Date of publication JUN-30, 2025, date of current version JUN-25, 2025
www.computingonline.net/ computing@computingonline.net

*Print ISSN 1727-6209*
*Online ISSN 2312-5381*
*DOI 10.47839/ijc.24.2.4015*

# Implementation of Test-Driven Approach to Empower Self-Learning in PHP Web Programming Practice

**YAN WATEQULIS SYAIFUDIN[1*], TRIANA FATMAWATI[1], AMAL DERMAWAN UDJIR[1], PRAMANA YOGA SAPUTRA[1], NOBUO FUNABIKI[2], MUSTIKA MENTARI[2], CHANDRASENA SETIADI[3], ALFIANDI AULIA RAHMADANI[1]**

[1]Department of Information Technology, Politeknik Negeri Malang, Malang, Indonesia
(e-mail: qulis@polinema.ac.id, triana@polinema.ac.id, amaludjir@gmail.com, pramanay@polinema.ac.id, alfirahma4@gmail.com)
[2]Department of Electrical and Communication Engineering, Okayama University, Okayama, Japan
(e-mail: funabiki@okayama-u.ac.jp, p2918093@s.okayama-u.ac.jp)
[3]Department of Electrical Engineering, Politeknik Negeri Malang, Malang, Indonesia
(e-mail: chandrasenasetiadi@polinema.ac.id)

**ABSTRACT** Web applications have become immensely popular and widely utilized across various sectors due to their accessibility, scalability, and ability to provide real-time updates, driving the demand for skilled web programmers. Approximately 75.9% of websites that use server-side languages rely on PHP, a widely adopted and flexible scripting language that integrates seamlessly with databases, making it suitable for dynamic website development. As the complexity of web development increases, self-learning systems become essential for fostering independent programming skills through platforms such as learning management systems (LMS) and MOOCs; however, many of these tools still require human intervention for code verification and lack automated feedback mechanisms. This study proposes a self-learning framework that leverages test-driven development (TDD) to enhance PHP programming education. The framework enables students to participate in structured learning modules, receive real-time feedback, and refine their coding abilities. An evaluation with 150 first-year IT students in Indonesia demonstrated a 100% success rate in all modules, where success was defined as the ability of students to pass all test cases in the predefined test suites after multiple attempts. This highlights the framework's effectiveness while also noting areas for improvement in content depth and instructional clarity.

**KEYWORDS** self-study; code verification; PHP; web programming; automated testing; test-driven development

## I. INTRODUCTION

Web applications have become highly popular in the current era of the internet [1], being widely used in various sectors. They offer several advantages, such as being accessible from any internet-connected device, efficient scalability, and easy deployment, enabling real-time updates without user involvement. For example, e-commerce platforms allow consumers to conveniently shop around the world, and business information systems optimize operations and decision-making in organizations [2]. The growing reliance on web applications has heightened the demand for skilled web developers who can design, implement, and maintain these critical tools. As a result, numerous educational institutions have made web programming a mandatory subject in their IT departments [3]. Universities are providing students with web development education to meet the growing demands of a digital economy and foster the innovation that propels it [4], [5].

Approximately 75.9% of websites that use a server language rely on PHP [6], a popular scripting language in web programming [7] due to its user-friendliness and flexibility [8]. Its popularity comes from seamless database integration, making it ideal for developing dynamic websites. PHP offers many benefits, including numerous preexisting features,

frameworks such as Laravel and CodeIgniter that accelerate development [9], [10], and a large supportive community providing ample resources. Major websites like Facebook, WordPress, and Wikipedia use PHP [11], underscoring its robustness for various web tasks.

Self-learning systems in computer programming are increasingly important due to the complexity and evolving nature of web development, which require various languages and tools [12], [13]. As the demand for qualified web developers increases, self-directed learning platforms such as Learning Management Systems (LMS) [14], MOOCs [15], and automated assessment tools such as APAS [16] and ProgEdu4Web [17] are essential to provide flexible, personalized educational experiences. However, many of these platforms require human intervention for code verification and lack automated assessment features specifically designed for web programming tasks. Research indicates a significant need for real-time feedback mechanisms tailored to server-side programming. The implementation of a self-learning system that offers real-time automated feedback could enhance learner engagement and improve comprehension and skill development in PHP through immediate error identification and contextual insights.

This study presents a novel approach to enhance PHP programming education through a self-learning framework that adopts test-driven learning. The test-driven development (TDD) method realizes the creation of tests before coding [18], improving the quality of the code and the logical reasoning. This integrated approach enables learners to test coding solutions, receive real-time guidance on errors, and correct mistakes, thereby deepening their understanding of programming concepts. The web-based framework focuses on the basic syntax of PHP, defining specific learning objectives to guide students to master the fundamentals of PHP through structured modules, practical exercises, and customized test codes using PHPUnit to verify accuracy and functionality [19].

The system evaluation demonstrated high effectiveness, with the 150 students successfully completing six modules and achieving a success rate of 100% on the tests cases. While students noted improvements in their understanding and confidence, they also reported technical challenges and requested additional practical examples and multilingual support. Future enhancements should address these issues by providing detailed explanations of complex concepts, incorporating gamified collaborative learning, and including cybersecurity principles to better prepare students for real-world challenges. The framework's design also allows for future AI integration to further personalize learning and enhance feedback mechanisms.

## II. RESEARCH SIGNIFICANCE

This section presents recent studies related to this study, test-driven development method, and the idea of test-driven learning for web programming using PHP.

### A. WEB PROGRAMMING WITH PHP

PHP (Hypertext Preprocessor) is a widely-used open-source server-side scripting language designed primarily for web development. It enables developers to create dynamic web pages and web applications efficiently. The language is embedded within HTML, making it easy to integrate with front-end technologies and databases such as MySQL. Learning PHP involves various topics, including understanding PHP syntax, handling forms, working with databases, managing sessions, and implementing security practices. Additionally, concepts such as object-oriented programming, web frameworks (e.g., Laravel, Symfony), and content management systems (e.g., WordPress) may also be covered. Mastery of these topics is essential for developing robust applications and ensuring seamless interaction between the server and client-side elements of a website.

