International Journal of Research Publication and Reviews

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

Let Me Code: An Integrated and Intelligent Web-Based Learning Environment for Programming Education

Gyanendra Dwivedi[1], Ankit Chandra[2], Rounak Agrahari[3], Anukul Sinha[4], Rajeshwar Dewangan[5]

Computer Science and Engineering Department, Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India
gyanudwivedi735@gmail.com[1];ankitchandrapihu@gmail.com[2];agraharirounak17@gmail.com[3] anukulsinha1332003@gmail.com[4];
rajeshwarkd@gmail.com[5]

ABSTRACT–

Let Me Code is a next-generation e-learning platform that integrates interactive technologies to teach coding through an immersive and hands-on web interface. Utilizing tools like the Monaco code editor, Piston API, OpenAI, and Gemini, the platform enables real-time coding, AI-powered debugging, and interactive assessments. Designed with inclusivity and accessibility in mind, the platform caters to learners of all levels—from beginners to advanced programmers and educators. This paper presents the platform's architecture, features, and future potential as a holistic coding education tool.

Keywords—Coding education, Interactive learning, Web application, Monaco Editor, Piston API, AI-based assistant, Next.js, JavaScript.

I. INTRODUCTION

In today's digital age, coding has evolved from a specialized skill to a foundational capability required across numerous industries. With the increasing importance of software in domains ranging from healthcare to finance, the demand for effective coding education has surged. Traditional methods of teaching programming, which often rely heavily on theoretical instruction, fail to offer the hands-on experience that learners need to grasp practical concepts effectively.

To address this challenge, we introduce 'Let Me Code'—an innovative e-learning platform that redefines programming education through immersive, interactive learning experiences. This platform integrates real-time code execution, AI-driven feedback, and gamified assessments within a single web application, enabling learners to practice, learn, and improve seamlessly. Built with technologies such as Next.js, the Monaco code editor, and APIs like Piston and OpenAI, 'Let Me Code' bridges the gap between theory and application, promoting a learn-by-doing philosophy.

II. SYSTEM ARCHITECTURE AND TECHNOLOGIES

'Let Me Code' leverages a modern technology stack to provide an intuitive, scalable, and robust platform for programming education. The system architecture includes the following key components:

A. Frontend:

- HTML, CSS, and JavaScript for responsive user interfaces.

- Next.js framework for server-side rendering and seamless routing.

- Monaco code editor for in-browser code writing with features like syntax highlighting and auto-completion.

B. Backend:

- Node.js with integrated APIs to handle code execution, debugging, and AI features.

- Piston API to support multi-language code execution in real time.

- OpenAI and Gemini APIs to provide intelligent suggestions and explanations.

C. Database:

- MongoDB is used for managing user profiles, session history, quiz results, and activity logs.

D. Deployment:

- The platform is hosted on DigitalOcean, with CI/CD pipelines managed via GitHub Actions for streamlined updates.

- Monitoring tools like Grafana and New Relic ensure platform stability and scalability.

III. METHODOLOGY

The development of the "Let Me Code" platform followed a structured and iterative approach grounded in the Agile Software Development Life Cycle (SDLC). This methodology was chosen to accommodate evolving requirements, enable rapid feedback integration, and ensure continuous delivery of functional features. The methodology involved the following key stages:

A. Requirement Analysis and Planning

Initial planning involved identifying the needs of various user groups including beginners, intermediate learners, and educators. Based on this analysis, the platform was conceptualized to provide an all-in-one coding learning environment with real-time execution and AI-driven support.

B. Design and Architecture

A modular architecture was designed to support flexibility and scalability. The frontend was structured using Next.js and integrated with the Monaco Editor for code writing. Backend services were designed to interface with Piston API for code execution and OpenAI API for AI features. A NoSQL database (MongoDB) was selected for its flexibility in managing diverse user data and session histories.

C. Agile Development and Sprints

The project was divided into multiple sprints, each lasting two weeks. Each sprint focused on implementing specific features, such as the code editor, quiz engine, or user dashboard. Regular sprint reviews and retrospectives ensured that feedback was continuously incorporated, improving usability and system performance over time.

D. Implementation and Integration

Feature components were developed independently and then integrated. Key implementations included:

- Real-time code execution using the Piston API.
- Contextual AI assistance using the OpenAI API.
- Interactive quizzes with automated evaluation.
- Secure user authentication and session management.

E. Testing and Evaluation

A comprehensive testing strategy was employed:

- **Unit testing** for individual modules.
- **Integration testing** to ensure component interaction.
- **User acceptance testing (UAT)** with sample learners and educators to validate functionality and usability.

F. Deployment and Monitoring

The platform was deployed using DigitalOcean virtual private servers. CI/CD pipelines via GitHub Actions were used for automated testing and updates. Monitoring tools such as Grafana were configured to observe server performance, user activity, and application health.

