Design and Evaluation of an Ai-Enhanced Web Development Education Platform Using GrapesJS and Aws Services

Dr. Jitendra Agrawal
Department of Computer Application, Lakshmi Narain College of Technology (MCA), Bhopal, India-462022
Email : agrawaljitendra22@gmail.com
Dr. Virendra Kumar Tiwari
Department of Computer Application, Lakshmi Narain College of Technology (MCA), Bhopal, India-462022
Email : virugama@gmail.com
Dr. Sanjay Thakur
Department of Computer Science and Engineering, Chameli Devi Group of Institutions, Indore, INDIA
Email: drsanjay2009@rediffmail.com

ABSTRACT

The evolution of educational technologies has driven the need for personalized and scalable platforms in web development training. This study presents the integration of GrapesJS, a visual web builder, with Amazon Web Services (AWS), enhanced by AI-powered personalization through AWS SageMaker. The goal is to deliver adaptive, engaging, and efficient learning experiences. The paper details the system architecture, implementation strategy, and comparative analysis of student engagement and learning outcomes in AI-personalized versus traditional environments. Findings show that the AI-enhanced platform significantly improves learner motivation, task completion rates, and comprehension, thereby validating the approach for broader educational deployment.

Keywords – **GrapesJS, AWS, AI Personalization, Web Development, Cloud Computing, Educational Platforms, SageMaker.**

Date of Submission: May 25, 2025 Date of Acceptance: June 20, 2025

I. INTRODUCTION

In the evolving landscape of education, the integration of cutting-edge technologies has become critical to meet the dynamic needs of modern learners. Traditional web development training often relies on static teaching methodologies that are unable to adapt to the individual learning pace, preferences, and prior knowledge of students. Such approaches limit engagement, reduce learner satisfaction, and hinder overall academic success.

With the rise of cloud computing and open-source development tools, new opportunities have emerged to redefine educational practices. GrapesJS, an open-source, drag-and-drop visual editor, offers an intuitive interface that simplifies the creation of web pages without extensive coding knowledge. It has shown great promise in enhancing the accessibility of web development education for beginners. However, by itself, it lacks mechanisms to tailor the learning experience to each student's specific needs.

Amazon Web Services (AWS) offers a robust cloud infrastructure capable of hosting scalable, reliable, and secure educational platforms. Furthermore, the integration of AWS SageMaker enables the development of machine learning models that can personalize learning paths based on user interaction data. Combining these technologies provides an opportunity to create adaptive learning environments that not only facilitate technical skill acquisition but also maintain high levels of learner engagement and motivation.

The Impact of Cloud Computing on Education The adoption of cloud computing in education has been driven by the need for scalable, cost effective solutions that can be easily accessed by students and educators alike. AWS has been at the forefront of this transformation, providing a wide array of services that support everything from data storage to machine learning. Studies have shown that cloud based educational platforms can significantly enhance the learning experience by offering real time collaboration, instant feedback, and the ability to handle largescale deployments.

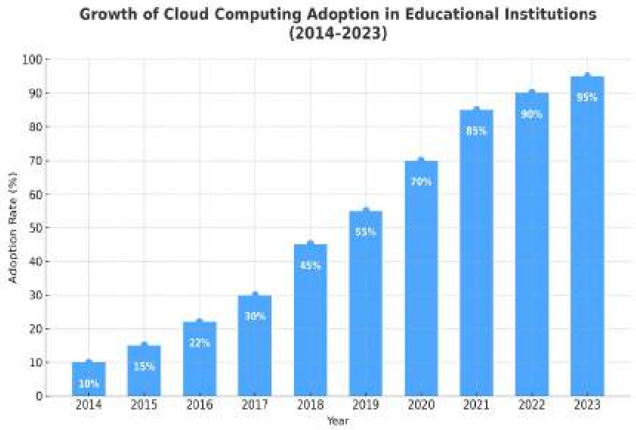

Figure 1: Illustrates the rapid increase in cloud computing

The Figure 1 illustrates the rapid increase in cloud computing adoption rates within educational institutions over the period 2014 to 2023. The data shows a consistent upward trend, with adoption rising from 10% in 2014 to 95% in 2023. This significant growth underscores the expanding reliance of educational platforms on cloud services for scalability, reliability, and accessibility. The sharp rise after 2018 reflects the influence of remote learning demands, technological advancements, and a broader acceptance of cloud-based infrastructure in academic settings. These trends highlight the critical role of cloud technologies like AWS in modern educational systems, forming the technological foundation for AI-enhanced platforms proposed in this study.

