**Evolving WebStack Information System with the Intelligent Cloud Integrated Development Environment**

Chew Shen1, a), Sin-Ban Ho1, b), and Chuie-Hong Tan2, c)

1Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Malaysia.

2Faculty of Management, Multimedia University, 63100 Cyberjaya, Malaysia.

b) Corresponding author: sbho@mmu.edu.my

a) chewshen012@gmail.com  
c) chtan@mmu.edu.my

**Abstract.** The growing complexity of code structures in information system development creates important challenges for both education beginners and professionals. They attempt to understand program code with debugging and collaborative project organization. The current Integrated Development Environments available on the market demonstrate inferior combinations between user-friendly operations and flexible specifications, alongside their seek for artificial intelligence (AI) support to bridge the software development gaps. Hence, the eCloud WebStack system is built and researched as an answer to Integrated Development Environment deficiencies by delivering AI-assisted cloud-based editing tools to strengthen programming development with both educational assistance and automated methodology. The system includes code visualization and AI chatbot with natural language capabilities to offer contextual help with code explanations towards proper developed workflows. It provides features through a user-friendly design, which serves users ranging from beginners to senior developers. The multiple features include live collaboration and secure file handling well considered across the Django and ReactJS frameworks that utilize AWS S3 cloud storage services. The research experienced along with iterative prototyping, user-focused evaluation, and evolved database technology can be good guidance for the diversified systems encountered. The usability tests involve participants who had expertise in education and development work reveal useful features aligned with their dual occupational expectations. This paper merges educational with professional assistance. The solution plays an essential role in closing the learning-environment to professional-industry transition, thus proving its worth to education and evolving information system practices.

# INTRODUCTION

To the information system evolution, eCloud WebStack System (ecWS) is the solution-based technique to enhance web development environment. It tackles problems that are challenging in information system developers like full stack, front-end, and back-end challenges. Through this system, the workflows are proposed to make the learning process easier and better oriented for individual users’ needs with artificial intelligence (AI) support, meanwhile making it easier for them to improve their skills more efficiently. The very basis behind the system is a total and personalised web development technique. Such features as automation tools with as low a learning curve as possible, as well as a flexible and intuitive user interface, allow to reduce a steep learning curve for beginners and increase productivity of the experienced developers [1]. The environment it builds encourages interaction where users can work through web development concepts and run experiments, evaluate, and understand the concepts better through direct practice. Unlike other popular integrated development environments (IDEs), ecWS is made to fit the requirements of students, educators, novices, and experienced users. Because of its intended use for teaching and professional work, ecWS can bridge an important gap that exists among current software development tools.

This evolved system possesses reshapes features with a robust valid and friendly interface to support practical outcomes. The authors make it easy to explore full stack, front-end, and back-end development components, with customisation over learning or task management [2]. Automated functionalities also make web development in the components much easier and more usable [3] [4]. The general techniques include researching workflow personalisation, designing approach to manage the way the development stacks and the research elements are being built. The authors embrace a practical and efficient system that both the learners and experienced developers need. The goals to disrupt some parts of the web development education and professional workflows by combining personalisation, automation, and intuitive learning.

This research involves developing a generic WebApp or Mobile App using modern frameworks such like Django, which were a is a free Python-grounded web frame with open source following architecture pattern model-template design developed. Because of its rapid-fire development, it also in high demand in the current request [5]. ReactJS is integrated into the system, so that user can have a real-time data synchronisation and modification. ReactJS also provide a solution where numerous users can interact at the same time through technologies like WebSocket or WebRTC [6]. The system relies on two AIs: Rasa NLU which manages the structured advice for users and Gemini API which provides help with coding and displays this visualised code through Mermaid.js. This combination allows easy use as well as helpful programming assistance in just one platform. These factors ensure that the system helps users to gain a better appreciation of some of the key conceptual pillars of web development, thus equipping them with the tools they needed to create more effective applications that met actual human requirements.

# LITERATURE REVIEW

The role of code visualization stands essential for these domains since it helps programmers and novices understand how software systems function. Utility programs display code information in visual representations that simplify complex programming structures for better understanding. The implementation of UML (Unified Modelling Language) diagrams flowcharts along with dependency graphs helps developers immediately find component relationships so they can speed up their debugging efforts while maximizing their work output [7]. These tools enable distributed teams to collaborate through their provision of shared system architecture understanding. The interactive visual documentation features decrease both the time new team members need to onboard and improves team member communication performance. Technology tools designed to support collaborative work operations have become essential elements for modern academic and professional software engineering practices.

