintained to display participant rankings based on performance. This feature encourages competition and allows users to track their progress during events.

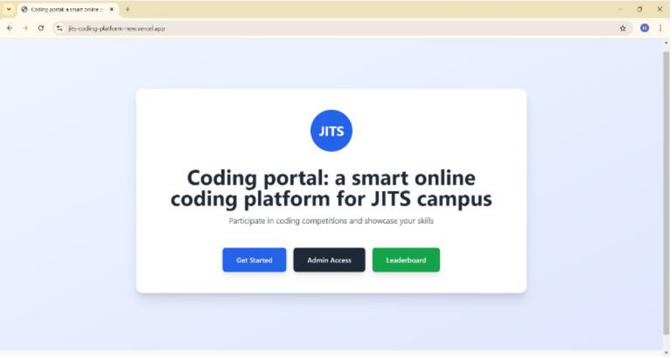
2.5.6. Result and Performance Tracking

The system stores submission results and allows users to view their performance history. This helps students analyze their coding skills and improve over time.

These features collectively ensure that the Coding Portal provides an efficient, reliable, and user-friendly platform for conducting coding competitions and enhancing programming skills.

2.6 Results and Discussion

The developed Coding Portal was successfully implemented and tested within the campus environment. The system demonstrates a complete workflow, including user authentication, event creation, participant management, code execution, and automated evaluation. The results confirm that the platform operates as an efficient and scalable solution for managing coding competitions within an academic setting.



Coding portal: a smart online coding platform for JITS campus

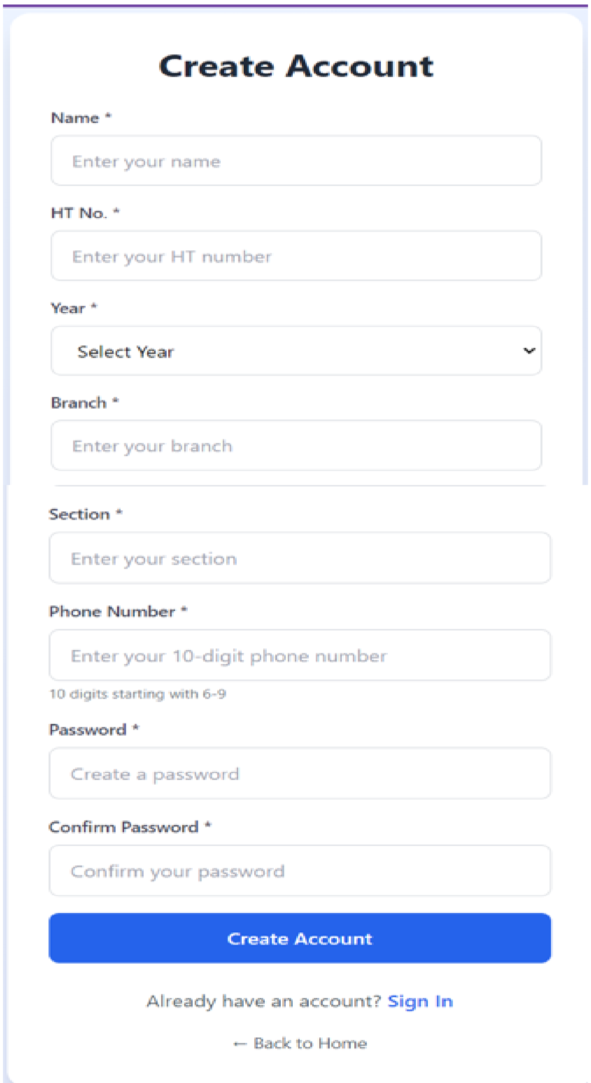
Participate in coding competitions and showcase your skills

Coding portal: a smart online c
jits-coding-platform-new.vercel.app
JITS
Get Started
Admin Access
Leaderboard

Fig -2: Landing Page of the Coding Portal

Fig.2 describes the landing page of the Coding Portal, which serves as the main entry point of the system. It provides a clear and user-friendly interface that allows students and administrators to easily navigate the platform. The page displays the platform title along with a brief description encouraging users to participate in coding activities. It includes key options such as Get Started, Admin Access, and Leaderboard, enabling users to access login, manage events, and view rankings. The design is responsive and ensures smooth accessibility across different devices, making it easy for users to interact with the system.

