

NexaClass: An AI-Powered Virtual Classroom for Interactive Programming Education

Y. Divya Vani¹; P. Pujitha²; M. Suma Sree³; B. Sai Pradeep⁴; K. Manoj Vamsi⁵;
P. Jashuva⁶

^{1,2,3,4,5,6}Sri Vasavi Engineering College, Department of Computer Science Engineering Tadepalligudem, Andhra Pradesh India.

Publication Date: 2026/03/25

Abstract: NexaClass is an AI-enhanced virtual classroom platform built to improve online programming education by combining interactive assessments, an integrated code editor, and intelligent learning support. The platform enables faculty to design quizzes, coding tasks, and slip tests, while students can write, compile, and test code directly within the system and maintain organized digital notes. A conversational AI assistant helps students debug code and clarify concepts from their notes, providing personalized, on-demand guidance. To uphold academic integrity during timed assessments, NexaClass incorporates tab-switch detection that alerts instructors when a student navigates away from the test interface. By integrating collaborative teaching tools with automated assistance and integrity-monitoring features, NexaClass aims to bridge gaps found in traditional e-learning environments and deliver a scalable, engaging solution tailored to programming courses.

Keywords: AI in Education; Virtual Classroom; Code Editor; Tab-switch Detection; Online Assessment; EdTech.

How to Cite: Y. Divya Vani; P. Pujitha; M. Suma Sree; B. Sai Pradeep; K. Manoj Vamsi; P. Jashuva (2026) NexaClass: An AI-Powered Virtual Classroom for Interactive Programming Education. *International Journal of Innovative Science and Research Technology*, 11(3), 2116-2120. <https://doi.org/10.38124/ijisrt/26mar1276>

I. INTRODUCTION

In recent years, digital learning has become a vital part of education, especially after the rise of online teaching platforms. However, most existing platforms only focus on delivering content rather than creating an interactive and engaging learning environment. Students often face difficulties in clarifying doubts, practicing programming, or maintaining concentration during assessments. On the other hand, faculty members struggle to monitor student activities and ensure fairness in online evaluations.

To address these challenges, our team has developed NexaClass, an AI-powered virtual classroom platform that aims to make online education more interactive, intelligent, and student-friendly. NexaClass allows faculty to conduct quizzes, coding tasks, and slip tests in real time while students can practice programming within a built-in code editor and maintain digital notes for new concepts they learn.

The most innovative aspect of NexaClass is its AI-driven chatbot, which assists students in debugging code and answering their queries based on their notes. This makes the learning process more personalized and reduces dependency on continuous teacher availability. To maintain academic honesty during tests, NexaClass also integrates a tab switch detection feature that identifies

when a student navigates away from the test window, thereby ensuring fair evaluation. Overall, NexaClass bridges the gap between traditional classrooms and digital education. It combines real-time interactivity, artificial intelligence, and security mechanisms into a single unified platform, helping both students and teachers experience a smooth and efficient online learning process.

➤ Contributions

The NexaClass project was developed as a team effort where each member contributed to different parts of the system's design and implementation. The main contributions of our work are summarized as follows:

- Integration of AI Assistance
- Built-in Code Editor
- Tab Switch Detection for Integrity
- Digital Notes Management
- Comprehensive Faculty Dashboard

II. LITERATURE REVIEW

In recent years, researchers and educators have explored several ways to improve the effectiveness of online learning platforms. Many studies have focused on creating virtual classrooms that replicate traditional teaching environments while introducing new forms of interactivity and automation.

Md. Mahmudul Hasan and Md. Zobair Khondaker Rion (2015) proposed and implemented a web-based virtual classroom that aimed to make remote teaching more accessible and measurable. Their results showed that virtual classrooms significantly improved accessibility and learning outcomes, especially for students from remote areas.

