Next-Gen Streamlining of Practical Examinations of Programming Courses with AI-Enhanced Evaluation System

Jayesh Sarwade[a], Afrin Shikalgar[b], Chaitanya Modak[c], Nikita Nagwade[d],
Dnyaneshwari Devmankar[e] and Ganesh Arbad[f]
Department of Information Technology, JSPM'S Rajarshi Shahu College of Engineering, Pune, Maharashtra, India

Keywords: Programming Assessment, Multi-Language Compilers, AI-based Proctoring, Machine Learning Evaluation, Decision Trees, Support Vector Machines, MERN Stack, Browser Lock API, Dialogflow Chatbot, Convolutional Neural Networks (CNN), Exam Integrity, Cheating Prevention, Scikit-Learn, OpenCV, TensorFlow.

Abstract: The final assessment of any course must reflect its goals and contents. An important goal of our foundational programming course is that the students learn a systematic approach for the development of computer programs. Since the programming process itself is a critical learning outcome, it becomes essential to incorporate it into assessments. However, traditional assessment methods e.g. oral exams, written tests, or multiple-choice questions are not well-suited for evaluating the process of programming effectively. Additionally, in educational institutes, teachers often use physical chits to distribute problem statements among students, then students perform them on computers in college labs, which is time-consuming. If essential compilers or development tools are missing on college computers, students resort to online compilers, increasing the risk of internet misuse for copying solutions. There is also a possibility of students using USB drives to share unauthorized code, compromising exam integrity. Therefore, there is a growing need for a fair and standardized evaluation process that accurately assesses students based on their coding abilities, eliminating the risk of cheating or unfair advantages. To address these challenges, this paper proposes a comprehensive software solution to modernize practical exams. The system automates problem statement allocation and integrates multi-language compilers through APIs like JDoodle and HackerRank. It ensures exam integrity by enforcing a full -screen mode using Browser Lock APIs, disabling copy-paste functionality, and adding watermarking for security. The solution includes AI-based chatbots for guidance, powered by Dialogflow, and AI-powered proctoring with OpenCV and TensorFlow, utilizing Convolutional Neural Networks for face detection. For automated and fair evaluation, machine learning models developed with scikit-learn are employed, using algorithms such as Decision Trees and Support Vector Machines. The platform is built on the MERN stack, comprising MongoDB, Express.js, React, and Node.js, to ensure a robust, scalable, and efficient examination process.

1 INTRODUCTION

Practical examinations remain a vital approach to evaluate students' competencies in subjects like programming and data structures. However, traditional methods often lead to inefficiencies, security vulnerabilities, and instances of academic dishonesty. To resolve these issues, this document introduces a comprehensive software solution designed to modernize practical evaluations and deliver measurable results. The proposed system addresses key concerns of standard techniques by enhancing efficiency, safety, and fairness. With a user-friendly interface, educators can easily design, schedule, and manage examinations, specifying crucial details such as date, time, course, division, and problem descriptions. The ability to upload multiple problem statements simultaneously using Excel sheets significantly reduces preparation time, thereby streamlining the entire exam setup process. Students benefit from a robust platform to undertake assessments, review completed evaluations, and manage their profiles. Additional features, like multi-language compilers and an AI-based chatbot, aid in

Sarwade, J., Shikalgar, A., Modak, C., Nagwade, N., Devmankar, D. and Arbad, G.
Next-Gen Streamlining of Practical Examinations of Programming Courses with AI-Enhanced Evaluation System.
DOI: 10.5220/0013590100004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 2*, pages 246-252
ISBN: 978-989-758-763-4

comprehending problem statements and provide instant assistance, reducing misunderstandings. The software's full-screen exam functionality and automatic question generation features foster fairness and lessen opportunities for cheating, thereby minimizing dishonest practices. By automating examination procedures and integrating advanced technologies, this software solution not only saves time but also enhances the overall test experience. It advocates for eco-friendly practices by eliminating paper-based assessments, mitigating grading logistical challenges, and promoting academic integrity. Implementing this software could revolutionize practical evaluations in educational environments, enabling students to effectively demonstrate their skills in a secure and equitable digital setting. This system sets a new standard for practical assessments, indicating significant improvements in exam efficiency, security, and learning outcomes. To ensure the system's reliability and continuous enhancement, AI-driven evaluation methodology is employed to analyze performance metrics, identify improvement areas, and provide actionable insights for refining exam content and processes.

