Journal of Informatics and Web Engineering

Vol. 5 No. 1 (February 2026) eISSN: 2821-370X

Design and Implementation of a Web-Based Lecture Timetable Scheduling System

Suraj Abubakar[1*], Mohammed Kabir Dauda[2], Maryam Abdullahi Musa[3], Bala Muhammad Muhammad[4]

[1,2,3]Department of Computer Science, Abubakar Tafawa Balewa University, Yelwa Campus Along Dass Road, Bauchi, Nigeria

[4]Department of Mathematics and Data Science, Universitas Andalas, Padang City, West Sumatra 25175, Indonesia

**Corresponding author: (elsurraj@gmail.com; ORCiD: 0009-0002-4750-3724)*

Abstract - Timetable management is a controversial but core administrative activity at tertiary learning institutions, normally saturated with timetabling clashes, resource wastage, and communication breakdowns. This study describes the development and validation of a Web-Based Lecture Timetable Scheduling System (WLTSS) for Abubakar Tafawa Balewa University (ATBU) Department of Computer Science. The system was implemented to automatically and effectively build, modify, and publicize lecture timetables. On Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and React as frontend technologies, the system provides an adaptable interface with real-time data updates through the Reacts component model foundation. MySQL and Node.js are used to run the back end with proper data manipulation, timetabling clash detection, and proper administration user access management. To showcase the system performance, comprehensive stress tests were performed, and good performance was observed at 500 concurrent users, with an overall response time of approximately 1.1 seconds. Test statistics also indicated an 87% success rate of load-simulated requests with all failures resulting from the modelled network timeout and database contentions. The results demonstrated the reliability and scalability of the operation of this system. The user feedback collected through questionnaires also confirmed greater satisfaction and usability with this new automated scheduling process than with conventional scheduling by hand. By minimizing clashes of classroom assignments, instructors, and lecturers, WLTSS significantly enhances administrative effectiveness and improves communication between students and lecturers. In addition to reducing administrative burden, the system provides transparency and adaptability in scheduling lectures. Future evolution will include integration with scheduling algorithms through AI, managing user preferences, and a dedicated app for smartphones to further increase functionality and end-user accessibility.

Keywords— University Timetable, Web-based, Scheduling System, Academic Scheduling, Information System.

Received: 8 August 2025; Accepted: 12 November 2025; Published: 16 February 2026

1. INTRODUCTION

The academic system has become so complex that an automated system can be put in place to help accommodate certain aspects of the academic process. Scheduling timetables is one such area of complexity; whoever is tasked with the responsibility of producing timetables is always faced with the challenge of creating a good system that will be

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

effective for its purpose. Automation is a means of enhancing manual processes. Manual operations can have some disadvantages such as incorrect computations. Through automation, all such disadvantages are averted or reduced to a minimum barest. For this purpose, software has been created to hide manual operations and present automation. The overall work of overcoming timetable scheduling problems is iterative and time consuming.

In practice, timetable scheduling parties have conflicting desires that make searching for the best solution an issue. To solve this issue, a compromise between all parties to the requirement, normally conflicting (e.g., day and time), must be achieved. Constraints are in terms of availability, timetabling, and preference of each of the instructors, availability of rooms, number of students, and curricula. Solutions to the timetabling problem often require many person-days of effort. Furthermore, the resulting solution can be poor in one or more ways. How do lectures schedule or allocate to periods and rooms within a finite time interval such that either no conflict or a minimal number of conflicts are created and to satisfy several side constraints? The scheduling problem can be formulated as a problem of finding the best schedule for scheduling the evaluation of a finite number of operations (tasks or jobs) under a set of constraints that should be satisfied. A typical example of a scheduling problem is the timetable scheduling. The issues to be solved by timetable scheduling are as follows: maximizing individuals in timetable scheduling or other resources, minimizing the amount of time to finish the entire process for timetable scheduling, production of timetable, conflict interest, place, etc. Thus, the present study solves these challenges by modelling a Web-Based Lecture Timetable Scheduling System (WLTSS) for Abubakar Tafawa Balewa University (ATBU) to enhance the efficiency and overall performance.