Michalis Xenos (2018) discussed how existing online classroom platforms could overcome the limitations of traditional teaching methods. He analysed common features such as live video, chat, and content sharing, concluding that many advanced features are underutilized due to poor adoption or lack of user awareness.

S. K. Sharma and P. Kumar (2021) designed an AI-integrated virtual classroom that provided real-time quizzes, chatbot-based question answering, and adaptive learning paths. Their experimental results showed higher student engagement and improved learning outcomes when AI was used to support personalized learning. Lasha Labadze et al. (2023) conducted a systematic review on the role of AI chatbots in education. They found that chatbots help reduce teacher workload and provide immediate assistance to students, although challenges such as accuracy and trustworthiness still remain.

From these studies, it is clear that while virtual classrooms have evolved rapidly, there is still a need for systems that combine intelligent assistance, code practice environments, and academic integrity features. Our project, NexaClass, builds upon these existing ideas by integrating an AI chatbot for real-time doubt solving, a built-in code editor for programming practice, and tab-switch detection for secure assessments. This combination addresses both engagement and ethical issues in online education

➤ Existing Systems

Most of the currently available virtual classroom platforms such as Google Classroom, Moodle, and Microsoft Teams primarily focus on content delivery, file sharing, and communication. While they provide a convenient medium for conducting online classes, they lack intelligent features that promote personalized and interactive learning. In existing systems, students can attend lectures and submit assignments, but they rarely get real-time coding practice or automated assistance when facing programming errors. Faculty members also find it difficult to evaluate students' practical coding skills directly within these platforms. As a result, online assessments often depend on manual evaluation or third-party tools. Another major drawback of existing systems is the absence of an integrated note-taking and learning support feature. Students have to use separate applications for writing notes, practicing code, and clarifying doubts. This creates a fragmented learning experience and leads to lower engagement. Therefore, while existing virtual classrooms serve as basic teaching tools, they fail to provide a complete, intelligent, and secure learning environment. These limitations motivated the development of NexaClass, a platform that addresses all these gaps through AI assistance, real-time code execution, and integrity monitoring.

➤ Proposed Systems

The proposed system, NexaClass, is an AI-powered virtual classroom platform designed to overcome the limitations of existing online learning systems. It provides an all-in-one environment where faculty and students can interact in real time through quizzes, coding tasks, digital notes, and AI-driven learning support. The main objective of NexaClass is to enhance engagement, simplify online assessments, and provide personalized assistance through artificial intelligence. In NexaClass, faculty members can create and conduct quizzes, coding exercises, and slip tests directly from their dashboards. These assessments are automatically stored in the database and can be accessed by students after logging into their accounts. Students, in turn, can write, compile, and execute code using the built-in code editor integrated into the platform. This feature allows them to practice and test their knowledge without switching to any external application. A key component of the proposed system is the AI chatbot, which assists students in debugging code and clarifying doubts based on their stored notes. This creates a personalized learning experience similar to having a private tutor available at any time. The chatbot is integrated using local AI models to ensure data privacy and real-time performance without relying on external APIs. To maintain fairness during online tests, NexaClass incorporates a tab-switch detection mechanism that identifies when students navigate away from the exam tab. This helps instructors monitor student activity and prevent malpractice during assessments. The entire system is built using a Spring Boot backend with MySQL as the database. The frontend is developed using React.js, and the application is deployed in a Docker container for better scalability and performance. Security is ensured through JWT-based authentication and role-based access control, allowing separate interfaces for students and faculty. By combining these modules, NexaClass provides a seamless, interactive, and secure digital classroom experience. It bridges the gap between traditional learning and modern e-learning by blending AI assistance, code execution, and academic integrity into one unified platform

➤ Objectives

The main objectives of the NexaClass project are to enhance the quality and effectiveness of online education through intelligent features and real-time interactivity. The system is designed with the following specific goals.