2 ENHANCEMENTS OVER TRADITIONAL PRACTICAL EXAMINATION APPROACHES

2.1 Automation and Efficiency

There is always a lot of instructional work involved in getting ready for the test, including scheduling the time, assigning questions, and creating study notes. Instructors must handle submissions, submit questions, and keep track of student registration, which can be a time-consuming and possibly disruptive procedure. Using this book can be a slow, ineffective, and poorly managed procedure. With the help of the interactive user interface, teachers can quickly design tests by entering information like the date, time, group, and explanation questions. Manual data entry is eliminated by automatically managing student records and uploading problem descriptions in batches. Teachers can concentrate on teaching instead of handling the burden since automation streamlines the testing process, lowers the possibility of human error, and simplifies workload management.

2.2 Dynamic Problem Allocation

Conventional Method: Issue statements are typically provided manually in conventional situations, often with the aid of static templates or predefined lists. Because of this regularity, students could foresee questions or even share information with others in advance. Additionally, because some students might be given easier or more identifiable problems, manual assignment raises the risk of prejudice. To address these problems, the system employs a dynamic problem allocation technique. Students in the same batch are given problems at random to ensure that no two students receive the same set of questions. This randomization reduces the likelihood of cheating because pupils are unable to discuss answers or prepare for specific problems. Given that every student is assigned a distinct task.

2.3 Real-Time Monitoring

Conventional Method: Invigilators have historically kept an eye on student behavior during practical tests. This approach is not infallible, though, as one invigilator would not be able to supervise every student efficiently, particularly in bigger groups or online environments. Due of human oversight’s unpredictability and susceptibility to diversions, dishonest operations may go undetected. The suggested software makes use of technology to provide reliable, non-intrusive real-time monitoring. Throughout the test, features like facial detection confirm the student's identification at regular intervals, guaranteeing that the registered student stays at the workstation. Voice surveillance also picks up on audio irregularities that can point to communication with unapproved parties. Secure testing is ensured by this real-time, AI- enhanced surveillance, which also serves as a disincentive against dishonest behaviour and preserves exam integrity.

2.4 AI-Assisted Evaluation

Conventional Method: Practical test evaluation is usually done by hand, which can be a time-consuming and labour-intensive procedure. Because various evaluators may interpret and assess student submissions differently, manual grading carries the danger of prejudice, inconsistency, and human error. This lack of uniformity may result in unfair evaluation results. The suggested system incorporates automatic evaluation driven by AI, which reliably and efficiently grades student work. The AI is able to assess code according to preset standards, including efficiency, functionality, and conformity to best practices. Subjectivity is removed, guaranteeing that every student is evaluated using the same criteria. Teachers are relieved of the tiresome chore of manual assessment since the automated evaluation drastically cuts down on the amount of time needed for grading, enabling findings to be processed and distributed more quickly.

2.5 Multi-Language Support for Programming

Conventional Method: Students are often restricted to using specific programming languages that are easy for examiners to evaluate, which may make it harder for the students to demonstrate their abilities, especially if they speak languages that are not supported. Students who are more comfortable speaking one language but are forced to use another language may also find it challenging. Proposed System: The software's multilingual capability allows students to write their code in the language they are most comfortable with, allowing them to capitalize on their strengths and resulting in a more accurate assessment of their abilities. The platform supports multiple programming languages, accommodating a wider range of technical competencies and different learning requirements.

2.6 Intelligent Assistance During Exams

Conventional Method: Students may have trouble comprehending problem statements on traditional tests, which could impair their performance. aid is frequently scarce, and invigilators may not be able to aid with complicated queries without inadvertently giving away too much. Proposed System: Students who might require assistance comprehending problem statements can receive real-time support from the integrated AI chatbot. It can provide clarifications or explanations without giving away answers, enabling pupils to move forward with assurance.