Fig -3 describes the student registration interface, which allows new users to create an account in the Coding Portal system. The form collects essential details such as name, hall ticket number, academic year, branch, section, phone number, and password. Input validation ensures that all required fields are correctly filled, preventing incomplete or invalid entries. After successful registration, user data is securely stored in the database, with passwords protected using encryption techniques to ensure security and privacy. This module simplifies the



Create Account

Name *

Enter your name

HT No. *

Enter your HT number

Year *

Select Year

Branch *

Enter your branch

Section *

Enter your section

Phone Number *

Enter your 10-digit phone number

10 digits starting with 6-9

Password *

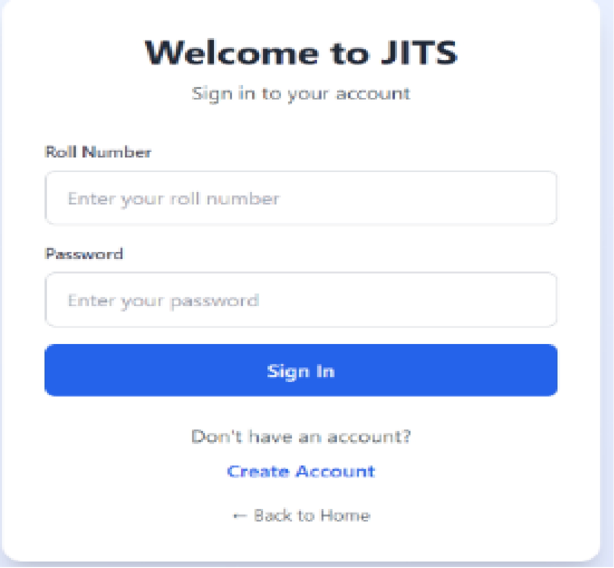
Create a password

Confirm Password *

Confirm your password

onboarding process and ensures that only valid users can participate in coding competitions

Fig -3: Student Registartion Page of the Coding



Welcome to JITS

Sign in to your account

Roll Number

Enter your roll number

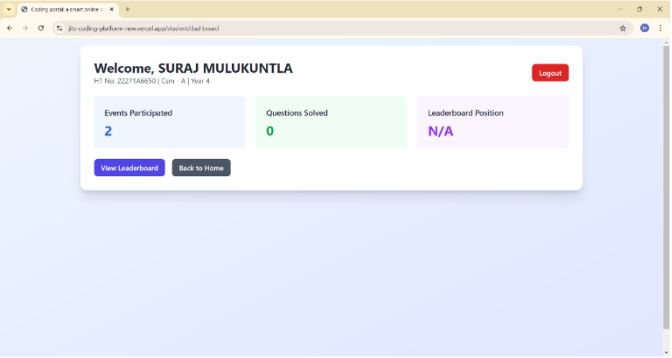
Password

Enter your password

Portal

Fig -4: Student Login Page of the Coding Portal

Fig -4 describes the student login interface, which allows registered users to securely access the Coding Portal platform. Students enter their roll number and password to authenticate their identity, ensuring that only authorized users can access coding competitions and system features. The authentication process uses JSON Web Tokens (JWT) to maintain a secure session after successful login. If incorrect credentials are entered, the system displays an error message and restricts access, protecting user data. The interface is designed with a simple and clean layout, enabling students to log in quickly and participate in coding events without difficulty.



Coding portal: a smart online c
jits-coding-platform-new.vercel.app/student/dashboard

Welcome, SURAJ MULUKUNTLA

HT No: 22271A6650 | Csm - A | Year 4

Events Participated 2

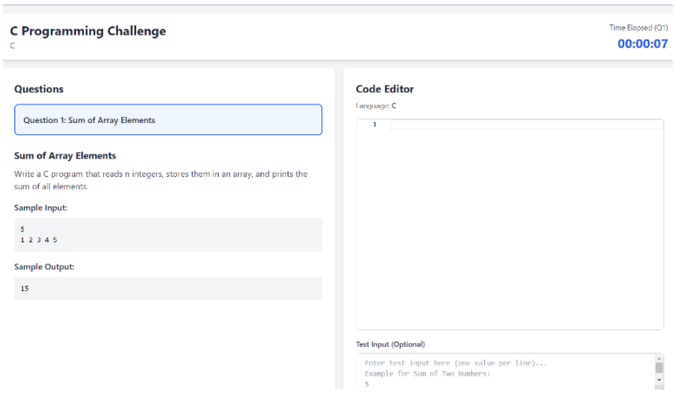
Questions Solved 0

Leaderboard Position N/A

Logout
View Leaderboard
Back to Home

Fig -5: Student Dashboard of the Coding Portal

Fig -5 describes the student dashboard, which provides a personalized interface for students after successful login. It displays key information such as number of events participated, problems solved, and leaderboard ranking, helping students track their performance. The dashboard also shows active coding events, allowing users to join competitions directly without navigating multiple pages. In addition, it includes options to view the leaderboard and return to the homepage. The interface is designed to present information clearly and efficiently, improving user experience and enabling students to monitor their progress within the platform.