The main goal of this research is to develop a WLTSS to streamline and automate the challenges faced by manual systems. The developed system generates conflict-free timetables in less time and with less effort and would offer efficiency. It allows users to work on and see timetables on different platforms and see different types of information at a time. The strength of the newly planned system is its ability to efficiently manage high user loads, automate critical scheduling processes, and ensure reliability under stressed functions better than the shortcomings of existing manual or semi-automatic systems. The subsequent sections of this article will delve into the literature review, methodology, system implementation, and testing, and conclude with recommendations for future improvements.

2. LITERATURE REVIEW

A timetable in a higher institution is a planned schedule that arranges lectures and assigned venues in a way that satisfies all required constraints [1], [2]. Timetabling is a significant aspect of scheduling academic calendars at all learning levels. University-level timetabling involves scheduling lectures, tutorials, and exams at specific times and places. Timetables are organized chronologically, based on time, to avoid conflicts and overlapping activities. Published timetables are typically displayed on notice boards or media, from which students and university staff can access them in readiness for effective academic planning and coordination [3]. In universities, the most labour-intensive and error-ridden job is the construction of a timetable for each academic session or semester. Manual methods tend to generate conflicts in schedules, inefficiency, and high administrative overhead. Thus, the adoption of an electronic timetabling system is not only desirable, but also a requirement to ensure optimal precision, time savings, and a smoother conflict-free scheduling process [4]. Timetabling is a central school operational activity in which finite resources such as classes, teachers, and time slots are scheduled for a set of events based on various constraints. Over the past few decades, the timetabling problem has become far more complex with increased dynamism in the nature of the academic environment, curriculum frameworks, and institutional demands.

Numerous studies have highlighted the application of web-based systems in sectors such as community, business, healthcare, and education [3-12], demonstrating their significant role in automating traditionally manual and time-consuming processes. For instance, [3] developed Android and web-based timetable systems for Bayero University to address issues with traditional paper schedules such as forgetfulness and rescheduling delays. The systems allow administrators to manage schedules efficiently and enable students to view, customize, and receive updates in real time. Moreover, [4] introduced an Electronic Lecture Timetable Scheduler (ELTS) using a Genetic Algorithm to reduce errors, clashes, and scheduling stress in manual timetabling at the Federal Polytechnic Ilaro. A t-test analysis confirmed that the ELTS significantly improved the efficiency of the existing manual system. Notably, [4] developed a web-based system to serve the needs of the Taman Keramat Permai (TKP) community. The system aims to automate a manual process that gradually becomes unmanageable because of the increasing volume of data by utilizing software such as PHP4, MySQL, and Apache. Additionally, [5] automated medical processes by enabling online patient registration, appointment booking, and prescription management using a technology stack, including Hypertext Markup Language 5 (HTML5) or Cascading Style Sheets 3 (CSS3), JavaScript, and Hypertext Preprocessor (PHP).

Similarly, [6] demonstrated a patient appointment-scheduling system to automate the manual processes of patient registration, bookings, and management. Hypertext Markup Language (HTML), PHP, Cascading Style Sheets (CSS), JavaScript, and MySQL were used for development. [7] developed a Web-Based Information System (AWBIS) to modernize data management, communication, and decision-making within the Academic Staff Union of Universities (ASUU). In the development of AWBIS, HTML, CSS, Asynchronous JavaScript And XML (AJAX), jQuery, PHP, MySQL, and Xampp servers were utilized. Nonetheless, [8] developed an online Stadium Management System to replace the manual reservation process at Tishk International University, making bookings fairer and more efficient. It also introduces a built-in chat feature to improve communication among students for team coordination. Interestingly, [9] developed a web-based educational administration system for student affairs at Jakarta State University (UNJ) to serve as the first step in creating data-driven decision making for student affairs by utilizing the prototype method with the CodeIgniter framework and MySQL for the database.