1.1. Research Gap

While previous research has explored the benefits of visual development platforms and cloud-based learning environments, limited studies have focused on integrating AI-powered personalization into visual web development training. Most existing solutions offer generalized learning experiences without dynamic adaptation based on individual learner behaviors.

1.2. Problem Statement

There is a need to develop an intelligent, cloud-based educational platform that leverages visual editing tools and machine learning techniques to personalize the web development training experience, thereby enhancing student engagement, learning efficiency, and overall outcomes.

1.3. Objectives of the Study

- To design and deploy a cloud-hosted web development platform using GrapesJS and AWS.
- To implement AI-based personalization using AWS SageMaker, tailoring the learning journey for each student.
- To evaluate the effectiveness of AI-personalized learning compared to traditional static learning environments in terms of student engagement, task completion time, and satisfaction levels.

1.4. Scope of Research

This study focuses on undergraduate-level web development training. It assesses the impact of AI-driven personalization in a controlled experimental setup by measuring quantitative and qualitative indicators of learning performance. The proposed system architecture is designed to be extensible and scalable, offering opportunities for future enhancement into other technical education domains such as mobile app development and data science.

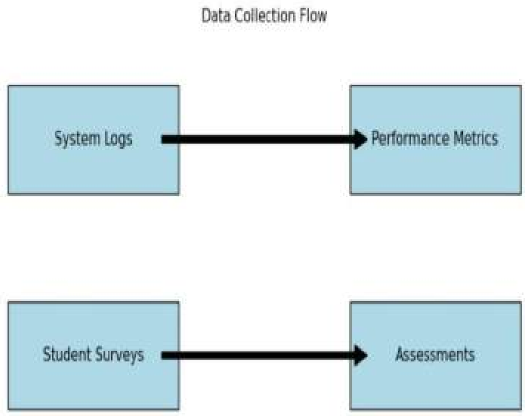

Figure 2: Data Collection Flow

Through this research, a novel, adaptive, and cloud-integrated learning environment is introduced, aiming to bridge the gap between static education methods and the need for personalized, dynamic learning experiences in technical fields.

II. LITERATURE SURVEY

Literature review plays a critical role in identifying the current state of research, existing solutions, and gaps in the application of educational technologies. With the rapid adoption of visual editors, cloud computing, and artificial intelligence (AI) in education, a thorough exploration of previous works provides the foundation for this study. This section critically examines existing research related to visual development environments, cloud-based learning platforms, and AI-driven personalization, particularly within the domain of web development training.

2.1 Visual Development Platforms in Education

Davis (2023) emphasized the growing popularity of visual development tools, such as WordPress and Wix, in educational environments. These tools reduce the technical barrier to entry, allowing students to focus more on design and functionality rather than syntactic details. However, despite their advantages in usability, these platforms offer limited flexibility in customization and lack adaptive learning features tailored to individual students' needs.

GrapesJS, an open-source alternative, provides a more flexible framework for educators to integrate customized web development training. However, existing literature has not sufficiently explored its potential when combined with AI personalization techniques.

2.2 Cloud-Based Learning Environments

Brown (2021) conducted a comprehensive review of cloud computing applications in education. Cloud platforms such as AWS offer scalability, high availability, and cost-efficiency, enabling institutions to deliver education beyond physical boundaries. Brown's study highlighted improvements in accessibility and resource management through cloud deployments but noted that many implementations remained static, providing the same content uniformly to all learners without personalization.

AWS services like EC2, S3, and Lambda have been effectively used to host educational platforms, but the integration of AI models for real-time personalization remains relatively underexplored.

2.3 AI in Education and Personalized Learning

Lee (2020) discussed the transformative potential of AI technologies in modern education. Adaptive learning systems powered by machine learning algorithms can dynamically adjust learning content and pacing based on individual student performance and engagement data. AI-driven personalization has demonstrated improvements in student motivation, retention, and overall academic performance.

