AI-Powered Inquiry based Instructional Design for Laboratory Courses in CS/IT Programs: A Case Study

[1]Jeyamala Chandrasekaran, [2]Thiruchadai Pandeeswari S, [3]Pudumalar S, [4]Anitha D
[1,2,3,4] Thiagarajar College of Engineering, Madurai, Tamil Nadu, India
[1] jeyamala@tce.edu
[2] eshwarimsp@tce.edu
[3] spmit@tce.edu
[4] anithad@tce.edu

***Abstract*—These instructions give you guidelines for preparing papers for the Journal of Engineering Education Transformations (JEET). This template is based on the IEEE Transactions template. Use this document as a template if you are using Microsoft Word 6.0 or later. Otherwise, use this document as an instruction set. Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title. The abstract must be a concise yet comprehensive reflection of what is in your article. An abstract needs to be written in a way that captures the attention of a potential reader and gives them enough information to know that the paper is relevant and leaves them wanting to explore the paper in more detail. In particular, the abstract must be self-contained, without abbreviations, footnotes, or references. It should be a microcosm of the full article. The abstract must be between 150–250 words. Be sure that you adhere to these limits; otherwise, you will need to edit your abstract accordingly. The abstract must be written as one paragraph and should not contain displayed mathematical equations or tabular material. The abstract should include three to six different keywords or phrases, as this will help readers to find it. It is important to avoid over-repetition of such phrases as this can result in a page being rejected by search engines. Ensure that your abstract reads well and is grammatically correct.**

***Keywords*—AI based Instructional Design; Inquiry based approach; Code generators; HOTS; Information Security; Laboratory Courses.**

***JEET Category*—Research**

***ICTIEE* Track*:* Technology Enhanced Learning**
ICTIEE Sub-Track: Transforming Education Through Technology: Best Practices and Case Studies

I. INTRODUCTION

Artificial intelligence (AI) is a technology that enables machines to mimic human cognitive abilities, including learning, reasoning, and problem-solving. AI systems can process data, detect patterns, and make decisions with minimal human intervention. In education, AI refers to the application of these technologies to improve teaching and learning. AI tools are used to enhance educational outcomes, tailor learning experiences to individual needs, and simplify administrative tasks.AI-based instructional design utilizes these technologies to develop, implement, and refine educational content and learning experiences. This approach aims to increase the effectiveness and personalization of instruction through advanced AI tools and methods.

Laboratory courses in Computer Science and Information Technology (CSE/IT) have traditionally been a cornerstone of practical education, bridging the gap between theoretical knowledge and real-world application. However, the evolution of coding practices, driven by advancements in technology such as code generators and generative AI, has introduced new challenges to the traditional approach. This section explores the inherent challenges in the conventional conduct of lab courses, emphasizing the impact of modern tools and the necessity for pedagogical innovation.

One of the primary challenges with traditional lab courses is the reduced emphasis on understanding fundamental concepts. Historically, lab exercises were designed to reinforce theoretical knowledge by requiring students to write code from scratch. This process helped students internalize programming principles and problem-solving techniques. However, with the advent of code generators and AI tools, students can now easily generate functional code snippets with minimal understanding. This shift often leads to superficial learning, where students might execute tasks without grasping the underlying algorithms or data structures. he practice of debugging and iterating code helped develop resilience and a deeper comprehension of programming challenges. In contrast, code generators and AI tools can automate problem-solving processes, potentially leading students to bypass critical thinking stages. As a result, students might struggle with problem-solving in real-world

Jeyamala Chandrasekaran
Thiagrajar College of Engineering
jeyamala@tce.edu

scenarios where they cannot rely on automated tools, thus weakening their overall computational thinking skills. The increasing reliance on code generators and generative AI tools can create a dependency that hampers students' ability to code without external assistance.