These tools combine various development utilities into a unified interface which functions as an integrated development environment (IDE) platform. Integrated development environments comprise code editor and compiler and building automation tools and debuggers to create a single interface that boosts developer productivity. Visual Studio Code and IntelliJ IDE provide contemporary programmers with version control and code completion functionalities through syntax highlighting as well as development workflow simplification. Integrated development environments represent essential tools that serve both enterprise establishments and educational settings for novices who need programming knowledge through an easy-to-use organized platform. Integrated development environments provide a unified package that lowers user mental strain and speeds up programming development along with debugging processes.

Cloud-based integrated development environments provide developers with transformative capabilities that build accessibility features and enable better collaboration. Through cloud platforms which include AWS Cloud9, GitHub Codespaces together with Visual Studio Code for the Web developers gain access to robust development tools immediately in their browser interface [8]. Cloud-based IDE enable users to code and execute their programs with bugs checking on any internet connected devices. Cloud-based IDEs generate three major advantages through their ability to enable effortless teamwork together with self-maintaining backup systems and instant task collaboration access [9]. However, there are notable limitations. The performance suffers from internet speed fluctuations and delays because it creates problems during AI model training when performed in real time as well as in simulations involving heavy resource usage [10]. Data security together with privacy are essential issues for users due to the necessity of protecting sensitive data and proprietary code. Datapoint.com indicates that data breaches remain common alongside API security issues and the lack of standard security protocols in cloud IDEs [11]. Data protection mechanisms combined with stricter frameworks need to be implemented because of these emerging privacy-related concerns.

Web development frameworks serve as essential components for process definition because they help developers work faster. Django serves as a full-stack Python-based framework with "batteries-included" design approach by delivering ORM (Object-Relational Mapping) to interact with databases like PostgreSQL and OpenGauss. This framework creates an excellent environment for the intense data requirements that Instagram and Spotify use. The microframework Flask provides enhanced flexibility because it caters particularly to lightweight applications along with APIs. Python-based Query Prototyping and Machine Learning tasks use Django because of its management costs are relatively low and it remains easy to use. Laravel provides medium and small organizations with PHP-based development which unites sophisticated syntax with hard security components [9][10]. The selection between these frameworks stems from research complexity as well as performance needs and the technical abilities of developers respectively.

The evaluation of current cloud integrated development environment demonstrates the lack of suitable features for educational teaching and collaborative work environments [12]. Cloud9 connects well with AWS services yet forces users using AWS platforms to understand their environment. GitHub Codespaces provides quick startup and GitHub-integrated teamwork features although it depends heavily on GitHub practices. The web-based version of VS Code allows users to access desktop-like features from their browser, yet it lacks implicit collaborative capabilities which need extensions like Live Share to function. The tools demonstrate professional developer-focused strengths, but they sacrifice educational usability while providing inadequate assistance from AI and visual interface features that help novice users. This research introduces multimodal AI tools in education with both personalised learning and intelligent tutoring support. It investigates what Google Gemini is capable of and realise the use of visual, textual, audio and interactive features serves as a strong base for future developments in education. Rather than using a traditional IDE, Gemini is designed to help students by offering solutions, giving instant feedback and creating learning materials in multiple ways [13].

The eCloud WebStack System represents an integrated technique, which unites development with visualization along with AI-powered support to close current tool deficiencies. This system provides simultaneous benefits for development proficiency and educational progress beyond standard integrated development environment solutions. It offers multiple smart functions such as code visualization through diagrams and AI-powered chatbots for on-the-spot help and integrated teamwork features. The interface of this platform matches well with users including novices along with teachers and first-time developers who might find other integrated development environments complicated to use. Although the system design is a multipurpose approach, it serves both developing software systems and educate users more efficient than conventional integrated development environments.

# RESEARCH METHODOLOGY

The design-based strategy used for developing the eCloud WebStack System follows a structured method to solve usability together with collaboration and educational weaknesses found in modern cloud-based integrated development environment. The research initiative started through assessments of existing platforms during background research and literature review to uncover their shortcomings particularly regarding their code visualization features along with AI-based support functions. The investigation produced essential findings which enabled the development of an integrated system combining development resources with learning support elements under one unified platform [14] [15].

