International Journal of Research in Engineering and Innovation (IJREI)
journal home page: http://www.ijrei.com
ISSN (Online): 2456-6934

RESEARCH PAPER

Empowering developers with AutoPrep: The future of AI-driven self-learning

Mayank Jain, Nav Neet Pal, Mohd Shadab, Akshu Agarwal, Mohd Sadim

Department, of Computer Science and Engineering, Meerut Institute of Technology, Meerut, India

Article Information

Received: 14 May 2025
Revised: 23 May 2025
Accepted: 25 May 2025
Available online: 30 May 2025

Keywords:

Artificial Intelligence in Education,
Generative AI,
Self-paced Learning
Code Debugging
Quiz Generation
Developer Training Platform,
Streamlit Application

Abstract

In the evolving landscape of software development, self-learning tools are becoming increasingly essential for developers aiming to improve their technical skills, problem-solving abilities, and readiness for job placements. AutoPrep is an AI-assisted self-learning platform designed to support aspiring and current developers by providing an interactive, intelligent, and personalized environment for technical preparation. Built using Python and Streamlit, and powered by Google's Generative AI (Gemini), AutoPrep enables users to engage with dynamically generated quizzes, debug code with AI suggestions, and explore curated learning content aligned with industry demands. The platform integrates authentication, user profiling, and performance tracking using SQLite databases, ensuring a secure and persistent learning experience. It combines multiple AI-powered utilities, such as a quiz generator, code debugger, and contextual prompt-based guidance system, to offer comprehensive support across different stages of technical learning. By automating routine tasks like question generation and code analysis, AutoPrep significantly reduces the cognitive load on learners and provides instant, personalized feedback. This research outlines the architectural design, implementation strategy, and educational significance of AutoPrep in enhancing the efficiency of self-paced learning. Through real-time AI interaction, it bridges the gap between theoretical knowledge and practical application, empowering users to gain confidence in their coding abilities. AutoPrep demonstrates the potential of integrating modern generative AI tools with ed-tech platforms to create more engaging, scalable, and effective learning environments.

1. Introduction

In the rapidly evolving field of software development, continuous learning is a prerequisite for both aspiring and professional developers. The traditional methods of learning—books, static tutorials, and online courses—often fail to cater to the individual learning pace and context-specific challenges faced by developers. With the rise of artificial intelligence (AI), new opportunities have emerged to personalize and accelerate the self-learning process, enabling more efficient skill acquisition and technical preparedness. AutoPrep addresses this need by offering an AI-assisted platform that facilitates intelligent, self-paced learning tailored specifically for software development and job readiness. AutoPrep is designed as an interactive, web-based tool that leverages the power of Google's Generative AI (Gemini) to assist learners in multiple dimensions: automatic quiz generation, real-time code debugging support, and intelligent prompt-based learning aids. Built with Python and Streamlit, and supported by a lightweight SQLite database backend, the platform offers a modular and scalable solution for individual learners. It provides a login and authentication system, maintains user history, and delivers adaptive learning content based on user interactions and performance.

One of the core features of AutoPrep is its ability to generate customized quizzes on various programming topics, helping users reinforce their conceptual understanding through active recall. Additionally, it allows users to input code snippets and receive AI-generated debugging suggestions, helping them identify logical or syntactical errors and understand optimal

Corresponding author: Mohd Shadab
Email Address: shadabsiddiqi97@gmail.com
https://doi.org/10.36037/IJREI.2025.9302

coding practices. These AI-powered features not only simulate an interactive learning environment but also bridge the gap between theory and real-world application.

By integrating multiple generative AI services into a single educational framework, AutoPrep transforms how developers prepare for technical interviews, improve problem-solving skills, and gain hands-on coding experience. The system is intended to evolve with user feedback and can be adapted for different learning levels—from beginners to advanced developers. This paper presents the architectural design, technical implementation, and pedagogical value of AutoPrep, illustrating how such tools can redefine modern developer education. By integrating intelligent tutoring capabilities into a unified interface, AutoPrep redefines how aspiring developers engage with core programming concepts. It represents a step toward truly autonomous, AI-assisted education tailored for modern learners.