3 ALGORITHM AND SOFTWARE

The development of online examination software involved the integration of advanced technologies to guarantee the security, scalability, and efficiency of the system. Python was selected as the primary programming language for its adaptability and extensive library support, enabling seamless backend development, management of intricate logic, data processing, and integration with other system components. The incorporation of Natural Language Processing (NLP) aimed to enhance the precision of automated assessments. NLP algorithms were used to assess textual responses, scrutinize content, grammar, and structure, automating the grading of essay-type questions and identifying plagiarism by cross-referencing submissions with known sources to uphold academic integrity (Nayak, Surabhi, et al. , 2022)

NLP was utilized to automatically evaluate student answers, emphasizing content relevance, grammatical accuracy, and structural quality, ensuring consistent and impartial grading of essay-style questions. Additionally, NLP was employed for plagiarism detection by comparing responses with a repository of previously known sources, ensuring academic honesty and reducing manual intervention in grading, thereby streamlining the process for educators. (Prathyusha, Premasindhu, et al. , 2021)

A robust infrastructure for server-side operations was established using the Django web framework, offering built-in features for user authentication, session management, and security. This framework efficiently managed workflows such as exam creation, delivery, and submission. MySQL was chosen as the database system due to its capability to handle large volumes of structured data, such as exam questions, student records, and submissions, ensuring rapid and efficient data retrieval during the exam process. (Kumar, Choubeya, et al. , 2020)

To facilitate flexible deployment, the software was containerized using Docker, ensuring consistent deployment across different environments by packaging both the software and its dependencies, minimizing configuration issues and enhancing scalability. Git was utilized for version control, enabling developers to track the development process, manage collaboration, and maintain a history of changes, thereby facilitating efficient project management and debugging. (Brkic, Mekterovic, et al. , 2020).

4 METHODOLOGY

4.1 Project Scope and Requirements

Defining Core Modules and Features: The project began with a solid understanding of the basic elements that comprise the practical test software. These comprised teacher and student modules, test design, scheduling, coding interfaces, issue statement presentation, incorporating an AI-powered chatbot, evaluation, and results output. All requirements were accurately documented because each of these components was well defined. Analysis of Skills and Traits: Identifying the specific attributes needed for each module was an essential initial step.

4.2 Conduct Market Research

Examining Current EdTech Solutions: In order to comprehend the current state of educational technology, extensive market research was carried out, with a special emphasis on platforms that provide coding tests, AI-assisted learning, and experiential learning settings. Finding opportunities and gaps where the suggested method may provide clear benefits was the aim.

Analysis of Competitors: The strengths, shortcomings, and areas of differentiation of competitor products were evaluated. For example, current platforms may provide coding interfaces without support for multi-language compilers or automated grading yet lack strong security features. The creation of distinctive characteristics that would distinguish the program was guided by this analysis.

4.3 Develop a Project Plan

Thorough Planning: A thorough project plan was made that included all of the tasks, due dates, and resource allocations. Every stage of the project lifecycle, from original design and development to deployment and maintenance, was addressed in the plan. Better project management and budgeting were made possible by the inclusion of time and cost estimates.

Establishing Milestones and Deliverables: For every stage of development, specific deliverables were set.

4.4 Design the Software Architecture

Scalable and flexible Design: Future updates and expansions are made possible by the software architecture's flexible design. Every module, including the user dashboards, exam management, and problem statement repository, was created as an independent part that could be changed without impacting the system as a whole.

Structural Planning: The architecture contained information about the processing, retrieval, and storage of data.