Assessing students' true coding competence has become more challenging with the use of generative tools. Traditional lab assessments focused on evaluating students' ability to design, implement, and debug code independently. However, when students use code generators, it becomes difficult to gauge their understanding and skills accurately. Educators face the dilemma of distinguishing between students who genuinely comprehend the material and those who rely heavily on automated solutions. This challenge necessitates a revaluation of assessment strategies to ensure that they effectively measure students' proficiency and learning outcomes. The ease of obtaining pre-written code from generative tools or online sources raises concerns about academic integrity. In traditional lab courses, students were required to write code individually, which helped ensure originality and prevent plagiarism. The accessibility of automated solutions can lead to increased instances of students submitting work that is not their own, thereby compromising the integrity of the learning process.

All the above-mentioned challenges including reduced emphasis on understanding fundamentals, diminished problem-solving skills, lack of hands-on experience, dependence on automated solutions, difficulties in assessing true competence, potential for academic dishonesty, limited exposure to real-world challenges, inadequate preparation for collaborative work, and the need to adapt to rapid technological changes have necessitated the innovations in the design of lab courses. These innovations should primarily incorporate strategies that promote deeper learning, critical thinking, and practical skill development while leveraging technological advancements to enhance the educational experience.

II. Literature Review

In this section, a thorough study on prominent existing research works that have explored the use of AI in various aspects of education is carried out. The literature survey is carried out in three AI for Instructional design, AI for assessments and the impact of AI on students' learning outcomes. Abernethy (1998) highlights the benefits of combining traditional inquiry-based learning with AI to enhance problem-solving skills in computer science. Chng (2023) discusses integrating AI into the ADDIE instructional design model, showing that AI can improve the overall learning experience. Tennyson (2024) explores AI's role in project-based learning, noting its limited use in inquiry-based approaches.

Brett et al. (2023) examines the impact of AI tools in introductory programming classes, finding that while these tools support beginners and equalize learning opportunities, they also risk fostering over-dependence and compromising academic integrity. Gianni Fenu (2023) further investigates how AI tools in programming training provide real-time feedback and code suggestions, suggesting a need for adjustments in teaching methods to balance AI assistance with traditional instruction. Valentine et al. (2023) study the use of AI assistants in computer science education, demonstrating their positive impact on students and instructors through personalized feedback and reduced administrative burdens. Adele et al. (2023) explore AI's role in automating and enhancing assessment practices, noting its potential to reduce instructor workload while also raising concerns about accuracy and bias. Collectively, these studies underscore AI's significant potential to transform education, while also acknowledging the challenges and ethical implications that accompany its integration.

Though there is a substantial amount of literature existing on AI for education, there is a minimal amount of research work on how to use artificial intelligence for the conduct of laboratory programs especially in Computer Science/ Information Technology domains. The laboratory course on Information security has been taken as a pilot study. For all the experiments listed in the syllabus, the authors have explored various ways to incorporate the use of generative AI and are presented in section V. The impact of generative AI on achieving higher order thinking skills and student satisfaction is reported in section VI. The strategies to prevent unethical use of AI in completing assessment items are presented in VII. The article is concluded with possible extensions for the proposed educational research

III. Research Questions

As per the discussion on the need of this study, the following research questions have been formulated for the educational study.

RQ1. How can inquiry-based learning be refined to accommodate generative AI in all dimensions of learning from the stage of problem identification to result analysis?
RQ 2: What are the ways by which generative AI tools can be included in the instructional design of laboratory courses?
RQ 3: What is the impact of generative AI in promoting higher order learning outcomes and enhancing engagement and satisfaction level of the learners?
RQ 4: What are the strategies to prevent the unethical use of Generative AI by students in completing assessment items?