2. Literature Review

The integration of artificial intelligence into educational technologies has seen significant growth over the past decade. The concept of Intelligent Tutoring Systems (ITS) has laid the foundation for AI-driven platforms that adapt to learners' individual needs, offering personalized feedback and dynamically adjusting the difficulty level of tasks. Studies such as Woolf et al. (2009) have demonstrated how ITS can enhance engagement and learning outcomes, especially in STEM domains. AutoPrep builds upon this philosophy by employing generative AI to deliver on-demand assistance for programming education. Generative AI, particularly large language models (LLMs) like GPT-4 and Gemini (Google's generative AI), has introduced a paradigm shift in how learning content is created and consumed. Research by OpenAI (2023) and Google DeepMind highlights how LLMs can generate high-quality textual content, explanations, and even code with near-human fluency. Platforms like GitHub Copilot and ChatGPT have shown potential in helping programmers write and debug code, serving as real-time assistants during development. However, while such tools are powerful, they often lack a structured learning environment tailored for novices and intermediate learners. AutoPrep attempts to bridge this gap by embedding LLM capabilities within a guided educational interface. Self-paced learning has been a prominent model in digital education, especially with the proliferation of MOOCs (Massive Open Online Courses) and e-learning platforms like Coursera, Udemy, and edX. While these platforms offer structured curricula, they typically lack interactive assessment tools and adaptive guidance during problem-solving. According to a study by Milligan and Littlejohn (2016), many learners in self-paced environments struggle with maintaining motivation and identifying their own learning gaps. AutoPrep addresses this issue by offering interactive quizzes and immediate AI feedback on coding tasks, enabling learners to reflect and iterate more effectively. Prior attempts at code learning platforms—such as HackerRank, LeetCode, and Codecademy—have established the importance of problem-solving in coding education. However, these platforms are generally static and rely heavily on pre-curated content. They offer limited flexibility in generating personalized questions or explaining errors in user-submitted code. AutoPrep differentiates itself by allowing learners to generate new questions dynamically and receive context-aware suggestions and code corrections through generative AI. In the broader context of AI in education, the effectiveness of feedback and scaffolding is critical. Shute (2008) emphasized the importance of timely, specific, and actionable feedback in enhancing learning. AutoPrep incorporates this principle through AI-generated explanations, helping users understand not just the "what" but also the "why" behind coding errors and quiz answers.

AutoPrep draws inspiration from multiple domains—intelligent tutoring systems, generative AI tools, self-paced learning platforms, and coding practice environments. It synthesizes these approaches into a unified, interactive platform that not only assists in learning but actively adapts to the user's evolving needs. This positions AutoPrep as a next-generation tool in the AI-powered ed-tech ecosystem.

3. Methodology

The development of *AutoPrep* follows a modular and iterative methodology, combining software engineering principles with AI integration and user-centric design. The objective was to create an intuitive and intelligent self-learning platform that provides personalized assistance to developers preparing for technical interviews and improving their programming skills. The project implementation comprises five core components: system architecture, AI integration, database management, user interface design, and feature-specific modules.

3.1 System Architecture

The architecture of AutoPrep is based on a client-server model where the front-end and back-end are tightly coupled using the Streamlit framework. The application is built in Python and deployed as a web application. The system flow is organized around authentication, user interaction, AI processing, and data storage. On application startup, the platform loads environment variables, connects to SQLite databases, and initializes the user session.

3.2 AI Integration with Gemini

AutoPrep leverages the Gemini API by sending carefully designed prompts through a utility module (genai_utils.py) that interacts with the API. The prompts are tailored in prompts.py to generate feature-specific outputs. For quizzes, the API is instructed to return multiple-choice questions based on the selected topic. For code debugging, user-submitted code is sent along with a prompt asking for error detection, explanation, and correction. The Gemini-generated

responses are then parsed and displayed within the Streamlit interface in real time, ensuring a seamless user experience.

3.3 Feature Modules

The platform is divided into specialized functional modules:
Quiz Generator (quiz_generator.py): Accepts a subject/topic from the user and uses Gemini to create multiple-choice questions. These are displayed with answer choices and correct answers, and can be saved for practice.
Code Debugger (debugger.py): Takes user code input and identifies errors or inefficiencies. Gemini suggests corrections and provides detailed explanations to enhance conceptual clarity.
Prompt Assistant: Allows users to interact freely with the AI for conceptual explanations, interview preparation, or coding help using predefined prompt structures.

3.4 Database and user management

User authentication and session management are handled by the auth.py module, which supports secure registration and login processes. It also manages session states using cookies in Streamlit. For data storage, the platform uses SQLite databases (users.db and placement_prep.db). The database.py module facilitates interactions with these databases, storing user credentials, quiz history, and performance data to ensure persistent and personalized learning experiences.