4.5 Discuss the Implementation

The implementation of the system resulted in a highly efficient and secure platform for conducting practical exams, addressing the inefficiencies of traditional methods. By leveraging the MERN stack, the system provided a robust backend for seamless data processing and a responsive frontend for user interaction. Faculty could effortlessly create, schedule, and manage exams with bulk uploads of problem statements, saving significant time. Students benefited from an intuitive interface, enabling distraction-free coding with secure features like full-screen mode and disabled copy-paste functionality. AI-powered evaluation ensured unbiased and accurate grading, while automated result generation significantly reduced manual effort and errors. The modular architecture enhanced scalability and allowed the system to handle concurrent exams efficiently, achieving faster load times and higher user satisfaction. Feedback-driven refinements improved usability and reliability, ensuring a future-proof solution that streamlined practical examinations and upheld academic integrity.

4.6 Present Use Cases and Scenarios

Examples of Practical Uses for Instructors and Learners: The system was designed with real-world scenarios in mind. For example, educators can quickly create assessments, schedule tests, and monitor ongoing sessions by inserting issue statements as Excel files into their dashboards. After checking in and utilizing a unique code to access their evaluations, students were able to focus on coding tasks in a secure, distraction-free setting.

Common processes and interactions within the system: Provide instances of standard practices such as student registration, exam scheduling, problem statement dissemination, and automatic grading.

These scenarios ensured that every interaction was straightforward and seamless by focusing on the user experience.

4.7 Propose Evaluation Metrics

Creating Success Criteria: Assessment tools were created to determine the software's effectiveness in several domains. These metrics included user satisfaction, dependability, usability, and system performance. Quantitative measures including average load times, error rates, and successful exam completions were integrated with user feedback.

The following key performance indicators (KPIs) were prioritized: scalability (e.g., the ability to handle multiple concurrent exams), usability (e.g., ease of navigation), and security (e.g., successful detection of illegal activities). Other KPIs included the speed and accuracy of AI-assisted tests and the ability to support many languages without performance degradation.

User satisfaction and continuous improvement: Teachers and students took part in surveys and feedback sessions to find out more about user satisfaction.

5 LITERATURE SURVEY

Table 1: Literature Survey.

Sr. No	Publisher and Year	Title	Technologies	Benefits	Drawbacks/ Limitations
1.	IRJET, June 6, 2022	Online Examination System Using AI	Machine Learning, Pattern Matching, Naive algorithm. Linguistic Analysis Algorithm	Malpractice can be detected easily.	Only applicable for theory questions
2.	eLifePress, 2022	Online Exam Portal System Using ML algorithm	Machine Learning, Python	Used by students who are studying for examinations to practice and track their progress.	No API that meets the requirements.
3.	IJIRT, 2021	An Examination System Automation Using NLP	PYTHON, NLP	Immediate Feedback for errors, Provided solutions can be accessed.	Applicable only for multiple choice type of questions.
4.	ICAISC, 2020	A Study on Web based Online Examination System	JS programming language, Ajax technics, Mysql	The system's effectiveness as they can rapidly select the finest reply given, minimizing time spent on each address.	Applicable only for MCQ questions.
5.	IEEE, Nov2020	Automatic Analysis and Evaluation of Student Source Codes	Machine learning, Roslyn API	Smooth review process, automatic assessment of submitted task	Only C# coding language is available no others
6.	IEEE, April 28, 2020	Building Comprehensive Automated Programming Assessment System	Python programming language, Django web framework, MySQL, Docker, Git version control system	Improved scalability, Reduction in grading time, Increased consistency in grading, Enhanced efficiency in assessing programming assignments	Limited adaptability, Lack of real-time feedback capabilities, Static nature of the system, Absence of adaptive learning features.
7.	JETIR, April, 2015	A Survey on Integrated Compiler for Online Examination System	MEAN stack, JVM, Graph mining	Conduct both subject quizzes and lab exams online	Focus only on conducting exams without providing an evaluation mechanism
8.	IEEE, Dec, 2006	Assessing Process and a Product- A Practical Lab Exam for an Introductory Programming Course	Programming languages, web development frameworks, and educational assessment tools	Hands-on learning, Real-world assessment, Collaborative problem-solving, Instructor feedback, Understanding best practices	Time-consuming grading, Limited scalability, Resource-intensive setup, Subjective evaluation