- To create an interactive virtual classroom platform that enables real-time communication and engagement between faculty and students.
- To integrate a built-in code editor that allows students to write, compile, and execute programs directly within the platform, promoting hands-on learning.
- To provide AI-based learning assistance through a chatbot that helps students debug code and clarify concepts based on their notes.
- To ensure academic integrity during assessments by detecting tab switching and preventing unfair practices.
- To enable students to maintain organized digital notes, improving knowledge retention and accessibility of learning materials.

- To develop a secure and scalable system using modern technologies such as Spring Boot, React.js, Docker, and MySQL.
- To bridge the gap between traditional classrooms and online education by creating a unified platform that supports both teaching and practical learning.

➤ *System Architecture*

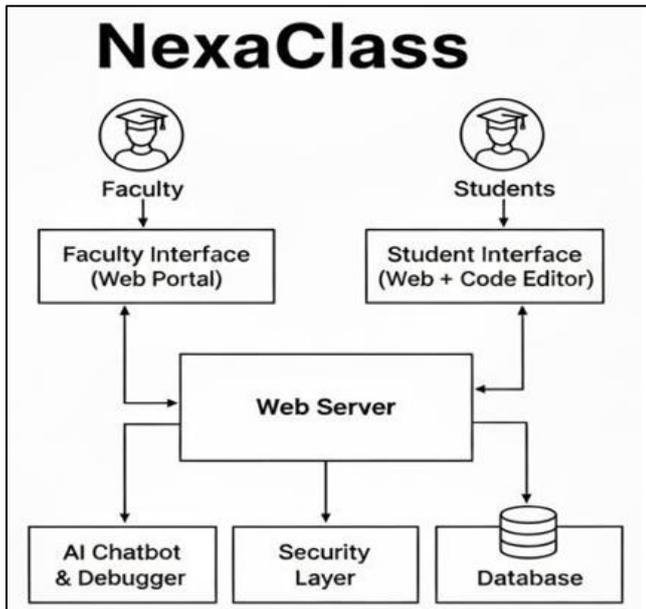


Fig 1 NexaClass

The architecture of NexaClass is designed using a client-server model that connects faculty, students, and the AI components through a secure backend. The system follows a modular structure, where each component handles a specific function to ensure scalability and maintainability.

At the top level, there are two main user interfaces: The Faculty Interface and the Student Interface.

- The Faculty Interface allows teachers to create quizzes, coding exercises, and slip tests, as well as monitor student performance.
- The Student Interface enables students to log in, access assessments, write and execute code, chat with the AI assistant, and maintain digital notes.

All requests from the frontend are processed through the Spring Boot backend, which acts as the core of the system. It manages authentication, data handling, and communication between the client and database. The backend uses JWT (JSON Web Token) for secure authentication and role-based access management.

The AI Chatbot is integrated as a separate module using local AI models via Ollama, ensuring that doubt resolution and debugging support work even without an internet connection to external APIs. The chatbot interacts with both the student interface and the database to analyze code and provide relevant feedback.

The Database Layer, built using MySQL, stores user credentials, quiz details, code submissions, and digital notes. The system also includes a Tab Switch Detection mechanism at the frontend using the Page Visibility API, which sends alerts to the backend if a student switches tabs during an assessment. To ensure reliable deployment, the entire application is containerized using Docker, making it easier to manage across different environments. The architecture ensures high performance, security, and smooth interaction between all module.

III. METHODOLOGY

The development of NexaClass followed a structured methodology that combines software engineering principles with AI-based techniques to ensure scalability, accuracy, and user interactivity. The system is composed of multiple modules, each employing specific algorithms and technologies

By combining Spring Boot, React.js, JWT security, NLP-based chatbot intelligence, and Docker sandboxing, the methodology ensures that NexaClass is secure, scalable, and efficient

➤ *User Authentication Module:*

- *Technique Used:*
JWT (JSON Web Token) Authentication

- *Description:*

The authentication process is handled using Spring Security with JWT tokens. When a user (faculty or student) logs in, the system verifies the credentials using BCrypt hashing and generates a signed JWT token.

