Student AI Hub

Namit Jagadeesh[1], Prof Neetha[2], Abhishek S Thayyil[3], Karthik Ajay[4], Mahammad Mashood[5]
[1]*Leader, Srinivas Institute of Technology*
[2]*Guide, Srinivas Institute of Technology*
[3,4,5]*Member, Srinivas Institute of Technology*
doi.org/10.64643/IJIRTV12I12-206744-459

***Abstract*—The use of Artificial Intelligence (AI) in the education industry has seen the introduction of numerous systems supporting students with coding, summarization, writing reports, and planning academics. Nevertheless, these systems operate separately, leading to scattered workflow and decreased productivity. The proposed solution of Student AI Hub seems to address this issue by offering a comprehensive multi-model AI platform incorporating multiple education systems under a centrally accessible platform. The platform combines functionality such as code generation, code debugging, summarization of both text and video content, analysis of resumes, and construction of roadmaps, thus making it easier for students to conduct academics more efficiently and easily. Created with the PERN tech stack and incorporated with Large Language Models such as LLaMA, DeepSeek, and Cerebras, the platform offers the capability to use the model freely, with a high level of accuracy and the facility of processing the query in the blink of an eye.**

***Index Terms*—Artificial Intelligence in Education, Code Generation, Video Summarization, Debugging Tools, Multi-Model AI Platform, Student Productivity, PERN Stack, Resume Analysis, Roadmap Generation.**

I. INTRODUCTION

AI has drastically revamped learning environments by directly providing personalized and automated support to students. As academic workloads continue to mount and digital study materials become a necessary prerequisite, students increasingly rely on online tooling that eases learning tasks. However, such tooling is often fragmented across numerous platforms, and therefore students find themselves constantly switching applications-to summarize a lesson, find a bug in code, write a resume, or draft a learning roadmap. This can be highly disruptive to work, creates inefficiencies, and raises the cognitive load of academic activities.

They encounter a series of pain points when debugging errors, comprehending complex technical concepts, processing long videos or documents, and managing study resources. Traditional models of education depend too much on manual learning and feedback delay, and they cannot fit real-time academic dilemmas. Besides, AI tools published online are not integrated; in other words, they are not designed to meet the integrated needs related to learning workflows. Hence, a learner spends more time in managing those tools rather than actually learning.

In the last few years, rapid growth in generative AI models has brought some thrilling functionalities, including natural language comprehension, automated reasoning, and context-aware content generation. Capabilities related to the automation of challenging academic tasks have become possible; code synthesis, for example, error detection, summarizing large-scale content, and even personalized guidance. Nonetheless, the greatest disadvantage in practical academic use is the absence of a common platform integrating these features. Because of a lack of integration, students cannot exploit their combined potential.

Further, the fact that students' learning styles and academic objectives increasingly differ emphasizes the necessity for educational systems to be adaptive and flexible. Whereas some learners need a detailed explanation of tasks step by step, others prefer to read just a summary or learn through examples. A single-model or one-tool approach cannot satisfy these demands. Thus, a multi-model system, where users can select among different AI engines depending on the task's complexity, quality of response, and personal preference, would be of great value in modern education.

Student AI Hub was thereby designed to address these challenges by integrating key academic AI tools into a single, cohesive platform. It reduces fragmentation in

this manner and increases the learning experience by allowing students to conduct various activities without ever having to leave the system. It supports multiple AI models and allows users to select the most suitable model for each task at hand, therefore giving flexibility and better accuracy of the outcomes. This platform integrates AI-driven summarization, coding assistance, debugging, planning, and productivity tools that foster independent learning, effective time management, and smoother academic workflows. The main driver in the development of Student AI Hub is to build a scalable, secure, and user-friendly academic support system, which caters to the rapidly changing needs of higher education. By ensuring seamless communication between the frontend and the backend, maintaining secure AI interactions, and providing consistent output format, the platform can be used as an all-encompassing digital companion by the students. In the long run, it aims to support the facilitation of reduced cognitive overhead, enhanced learning efficiency, and thereby academic success and professional preparedness for the learners using the facility.