IV. RQ1: AI Enabled Inquiry Based Learning

This work attempts to integrate use of AI into Inquiry based learning framework and adapt the same for the conduct of lab courses. A comprehensive hybrid approach to incorporate inquiry-based learning in lab courses of undergraduate engineering courses is presented in the work by Chandrasekaran. J (2024). As per the hybrid approach, for lab courses, each experimentation component in the lab courses must contain four phases viz (1) Identifying open-ended problems, (2) acquiring the necessary background knowledge to solve the problem, (3) designing the logical solution and development of code using the appropriate set of tools and techniques and (4) obtaining experimental results and drawing interpretations from the same. In each of these phases, the instructors need to intervene and provide the learners with certain components that are listed under 'Confirmation' column in fig 1 and the learners are expected to explore, acquire and do certain components that are listed under 'Inquiry' column in fig 1.

Experimentation Component	Confirmation	Inquiry
Problem Identification	Guidelines on Subject area	Application
Background	Relevant Theory concepts	Specific Application related concepts
Design of Solutions	Set of Tools and Technologies	Select and Use of appropriate tools
Results and Discussion	Expectations	Inferences

Fig. 1. Hybrid Inquiry based approach for lab courses

To integrate AI into this Inquiry-based learning framework, it is essential to identify activities that can be conducted with the assistance of AI at each of the phases. While AI can be leveraged to enhance the learning experience and outcomes, it is significant to ensure that the inclusion of AI assistance does not replace the learning itself. Hence careful planning is required to include AI assisted activities into the distinct phases of Inquiry-based approach for lab courses. This provides the context for research with respect to RQ1. At each of the four phases of experimentation, both the instructor and learners could leverage AI assistance to enhance the learning experience and outcome. They are systematically explained in the table I.

TABLE I
AI ASSISTED HYBRID INQUIRY BASED LEARNING

Experimentation Component	Confirmation (Expectation)	Confirmation (AI Assistance)	Inquiry (Expectation)	Inquiry (AI Assistance)
Problem Identification	Exposure to Subject Matter and its ability to solve real world problems	Development of reference reading materials, Generation of realistic case studies	Identification of a suitable real-world problem	Identification of references for market study, existing literature in a particular area, Synthesis of end-user feedback reports
Related Knowledge	Provide necessary knowledge on allied set of technologies, tools and techniques	Identification of scope, gaps in the existing body of reading materials	Gather additional knowledge relevant to the chosen problem and its proposed solution	Filling knowledge gaps through prompt-based exploration
Design and Development of Solutions	Provide exposure on actual implementation through demonstrations	Preparation of user manuals, Troubleshooting guidelines	Design logical solution and implementation of the same using a specific coding language	Understanding error messages and causes of error – Use of AI must strictly be avoided in the following aspects: Generation of pseudocode, Synthesis of logical solutions, Code optimization
Results & Interpretation	Clearly communicate the expectations	Building Rubrics	Obtain experimental results, analyse them, draw interpretations and improve	Identification of suitable and effective metrics, trend analysis, identification of hidden patterns etc.,

V. RQ2: AI ENABLED INSTRUCTIONAL DESIGN – A CASE STUDY

Information security as a laboratory course for undergraduate programs, requires a deep understanding of algorithms, security principles, and mathematical foundations. Traditional lab exercises often limit the scope of experimentation and exploration as students are required to manually write, debug, and execute code, often spending a significant amount of time overcoming basic programming challenges. This process involves configuring systems, running experiments, and manually analyzing results, all without automated support. While this firsthand approach is valuable for learning the fundamentals, it can be time-consuming and may restrict students from fully exploring the concepts due to steep learning curve. Instructors must provide extensive guidance, which can be challenging in large classes, leading to potential delays in students' understanding of key concepts. The focus often shifts from grasping the theoretical and practical aspects of the subject to merely getting the code to work, which can hinder deeper learning and limit the scope of experimentation. Table II presents the refinements done to the list of experiments in the laboratory course on Information security by involving AI tools such as code generators like chatGPT and CoPilot.