C Programming Challenge

C

Time Elapsed (Q1)

00:00:07

Questions

Question 1: Sum of Array Elements

Sum of Array Elements

Write a C program that reads n integers, stores them in an array, and prints the sum of all elements.

Sample Input:

```
5
1 2 3 4 5
```

Sample Output:

```
15
```

Code Editor

Language: C

```
1
```

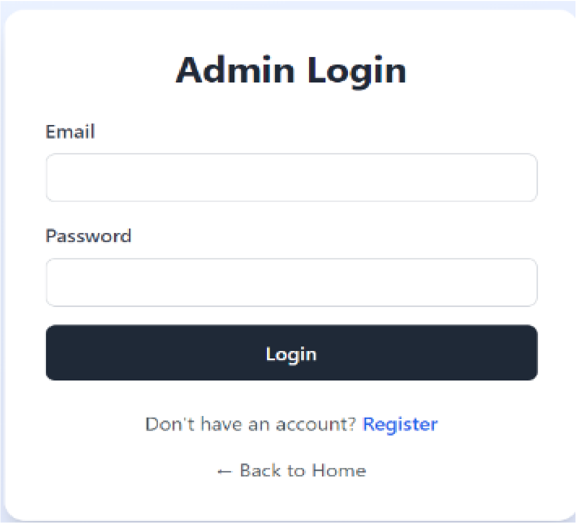
Test Input (Optional)

```
Enter test input here (one value per line)...
Example for Sum of Two Numbers:
5
```

Fig -6: Coding Challenge interface

Fig -6 describes the coding challenge interface, which is a key component of the Coding Portal system that allows students to read problems and write solutions within the browser. The interface is divided into two sections, where the left side displays the problem description along with sample input and output, and the right side provides a Monaco code editor for writing programs. The editor supports multiple programming languages and includes features such as syntax highlighting and code formatting. Students can also test their code using custom inputs before submission. Once submitted, the system forwards the program to the backend for execution and evaluation. This interface provides an interactive environment for coding practice and competition.

ISSN: 2583-6129
DOI:10.55041/ISJEM05744



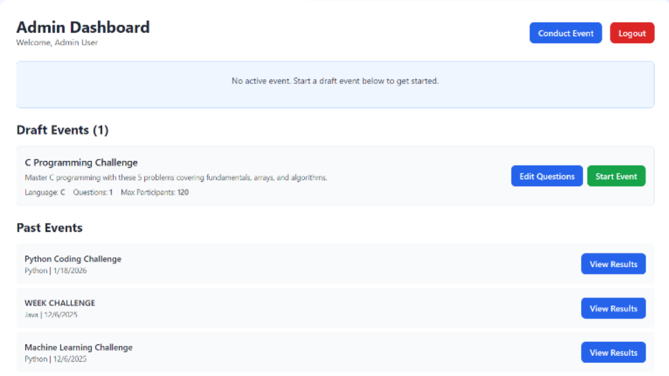
Admin Login

Email

Password

Fig -7: Admin Login Page of the Coding Portal

Fig -7 describes the administrator login interface, which enables authorized administrators to access the management features of the Coding Portal system. The login process verifies credentials using secure authentication mechanisms, ensuring that only users with administrative privileges can enter the system. After successful login, administrators are redirected to the dashboard, where they can manage coding events, questions, and monitor student performance. This module ensures that all administrative operations are protected and accessible only to authorized personnel.



Admin Dashboard
Welcome, Admin User

No active event. Start a draft event below to get started.