II. LITERATURE REVIEW

Generative AI has had a remarkable impact on the education of computing since it offers an automated support system for programming and understanding.
The application of generative tools like code assistants and explanation tools in programming classes has been considered by Denny et al. [1]. According to this particular study, using AI-based scaffolding helps in improving ways of debugging, conceptual understanding, and motivating students, especially in crowded classrooms where teacher interaction with every student separately is not possible. At the same time, it has also been cautioned against possible risks such as plagiarism, dependence on AI-based tools, and biases in programmed codes.

Researchers Chen et al. [2] examine transformer-based large language models, including those reliant on massive code repositories, in terms of their efficiency and effectiveness in code completion, generation, and translation processes. In their analysis, they observe more syntactically correct and passing instances of codes provided by larger language models, thus deeming them utterly useful for automatic code generation. Though effective, however, it is noted by the authors that inconsistency in logic, high computation, and reliance on voluminous datasets are defects of these language models.

Zastudil et al. [3] examines the opinions and attitudes of learners and teachers on the use of AI technology in program learning by using the process of surveys, interviews, and controlled studies. The study shows the increased learner confidence, increased speed during debugging activities, and enhanced learning ability of learners when using AI learning technology. However, it ignores the increased cheating rates and decreased learner efforts reported by the teachers. This study is biased towards self-reported information with a localized population.
A comparison study of various neural network architectures, such as hybrids of CNN and LSTM, and transformer-based methods in the context of educational video summarization is conducted by Eissa et al. [4]. It is found that while transformers outperform other methods because they are capable of handling long-range dependencies, an important aspect for lecture video summaries, multimodal fusion is crucial. However, such high-computational-cost methods with possible existing dataset mismatches as well as misalignments with educational objectives fail to be very useful in an educational setting.

The SpringerOpen Editorial Board [5] investigates whether AI literacy is an indicator of predictive power in utilizing AI learning tools. Results indicate that individuals possessing better AI skills are more capable of effective programming statement writing, interpreting results, and utilizing debugging techniques. This study faces a small sample size and self-evaluation methodologies that fail to establish causes. It is essential to have training programs in AI literacy to guarantee proper utilization of AI technology.
A thorough analysis of AI applications within the higher education sector is carried out by Crompton and Burke [6]. The analysis covers AI-based adaptive learning, automatic assessment, student interventions, and governance. The paper highlights the need for training faculty, ethics, and effective infrastructure for the effective implementation of AI. The paper is full of strategic insights, but it is less technical, and because of advancements in AI, it seems outdated.

Hu et al. in [9] offer transformer models for automated program repair through edit-based learning and test case validation. This work indicates enhanced debugging productivity and robust performance on the benchmark, but also points to very important weaknesses: the production of semantically flawed repairs that are test-passing, bias in the dataset, and learning degradation when explanation was absent. The authors strongly emphasize the importance of explanation in debugging systems and comprehensive test frameworks.

Billur et al. describe a "systematic review of the state of the art" regarding the comparison of the abstractive and extractive methods of video summarization using various paradigms of technical approaches [10]. The paper provides a categorization of methods and the criteria of evaluation, which helps to distinguish the appropriate methods to be used depending upon the type of video chosen for processing. The major deficiencies highlighted in the paper lie in the diversity of processing datasets and the complexity of the transformation technique with a larger computational overhead.

The authors emphasize the necessity of relevance to the learning objective of the teacher, which cannot be accomplished using only the processing metrics of the in a more recent study, Sharma and Saini [11] examined the value of video summarization in online learning environments in regard to its usefulness in mitigating cognitive overload with a focus on revisions in critical concepts. Slide-speech alignment and keyword navigation strategies proved effective with video material. While the findings tend to be more suggestive than concrete, they fail to produce compelling evidence about any potential causal link between video summarization and learning outcomes.