Practical skills are crucial in programming, as they bridge the gap between theoretical knowledge and real-world application. Students must acquire practical experience through assignments that challenge them to apply their knowledge in tangible ways. Assignments not only reinforce learning, but also develop problem-solving capabilities and confidence in coding. To ensure the correctness and effectiveness of their solutions, code verification becomes an essential process. This involves using various testing methodologies, such as unit tests, to validate assignment answers against specified requirements. By implementing code verification, students can identify errors, understand debugging practices, and gain insights into writing clean, efficient code. This hands-on experience, coupled with test-driven learning, ultimately prepares students for successful careers in web development.

### B. TEST-DRIVEN LEARNING

Test-Driven Learning (TDL) is an educational methodology inspired by test-driven development (TDD) in software engineering [20], which emphasizes the use of tests as a primary driver for learning and understanding concepts. In TDL, learners first define a set of expected outcomes or behaviors for a particular concept or assignment, typically in the form of test cases. They then engage in self-directed exploration and coding to meet those predefined criteria, continually refining their solutions based on the feedback from the tests. This iterative process encourages active learning and critical thinking, as students are motivated to understand the underlying principles and syntax necessary to pass the tests. In addition, TDL promotes a hands-on approach to learning, allowing students to experiment and learn from mistakes in a structured way [21]. As a result, Test-Driven Learning not only enhances problem-solving skills but also fosters a deeper understanding of programming concepts, thereby empowering learners to become more proficient and confident in their coding abilities [22].

### C. RECENT STUDIES

There are some studies related to self-learning systems for computer programming, including:

- In 2020, Syaifudin et al. proposed an Android Programming Learning Assistance System [23], that is designed to achieve independent learning without the presence of teachers by adopting the test-driven development method [24]. This proposal was continued with the development of an online platform to automatically distribute, collect, and validate the answers.
- Mekterović et al. developed an Automated Programming Assessment System (APAS) to address the challenges of manual programming task evaluations by providing objective, efficient evaluations and timely feedback [16]. This study reviews the literature and APAS software, identifying key features needed to support all stages of assessment in computer science courses. Despite numerous publications, there are limited software options available.
- Research by Paiva et al. in 2020 focused on the types of exercises supported, the security measures used, the testing techniques used, the types of feedback generated, and the information provided to instructors to understand and facilitate learning optimization [25]. They emphasize that practical programming skills are crucial for computer science education and postgraduate preparation.
- Similarly, research by Barra et al. in 2020 [26] described the transformation of programming course assessments in higher education into fully online formats during the pandemic through student-centered automated assessment tools. The evaluation of student interactions and perceptions showed highly positive results, with most of the students preferring to use these tools and expressing a desire to implement them in other courses.
- Recently, Snowberger et al. applied Front-End Code Playground (FECP) tools in web development learning [27], particularly the integration of HTML, CSS, JavaScript, and additional libraries. Given that many web programming errors do not yield visible results in browsers, it has ability to consolidate web technologies into a single platform and provide immediate visual feedback. The study collects and analyzes data on various FECP tools, highlighting their differences and benefits for students.
- Furthermore, Duong et al. presented a recent proposal of ProgEdu4Web [17], an innovative Automated Programming Assessment System for web programming courses. This system integrates automatic source code evaluation tools for static code analysis, enabling instructors to efficiently assess student team collaboration, code quality, and web programming skills while providing timely feedback to students.

### D. TEST-DRIVEN LEARNING FOR WEB PROGRAMMING

Despite the advancements in educational tools, there remains a notable gap in addressing the specific needs of PHP programming learners. Current systems often provide front-end development and overlook back-end validation, underscoring a need for a PHP-centric educational framework that utilizes TDD and provides real-time automated feedback. Such a system could significantly enhance learner engagement, allowing for immediate identification and resolution of errors, fostering a deeper grasp of PHP programming fundamentals. A tailored feedback system would empower students to iteratively improve their coding skills [28], creating a resilient learning environment that adapts to individual needs and promotes proficiency in backend development.

Consequently, the implementation of a PHP web programming system utilizing Test-Driven Development (TDD) and real-time automated feedback is essential to address these deficiencies. This system could improve the participation of the learners and facilitate deeper comprehension and refinement of PHP programming skills by providing real-time error identification and resolution tools, providing immediate, context-specific insights that directly address the shortcomings of current systems. This customized feedback system enables students by facilitating iterative code enhancement, fostering a resilient learning environment that adapts to mistakes and individual learning speeds, thus addressing the intricate skills necessary for proficient PHP backend development.

## III. METHOD FOR SELF-LEARNING FRAMEWORK

This section proposes the self-learning framework for web programming using PHP by utilizing TDD method and automated testing.

### A. SELF-LEARNING MODEL

Computer Assisted Learning (CAL) is an educational approach that employs computers to deliver instructional content and enhance the learning process through multimedia, interactive modules, and digital resources, catering to diverse learning preferences. It allows students to interact with educational software for customized experiences, providing immediate assessment and feedback, which facilitates learning at their own pace [29]. As technology advances, CAL has evolved into self-learning systems that promote student autonomy and active engagement in their education [30].