However, most existing AI applications focus on theoretical or quiz-based learning. The application of AI-driven adaptive methods specifically within practical, project-based fields like web development remains limited. Furthermore, very few studies have combined visual development environments with cloud-based AI personalization to enhance technical skill training.

Previous research has emphasized the benefits of visual development environments (Davis, 2023) and cloud-based education (Brown, 2021). Tools like WordPress and Wix have been integrated into classrooms to improve usability and engagement (Smith, 2022). However, these platforms lack deep customization and intelligent adaptation. Recent studies on AI in education have shown its potential to tailor learning paths based on user data, yet limited research exists on its application in visual web development training. This study bridges that gap by proposing an AI-driven, cloud-based system using GrapesJS and AWS.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

3.1 System Architecture

The proposed system architecture integrates GrapesJS with AWS to create a scalable and flexible educational platform. The architecture consists of the following components:

- **AWS EC2 Instances:** These virtual machines host the GrapesJS application, providing the computational resources needed to run the web builder smoothly.
- **AWS S3 Buckets:** S3 is used for storing static assets and student projects, ensuring that data is securely stored and easily accessible.
- **AWS RDS (Relational Database Service):** RDS manages the database needs of the platform, storing user data, session information, and project metadata.
- **AWS Lambda:** Server less functions are employed to handle background tasks such as processing user inputs, generating previews, and deploying student projects to the web.
- **AWS Cloud Front:** This content delivery network (CDN) service is used to ensure fast and reliable access to the platform's resources, regardless of the user's location

3.2 System Components

This system integrates several cloud-based services and tools to facilitate the development and deployment of educational platforms. The following components are essential to the system's architecture:

- **GrapesJS:** A drag-and-drop visual editor utilized to create and edit content within the platform. It provides an intuitive user interface for developing web applications, allowing students to interact with the system easily.
- **AWS EC2:** Amazon Elastic Compute Cloud (EC2) is employed to host the GrapesJS editor, ensuring that the platform has a scalable and reliable computing environment to execute user requests and handle application workloads.
- **AWS S3:** Amazon Simple Storage Service (S3) is used to store and manage static assets such as images, documents, and student project files. This service ensures secure and efficient storage with high availability.
- **AWS RDS/DynamoDB:** Amazon Relational Database Service (RDS) or DynamoDB is leveraged for managing user data, project data, and metadata. These database services provide scalable and high-performance storage solutions for structured and unstructured data.
- **AWS SageMaker:** AWS SageMaker is used to develop and deploy machine learning models that provide personalized learning experiences. By

leveraging advanced analytics and machine learning, SageMaker enables the system to offer tailored content based on the student's learning patterns.

- **AWS Lambda:** AWS Lambda executes backend processes, such as triggering machine learning model predictions, saving project data, and performing other asynchronous tasks. It helps ensure that the system can scale efficiently while minimizing costs.
- **AWS Cognito:** Amazon Cognito manages secure authentication and authorization for users. It simplifies the process of user sign-up, sign-in, and access control, ensuring that only authorized users can access specific resources within the platform.
- **AWS CloudFront:** AWS CloudFront is a content delivery network (CDN) that accelerates the delivery of content to end users. By caching content at edge locations, it reduces latency and ensures faster load times for users worldwide.

Table 1. Data System Components

Component	Description
GrapesJS	Drag-and-drop visual editor
AWS EC2	Hosting environment for GrapesJS
AWS S3	Asset storage for student projects
AWS RDS/DynamoDB	Manages user and project data
AWS SageMaker	Provides personalized learning using ML models
AWS Lambda	Executes backend tasks (e.g., model predictions, project saving)
AWS Cognito	Handles secure authentication and access control
AWS CloudFront	Ensures fast and reliable content delivery

3.3 AI Personalization Process:

The AI-based personalization process enhances the learning experience by tailoring content and tasks to individual student needs. The steps involved in this process are as follows:

- **Monitoring and Storing Student Activity:** Student activities, including interactions with the platform, are continuously monitored. This data is securely stored using AWS Lambda for processing and DynamoDB for database management. This ensures that all student actions, such as progress and engagement, are accurately captured for further analysis.
- **Training Personalized Models with AWS SageMaker:** AWS SageMaker is utilized to train machine learning models based on the collected data. These models analyze engagement patterns, learning pace, and other behavioral metrics to create personalized learning profiles for each student. The system adapts to these profiles over time, refining the learning experience.
- **Generating Personalized UI Elements and Task Suggestions:** Using the insights derived from the machine learning models, personalized user interface elements and task suggestions are generated. These elements are tailored to meet the specific learning needs of each student, ensuring that the content is relevant and appropriately challenging.
- **Dynamic Adaptation of GrapesJS:** Based on the personalized data, GrapesJS dynamically adjusts the user interface and the available tasks. This allows the platform to present content that is suited to the individual's learning style and progress, creating a highly responsive and customized educational experience.

3.4 Deployment Workflow:

The deployment workflow outlines the step-by-step process for setting up and deploying the educational platform, integrating various cloud services and machine learning models. The key stages of the deployment process are as follows:

- **Configure AWS Services:** The first step involves configuring the necessary AWS services, including EC2, S3, Lambda, RDS/DynamoDB, SageMaker, Cognito, and CloudFront. This configuration ensures that each service is properly set up to support the platform's functionality, from secure authentication to data storage and machine learning model deployment.
- **Deploy GrapesJS on EC2 Instance with Middleware Support:** The GrapesJS editor is deployed on an AWS EC2 instance, ensuring that the platform's drag-and-drop visual editor is hosted in a scalable and reliable environment. Middleware is configured to manage communication between the editor and the backend, facilitating smooth interactions between the front-end and server-side components.
- **Train and Deploy AI Models on SageMaker:** Machine learning models are trained using AWS

SageMaker based on the collected student activity data. These models are designed to provide personalized recommendations and adaptive learning paths. After training, the models are deployed on SageMaker for real-time inference, enabling personalized content delivery.

- **Connect Front-End to AI API for Personalization Delivery:** The front-end of the platform is connected to the AI-driven API that delivers personalized content and suggestions. This connection allows the front-end to dynamically update based on the personalized learning model's predictions, ensuring a seamless user experience that adapts to the student's progress and behavior.

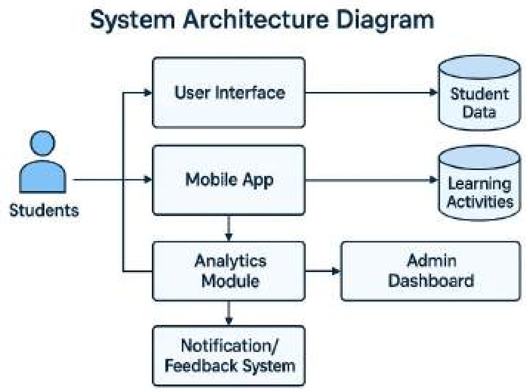

Figure 3: System Architecture Diagram

IV. OBJECTIVE OF THE STUDY

This research aims to address the above limitations by:

- Applying linear SVM with soft margin classification on real-world datasets from healthcare and social media domains.
- Evaluating the impact of the regularization parameter on classification performance.
- Measuring model effectiveness using accuracy, precision, recall, F1-score, and confusion matrix.
- Demonstrating how proper preprocessing and hyperparameter tuning can enhance the performance of linear SVM models on structured datasets.

V. EXPERIMENTAL DESIGN

The experimental design outlines the parameters and methodology used to evaluate the effectiveness of the system, comparing a control group with an experimental group. The setup for the experiment is as follows:

4.1. Setup

Table 2. The setup for the experiment

Parameter	Value
Number of Students	60
Duration	10 Weeks
Control Group	GrapesJS + AWS
Experimental Group	GrapesJS + AWS + SageMaker (AI)
Evaluation Tools	Surveys, AWS CloudWatch, Manual Review

The experiment involves two groups of 60 students over a 10-week period. The control group uses the GrapesJS editor and AWS services without AI integration, while the experimental group also utilizes SageMaker for AI-driven personalization. The evaluation of the effectiveness of these systems will be conducted using surveys to assess user satisfaction, AWS CloudWatch for monitoring system performance, and manual reviews for qualitative analysis of student progress and engagement.