The process of requirements gathering established fundamental operational capabilities needed in the proposed system. Real-time editing capabilities combined with interactive visual frameworks and a chatbot assisted queries as well as debugging mechanism would be among the system's functional requirements alongside its pair-programming support. The system prioritizes accessibility together with user interface design responsiveness and data privacy and multi-language adaptability as essential non-functional requirements for diverse user populations. Modern tools and frameworks operated together to create the development process. The frontend elements used ReactJS with Figma to build an interface that combined usability alongside a responsive design while the backend solution utilized Django as the Python web framework. The Monaco Editor equipped VS Code with performance-oriented flexibility that enabled coding through its interface. The RASA platform was chosen to create an AI chatbot that delivers real-time assistance and during the development process. AWS S3 served as the platform to manage and store cloud files because of its ability to scale, provide security measures and adapt to various integrations.

Besides using the Rasa NLU chatbot, ecWS offers a second chatbot system operated through the Gemini API. While Rasa provides rule-based system-level support (e.g., helping users navigate the interface or retrieve tool descriptions), the Gemini chatbot offers AI-generated programming assistance. With it, user can fix code issues, see the meaning of various coding elements and make visual flowcharts using Mermaid.js. Within the platform, each bot is its own module designed for different purposes: the first is designed to be powerful and accurate, while the other is about experimentation and discovery. Being separated in function enables ecWS to include both detailed structures and synthetic features in its user interface.

The system operation framework integrates multiple user capabilities which include protected authorization systems together with programming interface modification tools and graphic presentation features and robot dialog functions. Intent recognition and entity extraction in Rasa NLU help this chatbot structure conversations and provide system guidance and Gemini applies its language-generating skills to help users do and understand coding tasks. This dual-system approach support users so they can receive clear as well as helpful adjustments when needed.[16]. The visualization workflow allows users to create and modify diagrams from code that improves both the comprehension of software structure and logic. Next, the system is assessed through 30 potential users consisting mostly of programmers who had minimal stacking experience. The users highlighted the key importance of using simple interfaces in visualizing code together with AI support and teamwork functionality. The research data produced direct inputs for making consecutive design improvements. User feedback and system testing were used to enable development through an agile methodology throughout the development process [17]. The research approach supports our information system to uphold technological strength, and basic user needs while doing so. The system integrates AI functions with visualization technologies and collaborative features inside a cloud-based integrated development environment to connect educational requirements with professional development tools which create an improved interactive and productive programming environment [18].

# RESULTS AND DISCUSSIONS

The eCloud WebStack System uses user-centred design principles to combine various features which boost educational process development and learning experiences through its interface. A preliminary exercise was conducted with 30 participants primarily students with less than one year of coding experience to identify essential features for the eCloud WebStack System (ecWS). Table 1 shows the result, which verified our designs with priorities setting for each system features.

**TABLE 1.** Summary of user survey responses (n = 30)

|  |  |
| --- | --- |
| **Feature Evaluated** | **Percentage (%) rated “Important” or “Very Important”** |
| Ease of use | Over 90% |
| Code visualization tools (e.g., UML) | 74% |
| AI-based debugging and support | 83% |
| Real-time collaboration | 66% |
| Tutorials and educational tools | 100% |
| Desire for simplicity in interface | 90% |
| Use of AI chatbots in coding tasks | 100% used; 60% say it helps with debugging |
| Preferred IDE features for learning | Visual aids, simplified UX, AI assist |

The file dashboard in Figure 1, and code visualisation dashboard shown in Figure 2 support code organisation plus presentation, while reducing cognitive workload through dual-coding theory visualisations. According to the dual-coding theory information that combines both visual and verbal forms increases user understanding and code retention leading to easier interpretation of code. The main interface of the system integrates a real-time AI-powered chat feature that works next to the main coding interface as depicted in Figure 3 to provide instant debugging and learning assistance.

A screenshot of a computer

AI-generated content may be incorrect.

**FIGURE 1.** File page

A screenshot of a computer

AI-generated content may be incorrect.

**FIGURE 2.** Code visualisation dashboard

The system uses scaffolding philosophy to support inexperienced users before their abilities mature. Figure 4 features the code visualization for visual program understanding through its easy-to-use interfaces. This provides support which enables users at different experience levels to obtain better comprehension of software architecture as well as logical programming flow thereby improving their coding skills. The system determines support levels automatically through ZPD (Zone of Proximal Development) principles to match them with the learner's development stage until the user demonstrates adequate skills [19].