VI. RQ3: IMPACT ANALYSIS

From the experimentation, it could be inferred that generative AI when integrated with traditional lab exercises provide the following advantages:

- Code generators automate the initial stages of code creation, allowing students to focus more on understanding and experimenting with cryptographic concepts rather than getting bogged down in syntax and boilerplate code.
- By using code generators, students can produce well-structured and efficient code with minimal effort. This not only boosts their confidence but also allows them to explore more advanced concepts and experiment with different cryptographic techniques.
- The ability to generate and modify code quickly enables students to test various scenarios, analyze the outcomes, and refine their understanding of how different cryptographic methods work in practice.

It is significant to note that the laboratory work done for has been converted into a learning repository and is accessible at

https://crypto-lab-2024.web.app/. The repository includes simulation and cryptanalysis of various crypto algorithms. In the information security laboratory course, after each experiment, the best implementation from the students is selected based on criteria such as accuracy, efficiency, and innovation. These top implementations are then thoroughly tested to ensure they meet the security and functionality standards. Once validated, the selected applications are integrated into a unified system, forming a comprehensive learning repository. This repository serves as a valuable resource, providing students with access to practical examples and solutions for various information security challenges. The integration process involves standardizing the code, documentation, and user interfaces to ensure consistency and ease of use. Each application within the repository is self-explanatory to help students understand the underlying concepts and techniques. The repository is made accessible to all students in the course, enabling them to review, study, and learn from their peers' work.

TABLE II
SUMMARY OF AI ENABLED INSTRUCTIONAL DESIGN FOR COURSE ON INFORMATION SECURITY

S.No	List of Experiments	Conduct of experiments using traditional approach	Conduct of experiments with generative AI tools	Enhancements in Learning Experience
1	Implementation and cryptanalysis of Caesar Cipher	• Coding for encryption and decryption in Caesar Cipher • Coding for Brute force attack and frequency analysis attack	• Customize the encryption/decryption of code for multiple Indian languages. • Analyse the impact of varying file sizes and file formats on encryption time. • Analyse the time taken for crypt analysis for different languages.	• Prevention of plagiarism by ensuring each student works with unique inputs. • Improvement of communication skills through the presentation of findings
2	Implementation and Cryptanalysis of Hill Cipher	• Coding for encryption and decryption • Coding for Known Plaintext- Cipher text attack	• Adopt modular programming and identify the various functions and its prototypes. • Feed the prototypes as prompts to code generators. • Use code generators to identify two or more logics to implement any function. • Analyse the time taken for encryption/decryption with respect to varying key sizes	• In traditional method, the dimension of the input key matrices is limited to 2 or 3. With AI, the codes are expanded for higher order matrices. • Students understand the practical implications of cryptography, such as the trade-offs between security and performance. • Exposure to different problem-solving techniques such as iteration/recursion and select the optimal solution. • Exposure to "Prompt Engineering"
3	Implementation of RSA algorithm	• Coding for key generation, Encryption and Decryption	• Integrate the key generation/ encryption/decryption codes generated into a web application. • Analyse the time taken for different key sizes such as 512 bits and 1024-bit keys • Analyse the time complexity of primality testing methods (Stochastic and deterministic methods) • Develop Codes for factorization and analyse the complexity for small and big integers	• Experience the challenges involved in using asymmetric systems for real time applications. • Practical understanding of complexities of primality testing and factorization • Usage of libraries for mathematical operations involving big integers
4	Implementation of Advanced Encryption Standard	• Coding for key generation, encryption and decryption using AES	• Analyse the time complexity of the codes developed using code generators and crypto libraries in programming languages such as Java and Python	• Exposure to code walk-through and review • Ability to write critical reviews for different implementations.