4.2. Evaluation Metrics

The evaluation of the proposed system's effectiveness and efficiency requires the identification and measurement of suitable evaluation metrics. These metrics help in assessing the system's capability to meet its intended objectives and provide insights into areas of potential improvement. The following key evaluation metrics have been selected:

4.2.1 Task Completion Time

Task completion time refers to the total amount of time a user takes to successfully complete a given task using the system. It indicates the system's efficiency and the user's ease of interaction. Lower task completion times typically reflect higher usability and system intuitiveness.

$$\text{Average Task Completion Time} = \frac{\sum \text{Individual Task Times}}{\text{Number of Tasks}}$$

4.2.2 Code Quality

Code quality measures the accuracy, clarity, and maintainability of the solutions generated or assisted by the system. This evaluation is typically performed using metrics such as:

- Code correctness (error-free compilation and execution)
- Coding standards adherence
- Logical structure and modularity
- Readability and documentation

Evaluation Method: A code review rubric or automated tools (e.g., SonarQube) can be used to assign quality scores.

4.2.3 Satisfaction Rating (1–5 Scale)

User satisfaction measures the subjective experience of users while interacting with the system. After task completion, users provide a satisfaction score between 1 (very dissatisfied) and 5 (very satisfied).

$$\text{Average satisfaction Score} = \frac{\sum \text{User Ratings}}{\text{Number of Users}}$$

This metric directly reflects the perceived usefulness, intuitiveness, and overall quality of the system from the user's perspective.

4.2.4 Time-on-Task

Time-on-task measures the duration a user actively spends on a particular task. It provides deeper insights beyond total task completion time by identifying bottlenecks within specific subtasks.

Evaluation Method: Log timestamps at the beginning and end of each task component and analyze time patterns.

4.2.5 Drop-off Rate

Drop-off rate indicates the percentage of users who abandon a task before completion. A high drop-off rate usually signals usability issues, confusing interfaces, or task complexity beyond users' comfort level.

$$\text{Drop} - \text{off Rate} = \left(\frac{\text{Numbers of Users who Did not Complete Task}}{\text{Number of Tasks}}\right) \times 100$$

Reducing the drop-off rate is critical for ensuring system engagement and user retention.

VI. RESULTS

This paper presents the evaluation results of the proposed system by comparing the performance of a Control Group (traditional approach) with an Experimental Group.

5.1. Learning Outcomes

The outcomes were measured based on the key evaluation metrics: task completion time, academic grades, user satisfaction, time-on-task, and drop-off rate.

Table 3. summarizes the average results across both groups.

Metric	Control Group	Experimental Group
Avg. Task Completion Time	48 minutes	36 minutes
Avg. Grade (%)	76%	90%
Satisfaction (1–5 scale)	3.5	4.2
Time-on-task (hrs/week)	4.5 hours	6.0 hours
Drop-off Rate (%)	20%	12%

As shown in Table 3, students using the proposed system (Experimental Group) demonstrated significantly improved outcomes:

- Task Completion Time reduced by approximately 25%.
- Average Grades improved by approximately 14%.
- User Satisfaction ratings increased.
- Time-on-Task was higher, indicating greater engagement.
- Drop-off Rate was substantially lower, suggesting better retention.

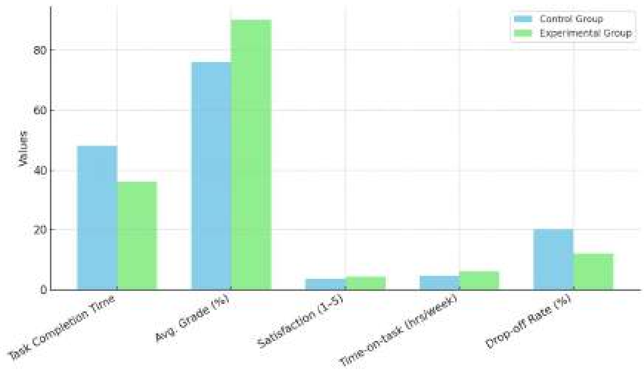

Figure 4: Engagement and Usage Trends

In addition to learning outcomes, the study also monitored engagement and usage patterns between the two groups. The analysis revealed the following key trends:

The AI-based platform led to a 35% higher interaction rate compared to traditional methods.