The proposed self-learning model leverages CAL with computer-generated assistance that can significantly streamline educational processes [30]. This model integrates the TDD method approach, encouraging students to first engage with test suites as part of CAL's structured learning resources. In the learning process, students can verify the written code against predefined test suites, strengthening understanding and facilitating a deeper grasp of PHP coding practices, as illustrated in Figure 1 The learning process in the proposed model includes steps:

- **Accessing Learning Resources**: Students receive a variety of comprehensive learning resources, including instructional guidance and structured test suites, aimed at grounding them in PHP programming and equipping

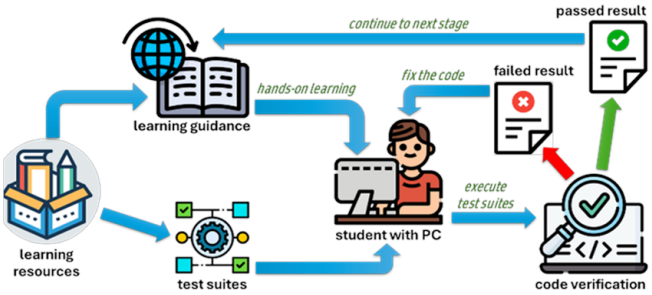


Figure 1. The proposed model of self-learning system

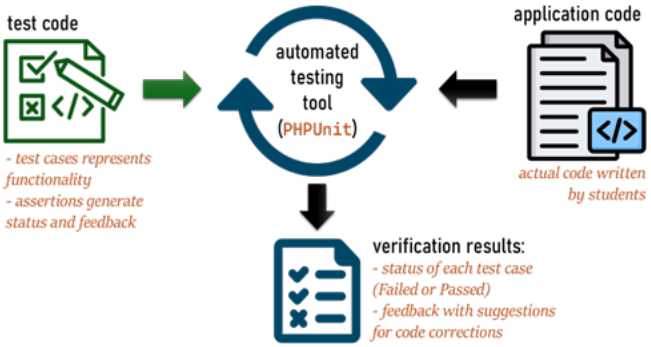


Figure 2. Automated code verification workflow

them with both theoretical knowledge and practical instructions.

- **Coding Practice**: After reviewing the guidance, students engage in coding exercises within a PHP environment (Visual Studio Code), applying theoretical principles to real-world scenarios, which enhances their problem-solving skills and provides essential hands-on experience.
- **Code Verification**: Upon completing their coding assignments, the students verify their code using predefined test suites to assess accuracy and effectiveness [31]. This process ensures that their code meets requirements and offers prompt feedback for future improvements.
- **Passed the Challenge**: If a student's code passes all tests, it indicates compliance with the predefined specifications, allowing them to progress to the next stage of the learning path, thereby promoting continuous learning and achievement throughout the curriculum.
- **Learning from Feedback**: If the code does not meet the requirements, the learning model provides constructive feedback detailing specific code flaws and suggestions for improvements, encouraging students to revise their work.

### B. AUTOMATED TESTING FOR PHP

Automated testing refers to the use of specialized software tools to run tests on code automatically [32], helping to ensure that applications function as expected without the need for manual intervention. Figure 2 illustrates that automated code verification utilizes test code to define test cases that contain assertions, which check the expected outcomes of the application code that is being verified. The application code represents the actual functionality and logic to be tested for correctness. An automated testing tool executes the test code, assessing both the functionality and syntax of the application code. The results of this process generate a status indicating whether each test case has Passed or Failed, along with detailed feedback for any failure.

There are several unit testing tools available for developers [33], including PHPUnit, Codeception, Behat, and SimpleTest, each offering unique features for testing applications. Among these, PHPUnit is regarded as one of the most popular tools for PHP development, recognized for its versatility and effectiveness in ensuring code quality. One of PHPUnit's key privileges is its user-friendly syntax, which makes it easy for developers to write and maintain tests. Additionally, it provides a rich set of assertions that allow for detailed validation of expected outcomes, along with robust mocking capabilities to isolate and test individual code components efficiently.

PHPUnit offers several key features, including unit testing, assertions, and test suites, that enhance the testing process for PHP applications. Unit testing focuses on discrete components, such as functions and classes, in isolation, improving the accuracy and robustness of the application. Assertions play a crucial role in this framework by allowing developers to evaluate conditions and confirm expected code behavior through various methods such as 'assertEquals()' and 'assertTrue()'. Additionally, PHPUnit enables the organization of test cases into test suites, which are collections that can be executed together, facilitating the management of large projects. This feature allows developers to streamline their testing processes by grouping related tests, making it easier to track functionality and ensure comprehensive test coverage.

### C. GENERATING INSTANT FEEDBACK

Each test suite is organized into specific test cases, which use various assertion methods to validate that the actual output matches the expected results and delivers failed messages. Automated code verification enables students to receive immediate feedback and effectively address mistakes in code. The test results are typically classified into two conditions: passed and failed. A passed condition indicates that the software operates as expected and meets the predefined requirements, affirming its correctness. In contrast, a failed condition signifies that the code has not performed as intended, highlighting specific areas that require attention. When a test fails, it generates informative feedback that elucidates the nature of the failure, providing essential insights to identify and rectify the underlying issues. Figure 3 illustrates a failed result when executing a test case following a specific error message, marking the location of the failure and providing clues about the cause of the error.

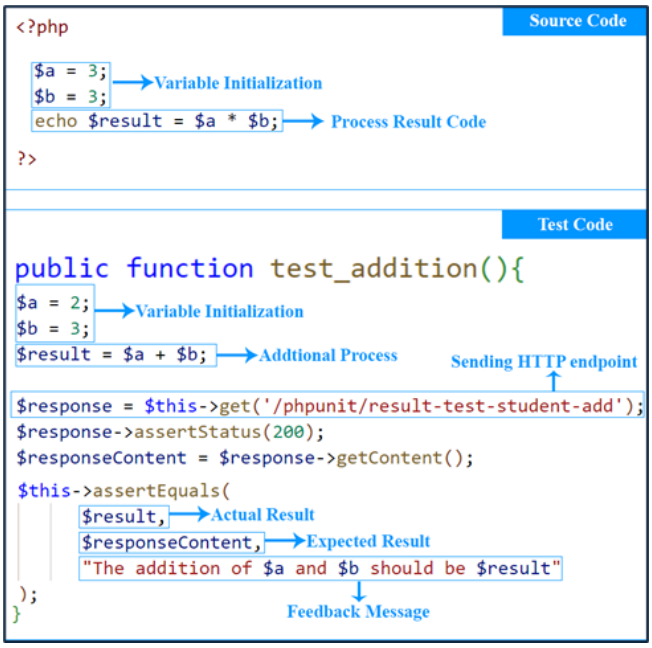


Figure 3. Source code and test case with feedback

## IV. IMPLEMENTATION OF SELF-LEARNING SYSTEM FOR PHP BASIC SYNTAX

Author names are to be centered beneath the title and printed in Times New Roman 11-point, boldface type. Author affiliations should be centered beneath author names and printed in Times New Roman 11-point, non-boldface type.