Authors Farooq et al. [20] study the personalized learning roadmaps created by AI, taking into consideration the proficiency and desired goals of the students. The conclusion drawn from the study is that there is an increase in the clarity of goals, self-regulation, and motivation of the learners. However, there is room for enhancement in the accuracy of the roadmap, depending on the accuracy of the data provided, and the problem of interdisciplinary learning is not easily solved.

III. REQUIREMENTS

Defining the requirements of the system is imperative for the successful design, development, and implementation of the Student AI Hub. This section provides the hardware and software requirements necessary for the successful operation of the Student AI Hub platform. The requirements outlined in this section will ensure that the system is able to handle the processing of AI and the integration of different models, while at the same time being stable.

I. Hardware Requirements

The system is expected to run using a moderately powered computer that is capable of running its various tasks. This is achieved by using a computer processing unit such as the i5 or Ryzen 5. The computer is also required to have a memory of at least 8 GB, which is vital for the smooth running of both the frontend and backend processes. The computer is also required to have storage capacity measured at about 20 GB, which is necessary for the storage of all software dependencies. The computer is also required to have a stable internet connection.

II. Software Requirements

The system is developed using the PERN stack, which comprises PostgreSQL as the database management system, Express and Node.js as the backend framework, and React as the graphical user interface. The system can run efficiently in a Windows, Linux, or macOS setup. Various libraries such as Axios, pdf-parse, and mammoth handle secure data transfer and processing. Integration with the AI models utilizes OpenRouter, LLaMA, DeepSeek through Ollama, and Cerebras, which provide varied and correct responses. The system is developed using Visual Studio Code and Git as the version control system.

IV. METHODOLOGY

Rather, the Student AI Hub approach is centered on developing an efficient and effective workflow system that encompasses user interaction, data preprocessing, multiple model AI processing, and result display. Indeed, the proposed process ensures that all forms of educational tasks, ranging from code generation, debugging, summarization, and planning, among others, are undertaken in an effective and error-free

manner. This discussion is expected to outline the varying operations involved in the system from input acquisition to final output delivery.

I. Input and Validation

Beginning with user input-submission via the frontend interface, students are able to input text, source code, documents, or video URLs or task-related queries based on the chosen tool. In fact, it is the responsibility of the frontend interface to validate whether or not the input is in an acceptable format corresponding to the chosen tool or functionality. This saves unnecessary work being done at the backend by rejecting all requests that are inappropriate and are hence irrelevant at that stage.

II. Preprocessing

Once the inputs are validated, the data is sent to the backend for preprocessing. This helps the inputs to be aptly prepared for interaction with the AI models. Text inputs like PDF and DOCX files have their texts extracted and normalized for eliminating any irrelevant formatting information.

Video inputs have their texts extracted from the speeches by utilizing the speech-to-text functionality, thereby making it easy for summarization. The source code input is sanitized by eliminating any irrelevant characters while retaining the syntax and structural information. This preprocessing is an essential task for the improvement of the quality of the AI output.

III. AI Model Integration

The Student AI Hub utilizes an AI model integration layer for managing interactions with multiple AI models, such as LLaMA, DeepSeek, and Cerebras. According to the choice made by the user or possibly the system configuration option, the queries are processed through the desired model with appropriately designed prompts for addressing the targeted task.

The AI model integration layer addresses functionalities like login credentials, data formatting for transmitting queries, fetching results, and fallback provisions in the event of API errors for enhanced performance and all-around flexibility as multiple AI models are utilized for addressing different queries from the students for gaining optimal results.

IV. Output Processing

Once the result is generated from the AI model, the backend system is responsible for processing the result to make it more coherent, consistent, and usable. The process includes refining the AI-generated content by eliminating unnecessary information, structuring the response into a logical format, and code output into an easily readable format with correct indentations. Summarized information is trimmed to focus on essential highlights, and learning roadmap and resume analysis is arranged in a consistent sequence. This enables AI-generated solutions not only to be correct but also easily understandable.