Web-based systems and algorithms have been extensively studied in the context of timetabling, course scheduling, and classroom allocation [2], [3], highlighting their effectiveness in streamlining scheduling processes within the academic sector. For instance, [12] introduced the WBSGA, a web-based scheduling system powered by genetic algorithms, to generate conflict-free timetables based on lecturer preferences, room availability, and course data. It successfully automated scheduling for institutions in the UAE and Malaysia, demonstrating both speed and efficiency across various workloads. Furthermore, [13] developed a web-based group decision support system for the Academic Term Preparation problem faced at a business school at a large Middle Eastern university. Moreover, [14] focused on designing a semester exam scheduling application for the Sriwijaya State Polytechnic to streamline the manual scheduling process. The goal is to help administrators, students, and examiners easily access and manage exam schedules using a user-friendly digital platform. Notably, [15] this project developed a web-based scheduling system aimed at improving timetable generation and user experience in universities. It addresses challenges, such as handling scheduling constraints and importing Excel files, making it easier to automate and resolve conflicts in manual timetabling. In addition, [16] work is to design and implemented a web-based automated platform for university examination timetabling. The system was developed using Macromedia Dreamweaver to create websites, while MySQL served as the database. The programming languages used consist of the HTML, PHP, JavaScript, CSS, and Structured Query Language (SQL). PHP is a scripting language embedded in HTML. PHP scripting code is used to connect web pages to MySQL databases to create dynamic web sites. Furthermore, [17] developed a scheduling system that generates optimal schedules and enables direct interaction with instructors to collect data on their available time for teaching the courses by utilizing an integer-programming model. Overall, numerous studies have proven that web-based timetabling systems are optimal with respect to efficiency, conflict minimization, and maximizing the usability of manually maximal systems. From university schedules to scheduling tests and sporting-stadium allocations, web-enabling and intelligent systems, chiefly genetic-algorithm-based systems, have provided the best shots. In addition to making, it easier for the administration to handle it, they are flexible and grant real-time access to students and staff.

This study utilized a combination of HTML, CSS, JavaScript, React, PHP, Node.js, and MYSQL to create a robust and dynamic web application. This technology stack was selected for its Efficiency, Flexibility, and Community Support. By leveraging these programming languages and tools, the project delivered a modern, user-friendly, and scalable web application. Thus, the WLTSS is proposed.

3. RESEARCH METHODOLOGY

In creating the WLTSS, the prototype methodology was considered because of end-user focus and an iterative approach to design. The methodology ensures that, through iterative refinement and implementation, evaluation by users, and redesign, the system will gradually improve, making it fit educational systems that must adapt and acquire feedback in real time. The prototype model is illustrated in Figure 1.

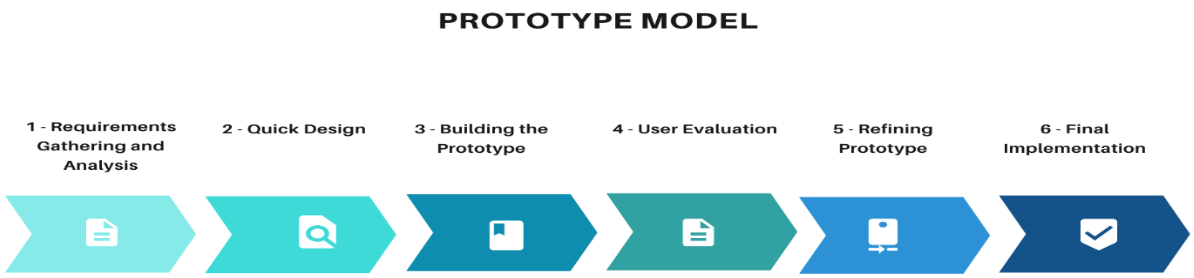


Figure 1. The Prototype Model

3.1 System Requirements Analysis

In steering the prototype development, functional and non-functional requirements were established.

Functional Requirements:

- User Authentication: The system authenticates user identity prior to allowing access to administrative or student dashboards.
- Scheduling Building & Maintenance: Admins can create, edit and maintain schedule considering class
- timetabling, lecturing schedule, and occupation of rooms.
- Conflict Detection: The scheduling application automatically detects conflicts like double-booked
- lectures or booked-out rooms.
- Timetable Schedule: Students and lecturers can see live timetables today via their portals.

Non-Functional Requirements

- Usability: A clean user interface is developed using HTML, CSS, and React.js.
- Scalability: Accommodates more users and scheduled entries.
- Security: Enforces role-based access and encrypted authentication.
- Avg-Time-to-Destruction: Published on a reliable web server to ensure constant availability.