### A. FUNDAMENTAL OF PHP WEB PROGRAMMING

To master web development with PHP, understanding the fundamental code structure and syntax is essential, even when frameworks are employed to streamline and expedite the process [34]. Native PHP code structures encompass the organization and writing of PHP without any frameworks, emphasizing the essentials such as PHP tags, which denote the start and end of PHP code within a file, enabling PHP to be embedded within HTML. Understanding syntax includes familiarity with comments for code documentation, variables that adapt dynamically to data types, and functions that encapsulate task-specific code blocks. Key programming constructs, such as conditionals and loops, help control the flow and repetition of code execution. Furthermore, arrays provide a method to manage multiple values within a single variable, supporting both numeric and associative types.

### B. LEARNING OBJECTIVES

For an effective self-learning system for students, a set of learning objectives is proposed that are adapted from various learning materials [35]. Table 1 shows the eight learning objectives for the learning topic of Basic Syntax.

**Table 1. Learning objectives for basic syntax learning**

| No. | Objective |
|---|---|
| LO1 | Learn how to set up a PHP environment, including installing PHP, a local XAMPP server, and a text editor with Visual Studio Code. |
| LO2 | Learn to open and close PHP tags within a file to distinguish PHP code from HTML. |
| LO3 | Comprehend the standard structure and syntax rules for writing PHP scripts. |
| LO4 | Understand how to declare variables and recognize different data types like strings, integers, and arrays. |
| LO5 | Learn how to use arithmetic, assignment, comparison, and logical operators to perform operations. |
| LO6 | Understand how to use conditional statements (if, else, switch) to control the flow of the program based on conditions. |
| LO7 | Learn to implement loops (for, while, foreach) to execute code repeatedly based on certain conditions. |
| LO8 | Learn to create and manipulate arrays and use built-in array functions for processing array data. |

**Table 2. Six learning modules for students**

| No. | Relation to Learning Outcomes |
|---|---|
| 1 | LO1: This module covers the complete installation process for setting up a PHP development environment, including Composer. |
| 2 | LO2: This module directly addresses how to correctly use PHP opening (`<?php`) and closing (`? >`) tags.<br>LO3: It also covers the fundamental syntax rules necessary for writing valid PHP scripts. |
| 3 | LO4: This module dives into declaring and utilizing various types of variables available in PHP, such as integers and strings.<br>LO5: Alongside variables, this module introduces different operators, enabling learners to manipulate data effectively. |
| 4 | LO6: The focus of this module is on conditional statements like `if`, `else`, and `switch`. |
| 5 | LO7: This module is dedicated to understanding and implementing various loop structures such as `for`, `while`, and `do...while`. |
| 6 | LO8: This module teaches students how to create both indexed and associative arrays. It covers accessing, modifying, and utilizing built-in functions for arrays. |

### C. LEARNING MODULES FOR STUDENTS

Table 2 describes six learning modules that cover eight learning outcomes.

### D. TEST SUITES

Test suites ensure the accuracy of student-submitted PHP code using PHPUnit. Each learning module includes tailored test suites specifically designed to assess the student's code, ensuring that their answers meet the required criteria [23]. When students submit their PHP code, the web application runs the test cases in test suite through PHPUnit, generating results that provide instant feedback on the accuracy and functionality of their submissions. This immediate feedback mechanism not only enables students to quickly identify and rectify errors. Each test suite demonstrates how to validate these elements: for instance, testing the basic structure confirms foundational functionality, while testing variables ensures the reliability of values. Additionally, tests on conditional statements and loops verify the program's logical

```php
public function test_loop_output() {
    $response = $this->get("/loop-assignment");
    $expectedOutput =
    "1 2 3 4 5 6 7 8 9 10 this is looping php example";
    $response->assertSee($expectedOutput);
    $this->assertIsString($response->getContent());
}
```

Figure 4. Test case to verify loop syntax

flow, and array tests check data integrity, as shown in Figure 4.

### *E. WEB INTERFACE FOR SELF-LEARNING*

The implemented web application serves as a dynamic learning interface for students studying PHP programming, allowing them to choose from various learning modules, including topics from installation to arrays. Each module includes a dedicated web interface that presents guidance documents aimed at facilitating hands-on programming practice. This resource offers detailed instructions and examples, promoting interactive exploration of programming concepts. A straightforward color-coded feedback system enhances the learning experience; a green box indicates successful code submissions, while a red box signals errors, encouraging students to make the necessary adjustments and improve their understanding.

In addition to the guidance function, the application features a code verification function that delivers immediate feedback on student coding submissions. Students upload their code alongside any explanatory comments and can initiate verification by clicking the "Verify the Answer" button, as shown in Figure 5. The system then assesses their submission and provides results that indicate whether the code has passed or failed. This seamless integration into the learning process offers automated feedback on coding attempts, with successful submissions resulting in a green box for affirmation, while failures trigger a red box that provides instant feedback to help students identify and correct errors, thereby reinforcing their learning of PHP programming concepts.

## V. PROPOSAL EVALUATION

This section presents the results of the self-learning platform evaluation by conducting initial testing and applying it to IT department students.