V. Frontend Rendering

This processed final output is handed over to the frontend part, and the display process takes place using an elegant and interactive interface. This frontend part uses proper formatting like syntax coloring for codes, title formatting for summaries, and roadmap and analysis formatting to display the final processed output. These interactive functionalities enable the users to analyze or copy the generated content easily for further use. With the main focus on readability and intuitive interaction, the frontend rendering process further improves the usability experience and helps the students easily understand the generated content from the AI component.

V. DESIGN

Modularity, easy scalability, and smooth interaction among system components are the core bases of the design in the Student AI Hub, including a wide range of academic tools driven by AI. In this architecture, there is effective communication between the user interface, backend business logic processing, database storage, and third-party AI model suppliers. The layered approach makes the design flexible, maintainable, and easily expandable for the future while consistently assuring performance and user friendliness.

I. Architecture

The architecture of the Student AI Hub system is a PERN layered system consisting of a three-tier framework, including the frontend, backend, and database. The frontend, designed with React, is a responsive platform through which students can

access a range of learning tools. The backend, designed with Node.js and Express, is a system processing platform that performs functions like input validation, preprocessing, communication with models, and output processing. The database, based on PostgreSQL, is supportive of basic student information like stored roadmaps, timer information, and basic student profiles. The system is designed in this way to allow for safe input handling, scalability, and system maintenance.

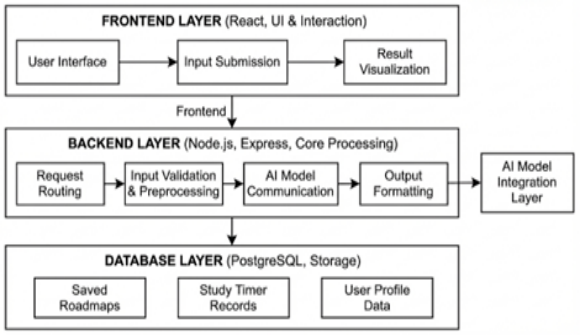

Fig 1: System Architecture

II. Data Flow

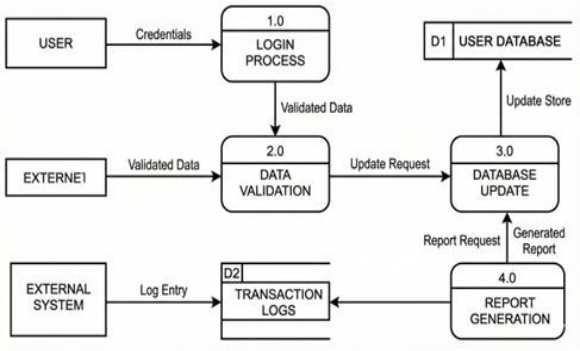

Fig 2: Data Flow Diagram

The flow of data in Student AI Hub is an organized process that guarantees smooth functionality when handling user requests. This begins when the user inputs their request via the frontend. This input is transmitted to the backend for validation and preprocessing according to the selected tool. This processed input is then directed to the AI model integration layer, which communicates with the selected AI tool in order to produce a response.

This process continues until an AI model returns an output, and it is then perfected and formatted by the backend before sending it back to the user via the frontend.

III. Module Design

The Student AI Hub is made up of autonomous yet interrelated modules, each capable of handling a particular academic task according to a standard interaction process. The code generation module is responsible for the translation of user descriptions into code snippets capable of being compiled according to the desired language. The code debugging module is designed to review submitted source code with the objective of finding any faults or defects and providing appropriate explanations.

The text and video summarization modules enable the compression of long pieces of information into their most abbreviated versions, which can be easily understood. Finally, resume analysis modules review resumes based on their format, skills, and relevance while the roadmap development module is responsible for developing learning paths depending on user goals and levels. Time management modules such as the study timer and to-do list enable students to manage time properly.