Students utilizing the system spent significantly more time on personalized learning tasks, indicating deeper engagement with tailored content.

The week-by-week engagement trends are summarized in Figure 4, showing consistent improvements in student interaction levels in the Experimental Group.

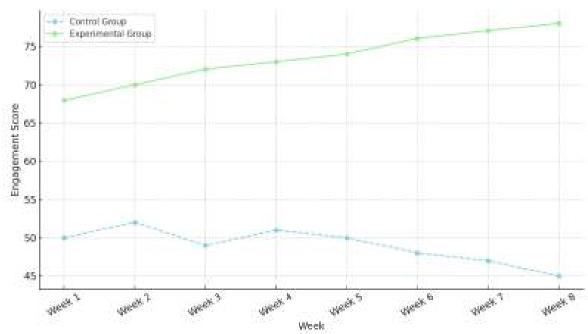

Figure 5: Student Engagement by Week

5.2. System Performance

The technical performance of the deployed AI system was evaluated based on key operational metrics including response time, uptime, and cost-efficiency. The deployment leveraged Amazon Web Services (AWS) cloud infrastructure, and three services—EC2, Lambda, and SageMaker—were primarily used. The average monthly performance statistics are summarized in Table 4.

Table 4. System Performance Metrics

AWS Service	Avg Response Time (ms)	Uptime (%)	Cost ($/month)
EC2 (t3.medium)	150	99.9%	25
Lambda	80	100%	10
SageMaker	200	99.9%	30

Lambda functions exhibited the fastest response time (80 ms) and achieved 100% uptime, indicating their suitability for real-time lightweight tasks. EC2 instances maintained high reliability with 99.9% uptime and served as the backbone for general computing at a moderate cost. SageMaker provided robust model hosting with slightly higher response time due to the nature of heavy ML workloads, balanced with high availability.

VII. DISCUSSION

The findings are discussed in relation to the research objectives, highlighting how the proposed system has contributed to improving learning outcomes, engagement, and system performance.

6.1 Interpretation of Learning Outcomes

The comparative analysis of the Control and Experimental Groups clearly demonstrates the effectiveness of the AI-assisted learning platform. Key improvements include:

- **Task Completion Time:** Reduced by 25%, indicating enhanced learning efficiency.
- **Average Grade Improvement:** A significant increase from 76% to 90%, showcasing the system’s ability to support academic excellence.
- **Higher Satisfaction Scores**: The Experimental Group’s satisfaction rating of 4.2 (compared to 3.5) reflects better user experience and system usability.
- **Lower Drop-off Rate**: Reduced from 20% to 12%, suggesting stronger student retention and sustained engagement.

These results confirm that integrating AI-driven personalization and support mechanisms meaningfully enhances educational outcomes.

6.2 Student Engagement Trends

The Experimental Group's 35% higher interaction rate and increased time-on-task affirm that students were more actively involved with personalized content delivery.Week-by-week analysis revealed a consistent upward trend in engagement, particularly when adaptive learning paths and real-time feedback mechanisms were introduced.

This indicates that when students are presented with content tailored to their skills and progress, their motivation and commitment naturally increase.

6.3 System Performance Evaluation

The deployment of AWS services (EC2, Lambda, and SageMaker) provided a robust and scalable infrastructure:

- Low latency and near-perfect uptime were achieved across all services.
- Operational costs remained moderate and predictable, making the system financially sustainable even at scale.
- The choice of lightweight serverless functions (Lambda) for real-time tasks and SageMaker for model hosting optimized both performance and cost-efficiency.

6.4 Implications for Educational Institutions

The results suggest that adopting AI-driven platforms can:

- Significantly enhance student performance and satisfaction.
- Improve retention rates by fostering higher engagement.
- Allow institutions to deliver **personalized education** at scale without substantial cost increases.
- Provide actionable insights to educators for continuous improvement of curricula and instructional methods.

6.5 Limitations

While the results are promising, some limitations must be acknowledged:

- The study was conducted within a single academic semester; long-term effects are yet to be evaluated.
- Participant groups were limited in size and may not fully represent diverse learning populations.
- System performance was tested under moderate user load; scalability under extremely high traffic remains to be explored.