### *A. EVALUATION SCENARIO*

The selected cohort of 150 students includes 50 individuals from each of the following programs: Information Systems, Informatics, and Web Development. This varied group provides multiple perspectives within the IT field, allowing for the evaluation of the system's adaptability and effectiveness across different academic disciplines. As these students will undertake a web programming course in their second year, early exposure to PHP syntax is vital to building a strong foundation in web development. Establishing this

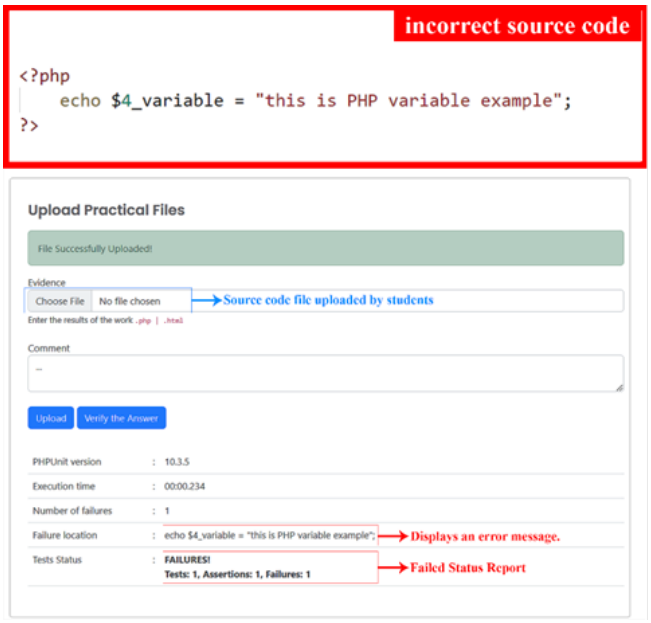


Figure 5. Web interface to submit answer and get feedback

**Table 3. Hardware and Software Specifications**

| No. | Hardware/Software | Description |
|---|---|---|
| 1 | Processor | Intel Core 2 Duo or equivalent |
| 2 | RAM | Minimum 4.00 GB |
| 3 | Storage | 120 GB HDD with at least 20 GB of available storage |
| 4 | Connection | LAN |
| 5 | Operating System | Windows 8.1 or newer |
| 6 | Text Editor | Visual Studio Code: Latest Stable Version Installed |
| 7 | Server | XAMPP Control Panel v3.3.0 with PHP 7.4 or newer |

groundwork is crucial for their future success in advanced web development and related computer science courses. This diverse group will provide valuable feedback to assess the system's adaptability and effectiveness in various IT areas.

Each student is provided with a personal PC to encourage individual engagement and hands-on experience. Table 3 details the hardware and software specifications. The evaluation is structured to be completed in one day, enabling students to concentrate fully on understanding and applying the concepts from each module. Student success was measured by their ability to pass all predefined test cases in the automated test suites for each learning module, achieving a "Passed" status after one or more attempts. The automated verification system ensured objectivity by requiring the code to meet specified criteria, including syntax correctness, functionality, and logical flow. Immediate feedback on failed test cases allowed students to iteratively refine their code until all tests were passed.

**Table 4. Completion Time Results on Each Module**

| Metric | Mod 1 | Mod 2 | Mod 3 | Mod 4 | Mod 5 | Mod 6 |
|---|---|---|---|---|---|---|
| Passed (students) | 150 | 150 | 150 | 150 | 150 | 150 |
| Failed (students) | 0 | 0 | 0 | 0 | 0 | 0 |
| Minimum submissions | 1 | 1 | 1 | 1 | 1 | 1 |
| Average submissions | 1 | 1 | 1.13 | 1.1 | 1.25 | 1.38 |
| Maximum submissions | 1 | 1 | 2 | 2 | 3 | 4 |
| Fastest (minutes) | 2 | 2 | 7 | 10 | 12 | 12.7 |
| Average (minutes) | 10.2 | 10.6 | 10.8 | 10.7 | 12.7 | 15.7 |
| Longest (minutes) | 20 | 24 | 26.7 | 30 | 25 | 48 |

### B. GENERAL RESULTS

During the evaluation, all students smoothly executed the technical aspects, with their computers set up without problems, leading to a smooth start to the learning activities. The web application, which served as the central platform for delivering learning modules and verifying submitted code, performed efficiently throughout the day, while Visual Studio functioned flawlessly, providing essential tools for assignments. Stable WiFi and internet connections ensured uninterrupted access to online resources. Notably, all students successfully completed the modules, demonstrating the effectiveness of the instructional design. Clear and accessible guidance allowed students to easily understand instructions, while the ability to approach assignments through their unique learning experiences fostered personal strategies and creativity in problem solving. Additionally, straightforward feedback allowed students to learn from and correct their errors effectively.

### C. COMPLETION TIME

All 150 students successfully passed each of the six modules, with no failures reported, as shown in Table 4. For modules 1 and 2, most students completed tasks with a single submission, while module 3 through module 6 saw a slight increase in average submissions, indicating more complexity or the need for retries, particularly in module 6, which required up to four submissions. The minimum number of submissions was consistent at one for all modules, while the maximum varied from one to four, reflecting different difficulty levels. Regarding completion time, the fastest students finished modules 1 and 2 in just 2 minutes, with subsequent modules taking longer, culminating in module 6 at 12.7 minutes. On average, students spent between 10.2 and 15.7 minutes on the modules, with the longest times ranging from 20 minutes in module 1 to 48 minutes in module 6, highlighting the increasing difficulty as they progressed through the modules, as shown in Figure 6.

