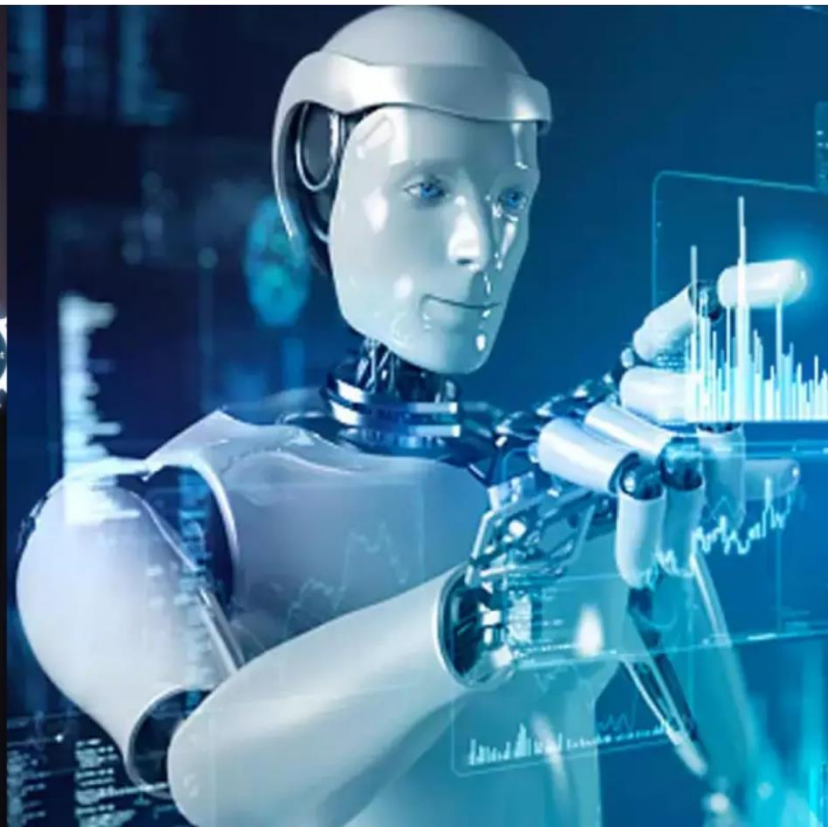
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*

**Impact Factor: 8.699**

**Volume 15, Issue 4, April 2026**

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504105|

# Vibecoding: An AI-Integrated E-Learning Platform for Comprehensive Web Development

**B.Shanmugapriya[1], V Bavadharani[2]**

Assistant Professor, Department of Artificial Intelligence and Data Science, The Kavery Engineering College (Autonomous), Mecheri, Salem,Tamilnadu, India[1].

UG Students, Department of Artificial Intelligence and Data Science, The Kavery Engineering College (Autonomous), Mecheri, Salem,Tamilnadu, India. [2]

**ABSTARCT:** Software development is undergoing a fundamental transformation as vibe coding becomes widespread, with large portions of contemporary codebases now being generated by Artificial Intelligence (AI). The disconnect between rapid adoption and limited conceptual understanding highlights the need for an inquiry into this emerging paradigm. Drawing on an intent perspective and historical analysis, we define vibe coding as a software development paradigm where humans and Generative AI (GenAI) engage in collaborative flow to co-create software artifacts through natural language dialogue, shifting the mediation of developer intent from deterministic instruction to probabilistic inference. By intent mediation, we refer to the fundamental process through which developers translate their conceptual goals into representations that computational systems can execute. Our results show that vibe coding redistributes epistemic labor between humans and machines, shifting expertise from technical implementation toward collaborative orchestration. We identify key opportunities, including democratization, acceleration, and systemic leverage, alongside risks such as black-box codebases, responsibility gaps, and ecosystem bias. We conclude with a research agenda spanning human-, technology-, and organization-centered directions to guide future investigations of this paradigm.