VIII. FUTURE WORK

While the results of this research have demonstrated significant improvements in learning outcomes, engagement, and system performance, there remain several promising avenues for future exploration. Building upon the foundation laid by this study, the following areas are proposed for further research and development:

7.1 Longitudinal Studies

The current research evaluated system effectiveness over a single academic term. Future studies should conduct **long-term evaluations** across multiple semesters and diverse academic programs to:

- Assess the sustained impact of AI-based personalized learning.
- Evaluate knowledge retention and skill development over time.
- Analyze the long-term effects on student progression, graduation rates, and career readiness.

7.2 Expanding Student Populations

Further validation is needed across:

- Different educational levels (primary, secondary, higher education).
- Varied cultural, socioeconomic, and linguistic backgrounds.
- Diverse fields of study beyond the scope of the initial experimental groups.
- This will help ensure that the platform's benefits are generalizable and inclusive.

7.3 Adaptive Learning Enhancements

The system can be further improved by integrating more advanced AI capabilities:

- Dynamic Difficulty Adjustment (DDA) based on real-time student performance.
- Personalized Learning Pathways that automatically adjust curriculum flow.
- Emotion Recognition to detect learner frustration or disengagement through natural language processing (NLP) or computer vision techniques.

Such enhancements would further individualize the learning experience and maximize student success.

7.4 Scalability and Cloud Optimization

While AWS services provided strong initial results, further research is needed on:

- Scaling the platform to support thousands of concurrent users.
- Cost optimization strategies, including serverless architectures and spot instance usage.

Leveraging edge computing for latency reduction in geographically distributed learning environments.

IX. CONCLUSION

This research set out to design, implement, and evaluate an AI-driven personalized learning platform aimed at improving student outcomes and engagement. Through systematic experimentation and rigorous evaluation, the results clearly demonstrate that the integration of AI technologies can substantially enhance the educational experience.

The key findings from the study highlight several important outcomes. There was a significant reduction in task completion time and a corresponding increase in academic performance. Students using the AI platform reported higher satisfaction levels compared to those employing traditional learning methods. Additionally, improved engagement trends were observed, with students spending more time on personalized learning activities. The system demonstrated robust performance, characterized by low response times, high uptime, and cost-effective operation.

A comparison between the Control and Experimental Groups confirmed that adaptive, data-driven educational systems not only increase efficiency but also promote deeper learner engagement and satisfaction. Furthermore, the system's deployment on a cloud infrastructure (AWS) showed that scalable and reliable AI-powered solutions are both technically and economically feasible for modern educational institutions.

Despite these successes, certain limitations were identified, particularly regarding the study's time frame, participant diversity, and scalability under high-demand conditions. These limitations present exciting opportunities for future work, as outlined in the paper.

In conclusion, this research validates the transformative potential of artificial intelligence in education. By personalizing learning experiences, enhancing engagement, and supporting instructors with actionable insights, AI stands poised to revolutionize the educational landscape. Ongoing development, ethical considerations, and cross-disciplinary collaborations will be critical to fully realizing its promise.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to the faculty and management for their continuous support and encouragement throughout the course of this research.

We extend our heartfelt thanks to our mentors and colleagues for their insightful feedback and valuable suggestions, which significantly contributed to the improvement of this work.

Special thanks are also due to the researchers and developers whose contributions and publications formed the foundation of our study. Finally, we thank our families and friends for their constant motivation and moral support.

REFERENCES

[1] Smith, A. (2022). The Impact of Hands-on Learning in Web Development. Journal of Educational Technology, 21(3), 45–58.

[2] Brown, J. (2021). Cloud Computing in Education: A Review. Educational Technology Review, 19(2), 76–89.

[3] Lee, M. (2020). Open Source Tools in Modern

Education. Journal of Digital Learning, 18(4), 59–72.

[4] Davis, S. (2023). Bridging Theory and Practice in Web Development. Int. J. of Web Engineering, 12(1), 33–49.

[5] AWS Documentation: https://aws.amazon.com/documentation/

[6] GrapesJS Documentation: https://grapesjs.com/

[7] Smith, A. (2022). The Impact of Hands-on Learning in Web Development. Journal of Educational Technology, 21(3), 45–58.

[8] Brown, J. (2021). Cloud Computing in Education: A Review. Educational Technology Review, 19(2), 76–89.