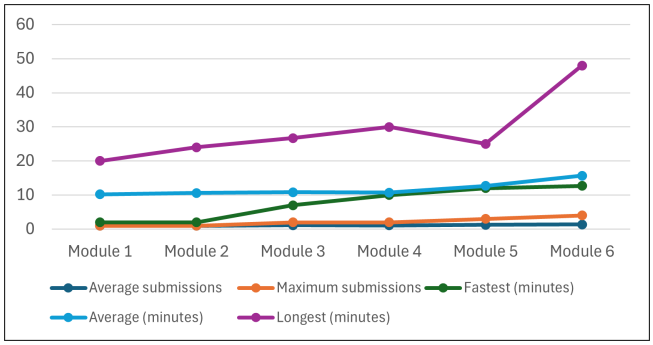


Figure 6. Completion time results for 150 students

**Table 5. Three Difficulty Levels for Six Modules**

| Level | Learning Modules |
|---|---|
| Easy | Module 1, Module 2 |
| Moderate | Module 3, Module 4 |
| Hard | Module 5, Module 6 |

### D. CLASSIFICATION OF DIFFICULTY LEVELS

The difficulty levels of the learning modules can be classified as follows, according to the results in Table 5:

- Easy: Modules 1 and 2 were completed with an average of one submission, and the fastest completion time was 2 minutes, with overall times remaining low.
- Moderate: Modules 3 and 4 required slightly more submissions, with some students needing two attempts. Average completion times increased modestly, indicating a moderate challenge.
- Hard: Modules 5 and 6 had higher average submissions and completion times. Module 5 required up to three attempts, while Module 6 was the most challenging, requiring up to four submissions and a maximum completion time of 48 minutes.

### E. ANALYSIS ON EVALUATION RESULTS

The evaluation results of the PHP programming self-learning system evaluation reflect a high level of effectiveness, with all 150 students successfully completing each of the six modules and no failures reported. The success rate of 100% reflects the fact that all students were able to pass all test cases in the predefined test suites for each module after one or more attempts. Success was measured based on the automated verification system, which required students' code to meet all specified criteria in the test suites. The seamless setup of technical components, including the web application and Visual Studio, facilitated a smooth start to the learning activities. The instructional design effectively helped students grasp PHP concepts, enabling creative problem-solving and personalized strategies. The rise in submissions for complex modules (5 and 6) highlights meaningful engagement with challenging material, while the clear feedback system facilitated error correction and skill improvement.

**Table 6. Students' Feedback on Proposed Framework**

| Types | Students' Comments |
|---|---|
| Positive Responses | - It helped me understand basic PHP concepts better, especially in practical applications;<br>- The testing method is very effective, helping me identify areas for improvement;<br>- The practical learning flow is clear and easy to understand;<br>- The instructions in the web are very clear, and I feel more confident using PHP. |
| Obstacles on Learning | - Some students experienced technical difficulties that extended the time;<br>- The guidance is in English. |
| Constructive Suggestions | -Provide additional practical examples before the trial for better preparation;<br>- Include more detailed explanations for complex concepts to aid understanding;<br>- Provide local language for better reading in guidance. |

### F. FEEDBACK FROM STUDENTS

Table 6 provides students' feedback during the evaluation consisting of positive responses, learning obstacles, and constructive suggestions. However, some feedback indicated areas that require attention to enhance the overall learning experience. Students reported experiencing technical difficulties that extended their completion times, highlighting the need for robust technical support to address such issues. While some students required additional submissions to achieve this, the iterative feedback mechanism enabled them to identify and correct errors effectively, ensuring eventual success. Additionally, the exclusive use of English in guidance materials could limit accessibility for non-native speakers, posing a barrier to comprehension [36]. To address these concerns, it is suggested that additional practical examples be included. Moreover, offering detailed explanations for complex topics and providing guidance in multiple languages would improve overall understanding and accessibility.

## VI. DISCUSSIONS

The proposed system demonstrates great potential to support the study of web programming using *test-driven learning* (TDL) and automated feedback, helping students master PHP through structured modules and real-time error identification. Its feasibility is demonstrated by the seamless integration of accessible tools such as XAMPP, Visual Studio Code, and PHPUnit, which work with standard hardware, making it adaptable for diverse settings. The modular design and scalability of the framework make it a viable candidate for development into an innovative educational product. By addressing the limitations, it could evolve into a comprehensive solution tailored to modern learners, bridging gaps in current platforms, and fostering self-directed web development learning.

While the framework demonstrated high effectiveness, several limitations were identified. Technical difficulties, such as software setup issues, highlight the need for pre-configured environments or offline support. Additionally, students requested more detailed explanations and advanced topics, indicating opportunities to expand content depth and scope. The feedback mechanism could be improved by providing contextual error explanations, and collaborative learning features, such as gamified challenges, could enhance engagement. Another aspect to consider is the absence of focus on cybersecurity principles and the implications of real-time updates in web applications [37]. Although the framework effectively imparts foundational PHP syntax and programming concepts, it currently lacks guidance on secure coding practices, including input validation, SQL injection prevention, and cross-site scripting (XSS) mitigation [38].

AI-powered tools have advanced programming education, however they often focus on general-purpose or front-end development, neglecting server-side needs like PHP. Additionally, AI-generated solutions lack structured learning and code accuracy assurance, critical for education and assessment. The proposed framework addresses this by integrating test-driven learning (TDL) with automated feedback, enabling iterative testing and debugging of PHP code. TDD ensures rigorous validation against predefined test cases, fostering core programming skills and logical reasoning. Furthermore, the modular design of the framework allows future AI integration, enhancing personalized learning through adaptive guidance and advanced error explanations.

## VII. CONCLUSION

This paper presented a proposal for a self-learning framework for PHP programming employing a test-driven learning approach to enhance students' coding skills and understanding of PHP fundamentals through direct interaction with test suites. This structured learning model encourages hands-on coding practice and self-verification against predefined test suites, promoting mastery of basic syntax while fostering critical thinking and problem solving abilities essential for progressing to advanced programming concepts. The automated testing tool, PHPUnit, offers real-time feedback that enables students to identify and correct errors. The evaluation of the framework, involving a diverse group of 150 students, revealed high engagement and successful completion rates, demonstrating its effectiveness and adaptability to various learning styles.