**KEYWORD:** Vibe coding Educational applications Pedagogy Innovation

## I. INTRODUCTION

The field of software engineering is currently undergoing a massive paradigm shift, driven by the integration of Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) into the software development lifecycle (Coello, Alimam & Kouatly, 2024). These models are no longer confined to autocompletion but are emerging as proactive virtual partners in application design, code implementation, refactoring, and testing (Kotsiantis, Verykios & Tzagarakis, 2024). This evolution has very recently given rise to a term “vibe coding” coined by Andrej Karpathy, which describes a software development methodology centered on a developer's intuitive, natural language expression of intent to an LLM (Karpathy, 2025). This practice lowers the entry barrier for code developers, making possible for individuals with minimal formal programming knowledge to create software artifacts. As such, vibe coding signifies a fundamental transition from programming as a formal, syntactic, computer engineering task to a conversational, Human-Computer Interaction (HCI) challenge (Gunatilake et al., 2024). Despite the widespread adoption of LLM-based coding assistants (Porter & Zingaro, 2024), because of the fast pace of development there is currently a significant lack of rigorous, comparative studies analyzing their performance and technical characteristics from a software engineering perspective (Shakya, Vadiee & Khalil, 2025) (Liang, Yang, & Myers, 2024). It remains unclear whether these tools are interchangeable or if they embody different underlying design philosophies that profoundly affect the software development process and the final product (Sergeyuk et al., 2025). The presented research addresses this gap by evaluating the two most commonly used generalpurpose LLM models, OpenAI's GPT-4o and Google's Gemini 2.5 Pro, on vibe coding of three typical frontend development tasks of increasing complexity. The trade-offs in vibe coding between the two models are analyzed and their implications discussed for novice developers, who are increasingly turning to these tools as their primary means of learning and building software, as well as senior developers who may turn to vibe coding for quick prototype building or assessment of new software technologies.

Ined the term “vibe coding” in a tweet (URL 2) as: There’s a new kind of coding I call “vibe coding”, where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It’s possible because the LLMs (e.g.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504105|

Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always,

**OVERVIEW**

With the rapid growth of the digital ecosystem, web development has become a critical skill. Traditional learning platforms often lack personalization and real-time guidance. VibeCoding addresses these challenges by integrating AI-driven tools that support learners throughout their journey.

The platform provides:

- Personalized course recommendations
- AI-based coding assistant
- Interactive development environment
- Performance tracking and analytics

## II. LITERATURE REVIEW

**1. AI in E-Learning Systems**

Recent studies highlight the role of Artificial Intelligence in transforming traditional learning platforms into adaptive systems. AI-driven platforms use machine learning algorithms to analyze user behavior and provide personalized recommendations.

- Researchers have demonstrated that AI can improve learner engagement through adaptive content delivery.
- Intelligent systems track user progress and dynamically adjust the difficulty level of content.

Limitation: Many systems lack real-time interaction and deep coding assistance.

**2. Intelligent Tutoring Systems (ITS)**

Intelligent Tutoring Systems simulate human tutors by providing step-by-step guidance and feedback.

- ITS platforms use Natural Language Processing (NLP) to understand student queries.
- They provide automated hints, explanations, and performance evaluation.

Limitation: Most ITS implementations are domain-specific and not optimized for complex tasks like full-stack web development.

**3. Online Coding Platforms**

Platforms like Codecademy, freeCodeCamp, and HackerRank focus on practical coding skills.

- Provide interactive coding environments
- Offer exercises and project-based learning
- Enable peer learning and community support

**Limitation:**

- Limited personalization
- Lack of AI-driven debugging support
- Feedback is often rule-based rather than intelligent

**4. Adaptive Learning Systems**

Adaptive learning systems personalize education based on learner performance and preferences.

- Use data analytics to identify strengths and weaknesses
- Modify learning paths accordingly

Limitation:

- Often limited to theoretical subjects
- Less integration with real-time coding environments

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504105|

**5. AI-Based Code Assistance Tools**
Modern tools like GitHub Copilot and AI code assistants provide real-time suggestions.
- Improve coding efficiency
- Assist in debugging and code generation

**Limitation:**
- Not specifically designed for structured learning
- May encourage over-reliance without conceptual understanding

**6. Gamification in E-Learning**
Gamification techniques such as badges, leaderboards, and rewards improve learner motivation.
- Encourages continuous engagement
- Enhances retention rates

**Limitation:**
- Does not address individual learning gaps effectively

**7. Cloud-Based Learning Platforms**
Cloud computing enables scalable and accessible learning environments.
- Supports real-time collaboration
- Provides storage and computational resources

**Limitation:**
- Security and privacy concerns
- Dependence on internet connectivity

## III. IMPLEMENTATION

**EXISTING SYSTEM**

**Traditional e-learning platforms**

Traditional online learning systems provide recorded video lectures and basic course materials. However, they often lack personalized learning recommendations and real-time doubt clarification support.