			• Document the reasons for performance enhancement of crypto libraries. • Customize the code to work for different file formats such as images, audio and video	• Exposure to image, audio and video encryption
5.	Exploring OpenSSL crypto libraries	• Practice OpenSSL commands for encryption/decryption using symmetric/asymmetric systems, hashing, generation, and verification of Digital Signatures • Practice problems given in a common worksheet.	• Every student develops a worksheet that includes questions on algorithms on confidentiality, integrity, authentication and non-repudiation using OpenSSL libraries with answers • Generate answers for the questions using generative AI tools like Co-Pilot and GPT. Write inferences on the answers by humans and generators.	• Understand the power/limitations of generative AI tools. • Exposure on giving feedback to elucidate correct responses from AI tools. • Comparison of performances of AI tools.
6.	Study on Password Based Authentication	• Explore any open-source password cracking tool like John the Ripper, Hashcat etc • Implement dictionary attack on 3 letter passwords.	• Analyse the time taken to generate dictionaries with character sets of varying lengths/ • Analyse the time complexity for dictionary attack for passwords of different lengths. • Generate codes using different data structures for password tables like dictionaries, two dimensional arrays, hash maps etc.	• Appreciate the strength of long and random passwords in real time applications. • Analyse the impact of using of different data structures for the same application. • Explore the ways to enhance the strength of password-based authentication or alternatives to password-based authentication
7	Design and Development of Intrusion Detection Set (IDS)	• Use any classification algorithm for the development of IDS. Compute the precision, recall, F-score, Accuracy.	• Generate Python code snippets for implementing the chosen algorithm using libraries like scikit-learn, TensorFlow, or PyTorch. for the standard Dataset • Collect real time data from the lab using packet capturing tools. Use generative AI tools to perform data processing such as missing values, feature selection, normalization, or encoding categorical variables. • Interpret the results of the model developed using generativeAI, explaining the significance of different metrics, how to improve performance, and potential reasons for underperformance.	• Exploration of Multiple Algorithms in the context of Intrusion Detection • Custom experimentation with modification of features, trying different model architectures, or applying advanced techniques like ensemble learning • Automated Documentation and Explanation. • Understand the challenges in collection and processing of realtime data.
8	Implementation of hashing algorithms like MD5/SHA etc	• Coding for preprocessing all the stages of Hashing	• Integrate the hashing algorithm into a real-world application, such as password storage, digital signatures, or file integrity verification. • Create custom test cases that challenge the robustness of implementation. Learners explain why the test cases are significant and how their implementation handles edge cases. • Peer review and collaboration where students present their implementations and explain	• Applying the algorithm in a practical context helps students understand its relevance and importance in cybersecurity. • Writing and testing with custom cases ensures students understand the algorithm's limits and strengthens their critical thinking skills. • Collaborative learning and peer feedback help students refine their understanding and learn

			their thought processes. Other teams have to critique each other's work, suggest improvements, and discuss alternative approaches.	from each other's perspectives.
9	Firewall configuration	• Configure firewalls by wring rules based on application layer and network layer	• Students set up a security system using traditional methods first, and then use AI tools to optimize or audit the configuration. • Once the firewall is configured, students use AI to monitor and analyse real-time traffic passing through the firewall. The AI can identify unusual patterns, potential threats, or non-compliant traffic and suggest adjustments to the firewall rules.	• Students will learn both the manual process of configuring firewalls and how AI can be used to enhance security by optimizing configurations and detecting issues that might not be obvious to a human administrator. • Exposure to dynamic firewall management, allowing for real-time adjustments and improving the firewall's effectiveness in protecting the network.
10	Reconnaissance using Nmap	• Practice nmap commands for port scanning, service detection and host detection	• Perform a basic Nmap network scan to identify active hosts and open ports and then use AI tools to analyse the scan results, suggesting additional scans or targeting specific hosts based on detected patterns. • Explore the ethical and legal implications of using Nmap and AI in network scanning. Topics such as privacy concerns, the legality of scanning certain networks, and how AI could be used responsibly in cybersecurity shall be presented	• Students will learn the fundamentals of Nmap and how AI can enhance the scanning process by identifying potential areas of interest that may require deeper investigation. • Students not only develop technical skills but also understand the ethical and legal frameworks within which these tools must be used

The repository is also continuously updated with new implementations from subsequent experiments, ensuring it remains a dynamic and evolving educational tool. This approach fosters collaborative learning and encourages students to strive for excellence in their work. The repository serves as evidence for attainment of higher order thinking skills and product development. To analyze the impact of generative AI in student engagement and satisfaction, a 5 point Likert scale questionnaire has been developed as shown in Table III for getting student responses. Around 94 students participated in the survey and the responses received are converted to a numeric value for a maximum value of 5 with the given equation

$$\text{significance} = 5 * \frac{\text{Count of responses with rating 4 and 5}}{Total\ count\ of\ responses}$$

The count of responses with different ratings are presented in the figure 2 and the statistical analysis are presented in Table IV.