6 PROPOSED MODEL

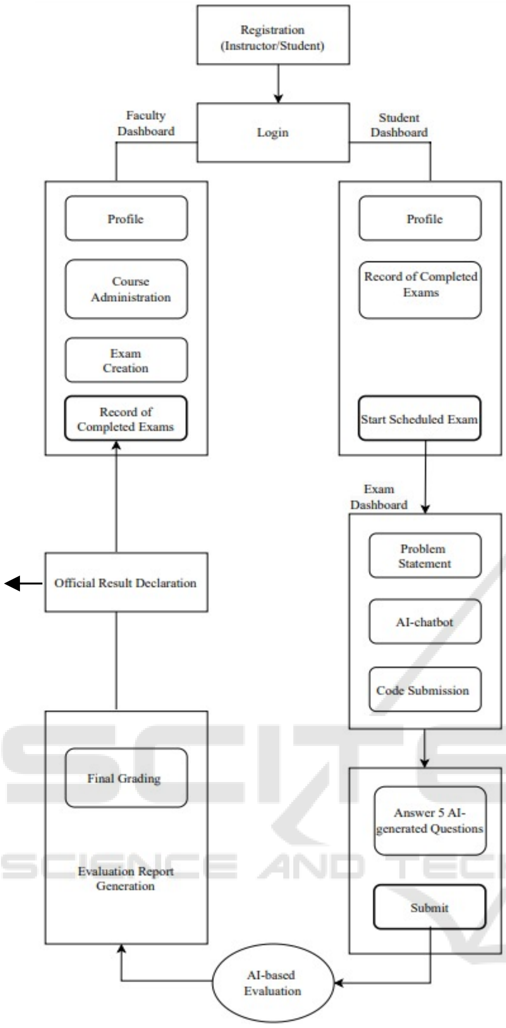

Figure 1: Proposed Model.

6.1 Instructor Dashboard

Instructors must initially register on the platform with a username and password, and they can subsequently log in using the same credentials. The dashboard serves as a centralized platform to manage assignments, personal information, and course administration. Instructors can view and manage their personal information. Course administration tasks, such as adding, viewing, and managing courses, are facilitated by the instructors. Additionally, instructors can create exams by specifying details such as the date, time, course, division, and batch. They can also upload problem statements individually or in bulk via an Excel sheet, and these problem statements are randomly assigned to students on the platform.

6.2 Student Dashboard

Students must also register on the platform with credentials, and they can log in to the system using these credentials. Just like faculty members, students have access to a homepage and profile section. On the dashboard, students can start their exams according to the assigned problem statement. They can execute their problem statement on a multi-language compiler, with an AI-powered chatbot available to assist in understanding the issued statement. After completing the main execution, students need to answer five AI-generated questions related to the problem statement they worked on. After submitting the exam, it is automatically evaluated by AI, ensuring accurate grading, and a detailed exam report is generated.

7 CONCLUSION

The concept presented here presents an intriguing opportunity to fundamentally change how educational institutions assess students: the creation of practical test software. This explanation aims to give a solid basis for the implementation phase by defining the project's characteristics, needs, and scope. The proposed initiative is to improve the process of creating assessments, boost student participation, and speed the administration of tests. It is designed to satisfy the needs of educators as well as learners. The proposal makes use of AI chatbots, issue statements, test design, scheduling, coding interfaces, and outcome evaluation in an effort to provide a comprehensive answer to current issues in education.

8 FUTURE SCOPE

Advanced AI models and offline functionality together enhance the security and accessibility of online exams. AI models leverage computer vision and machine learning to detect suspicious behaviors, such as unusual eye movements or excessive keyboard activity, ensuring a fair and secure environment with real-time alerts or automatic flagging. Simultaneously, offline functionality allows students to download exam materials and work without internet access, with automatic synchronization of completed work once connectivity is restored. This combination ensures exams are both inclusive and secure, catering to diverse regions and technological challenges while maintaining academic integrity.

REFERENCES

“Mr. Vasanth Nayak, Shreyas, Surabhi T, Dhrithi, Divyaraj K M” have implemented, “Online Examination System using AI” in “International Research Journal of Engineering and Technology. (IRJET)”, Vol:09, 2022.