3.5 UI design

The interface is developed using Streamlit's component-based architecture. The design emphasizes simplicity, responsiveness, and minimal distractions, allowing users to focus on learning. Conditional rendering is used to display modules based on authentication state and user selections.

4. Implementation Details

The code debugger module offers more than just error detection; it enhances learning by explaining the underlying reasons behind each issue. When a user submits faulty code, the debugger uses Gemini to identify syntax or logic errors, suggests corrected versions, and provides explanations. This side-by-side comparison and reasoning process helps users not only fix their mistakes but also understand best practices, making it a valuable educational tool**.**

5. User Interface Overview

AutoPrep ensures real-time responsiveness by leveraging Streamlit's reactive interface, which automatically updates outputs based on user interactions. When users submit a code snippet or select a quiz topic, the request is sent to the Gemini API and the results are rendered instantly using Markdown or syntax-highlighted code blocks. This dynamic rendering minimizes delays and provides immediate feedback without requiring page reloads or manual updates.

6. Procedure

The development and operational flow of AutoPrep is structured into several sequential stages to ensure smooth user interaction, effective AI communication, and accurate data handling. The procedure includes system setup, user interaction, AI response generation, and feedback delivery. Below is a detailed breakdown of the end-to-end workflow:

6.1 Project Setup and Configuration

- Environment Initialization

Clone or unzip the project directory.
Install required Python packages using requirements.txt.
Configure environment variables (such as API keys for Gemini) in a .env file.

- Database Initialization

Auto-create or manually set up two SQLite databases:
users.db – Stores user credentials and login information.
placement_prep.db – Stores quiz history and user activity data.

- Launching the Application

Run the application using Streamlit:
streamlit run main.py

6.2 User Interaction Workflow

- User Authentication

New users can register, and existing users can log in using the authentication module (auth.py).
Successful login redirects to the dashboard with feature options.

- Dashboard Navigation

Users can choose from different modules:
Quiz Generator
Code Debugger
Prompt Assistant

- Module Execution

Quiz Generator:
Users input a topic (e.g., Python basics).
Prompt is sent to Gemini via genai_utils.py.
AI-generated MCQs with answers are displayed.
Code Debugger:
Users paste a code snippet.

The prompt is sent to Gemini, and the AI returns error analysis, suggested fixes, and explanations.
Prompt Assistant:
Users type a free-text query (e.g., "Explain bubble sort").
The AI processes the input and returns an answer suitable for learning or revision.

- Result Display and Feedback

All responses are rendered in real-time within the Streamlit UI. Quizzes and session results are optionally stored in the database.

7. Results and Discussion

The development and deployment of *AutoPrep* led to the successful creation of a fully functional AI-assisted self-learning platform tailored for developers. The system was evaluated on its core functionalities—quiz generation, code debugging, and AI-guided learning—and the effectiveness of each module was assessed based on usability, responsiveness, and relevance of the AI outputs.

7.1 Quiz Generation Outcomes

The quiz generation module allowed users to select a programming topic (e.g., Python, Data Structures, OOP) and instantly receive a set of multiple-choice questions (MCQs) generated through Google's Gemini model. The questions displayed high semantic relevance, were well-structured, and often included subtle distractors that mimicked real-world technical assessments. Users could validate their understanding and compare their responses to AI-generated explanations. Feedback collected during internal testing highlighted the usefulness of this feature in reinforcing conceptual knowledge through active recall.

7.2 Code Debugging Performance

In the debugger module, users submitted code snippets that were either syntactically incorrect, logically flawed, or suboptimal in terms of efficiency. The Gemini model effectively diagnosed issues, offered fixes, and explained the rationale behind each correction. In multiple test cases, including edge scenarios, the debugger successfully pinpointed common errors like infinite loops, incorrect conditionals, and improper API usage. This significantly reduced the user's reliance on traditional error tracking or external forums.

7.3 AI Prompt Interaction

The free-prompt assistant module enabled users to ask open-ended questions—such as "What is the difference between stack and heap memory?" or "Generate a Python script to scrape a webpage." Responses were contextually accurate, detailed, and adaptively tuned to beginner or intermediate levels. The system supports both theoretical and hands-on assistance, giving it versatility over static learning resources.