Draft Events (1)

C Programming Challenge
Master C programming with these 5 problems covering fundamentals, arrays, and algorithms.
Language: C Questions: 1 Max Participants: 120

Past Events

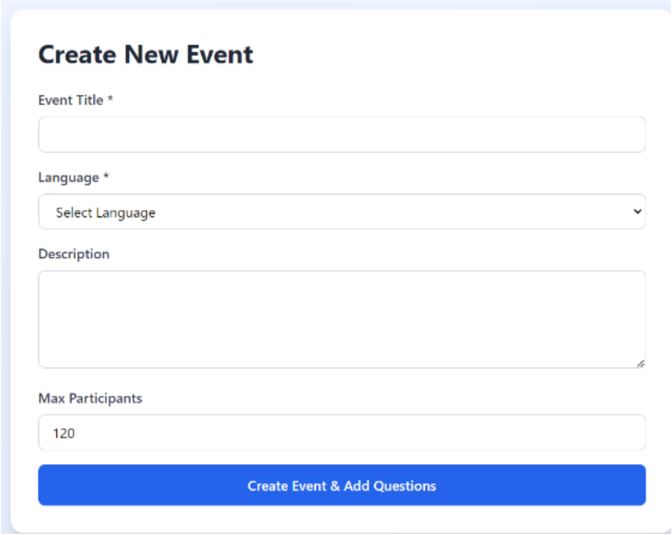
Python Coding Challenge
Python | 1/18/2026

WEEK CHALLENGE
Java | 12/6/2025

Machine Learning Challenge
Python | 12/6/2025

Fig -8: Admin Dashboard of the Coding Portal

Fig -8 describes the admin dashboard, which provides administrators with centralized control over the Coding Portal system. It displays details of coding events such as event title, programming language, participant count, and current status, allowing administrators to manage events effectively. Through this interface, admins can create, modify, start, or stop events, as well as manage programming questions and test cases. It also enables monitoring of participant activity and submissions in real time. This dashboard simplifies the overall management process and ensures smooth execution of coding competitions.



Create New Event

Event Title *

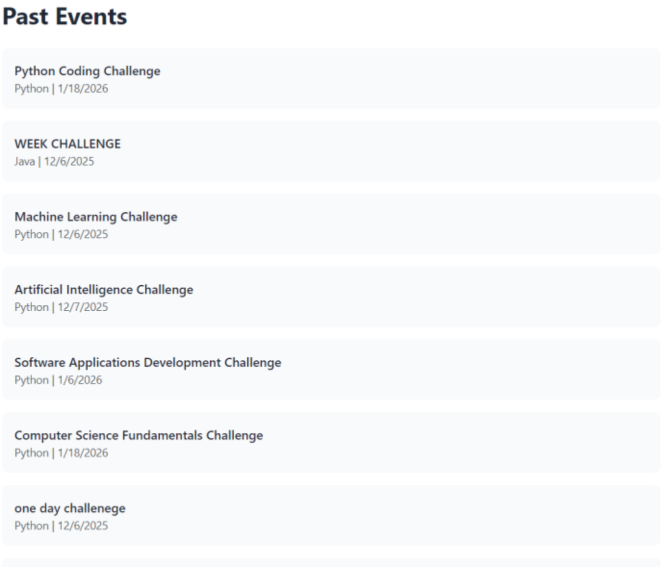
Language *
Select Language

Description

Max Participants
120

Fig -9 Event Creation Interface

Fig -9 describes the event creation interface, which allows administrators to organize new coding competitions within the Coding Portal. Administrators can define event details such as title, programming language, description, and participant limit. They can also add programming questions and test cases to ensure each event is properly structured. Once configured, the event information is stored in the database and becomes available to students when activated. This interface enables efficient planning and management of coding competitions.



Past Events

Python Coding Challenge
Python | 1/18/2026

WEEK CHALLENGE
Java | 12/6/2025

Machine Learning Challenge
Python | 12/6/2025

Artificial Intelligence Challenge
Python | 12/7/2025

Software Applications Development Challenge
Python | 1/6/2026

Computer Science Fundamentals Challenge
Python | 1/18/2026

one day challenege
Python | 12/6/2025

Fig -10: Details Of Past Events

Fig -10 describes the past events module, which maintains a record of previously conducted coding competitions on the platform. It displays details such as event title, programming language, and event date, allowing users to review earlier competitions. This module helps both students and administrators analyze results and track performance over time. By maintaining historical data, the system supports better evaluation of past events and helps improve future coding competitions.