"Siddhant Chaurasia, Govid Suryakant Shinde, Shubham Rasal, Sudhanshu Singh" have implemented, "Online Exam Portal System Using Machine Learning Algorithm. (eLifePress)", Vol:3, 2022.

“T.S.S.K. Prathyusha, K. Premasindhu, T. Ganga Bhavani, V. stuthimahima” have implemented, “Examination system Automation using natural Language Processing. (IJIRT)”, Vol:08, 2021.

“Avinash Kumar, Anjali Choubeya, Ayush Ranjan Behra c, Anil Raj Kisku d, Asha Rabidase, and Beas Bhadraf” have implemented, “A study on the Web-based Online Examination System”, in the International Conference on Recent Trends in AI, IOT, Smart Cities & Applications.

“Adam Pinter, Sandor Szenasi”, have implemented, “The Automatic Analysis and Evaluation of Student Source Codes”, in IEEE International Conference on Computational Intelligence and Informatics,2020.

"Ljiljana Brkic, Igor Mekterovic, Boris Milasinovic, and Mirta Baranovic", have implemented, "Building Comprehensive Automated Programming Assessment System", in University of Zagreb, 10000 Zagreb, Croatia,2020.

“Chetan Kumar, Chinmayee Madhukar”, have implemented, “A Survey on Integrated Compiler for Online Examination System”, Vol: 2, 2015.

“Jens Bennedsen, and Michael E. Caspersen”, have implemented, “Assessing Process and a Product - A Practical Lab Exam for an Introductory Programming Course”, at Aarhus University.

Jayesh Mohanrao Sarwade , Harsh Mathur “Scaling Of Medical Disease Data Classification Based On A Hybrid Model Using Feature Optimization” Webology, Volume 18, No. 6, 2021 Pages 3681-3696.

“Ashwini S, Bhuvan Kumar M S, Chandra Sai D; Sathya V”, have implemented “ProctorXam - Online Exam Proctoring Tool”, in “2024 International Conference on Electronics, Computing, Communication and Control Technology(ICECCC)”, doi:10.1109/ICECCC61767.2024.10593954

J. Sarwade, S. Shetty, A. Bhavsar, M. Mergu and A. Talekar, &quot;Line Following Robot Using Image Processing,&quot; 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1174-1179, doi: 10.1109/ICCMC.2019.8819826.

“Adiy Tweissi, Wael Al Etaiwi, Dalia Al Eisawi”, have implemented, “The Accuracy of AI-Based Automatic Proctoring in Online Exams”, at Princess Sumaya University for Technology (PSUT), Jordan.

Jayesh Mohanrao Sarwade, Harsh Mathur “Performance analysis of symptoms classification of disease using machine learning algorithms” Turkish Journal of Computer and Mathematics Education (TURCOMAT), Volume 11 Issue 3 Pages 2024-2032.

Shudhodhan Bokefode, Jayesh Sarwade, Kishor Sakure, Sandeep Bankar, Surekha Janrao, Rohini Patil “Using a Clustering Algorithm and a Transform Function, Identify Forged Images”, Vol 5, Issue 1, Pages781789 ,DOI:10.47857/irjms.2024.v05i01.0299

Sarwade J. M., Sandip Bankar, Janrao, S., Sakure, K., Patil, R., Bokefode, S., &amp; Kulal, N. (2023). A Novel Approach for Iris Recognition System Using Genetic Algorithm. Journal of Artificial Intelligence and Technology, 4(1),9–17. https://doi.org/10.37965/jait.2023.0434

“Shilpa Manohar Satre, Shankar M. Patil, Vishal Molawade, Tushar Mane, Tanmay Gawand, Aniket Mishra”, have implemented, “Online Exam Proctoring System Based on Artificial Intelligence”, in 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT).

“Aditya Nigam, Rhitvik Pasricha, Tarishi Singh, Prathamesh Churi”, have implemented, “A Systematic Review on AI-based Proctoring Systems: Past, Present and Future”.