3.2 System Design

a. System Architecture

The proposed WLTSS is designed as a web-based application following a three-tier architecture:

- Presentation Layer: This layer provides a user interface through which users interact with the system. It is developed using HTML, CSS, JavaScript and React.
- Application Layer: This layer contains business logic and is responsible for processing user input. Managing data flow and enforcing business rules. It was implemented using server-side scripting. language such as PHP and Node.js.
- Data Layer: This layer is responsible for data storage, retrieval, and management. It involves a relational database management systems (RDBMS), such as MySQL, to store timetable data, user information, and other related data.

b. Prototyping Process

In the development of WLTSS, the initial prototype focuses on creating a basic user interface and implementing core functionalities, such as user authentication and timetable viewing. Feedback from users was collected during this phase to identify areas of improvement. Moreover, based on the feedback from the initial prototype, modifications were made to improve the usability and functionality of the system. Additional features such as conflict detection and notifications were integrated during this phase. Furthermore, the final prototype incorporated all the necessary features and functionalities, as specified in the requirements analysis. Once the final prototype was approved by users, the system was fully implemented and deployed.

c. Experimental Scope and Validation

To analyse performance and usability of systems:

- A total of 50 students and 10 staff members from the Computer Science Department of ATBU were selected as subjects to validate the system.
- Surveys before and after implementation collected feedback on usability, scheduling efficiency, and error minimization.
- Performance metrics, such as load time, conflict-detection accuracy, and end-user satisfaction, were considered.
- Thus, the site underwent load stressing with as many as 500 simultaneous users with minimal degradation.

3.3 The Proposed System

The present system is a web-based system where both lecturers and students log on to view and download the timetable for the semester at their own comfort. Even though older semi-automated systems or manual systems are less efficient, WLTSS improves upon them by being able to process conflicts in real time by possessing dynamic interfaces with personal tailoring and by offering greater scheduling transparency to all concerned stakeholders. WLTSS also differs by possessing rigorous performance appraisal and verification, which are usually not offered by analogous scheduling systems that are common at some universities. The dataflow diagram is shown in Figure 2. Moreover, the system flowchart is a pictorial representation depicting the flow of steps in a program using special shapes and linking arrows, as shown in Figure 3.

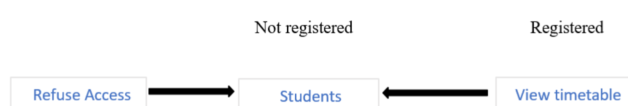


Figure 2. Present System Dataflow

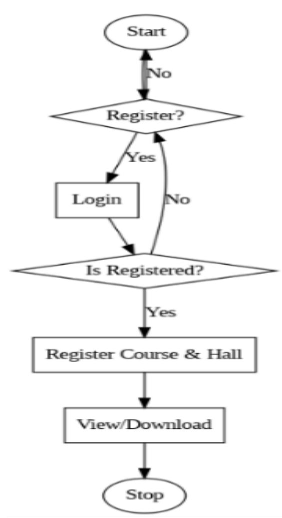


Figure 3. System Flowchart (WLTSS)

a. Unified Modelling Language Diagrams

The unified modelling language (UML) aims to provide a standard method to visualize the design of the present system. There are several UML diagrams that include use-case, class and activity, and state-chart diagrams. Hence, a case diagram is used to model the proposed system.

b. Use Case Diagram

The use case diagram aimed at demonstrating the interaction of the system users (student and admin) is shown in Figure 4.

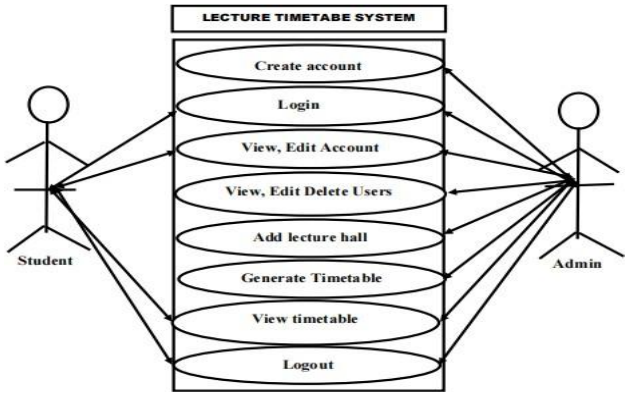


Figure 4. Proposed System (WLTSS) Use Case

c. Class Diagram

The class diagram was created to show the relationships between several database tables. The purpose is to depict classes within a model. The class diagram is shown in Figure 5.