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. |  |

**FIGURE 3.** Code editor page **FIGURE 4.** Code visualisation page

The system demonstrates personalisation and flexibility, as shown in the profile page in Figure 5 and Figure 6. This mechanism shows the system supports personalised setting management with Universal Design Learning (UDL) principles to enable all levels of learners to customise their experience [20]. In Figure 7, the System Assist interface provides further the interactive onboarding feature, which enlighten users’ understanding through multiple step-by-step prompts. This includes both quick overview and detailed tutorial choices to accommodate users who have different experience levels [21] plus learning speeds. The assistant provides clear explanations about pivotal features with code editing and visualisation. AI based diagnosis and co-managed functionality help users to effectively operate the system [22] from their initial engagement.

|  |  |
| --- | --- |
|  |  |

**FIGURE 5.** Profile page upper part **FIGURE 6.** Profile page lower part

Figure 8 highlights the AI Chat Assist feature to provide real-time interactive assistance through a conversational interface. Through the chatbot people can learn programming fundamentals while receiving instant debug recommendations [23] together with straightforward code language explanations. The AI-powered tutoring system operates through virtual instructor software which provides continuous guidance to novices while decreasing learning obstacles and improving program usage. The AI system provides step-by-step instruction for learners to understand defining functions in Python while explaining code operations then responding to additional questioning for in-depth clarification. The knowledge scaffolding process that happens through dialogue resembles teaching practices that intelligent tutoring systems implemented.

|  |  |
| --- | --- |
| A screenshot of a chat  AI-generated content may be incorrect. | A screenshot of a chat  AI-generated content may be incorrect. |

**FIGURE 7.** System assist pop up. **FIGURE 8.** Chatbot pop up.

# CONCLUSION

The eCloud WebStack System is aimed to mitigate the shortcomings of existing cloud-based integrated development environments, by providing features suitable for beginners as well as advanced users. These are enhanced and advanced code visualization, an artificial intelligent conversation along with debugging and learning, plus collaborative capability. Another integration of the system includes it work-effectively to improve the users’ productivity and their learning, as suggested by human trials together with user tests. Despite the issues like ensuring the chatbot effectiveness and managing its performance, the authors resolve these challenges through the experiments assessment. This work is indeed a valuable contribution to the web development techniques. It presents the consolidated and easily navigated environment that help novices to make the leap from academic world to practical use.