The survey indicates a strong preference for generative AI over traditional methods in information security labs. AI significantly enhanced understanding, application of concepts, engagement, and problem-solving, with students finding it particularly effective in fostering exploration and skill development. While the impact on task completion speed was mixed, the overall learning experience was overwhelmingly positive. A large majority of participants support the increased use of AI in future educational settings, highlighting its potential to improve learning outcomes.

TABLE III
IMPACT ANALYSIS SURVEY QUESTIONNAIRE

Q. No	Survey Questionnaire
Q1	How did your understanding of the information security concepts compare between using generative AI and traditional methods?
Q2	How did your understanding of the information security concepts compare between using generative AI and traditional methods?
Q3	How would you rate the speed at which you completed the lab tasks with generative AI versus traditional methods?
Q4	Did generative AI help you solve problems or complete tasks that you found difficult with traditional methods?
Q5	How did the use of generative AI affect your engagement and motivation during the lab compared to traditional methods?
Q6	Were you more or less likely to explore and experiment with different approaches using generative AI compared to traditional methods?
Q7	Do you feel that generative AI contributed more or less to developing your technical skills compared to traditional methods?

Q8	How would you compare your ability to troubleshoot and debug issues using generative AI versus traditional methods?
Q9	How did the overall learning experience with generative AI compare to that of traditional methods in terms of depth and comprehension?
Q10	Considering the benefits and challenges, do you believe generative AI should be used more frequently in future labs?

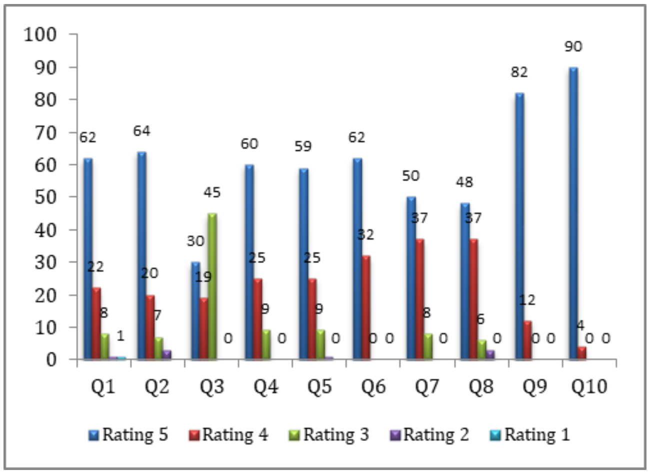

Fig 2 Impact Analysis

TABLE IV
STATISTICAL ANALYSIS OF SURVEY RESPONSES

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
4.5	4.5	2.6	4.5	4.5	5	4.6	4.5	5	5

VII. RQ4: PREVENTING UNETHICAL USE OF AI

In sections IV and V, the systematic inclusion of AI in the Instructional design of laboratory courses was presented and demonstrated with a case study. In both the sections, activities that can be carried out with the help of AI were highlighted. It is advocated to allow the use of AI to carry out certain activities only to enhance the learning outcomes and experience. At the same time, it is essential to be aware of the ill-effects the use of AI may have on the learning outcomes and experience as well. So, it is essential to identify the threats and challenges posed by usage of AI in learning environments and mitigate them using suitable strategies. This section delves into the threats and challenges posed by the unethical use of AI and some suggestions to eliminate them.