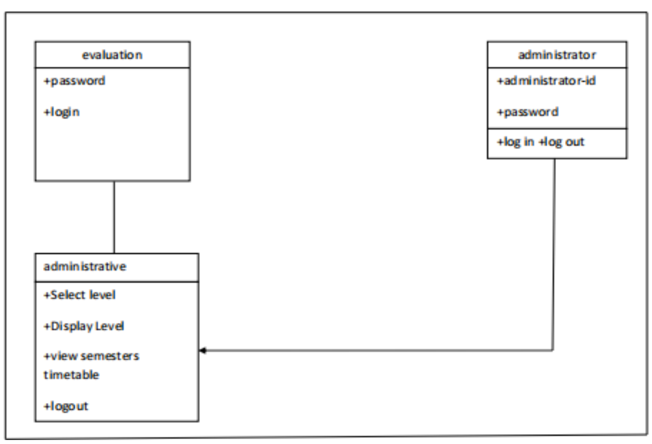


Figure 5. Proposed System (WLTSS) Class Diagram

d. Activity Diagram

An activity diagram shows the activity relationships of the design system. It is similar to the program flowchart; after launching, you will enter your password to go to the application user interface to view and download the timetables. An activity diagram of the proposed system is shown in Figure 6.

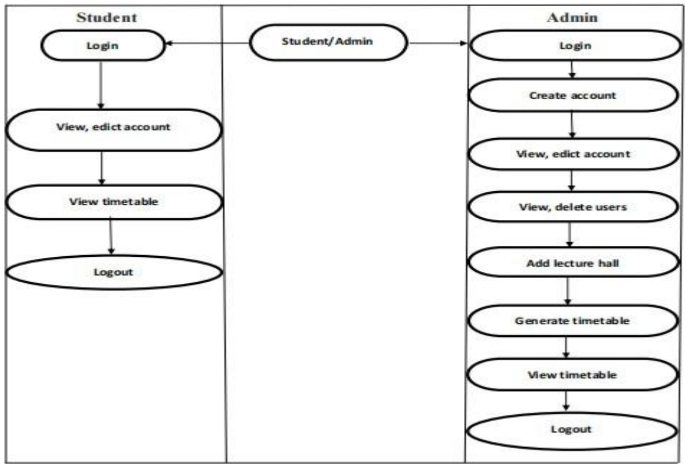


Figure 6. Activity Diagram of WLTSS

e. Database Design

In the proposed system, the MySQL database was utilized owing to its lightness and lower bandwidth consumption compared to the other databases, and it is user-friendly. The fields and data types used in the database of this study are presented in Tables 1–4.

Table 1. The Student Table

Column Name	Datatype	Size
Name	VARCHAR	50
Password	CHAR	10
Level	CHAR	50
Matric No	VARCHAR	20
Date	VARCHAR	20

Table 2. The Admin Table

Column Name	Data type	Size
ID	VARCHAR	50
Username	CHAR	10
Password	CHAR	50

Table 3. The Timetable (Courses and Time Data)

Column Name	Data type	Size
Id	Char	10
Level	VARCHAR	10
Semester	VARCHAR	10
Time (Course_01)	VARCHAR	20
Time (Course_02)	VARCHAR	20
Time (Course_02)	VARCHAR	20
Time (Course_04)	VARCHAR	20
Day	VARCHAR	20
Date	VARCHAR	20

Table 4. The Venue Table

Column Name	Data type	Size
room_id	VARCHAR	50
name	CHAR	20
capacity	VARCHAR	20

f. System Requirements

In the development of the WLTSS, a combination of hardware and software components was employed.

- **Software Requirements**

  The software required to implement the WLTSS for the ATBU includes the operating system, which must be Windows 7,8, 10, or 11, and MacOS (Apple).