**Basic Learning Management Systems (LMS)**

Existing LMS platforms allow course enrollment, content access, and online tests. However, they have limited AI integration, minimal student performance analytics, and no intelligent chatbot-based assistance.

### 3.1. DISADVANTAGES OF EXISTING SYSTEM

- Absence of instant help while coding
- No intelligent debugging support
- Learners must rely on external resources for problem-solving

### 3.2. PROPOSED SYSTEM

- The proposed system is an AI-integrated e-learning platform designed to provide comprehensive web development training through a personalized and interactive learning environment. Students can register and log in securely to access the platform.
- After logging in, learners can select their preferred web development courses. The system recommends and displays relevant video lessons and structured content to support effective and flexible learning.
- The platform includes an online assessment module where students can attend tests after completing lessons. Results are generated instantly, enabling learners to track their progress and improve their performance.
- An AI-powered chatbot is integrated to clarify doubts in real time and provide continuous learning support. An admin dashboard is available to monitor users, manage courses, and oversee overall system activities efficiently.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504105|

**ADVANTAGES**

- Heavy reliance on video lectures and static content
- Minimal hands-on practice
- Lack of immersive learning experience

**WORKING**

**1. User Registration & Login**

- The user creates an account and logs into the platform
- Basic details and preferences are collected
- User profile is initialized for tracking progress

**2. Skill Assessment**

- The system conducts an initial assessment test
- Questions include HTML, CSS, JavaScript basics
- AI analyzes performance to determine:
  - Skill level (Beginner / Intermediate / Advanced)
  - Strengths and weaknesses

**3. Personalized Learning Path Generation**

- Based on assessment results, AI generates a customized roadmap
- Topics are arranged in a logical sequence
- Difficulty level is adjusted dynamically

**4. Content Delivery**

- Users access:
  - Interactive lessons
  - Video tutorials
  - Coding examples
- Content is adaptive and updates based on user performance

**5. Interactive Coding Environment**

- Built-in code editor allows real-time coding
- Supports multiple languages (HTML, CSS, JS, etc.)
- Code execution happens instantly using a secure sandbox

**6. AI Coding Assistant**

- Acts as a virtual mentor
- Provides:
  - Code suggestions
  - Error detection and fixes
  - Step-by-step explanations
- Uses NLP to understand user queries

## IV. CONCLUSION

The working of VibeCoding ensures a smart, adaptive, and hands-on learning experience by combining AI, real-time coding, and personalized guidance. It transforms traditional learning into an interactive and intelligent process. This study explicates and explores the concept of “vibe coding,” or the use of AI platforms to teach relevant programming skills in a communication context. By integrating flow theory, the research presents a student-centered model for AI-augmented coding education, applying flow traits to achieve desired outcomes. Using student feedback from a mobile application development course, the study demonstrates how AI tools can enhance coding education by providing code samples, explanations, and troubleshooting assistance. The findings suggest that vibe coding can make coding more accessible and useful to a broader audience by applying the model in digital development education and practices.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504105|

## REFERENCES

1. (APA 7th edition style — note that URLs are direct where needed, with access dates suggested for final formatting.)
2. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from https://agilemanifesto.org 3. Business Insider. (2025). Vibe coding startups are pushing AI to generate most of their code. Retrieved March 2025, from https://www.businessinsider.com
4. Chen, M., et al. (2023). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.
5. Csikszentmihalyi, M. (1990). Flow: The psychology of optimal experience. New York: Harper & Row.
6. Dibia, V. (2022). Towards practical human-AI collaboration for rapid prototyping. Proceedings of the CHI Conference on Human Factors in Computing Systems.
7. Hutchins, E. (1995). Cognition in the wild. MIT Press.
8. Karpathy, A. (2025). Vibe coding definition [Social media post].
9. Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. Human Factors, 39(2), 230–253.
10. Ries, E. (2011). The Lean Startup. Crown Business.
11. Sanders, J., et al. (2024). Security vulnerabilities in AI-generated code: A systematic review. Journal of Systems and Software, 195, 111518.
12. Svyatkovskiy, A., et al. (2021). IntelliCode Compose: Code generation using transformer. Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.
13. Vaithilingam, P., et al. (2022). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. CHI Conference on Human Factors in Computing Systems.
14. von Hippel, E. (2005). Democratizing innovation. MIT Press.
15. Zhou, J., et al. (2024). Conversational programming: A comparison with traditional IDE workflows. arXiv preprint arXiv:2404.04567.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details