From a general teaching-learning perspective, students tend to resort to unethical practices due to some common causes which are listed below:

- *Overwhelming workload and poor time management*: When students are expected to meet unrealistic targets, they often adapt unethical practices to meet the expectations. Also, poor time management leads to a lot of constraints on student's performance
- *Lack of Proper guidance*: When students do not receive clear communication from the instructor about the expectations, guidelines on solution approaches and necessary background information, they look out for easily available unethical means to acquire the same.
- *Lack of awareness of Ethical Implications*: sometimes students are naive to understand the implications of unethical practices concerning intellectual property rights. These are common causes that lead to unethical practices like copying assignments, plagiarism, using unauthorized materials during exams, and fabricating false results etc., These unethical practices are easily checked through common measures like having randomized questions to avoid copying, use of tools to check plagiarism, providing proper guidance and instructions etc., But when a student outsources tasks like writing essays or preparing reports to AI and submit the works authored by AI, it easily slips the quality check. Also, the reports prepared by AI easily passes through the originality checks as well. This is a real threat as a student skip learning, however gets good grade. To restrict the use of AI, where the students need to provide original contribution following mitigation techniques presented in table 5 could be followed.

TABLE V
STATISTICAL ANALYSIS OF SURVEY RESPONSES

Mitigation Technique	Remarks
Include more real-time time-bound assessment items	Instructors may design set of small problems and assign to student teams randomly in class. The problems chosen should be small that they can be solved within 5-10 minutes and solving the problems should require application of specific concepts. Instructor may then validate the solutions provided by students. This type of assessment would be more appropriate for design problems that do not require the use of any external tools. Students may be asked to write pseudocode, find errors and design logical solution.
Creating sub-problems to test original contribution in coding	In CS/IT programmes, it is a common practice to include mini projects in the assessment items. As part of Mini-projects, students would solve slightly bigger problems and build applications. To test their original contribution, smaller sub-problems could be created, and they may be asked to solve the same in labs in proctored mode. For e.g. To test their contribution in a full stack application developed by them, they may be asked to connect database with front-end and perform simple CRUD operations.
Cross-validation of Submissions	When an Open-ended question needs to be included in the assessment and the students need to submit reports in asynchronous mode, students' original contribution could be checked by cross validation process. This process requires each submission to be followed by a short viva-voce. The instructor may use AI to generate two-three simple questions from the submission made and ask the students to answer orally. This will help the instructor assess the original contribution made by the student and the knowledge acquired as well. However, this is not feasible for large classes. In such cases, students may be asked to submit the report as teams. So, in a class of 60 there would be 20 teams approximately to validate.

Building necessary support systems for students	Every student in a class is unique and their needs are different. Avenues like one-to-one tutoring, peer-coaching, tinkering workshops, discussion forums should be made available. This will encourage students to reach out for help when they face difficulties in learning.
Use of tools to detect AI content	AI detection tools could be used to detect the AI content in the student submissions to discourage them from using AI unethically.
Bring in attitudinal changes	Making the students understand the importance of learning is an effective measure to curb unethical use of AI rather than any other measures. When students start valuing learning process, they are less likely to adapt unethical practices.
Focus on building learner-centric environment rather than grade centric environment	This will help the students to learn without pressure. When learners are asked to set the goal for themselves and base their learning journey based on the goals, they are more likely to participate in the learning journey authentically.

Conclusion

The research demonstrates the significant potential of generative AI tools in transforming the instructional design and conduct of laboratory courses in CSE/IT. The integration of AI into lab exercises not only enhances the learning experience by allowing students to focus on critical thinking and advanced concepts but also addresses the inherent limitations of traditional approaches. By automating routine coding tasks, AI enables students to explore a broader range of experiments, thus fostering a deeper understanding of complex subjects such as cryptography and network security. The development of a dynamic learning repository further supports continuous learning and the attainment of higher-order skills. However, the study also highlights the importance of implementing robust strategies to prevent the unethical use of AI, ensuring that students genuinely engage with the material and develop the necessary competencies. Overall, this research underscores the need for ongoing innovation in educational practices to fully harness the benefits of AI while maintaining academic integrity and rigor. The research work can be extended to explore the usage of AI-driven collaborative learning tools that facilitate group work, peer reviews, and collaborative problem-solving, aiming to improve teamwork and communication skills. Another promising research direction involves creating adaptive AI-based assessment strategies that personalize evaluations based on individual learning progress, ensuring a more accurate measure of student understanding. Ethical and responsible AI usage in education is another crucial area, focusing on frameworks that address academic integrity, data privacy, and the psychological impact of AI-assisted learning.