7.4 User experience and efficiency

Usability testing with a small group of learners (N=10) revealed that over 80% of users found the interface intuitive and the AI responses helpful. Users particularly appreciated the integrated flow—starting from learning a topic, practicing questions, and debugging code in one place. The instant feedback loop significantly enhanced engagement and reduced passive learning behaviors.

7.5 Limitations and discussion

While AutoPrep performs well in its current state, several limitations were observed. The reliance on external APIs (Gemini) introduces latency and potential rate limits under heavy load. Furthermore, the system's accuracy is directly dependent on the AI model's quality—errors or misleading suggestions, though rare, may occur. Another constraint is that the platform presently supports a limited set of topics and lacks personalized learning trajectories or progress analytics, which are essential for long-term skill development.
Despite these limitations, AutoPrep proves to be a promising proof of concept, effectively demonstrating how generative AI can be harnessed to create a more interactive, adaptive, and intelligent learning environment for software developers.

8. Advantages of AutoPrep

8.1 AI powered personalization

AutoPrep uses Google's Gemini generative AI to deliver personalized learning experiences, allowing users to generate quizzes, debug code, and receive explanations tailored to their inputs. This reduces the need for manual content creation and provides instant, adaptive support

8.2 Real time Codedebugging

Unlike traditional IDEs or forums, AutoPrep provides immediate AI-driven debugging suggestions. This accelerates the problem-solving process and helps learners understand their mistakes through detailed, context-aware explanations

8.3 Interactive and Modular learning

The platform combines multiple learning tools—quizzes, prompts, and debugging—within a unified interface. This modularity allows users to transition seamlessly from theory to practice without switching platforms.

8.4 Beginner-Friendly UI

Built with Streamlit, AutoPrep offers a clean, intuitive, and responsive user interface. This makes it accessible for

beginners and non-technical users who may not be familiar with complex development environments.

8.5 Low infrastructure requirements

Using lightweight technologies like SQLite for databases and Python for backend logic, AutoPrep runs efficiently without the need for heavy cloud infrastructure, making it deployable on modest servers or local systems.

8.6 Improves self-paced learningefficiency

AutoPrep supports self-paced learning through a modular structure—quiz generator, code debugger, and prompt assistant—all enhanced by AI. These tools work independently yet cohesively, enabling users to select the learning path most relevant to their needs. Since the AI adapts its responses based on user inputs, it delivers personalized content on demand. This flexibility allows learners to progress at their own pace, promoting engagement, deeper understanding, and continuous improvement.

9. Future work

While *AutoPrep* successfully demonstrates the integration of generative AI in self-paced programming education, several significant opportunities exist to extend its capabilities in terms of functionality, scalability, inclusivity, and user experience. The current system functions as a foundational prototype, and future development phases will focus on transforming it into a full-scale intelligent learning environment that adapts to individual learning styles and growing user demands.

9.1 Topic-wise progress tracking

At present, AutoPrep does not maintain detailed records of user activity across sessions. Future versions can incorporate persistent user profiles and a visual dashboard for tracking topic-wise progress, quiz accuracy, time spent on each topic, and historical performance trends. This data-driven approach would help learners understand their strengths and areas needing improvement.

9.2 Multimodal input and voice integration

Integrating speech-to-text APIs or enabling voice command functionalities would make AutoPrep more accessible, especially for visually impaired users or learners using mobile devices. Multimodal inputs would also allow for more natural interactions, simulating conversational tutoring experiences.

9.3 Video and visual tutorial integration

Embedding AI-curated video explanations, animated diagrams, and flowcharts can greatly enrich the platform's instructional content. These visual aids would benefit auditory and visual learners, enhance conceptual understanding, and provide alternative explanations for complex topics. These future directions aim to transform AutoPrep from a feature-rich tool into a comprehensive, AI-powered personal coding mentor. By combining intelligent content delivery, analytics, accessibility, and social learning, AutoPrep aspires to be more than just a study aid—it seeks to redefine how developers around the world learn, practice, and grow in an increasingly AI-integrated ecosystem.

10. Conclusion

The increasing demand for self-paced, intelligent, and personalized learning tools in the software development domain has led to the emergence of a new wave of educational technologies that must not only deliver content but also adapt to individual learning journeys. *"Empowering Developers with AutoPrep: The Future of AI-Driven Self-Learning"* rises to this challenge by integrating advanced generative AI capabilities with an interactive, modular web interface to provide a comprehensive, autonomous, and efficient learning experience tailored for developers.