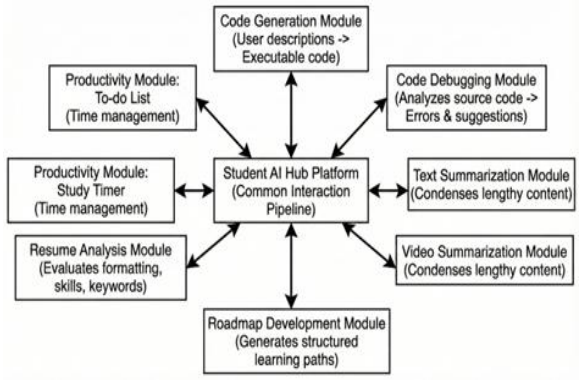

Fig 3: Module Design Diagram

VI. RESULTS

The correctness of the functionality, efficiency of performance, quality of user interaction, and stability of the system were assessed for the Student AI Hub platform. Particular evaluation concentrated on how well the integrated AI tools supported academic tasks like code generation, debugging, summarization, and planning.

In order to reach the design objectives and ensure the system works under real academic usage scenarios reliably, both functional and non-functional testing approaches were considered.

I. Functional Performance of the System
The functional efficacy of the Student AI Hub was also tested by using each of the integrated functionalities for different scenarios in academics. The code generation functionality of the module was able to generate code with the correct syntactic and logical structure with reference to the input prompt given by the user.

The code debugging functionality of the module was able to debug code with syntax and logical errors and also provide apt explanation to the user about the errors in the code. The functionalities of the summarization module, whether it be the text or video summaries, are able to summarize long pieces of academics into a concise summary with the required meaning intact.

II. Quality and Accuracy of AI-Generated Outputs
The performance of the AI-generated outputs had been measured based on the relevance, understandability, and academic utility. The coded outputs followed the conventional programming practices and resulted in accurate outputs for the majority of the cases. The summarized content included the essential elements and provided valuable information for immediate revision and understanding.
The debugging responses provided detailed reasoning and not just corrections for enhanced learning based on the concept. Various AI models were available for comparing the responses, further improving the quality of the outputs and the level of satisfaction for the users.

III. User Interface and System Responsiveness
The user interface of the Student AI Hub was tested for usability, navigation, and responsiveness. The interface was kept quite simple and organized in a manner that allowed smooth transitions for the user between different applications.

The input fields, upload components, and output sections of the applications were well-organized, and all tasks were performed quite efficiently. The applications were quite responsive on all devices and sizes of screens, from desktop to mobile browsers. The average time taken to generate responses for any AI-based task was three to five seconds.

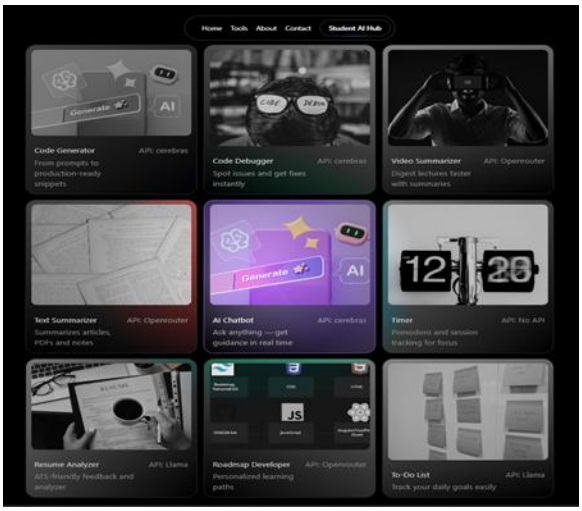

Fig 4: Tools Page

IV. System Stability and Reliability
System robustness has been tested under continuous usage and error-prone situations to check reliability. The system performed well on Student AI Hub with improper data inputs, network disconnection, and short-term failures in the AI API and did not halt or crash. Error messages were also conveyed to the users when an issue arose to provide transparency and retain user trust and confidence. Testing under stress conditions also indicated stable performance of the backend and had not impacted system response time.