References

S Becker, B. A., Craig, M., Denny, P., Keuning, H., Leinonen, J., Luxton-Reilly, A., Malmi, L., Prather, J., & Quille, K. (2023). Generative AI in Introductory Programming. In Computer Science Curricula 2023, Curricular Practices Volume (p. 25). ACM. https://csed.acm.org/wp-content/uploads/2023/12/Generative-AI-Nov-2023- Version.pdf

Cardona, M., Rodríguez, R. J., & Ishmael, K. (2023). Artificial Intelligence and the Future of Teaching and Learning. https://www2.ed.gov/documents/ai-report/ai- report.pdf

Ch'ng, L. K. (2023). How AI Makes its Mark on Instructional Design. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.8188576

Dwivedi, Y. K., Kshetri, N., Hughes, L., Slade, E. L., Jeyaraj, A., Kar, A. K., Baabdullah, A. M., Koohang, A., Raghavan, V., Ahuja, M., Albanna, H., Albashrawi, M. A., Al-Busaidi, A. S., Balakrishnan, J., Barlette, Y., Basu, S., Bose, I., Brooks, L., Buhalis, D., ... Wright, R. (2023). Opinion Paper: "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice, and policy. International Journal of Information Management, 71, 102642. https://doi.org/10.1016/j.ijinfomgt.2023.102642

Fenu, G., Galici, R., Marras, M., & Reforgiato, D. (2024). Exploring Student Interactions with AI in Programming Training. https://doi.org/10.1145/3631700.3665227

Hwang, G. J., Xie, H., Wah, B. W., & Gašević, D. (2020). Vision, challenges, roles, and research issues of Artificial Intelligence in Education. Computers and Education Artificial Intelligence, 1, 100001. https://doi.org/10.1016/j.caeai.2020.100001

Jianzheng, S., & Xuwei, Z. (2023). Integration of AI with Higher Education Innovation: Reforming Future Educational Directions. International Journal of Science and Research (IJSR), 12(10), 1727–1731. https://doi.org/10.21275/sr231023183401

Kamalov, F., Calonge, D. S., & Gurrib, I. (2023). New Era of Artificial Intelligence in Education: Towards a Sustainable Multifaceted Revolution. Sustainability, 15(16), 12451. https://doi.org/10.3390/su151612451

Kerres, M., & Buntins, K. (2020). Recommender in AI- enhanced Learning: An Assessment from the Perspective of Instructional Design. Open Education Studies, 2(1), 101–111. https://doi.org/10.1515/edu-2020-0119

MoldStud. (2024, January 24). The Future of Artificial Intelligence in Computer Science Education. https://moldstud.com/articles/p-the-future-of- artificial-intelligence-in-computer-science-education

Srinivasan, V. (2022). AI & learning: A preferred future. Computers and Education Artificial Intelligence, 3, 100062. https://doi.org/10.1016/j.caeai.2022.100062

Smolansky, A., Cram, A., Raduescu, C., Zeivots, S., Huber, E., & Kizilcec, R. F. (2023). Educator and Student Perspectives on the Impact of Generative AI on Assessments in Higher Education. https://doi.org/10.1145/3573051.3596191

Yang, W. (2022). Artificial Intelligence education for young children: Why, what, and how in curriculum design and implementation. Computers and Education

Artificial Intelligence, 3, 100061. https://doi.org/10.1016/j.caeai.2022.100061