The research findings demonstrate the high effectiveness of the proposed PHP self-learning framework, achieving a 100% success rate through structured test-driven learning (TDL) and automated feedback. The seamless integration of tools highlights its feasibility and adaptability for diverse educational settings. While the framework successfully fosters foundational PHP skills and creative problem-solving, limitations such as technical difficulties, language barriers, and the need for more detailed explanations and advanced topics were identified. Future work will focus on addressing these gaps by incorporating pre-configured environments, multilingual support, and secure coding practices, alongside expanding content depth and integrating AI-driven adaptive guidance.

## References


[1] S. Sotnik, V. Manakov, and V. Lyashenko, "Overview: Php and mysql features for creating modern web projects," International Journal of Academic Information Systems Research, vol. 7, no. 1, pp. 11–17, 2023.

[2] C. A. T. Romero, J. H. Ortiz, O. I. Khalaf, and A. R. Prado, "Web application commercial design for financial entities based on business intelligence," Computers, Materials & Continua, vol. 67, no. 3, p. 3177–3188, 2021.

[3] J. C. López-Pimentel, A. Medina-Santiago, M. Alcaraz-Rivera, and C. Del-Valle-soto, "Sustainable project-based learning methodology adaptable to technological advances for web programming," Sustainability (Switzerland), vol. 13, no. 15, pp. 1–25, 2021.

[4] M. Ferati and B. Vogel, "Accessibility in web development courses: A case study," Informatics, vol. 7, no. 1, pp. 1–15, 2020.

[5] V. Gor, Introduction to Web Development. Berkeley, CA: Apress, 2023, pp. 1–22.

[6] w3tech, "W3techs - extensive and reliable web technology surveys," 2022.

[7] J. Goetz and A. F. Marquez, "Web framework," International Journal on Engineering, Science and Technology, vol. 5, no. 4, pp. 353–375, 2023.

[8] P. Kyriakakis, A. Chatzigeorgiou, A. Ampatzoglou, and S. Xinogalos, "Exploring the frequency and change proneness of dynamic feature pattern instances in php applications," Science of Computer Programming, vol. 171, no. 15, pp. 1–20, 2019.

[9] P. R. Chavan and S. Pawar, "Comparison study between performance of laravel and other php frameworks," International Journal of Research in Engineering, Science and Management, vol. 4, no. 10, pp. 27–29, 2021.

[10] H. Abutaleb, A. Tamimi, and T. Alrawashdeh, "Empirical study of most popular php framework," in 2021 International Conference on Information Technology (ICIT), 2021, pp. 608–611.

[11] M. Zandstra, PHP 8 Objects, Patterns, and Practice: Mastering OO Enhancements and Design Patterns. Apress L. P., 2024, no. v. 1.

[12] M. Chen, "Reducing web page complexity to facilitate effective user navigation," IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 4, pp. 739–753, 2020.

[13] J. Swacha and A. Kulpa, "Evolution of popularity and multiaspectual comparison of widely used web development frameworks," Electronics (Switzerland), vol. 12, no. 17, pp. 1–23, 2023.

[14] I. Zhukov and Y. Kostyuk, "The system for automated assessment of knowledge and practical skills in programming," Informatics and education, vol. 38, no. 2, pp. 66–74, 07 2023.

[15] M. Elhayany, S. Serth, and C. Meinel, "Exploring the effectiveness of web-based programming environments for moocs: A comparative study of codeocean and openjupyter," in 2023 IEEE Learning with MOOCS (LWMOOCS), 2023, pp. 1–6.

[16] I. Mekterović, L. Brkić, B. Milašinović, and M. Baranović, "Building a comprehensive automated programming assessment system," IEEE Access, vol. 8, no. 1, pp. 81 154–81 172, 2020.

[17] H.-T. Duong and H.-M. Chen, "Progedu4web: An automated assessment tool for motivating the learning of web programming course," Computer Applications in Engineering Education, vol. 32, no. 5, p. e22770, 2024.

[18] V. S. Bhadauria, R. K. Mahapatra, and S. P. Nerur, "Performance outcomes of test-driven development: An experimental investigation," Journal of the Association for Information Systems, vol. 21, no. 4, pp. 1045–1071, 2020.

[19] M. Zandstra, Testing with PHPUnit. Berkeley, CA: Apress, 2010, pp. 379–405.

[20] M. T. Baldassarre, D. Caivano, D. Fucci, S. Romano, and G. Scanniello, "Affective reactions and test-driven development: Results from three experiments and a survey," Journal of Systems and Software, vol. 185, no. 1, pp. 111–154, 2022.

[21] M. A. Langford and B. H. C. Cheng, "Enki: A diversity-driven approach to test and train robust learning-enabled systems," ACM Transactions on Autonomous and Adaptive Systems, vol. 15, no. 2, May 2021.

[22] S. Marwan, B. Akram, T. Barnes, and T. W. Price, "Adaptive immediate feedback for block-based programming: Design and evaluation," IEEE Transactions on Learning Technologies, vol. 15, no. 3, pp. 406–420, 2022.

[23] Y. W. Syaifudin, N. Funabiki, M. Kuribayashi, and W. C. Kao, "A proposal of android programming learning assistant system with implementation of basic application learning," International Journal of Web Information Systems, vol. 16, no. 1, pp. 115–135, 2020.

[24] Y. W. Syaifudin, N. Funabiki, M. Mentari, H. E. Dien, I. Mu'Aasyiqiin, M. Kuribayashi, and W. C. Kao, "A web-based online platform of distribution, collection, and validation for assignments in android programming learning assistance system," Engineering Letters, vol. 29, no. 3, pp. 1178–1193, 2021.

[25] J. C. Paiva, R. Queirós, J. P. Leal, J. Swacha, and F. Miernik, "Managing gamified programming courses with the fgpe platform," Information (Switzerland), vol. 13, 2022.

[26] E. Barra, S. López-Pernas, A. Alonso, J. F. Sánchez-Rada, A. Gordillo, and J. Quemada, "Automated assessment in programming courses: A case study during the covid-19 era," Sustainability (Switzerland), vol. 12, no. 18, pp. 1–24, 2020.