VII. CONCLUSION AND FUTURE SCOPE

The Student AI Hub project offers an integrated AI platform that offers all the necessary learning features through the use of AI, including the generation of codes, debugging, text to video summary, roadmap development, resume review, and productivity features all in one location. This platform diminishes the need to use multiple isolated applications, thus performing functions in a more continuous and effective manner. This platform is more flexible and delivers high-quality results due to the application of the multi-model concept in AI, which allows users to choose the best model to use in the various tasks. This platform is scalable, reliable, and responsive due to its use of the PERN tech stack.

Future enhancements would be centered on increasing the platform's versatility and learning capabilities. Future enhancements would include enhancements

such as a learning app for support on the move, a group learning capability, and analytics for monitoring learning progression and making offers accordingly. Other enhancements would be voice interaction capability, multimodal input capability such as handwritten content and pictures, and offline execution of the AI model for improved accessibility and privacy of information. Future enhancements would allow the Student AI Hub platform to become a much more intelligent learning support system.

REFERENCES

[1] P. Denny *et al*., "Generative AI in computing education," *ACM Transactions on Computing Education*, vol. 23, no. 2, pp. 1–18, 2023.

[2] M. Chen et al., "Large language models for code generation," in Advances in Neural Information Processing Systems, vol. 34, pp. 1–12, 2021.

[3] C. Zastudil et al., "Student and instructor perspectives on generative AI," IEEE Transactions on Education, vol. 66, no. 4, pp. 412–420, 2023.

[4] M. H. Eissa et al., "Comparison of neural video summarization techniques," IEEE Access, vol. 13, pp. 20151–20165, 2025.

[5] SpringerOpen Editorial Board, "AI competence as a predictive factor for learning," International Journal of Educational Technology in Higher Education, vol. 21, no. 5, pp. 1–15, 2024.

[6] H. Crompton and D. Burke, "State of AI in higher education," Educational Technology Research and Development, vol. 70, no. 6, pp. 2763–2781, 2022.

[7] IJAIED Editorial Board, "Integrating AI into programming education," International Journal of Artificial Intelligence in Education, vol. 35, no. 1, pp. 25–41, 2025.

[8] IJME Research Group, "AI-based code suggestions in learning," International Journal of Mathematics and Education, vol. 18, no. 3, pp. 134–148, 2025.

[9] Q. Hu et al., "Deep learning models for code debugging," IEEE Transactions on Software Engineering, vol. 48, no. 12, pp. 5010–5024, 2022.

[10] D. D. Billur et al., "Comparative analysis of video summarization techniques," Multimedia Tools and Applications, vol. 82, no. 7, pp. 10345–10363, 2023.

[11] P. Sharma and R. Saini, "Video summarization in online learning," Education and Information Technologies, vol. 27, no. 4, pp. 5379–5395, 2022.

[12] E. Apostolidis et al., "Reinforcement learning for video summarization," IEEE Transactions on Multimedia, vol. 24, pp. 1150–1162, 2022.

[13] ACM UbiComp Editorial Team, "Lightweight educational video summarization," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 6, no. 4, pp. 1–18, 2022.

[14] S. Al Faraby et al., "Neural question generation in education," IEEE Transactions on Learning Technologies, vol. 17, no. 2, pp. 198–211, 2024.

[15] MDPI Psychology Editorial Board, "AI usage and student creativity," Psychology, vol. 14, no. 6, pp. 824–839, 2023.

[16] D. Clark and R. Martinez, "Personalized learning with AI tutors," Computers & Education, vol. 196, Art. no. 104709, 2024.

[17] Y. Zhang et al., "Explainable AI for educational code feedback," IEEE Access, vol. 11, pp. 87654–87667, 2023.

[18] J. Wilson and T. Brown, "Chatbot systems for academic support," Education and Information Technologies, vol. 27, no. 2, pp. 2135–2152, 2022.

[19] L. Russo et al., "Multimodal AI for learning content summarization," IEEE Transactions on Multimedia, vol. 26, pp. 3321–3334, 2024.

[20] S. Farooq et al., "Adaptive study roadmaps using AI," Computers *in Human Behavior Reports*, vol. 9, Art. no. 100195, 2024.