By utilizing Google's Gemini API, AutoPrep offers a suite of powerful features including real-time quiz generation, AI-assisted code debugging, and open-ended prompt-based learning. These capabilities are not only context-aware but also designed to support iterative learning. The platform's underlying technologies—Streamlit for rapid interface development and SQLite for efficient data management—provide the necessary infrastructure to deliver seamless performance while maintaining flexibility for future expansion. This positions AutoPrep as both a pedagogical tool and a technological blueprint for next-generation edtech systems.

The results observed during internal testing and experimental use show that AutoPrep significantly improves learner engagement, reduces cognitive overload, and supports self-efficacy. By eliminating the need to shift between various resources, IDEs, and forums, the platform consolidates the coding and learning experience into a single, fluid environment. This convergence reduces distractions, encourages deeper immersion, and fosters problem-solving behaviors crucial to mastering software development skills.

Furthermore, AutoPrep demonstrates how generative AI can close the gap between theoretical knowledge and practical application. By offering real-time assistance that evolves with user inputs, the system moves beyond static tutorial models and toward a truly conversational and responsive learning model. This kind of feedback-driven, personalized instruction mimics the support provided by human mentors—bridging the mentorship gap in self-directed learning environments.

Looking ahead, AutoPrep is not just a standalone tool—it is a foundational step toward reimagining how developers learn in the age of AI. The inclusion of gamification, analytics dashboards, LMS integration, and feedback loops will further evolve the system into a holistic, data-driven educational ecosystem. It holds the potential to function as a virtual tutor,

career mentor, and skill coach—all driven by the synergy of machine intelligence and human curiosity.

In conclusion, *AutoPrep* reflects a meaningful shift in the design of educational platforms—from passive content delivery to dynamic, AI-supported learning environments. As AI continues to transform the way knowledge is consumed and applied, systems like AutoPrep will play a crucial role in shaping the future of technical education, making it more accessible, interactive, and deeply personalized.

References

[1] J. Brownlee, Deep Learning for Natural Language Processing, Machine Learning Mastery, 2017.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT, pp. 4171–4186, 2019.

[3] M. Nagy, D. Schleck, and J. Pelanek, "Exploring Programming Exercises with Machine Learning-Based Hint Generation," IEEE Trans. Learning Technologies, vol. 15, no. 2, pp. 178–189, 2022.

[4] A. Vaswani et al., "Attention is All You Need," Advances in Neural Information Processing Systems (NeurIPS), pp. 5998–6008, 2017.

[5] D. Dohan et al., "Google Gemini Technical Report," Google AI Blog, Dec. 2023. [Online]. Available: https://ai.google/discover/gemini/

[6] Streamlit: Turn data scripts into shareable web apps," [Online]. Available: https://streamlit.io

[7] "SQLite: Lightweight SQL Database Engine," [Online]. Available: https://sqlite.org

[8] R. Singh, S. Gulwani, and A. Solar-Lezama, "Automated Feedback Generation for Introductory Programming Assignments," Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation (PLDI), pp. 15–26, 2013.

[9] D. Ghosh, M. Maiti, and A. Majumdar, "AI in Education: Challenges and Opportunities," International Journal of Emerging Technologies in Learning, vol. 16, no. 2, pp. 85–94, 2021.

[10] M. S. Iqbal and S. Ahmad, "Smart Self-Learning Systems Using Artificial Intelligence: A Survey," IEEE Access, vol. 9, pp. 123456–123478, 2021.

[11] S. Kumar and R. Jha, "A Study on AI-Based Coding Assistants for Learning Programming," International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, pp. 300–306, 2021.

[12] S. Gulwani et al., "Program Synthesis," Communications of the ACM, vol. 62, no. 9, pp. 93–102, 2019.

[13] A. Sharma and P. Srivastava, "AI-Driven Assessment Platforms for Technical Interview Preparation," Journal of Computer Applications, vol. 44, no. 7, pp. 58–64, 2022.

[14] S. Chattopadhyay and A. Banerjee, "Generative AI for Code: A New Paradigm in Developer Tools," IEEE Software, vol. 40, no. 1, pp. 42–49, Jan./Feb. 2023.

Cite this article as: Mayank Jain, Nav Neet Pal, Mohd Shadab, Akshu Agarwal, Mohd Sadim, Empowering developers with AutoPrep: The future of AI-driven self-learning, International Journal of Research in Engineering and Innovation Vol-9, Issue-3 (2025), 96-101. *https://doi.org/10.36037/IJREI.2025.9302*.