This token is used to authenticate future requests, ensuring stateless and secure communication between client and server.

- *Algorithm Steps:*

- ✓ User submits login credentials.
- ✓ Password is hashed and verified using BCrypt.
- ✓ If valid, a JWT token is generated and sent to the frontend.
- ✓ All subsequent requests are validated using this token

➤ *AI Chatbot Module:*

- *Technique Used:*
Natural Language Processing (NLP) with Local LLM (via Ollama)

- *Description:*

The chatbot processes student queries related to programming or notes. It uses token-based text embedding and context retrieval to understand the query and provide relevant solutions or debugging help.

This local AI model ensures low latency and high privacy since no data is shared externally.

- *Algorithm Overview:*

- ✓ User query is captured from chat interface.
- ✓ Text is pre-processed (tokenization and lowercasing).
- ✓ Query is compared with note keywords or code context.
- ✓ The AI model generates a natural-language response or debugging suggestion.

- *Code Execution Module:*

- *Technique Used:*

Sandboxed Code Execution (Multi-language Support)

- *Description:*

The built-in code editor allows students to write and execute programs securely. Each submission is executed in a Docker- isolated environment to prevent malicious operations.

- *Algorithm Steps:*

- ✓ Code submitted by the student is sent to backend API.
- ✓ Backend forwards it to a Docker container based on selected language.
- ✓ Code is compiled and executed within container limits (time, memory).
- ✓ Output or error is captured and sent back to the frontend.

- *Tab Switch Detection Module:*

- *Technique Used:*

Page Visibility API

- *Description:*

The Page Visibility API is used to monitor when a student switches browser tabs during an assessment. This helps maintain academic integrity.

- *Logic:*

- ✓ The system listens for visibility state changes in the browser.
- ✓ If a tab switch event is detected, an alert is sent to the backend.
- ✓ The event is logged, and the instructor can take necessary action.

- *Quiz and Assessment Module:*

- *Technique Used:*

MVC (Model–View– Controller) Design Pattern

- *Description:*

Faculty create quizzes and coding tasks that are stored in the database and served dynamically to students.

Results are automatically calculated and stored for performance tracking.

- *Algorithm Flow:*

- ✓ Faculty creates a quiz → stored in MySQL.
- ✓ Students attempt the quiz → responses submitted to backend.
- ✓ System validates and scores the attempt.
- ✓ Results stored and displayed in faculty dashboard.

- *Notes Management Module:*

- *Technique Used:*

CRUD Operations with Relational Database Model

- *Description:*

Students can create, read, update, and delete (CRUD) notes. Notes are linked to user IDs in the MySQL database for personalized access.

IV. CONCLUSION

In this paper, the NexaClass project successfully demonstrates how artificial intelligence and modern web technologies can be combined to create a smarter and more interactive virtual classroom environment. By integrating features such as an AI-driven chatbot, a built-in code editor, digital note management, and tab- switch detection, the platform addresses the key challenges faced in online education: lack of personalization, limited interactivity, and weak assessment integrity.

The system achieved its main objectives by providing a secure, scalable, and user- friendly solution for both students and faculty. The use of Spring Boot, React.js, MySQL, and Docker ensured smooth performance, while JWT-based authentication added robust security to user interactions. The AI chatbot played a significant role in improving student learning efficiency by offering instant support and guidance growing demands.

Although the platform meets current requirements effectively, there is still scope for further enhancement. In the future, the system can be extended with the following improvements:

- Integration of plagiarism detection for code submissions and notes.
- Addition of machine learning models for automatic student performance prediction.
- Support for real-time video sessions between faculty and students.
- Expansion to mobile application versions for better accessibility.

- *Declaration*

We, the undersigned, hereby declare that the project entitled “NexaClass: An AI- Powered Virtual Classroom for Interactive Programming Education” is a genuine work carried out by our team under the guidance of Mrs. Y. Divya Vani, Department of Computer Science and Engineering, Sri Vasavi Engineering college, Tadepalligudem, India.