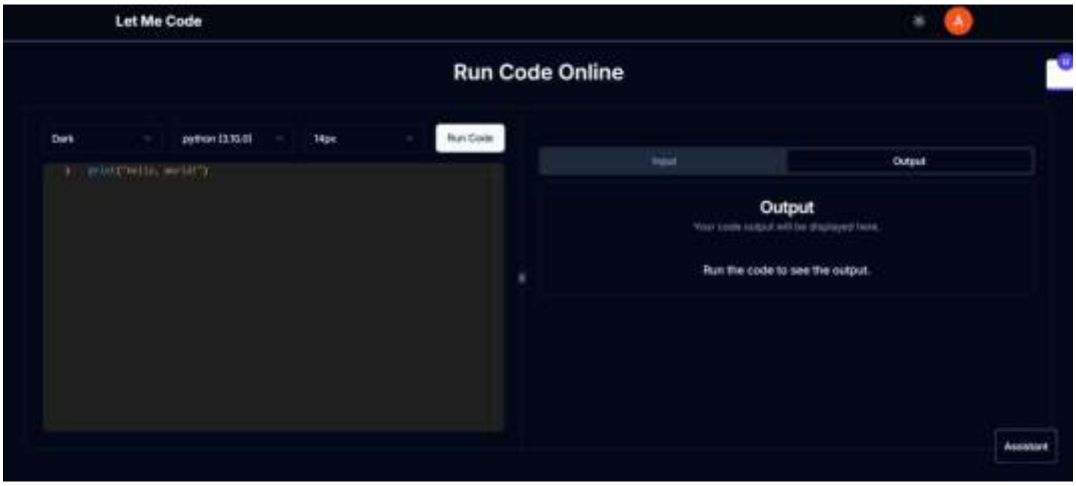
IV. PLATFORM FEATURES AND FUNCTIONALITY

The 'Let Me Code' platform is designed to provide a user-centric, feature-rich learning experience. Key features include:

A. Code Editor:

- Integrated Monaco editor with real-time feedback.

- Supports multiple programming languages, enabling learners to switch contexts easily.



Let Me Code

Run Code Online

Dark · python (3.10.0) · 14px · Run Code

```
print("Hello, world!")
```

Input · Output

Output

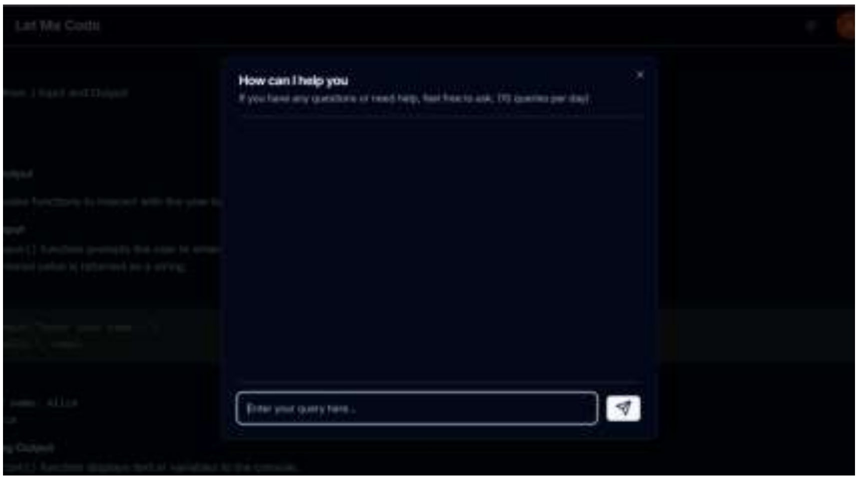
Your code output will be displayed here.

Run the code to see the output.

Assistant

B. AI-Powered Assistance:

- GPT-based support for code explanation, syntax correction, and debugging suggestions.



How can I help you

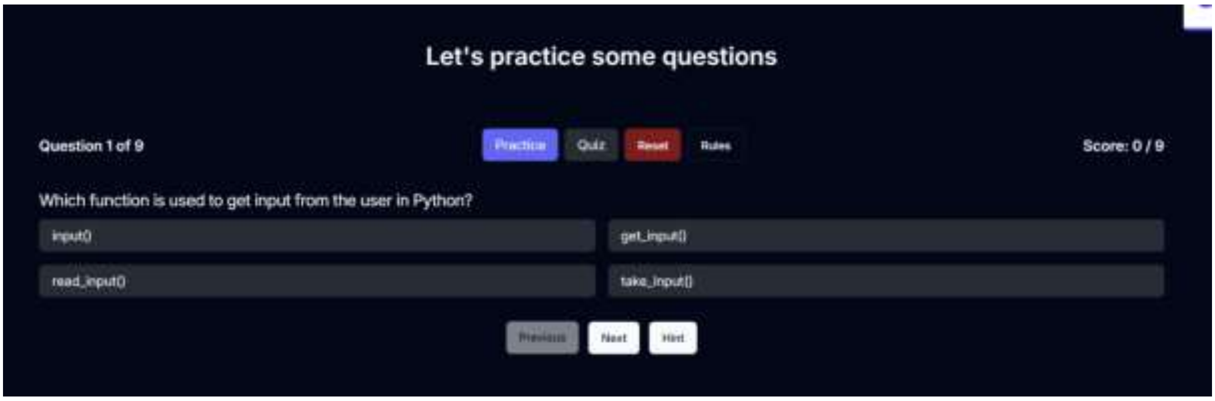
If you have any questions or need help, feel free to ask. (15 queries per day)

Enter your query here...