# REFERENCES

1. I. Ibriwesh, S.-B. Ho, I. Chai, and C.-H. Tan, “A Controlled Experiment on Comparison of Data Perspectives for Software Requirements Documentation,” Arab J Sci Eng **42**(8), 3175–3189 (2017). https://doi.org/10.1007/s13369-017-2425-2
2. S.-B. Ho, I. Chai, and C. H. Tan, “Leveraging framework documentation solutions for intermediate users in knowledge acquisition,” International Journal of Information Science 3(1), 13-23 (2013).
3. C.-W. Teoh, S.-B. Ho, K.S. Dollmat, and C.-H. Tan, “Ensemble-Learning Techniques for Predicting Student Performance on Video-Based Learning,” IJIET **12**(8), 741–745 (2022). https://doi.org/10.18178/ijiet.2022.12.8.1679
4. S.-B. Ho, E.-Y. Chew, and C.-H. Tan, “Streamlining Dental Clinic Management for Effective Digitisation Productivity and Usability,” JIWE **3**(2), 70–85 (2024). https://doi.org/10.33093/jiwe.2023.3.2.5
5. M. Sharma, M.S. Khan, and J. Singh, “Python & Django the Fastest Growing Web Development Technology,” in *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, (IEEE, Tandojam, Pakistan, 2024), pp. 1–9. https://doi.org/10.1109/KHI-HTC60760.2024.10482286
6. P. Keshari, P. Maurya, P. Kumar, and A. Katiyar, “Web Development Using ReactJS,” in *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, (IEEE, Greater Noida, India, 2023), pp. 1571–1575. https://doi.org/10.1109/ICAC3N60023.2023.10541743
7. “Predicting Student Performance from Video-Based Learning System: A Case Study,” JLISS, (2022). https://doi.org/10.33168/LISS.2022.0306
8. D.J. Malan, “Standardizing Students’ Programming Environments with Docker Containers: Using Visual Studio Code in the Cloud with GitHub Codespaces,” in *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2*, (ACM, Dublin Ireland, 2022), pp. 599–600. https://doi.org/10.1145/3502717.3532164
9. S.-B. Ho, I. Chai, and C. H. Tan, “An empirical investigation of methods for teaching design patterns within object-oriented frameworks,” International Journal of Information Technology & Decision Making **6**(04), 701-722 (2007). https://doi.org/10.1142/S021962200700271X
10. R. Ingabire, A. Bazco-Nogueras, V. Mancuso, L.M. Contreras, and J. Folgueira, “Clearing Clouds from the Horizon: Latency Characterization of Public Cloud Service Platforms,” in *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*, (IEEE, Kailua-Kona, HI, USA, 2024), pp. 1–9. https://doi.org/10.1109/ICCCN61486.2024.10637605
11. C.-W. Teoh, S.-B. Ho, K.S. Dollmat, and C.-H. Tan, “Machine Learning Prediction Model for Early Student Academic Performance Evaluation in Video-Based Learning,” Ijmst **10**(2), 1529–1544 (2023). https://doi.org/10.15379/ijmst.v10i2.1822
12. X.-X. Cho, S.-B. Ho, and C.-H. Tan, “Improving asthma treatment adherence by integrating weather information with responsive web technique,” (Virtual Conference, 2024), p. 030005. https://doi.org/10.1063/5.0216655.
13. M. Imran, and N. Almusharraf, “Google Gemini as a next generation AI educational tool: a review of emerging educational technology,” Smart Learn. Environ. **11**(1), 22 (2024). https://doi.org/10.1186/s40561-024-00310-z
14. R.K.J. Yik, S.-B. Ho, and C.-H. Tan, “Improving the prediction resolution time for mobile Eczema support system,” (Virtual Conference, 2024), p. 030007. https://doi.org/10.1063/5.0216658
15. M.E.K.K. Khairi, S.-B. Ho, and C.-H. Tan, “Real-time tracking and category filtering of an automated system for trading cards collections,” (Virtual Conference, 2024), p. 060001. https://doi.org/10.1063/5.0216659.
16. M.T.-T. Yong, S.-B. Ho, and C.-H. Tan, “Migraine Generative Artificial Intelligence based on Mobile Personalized Healthcare,” Journal of Informatics and Web Engineering  **4**(1), 275–291 (2025). https://doi.org/10.33093/jiwe.2025.4.1.20
17. S.B. Ho, S.L. Chean, I. Chai, and C.H. Tan, “Engineering Meaningful Computing Education: Programming Learning Experience Model,” in *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, (IEEE, Macao, China, 2019), pp. 925–929. https://doi.org/10.1109/IEEM44572.2019.8978920
18. S.-B. Ho, S.-K. Teh, I. Chai, C.-H. Tan, S.-L. Chean, and N.A. Ahmad, “Assessing Python Programming Through Personalised Learning Styles Model,” in *Computational Science and Technology*, edited by R. Alfred, H. Iida, H. Haviluddin, and P. Anthony, (Springer Singapore, Singapore, 2021), pp. 139–151. https://doi.org/10.1007/978-981-33-4069-5\_13
19. H. Xiao, Y. Sun, Z. Duan, Y. Huo, J. Liu, M. Luo, Y. Li, and Y. Zhang, “A Study of Model Iterations of Fitts’ Law and Its Application to Human–Computer Interactions,” Applied Sciences **14**(16), 7386 (2024). https://doi.org/10.3390/app14167386
20. J.-X. Kok, S.-B. Ho, and C.-H. Tan, “Enhancing Financial Literacy: A Progressive Web Application Approach for Malaysian Youth,” Journal of Informatics and Web Engineering  **4**(1), 53–69 (2025). https://doi.org/10.33093/jiwe.2025.4.1.5
21. S.B. Ho, I. Chai, and C.H. Tan, “AN EMPIRICAL INVESTIGATION OF METHODS, FOR TEACHING DESIGN PATTERNS WITHIN, OBJECT-ORIENTED FRAMEWORKS,” Int. J. Info. Tech. Dec. Mak. **06**(04), 701–722 (2007). https://doi.org/10.1142/S021962200700271X
22. W.-X. Ong, S.-B. Ho, and C.-H. Tan, “Enhancing Migraine Management System through Weather Forecasting for a Better Daily Life,” Journal of Informatics and Web Engineering  **2**(2), 201–217 (2023). https://doi.org/10.33093/jiwe.2023.2.2.15
23. J.-L. Goh, S.-B. Ho, and C.-H. Tan, “Weather-Based Arthritis Tracking: A Mobile Mechanism for Preventive Strategies,” Journal of Informatics and Web Engineering  **3**(1), 210–225 (2024). https://doi.org/10.33093/jiwe.2024.3.1.14