REFERENCES

- [1]. D. B., R. Chalon, and X. Zhang, "Virtual assistants (chatbots) as help to teachers in collaborative learning environment," *Proc. Int. Conf. on Interactive Collaborative Learning*, pp. 135–146, Cham: Springer International Publishing, Sept. 2022
- [2]. R. A. Biswas and S. Nandi, "Teaching in virtual classroom: Challenges and opportunities," *Int. J. of Engineering Applied Sciences and Technology*, vol. 5, no. 1, pp. 334–337, 2020.
- [3]. A. Y. Wang and M. H. Newlin, "Online lectures: Benefits for the virtual classroom," *THE Journal (Technological Horizons In Education)*, vol. 29, no. 1, pp. 17–17, 2001
- [4]. L. R. Porter, *Creating the Virtual Classroom: Distance Learning with the Internet*, John Wiley & Sons, 1997.
- [5]. R. M. Palloff and K. Pratt, *Building Online Learning Communities: Effective Strategies for the Virtual Classroom*, John Wiley & Sons, 2007.
- [6]. S. R. Hiltz, "Teaching in a virtual classroom," *Int. J. of Educational Telecommunications*, vol. 1, no. 2, pp. 185–198, 1995.
- [7]. S. P. J. Chiriboga, A. L. T. Burgos, M. M. R. Avila, J. L. C. Chida, K. J. A. Macias, Y. E. C. Morante, and A. P. B. Párraga, "Artificial intelligence and personalized learning in foreign languages: An analysis of chatbots and virtual assistants in education," *Revista Científica de Salud y Desarrollo Humano*, vol. 6, no. 1, pp. 882–905, 2025.
- [8]. H. B. Essel, D. Vlachopoulos, A. Tachie- Menson, E. E. Johnson, and P. K. Baah, "The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education," *Int. J. of Educational Technology in Higher Education*, vol. 19.
- [9]. D. Cohen and I. Sasson, "Online quizzes in a virtual learning environment as a tool for formative assessment," *JOTSE*, vol. 6, no. 3, pp. 188–208, 2016.
- [10]. E. DeSouza and M. Fleming, "A comparison of in-class and online quizzes on student exam performance," *J. of Computing in Higher Education*, vol. 14, no. 2, pp. 121–134, 2003. [10.20944/preprints202401.1817](https://doi.org/10.20944/preprints202401.1817).
- [11]. A. Mutawa and S. Sruthi, "Online proctoring and assessment in distance learning during COVID-19," in *Proc. EDULEARN21 Conf.*, pp. 5887–5895, IATED, 2021.
- [12]. P. A. Novick, J. Lee, S. Wei, E. C. Mundorff, J. R. Santangelo, and T. M. Sonbuchner, "Maximizing academic integrity while minimizing stress in the virtual classroom," *J. of Microbiology & Biology Education*, vol. 23, no. 1, pp. e00292–21, 2022.
- [13]. A. Lee-Post and H. Hapke, "Online learning integrity approaches: Current practices and future solutions," *Online Learning*, vol. 21, no. 1, pp. 135–145, 2017
- [14]. S. Kavitha, J. V. Anchitaalagammai, S. Murali, and S. Annapoorani, "Enhancing learning management systems with accessibility, integrity, and personalization through advanced technologies," in *Proc. 2025 Int. Conf. on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, pp. 1–6, IEEE, Mar. 2025.
- [15]. N. S. Reddy, N. P. Chaitanya, P. S. Varshitha, R. C. Varma, P. K. Reddy, and J. Dandu, "Intelligent chatbot for educational institutions," in *Proc. 2024 7th Int. Conf. on Circuit Power and Computing Technologies (ICCPCT)*, vol. 1, pp. 1337–1343, IEEE, Aug. 2024.