C. Interactive Quizzes and Assessments:

- Dynamic MCQs based on the learning session.

- Feedback and scoring mechanisms for self-evaluation.



Let's practice some questions

Question 1 of 9 · Practice · Quiz · Reset · Rules · Score: 0 / 9

Which function is used to get input from the user in Python?

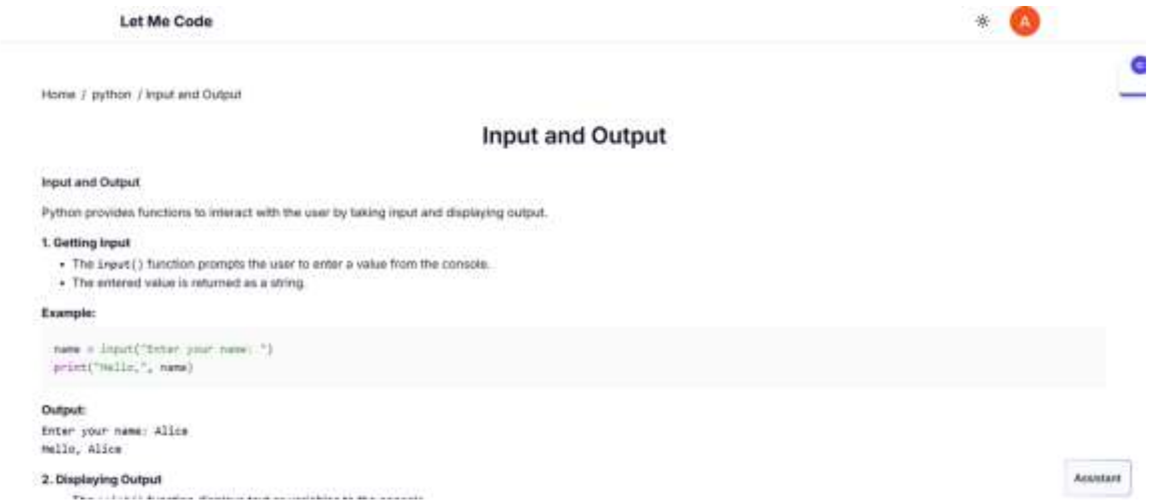
- input()
- get_input()
- read_input()
- take_input()

Previous · Next · Hint

D. Personalization:

- Theme and layout customization options.

- Progress tracking and resume-from-last-session features.



Home / python / Input and Output

Input and Output

Input and Output

Python provides functions to interact with the user by taking input and displaying output.

1. Getting Input

- The `input()` function prompts the user to enter a value from the console.
- The entered value is returned as a string.

Example:

```
name = input("Enter your name: ")
print("Hello,", name)
```

Output:

```
Enter your name: Alice
Hello, Alice
```

2. Displaying Output

E. Security and Authentication:

- Secure login with OAuth 2.0 authentication.

- Encrypted data transmission and storage.

V. RESULT AND DISCUSSION

Initial user testing of the platform demonstrated high levels of engagement and ease of use. Learners reported an improved understanding of programming concepts due to the instant feedback and the interactive nature of the platform. Educators also highlighted the potential of 'Let Me Code' as a supplementary tool in hybrid and online learning environments.

Key benefits observed include:

- Reduced cognitive load due to a unified learning environment.

- Higher retention rates due to active learning through coding challenges.

- Improved learner autonomy supported by AI-guided help and progress monitoring.

VI. CONCLUSION

'Let Me Code' stands at the forefront of educational innovation, offering a scalable, accessible, and intelligent solution for programming education. Its emphasis on experiential learning, coupled with AI support and real-time interaction, positions it as a comprehensive alternative to traditional learning methods. As digital literacy becomes increasingly vital, platforms like 'Let Me Code' can democratize access to coding education and inspire a new generation of developers.

REFERENCES

[1] Mozilla Developer Network, “HTML, CSS, and JavaScript Documentation.” https://developer.mozilla.org/

[2] Next.js Documentation, “React Framework for Production.” https://nextjs.org/docs

[3] Piston API, “Code Execution API.” https://github.com/engineer-man/piston

[4] OpenAI API Documentation. https://platform.openai.com/docs

[5] Gemini API Documentation. https://gemini.com/api

[6] MongoDB Documentation. https://www.mongodb.com/docs

[7] Agile Manifesto. https://agilemanifesto.org/

[8] DigitalOcean Documentation. https://www.digitalocean.com/docs

[9] Monaco Editor API Documentation. https://microsoft.github.io/monaco-editor/

[10] Grafana Documentation. https://grafana.com/docs