3. CONCLUSION

The coding portal developed in this project offers an efficient platform for carrying out coding competitions and programming assessments within an academic environment. The coding portal allows students to take part in coding competitions and assessments, solve coding problems using the coding interface provided by the system, and evaluate the code based on the test cases provided. This offers an efficient way of carrying out coding competitions and assessments since the system automates the execution and result generation for the students. Moreover, the coding portal offers a user-friendly interface for the students to take part in coding competitions and assessments. The coding portal also offers the required functionality for the system administrator to carry out the required operations using the interface provided by the system. The coding portal developed in this project has been able to achieve its objective of providing an efficient platform for carrying out coding competitions and assessments within an academic environment.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Dr. A. Manjula, Associate Professor, Department of Computer Science and Engineering, for her continuous guidance, valuable suggestions, and support throughout the development of this project.
We are also thankful to Dr. V. Neelima ,Professor and Head of the Department of CSE-AIML, for his encouragement and support.
We extend our gratitude to Dr. T. Anil Kumar, Principal of Jyothishmathi Institute of Technology and Science, Karimnagar, for providing a supportive environment to carry out our academic work successfully.
We would also like to thank all the faculty members and staff of JITS for their support and for providing the necessary facilities to complete this project. Finally, we express our gratitude to everyone who directly or indirectly contributed to the successful completion of this work.

M. SURAJ (22271A6651)
G. VAISHNAVI (22271A6652)
V. RAMANJALI (22271A6629)
N. VAMSHI CHARY (23275A6601)

1. Daramsenge Bilegjargal and Nien-Lin Hsueh "Understanding students' acceptance of online judge systems in programming education," *IEEE Access*, vol. 9, pp. 123456–123468, 2021.

2. R. Parvathy, M. G. Thushara, and J. M. Kannimoola, "Automated code assessment and feedback: A comprehensive model for improved programming education," *IEEE Access*, 2025.

3. Y. Watanobe, M. M. Rahman, T. Matsumoto, U. K. Rage, and P. Ravikumar, "Online judge system: Requirements, architecture, and experiences," *International Journal of Software Engineering and Knowledge Engineering*, vol. 32, no. 6, pp. 917–946, 2022.

4. M. Verma, A. Vishwakarma, and R. Ranje, "Online platform for coding competition," *International Journal of Research Publication and Reviews*, vol. 4, no. 4, pp. 5240–5244, Apr. 2023.

5. T. K. Chandru, M. Dinesh Kumar, S. Karthikeyan, and K. Saranya, "Interactive coding platform for students," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 4S, pp. 295–299, Nov. 2018.

6. R. Conejo, B. Barros, and M. F. Bertoa, "Automated assessment of complex programming tasks using SIETTE," *IEEE Transactions on Learning Technologies*, vol. 12, no. 4, pp. 470–484, 2019.

7. H. M. Chen, B. A. Nguyen, Y. X. Yan, and C. R. Dow, "Analysis of learning behavior in an automated programming assessment environment: A code quality perspective," *IEEE Access*, vol. 8, pp. 167341–167354, 2020.

8. L. Mekterovic, L. Brkic, B. Milasinovic, and M. Baranovic, "Building a comprehensive automated programming assessment system," *IEEE Access*, vol. 8, pp. 81154–81172, 2020.

9. F. J. Gambo, O. Gambo, R. O. Ogundokun, A. N. Babatunde, and C. C. Lee, "GRAD-AI: An automated

grading tool for code assessment and feedback in programming course," *Education and Information Technologies*, vol. 30, no. 1, pp. 9859–9899, 2025.

10. M. Messer, N. C. C. Brown, and M. Kölling, "Automated grading and feedback tools for programming education: A systematic review," *ACM Transactions on Computing Education*, vol. 24, no. 2, pp. 1–35, 2024.

11. Y. Watanobe, M. M. Rahman, T. Matsumoto, U. K. Rage, and P. Ravikumar, "IDL-LTSOJ: Integrated intelligent defect localization and lightweight task scheduling online judge," *High-Confidence Computing*, vol. 5, no. 1, p. 100185, 2025.

12. M. F. Li, S. K. Das, and H. Wang, "Clustering source code for automatic assessment in programming education," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 2, pp. 215–230, 2025.

13. L. Zhang, J. Chen, and Y. Liu, "ACCA: Automating the correctness assessment of AI-generated code," *Journal of Systems and Software*, vol. 210, p. 111432, 2024.

14. P. Garcia, R. Patel, and S. Kumar, "The future of grading programming assignments in education: Automation and large language models," *Computers & Education: Artificial Intelligence*, vol. 6, p. 100187, 2024.

15. H. Kim, A. Singh, and T. Nguyen, "An empirical study on online judge tests and their role in programming education," *Proceedings of the ACM on Software Engineering Education*, vol. 3, no. 1, pp. 45–59, 2024