[27] S. W. Aaron Daniel Snowberger, Semin Kim, "Analysis and application of front-end code playground tools for web programming education," Journal of Practical Engineering Education, vol. 16, no. 1, pp. 11–19, 2024.

[28] X. Zhong and Z. Zhan, "An intelligent tutoring system for programming education based on informative tutoring feedback: system development, algorithm design, and empirical study," Interactive Technology and Smart Education, vol. 22, no. 1, pp. 3–24, 2024.

[29] C. Uleanya, "Reconceptualising disabilities and inclusivity for the post-digital era: Recommendations to educational leaders," Education Sciences, vol. 13, no. 1, pp. 1–12, 2023.

[30] N. Bianchi, Y. Lu, and H. Song, "The effect of computer-assisted learning on students' long-term development," Journal of Development Economics, vol. 158, no. 1, pp. 10–29, 2022.

[31] S. Nayak, R. Agarwal, and S. K. Khatri, "Automated assessment tools for grading of programming assignments: A review," in 2022 International Conference on Computer Communication and Informatics (ICCCI), 2022, pp. 1–4.

[32] Y. Zhong, M. Shi, Y. Xu, C. Fang, and Z. Chen, "Iterative android automated testing," Frontiers of Computer Science, vol. 17, no. 1, pp. 175–212, 2023.

[33] D. Pramod, T. De Silva, U. Thabrew, R. Shariffdeen, and S. Wickramanayake, "Bugsphp: A dataset for automated program repair in php," in Proceedings of the 21st International Conference on Mining Software Repositories, 2024, p. 128–132.

[34] F. Kaddari, I. Ouahbi, and H. Darhmaoui, "Gamification approach in teaching web programming courses in php: Use of kahoot application," International Journal of Modern Education and Computer Science, vol. 13, no. 1, pp. 33–39, 2021.

[35] E. W. M. Barnard and S. McDonald, "Learning objectives and their effects on learning and assessment preparation: insights from an undergraduate psychology course," Assessment & Evaluation in Higher Education, vol. 46, no. 5, pp. 673–684, 2021.

[36] L. T. Yu, "A comparison of the autonomous use of technology for language learning for efl university students of different proficiency levels," Sustainability (Switzerland), vol. 15, 2023.

[37] A. Razaque, A. A. Ajlan, N. Melaoune, M. Alotaibi, B. Alotaibi, I. Dias, A. Oad, S. Hariri, and C. Zhao, "Avoidance of cybersecurity threats with the deployment of a web-based blockchain-enabled cybersecurity awareness system," Applied Sciences (Switzerland), vol. 11, 2021.

[38] K. Sambhus and Y. Liu, "Automating sql injection and cross-site scripting vulnerability remediation in code," Software, vol. 3, 2024.


YAN WATEQULIS SYAIFUDIN is an Associate Professor and Head of Applied Informatics Laboratory in Department of Information Technology, State Polytechnic of Malang. He received the bachelor degree in Informatics from Bandung Institute of Technology, Indonesia, in 2003, the master degree in Information Technology from Sepuluh Nopember Institute of Technology, Surabaya, Indonesia, in 2011, and Ph.D. degree in Information and Communication Systems from Okayama University, Okayama, Japan, in 2021, respectively. In 2005, he joined State Polytechnic of Malang, Indonesia, as a lecturer. His research interests include learning system and technology, database technology, blockchain, and smart farming. He is member of IEEE, IAENG, and Consumer Technology Society. He can be contacted at email: qulis@polinema.ac.id

TRIANA FATMAWATI completed her bachelor's degree in Informatics Engineering from Bandung Institute of Technology in 2023. Subsequently, she continued her master's degree in Industrial Engineering and Management at the same university, graduating in 2008. Triana began her career as a lecturer in the Information Systems program at Polytechnic of STMI Jakarta under the Ministry of Industry from 2005 until 2022. In 2023, she joined State Polytechnic of Malang as a lecturer in the Information Technology Department.

AMAL DERMAWAN UDJIR completed bachelor degree in Computer Engineering from the Indonesian School of Informatics and Computer (STIKI) Malang in 2012. Currently, he is a master student in Information Technology Engineering in Politeknik Negeri Malang. His research interest ara educational technology, programming, and information system.

PRAMANA YOGA SAPUTRA is a dedicated academic and researcher with a strong foundation in Informatics Engineering. He earned his Bachelor's degree in 2010 and his Master's degree in 2014, both from Institut Teknologi Sepuluh Nopember Surabaya, specializing in Informatics Engineering and Information Technology Management, respectively. Currently a Lecturer at Politeknik Negeri Malang and his research focuses on artificial intelligence, sentiment analysis, and system design.

NOBUO FUNABIKI received the B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with the System Engineering Division, Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000-2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimization algorithms, educational technology, and Web technology. He is a member of IEEE, IEICE, and IPSJ.

MUSTIKA MENTARI completed her bachelor's degree in computer science study program from Brawijaya University in 2011. Subsequently, she continued her master's degree in informatics engineering at Sepuluh Nopember Institute of Technology (ITS), graduating in 2014. Mentari has been active in her career as a lecturer in the Diploma 4 informatics engineering study program at the State Polytechnic of Malang since 2015. In 2023, she continued to take a doctoral degree in the Graduate School of Environmental and Life Science at Okayama University.

CHANDRASENA SETIADI holds a Bachelor's degree in Electrical Engineering from Brawijaya University (2011) and a Master's degree in Electrical Engineering from the State Polytechnic of Malang (2020). His research interests focus on the Internet of Things (IoT), computer network systems, and web applications, where he explores innovative solutions to enhance connectivity and optimize digital systems.

ALFIANDI AULIA RAHMADANI completed her Bachelor of Applied Electrical Engineering degree from the State Polytechnic of Malang, Indonesia, in 2023, followed by a Master of Applied Electrical Engineering degree from the same institution in 2024. In 2024, she joined the research team at the State Polytechnic of Malang.