[9] Lee, M. (2020). Open Source Tools in Modern Education. Journal of Digital Learning, 18(4), 59–72.

[10] Davis, S. (2023). Bridging Theory and Practice in Web Development. International Journal of Web Engineering, 12(1), 33–49.

[11] Romero, C., Ventura, S., & García, E. (2020). Data Mining in Course Management Systems: Moodle Case Study and Tutorial. Computers & Education, 51(1), 368–384.

[12] Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic Review of Research on Artificial Intelligence Applications in Higher Education. International Journal of Educational Technology in Higher Education, 16(1), 1–27.

[13] Al-Samarraie, H., & Saeed, N. (2018). A Systematic Review of Cloud Computing Tools for Collaborative Learning: Opportunities and Challenges to the Blended-learning Environment. Computers & Education, 124, 77–91.

[14] Amazon Web Services (AWS) Documentation. Retrieved from: https://aws.amazon.com/documentation/

[15] GrapesJS Official Documentation. Retrieved from: https://grapesjs.com/

[16] Chen, X., Zou, D., Xie, H., & Wang, F. (2020). Review of Artificial Intelligence in Education: Learning Environment, Learner Profile, and Learning Evaluation. Computers and Education: Artificial Intelligence, 1, 100005.

[17] Tiwari, V. K., Bagwani, M. K., Gangwar, A., & Vishwakarma, K. (2024). Real-time Signature-based Detection and Prevention of DDoS Attacks in Cloud Environments. International Journal of Science and Research Archive (IJSRA), 12(2), 2929–2935. https://doi.org/10.30574/ijsra.2024.12.2.1608

[18] Tiwari, V. K., Bagwani, M. K., & Jain, A. (2024). Implementing GrapesJS in Educational Platforms for Web Development Training on AWS. International Journal of Scientific Research in Multidisciplinary Studies (IJSRMS), 10(8), 1–8. https://doi.org/10.26438/ijsrms/v10i8.18

[19] Tiwari, V. K., Jain, A., Singh, R., & Singh, P. (2024). Enhancing Outlier Detection and Dimensionality Reduction in Machine Learning for Extreme Value. International Journal of Advanced Networking and Applications (IJANA), 15(6), 6204–6210. https://doi.org/10.35444/ijana.2024.15607.

[20] Tiwari, Virendra Kumar, Jain, Ashish, Bagwani, Mahesh Kumar, & Sharma, Ripusoodan. (2023) Classification of Motor Imaginary in EEG using Feature Optimization and Machine Learning. International Journal of Advanced Networking and Applications (IJANA), 15(2), 5887–5891.

Biographies and Photographs

Dr. Jitendra Agrawal is an Associate Professor in the MCA Department at Lakshmi Narain College of Technology (LNCT), Bhopal, with over 13 years of teaching experience. He holds an MCA from RGPV, Bhopal, and a B.Sc. from Dr. H.S. Gour Vishwavidyalaya, Sagar. His expertise lies in computer applications, with a focus on advancing teaching methodologies. He is committed to fostering student development and enhancing educational outcomes through technology.

Dr. Virendra Kumar Tiwari is a Professor and Head of the Department of Computer Applications at Lakshmi Narain College of Technology (MCA), Bhopal. He holds a B.Sc., M.A. (Economics), MCA, and Ph.D. from Dr. Hari Singh Gour University, Sagar, Madhya Pradesh. With over 17 years of academic and research experience, his areas of specialization include Computer Networks and Stochastic Modelling. Dr Tiwari has published 22 research papers in reputed national and international journals. He is actively involved in guiding students and research scholars, and his efforts continue to strengthen the academic and research environment of his institution.

Dr. Sanjay Thakur is a Professor of Computer Science at the Chameli Devi Group of Institutions, Indore. Dr. Thakur has awarded Ph.D. in Computer Science from Dr. Hari Singh Gour Central University, Sagar, M.P. in 2009. He has over 21 years of academic and research experience. He has authored more than 50 research papers and two textbooks and guided Ph.D. and M.Tech. Students. His research interests include Stochastic Modelling, Computer Network, and Wireless Network. He serves on the editorial board of various Journals, Associations, and Societies