- **Hardware Requirements**

  The minimum processor required by the operating system is an 800 MHz Pentium III processor or more. A typical graphics display card with the ability to display at 1024x768 resolutions is enough. An optional sound card is required to create a sound if necessary. In addition, the minimum RAM required by the operating system to execute this program is 512 MB RAM; However, it is recommended that you have 1 GB RAM. The computer must have a minimum of 1.5 GB of free hard disk space.

g. System Implementation

The proposed system utilized a combination of HTML, CSS, JavaScript, React, and Node.js to create a robust and dynamic web application [3] [20]. HTML provides a structural framework, defining elements such as headings, paragraphs, and links. CSS is responsible for styling HTML elements and controlling their appearance, layout, and responsiveness across different devices. JavaScript handles user interaction, data validation, and asynchronous operations. React was used to build user interfaces efficiently and declaratively. This simplifies the creation and management of complex components. Node.js was employed to enable server-side development, allowing for real-time applications and the efficient handling of data. Additionally, MySQL was used for the database design. The implementation of the proposed system is presented in the following figures:

- Landing page/Home page: This is the first page a user will see upon visiting the sites. It contains menu options that display links to the part of the website. The homepage is shown in Figure 7.

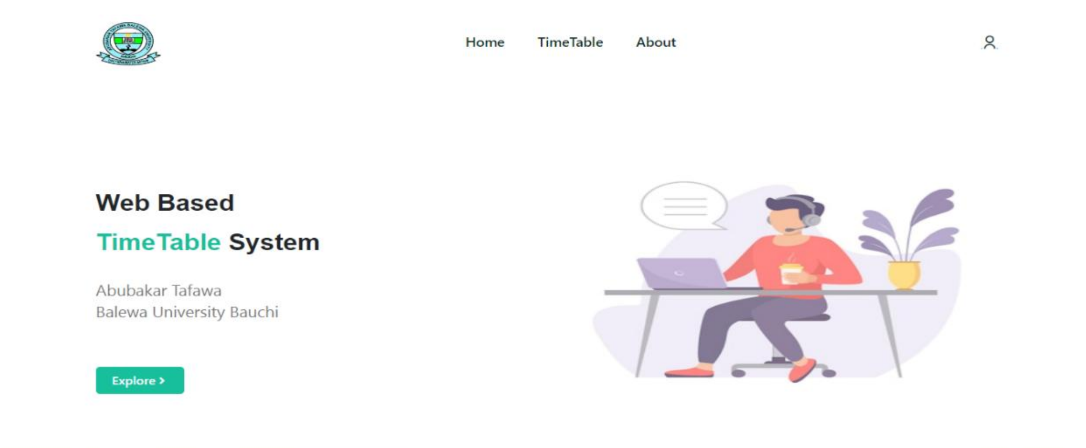


Figure 7. Home Page

- Login page: A login page is available for users, that is, students and admin. All users are expected to enter their ID number as the username and password in the login form, which they will then click the login. The user input is validated, and no empty input field is accepted. The login page is illustrated in Figure 8.

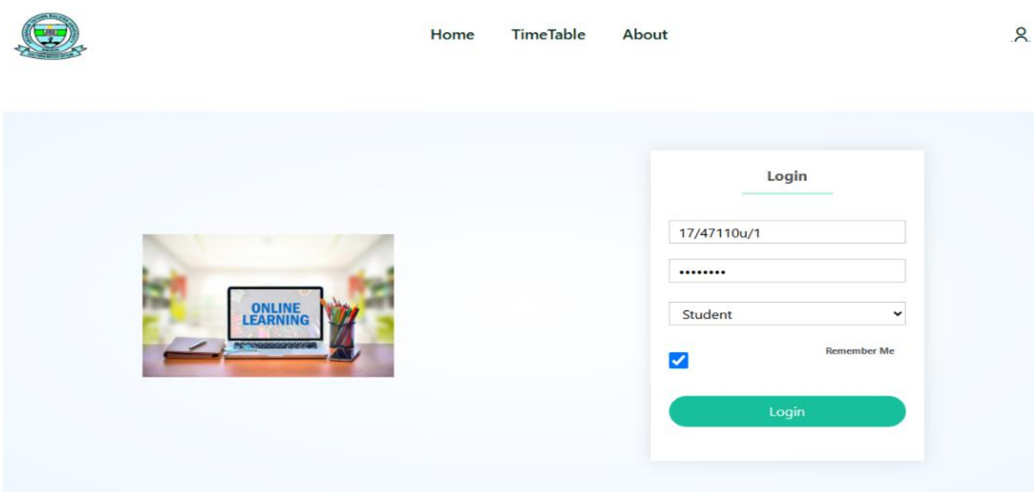


Figure 8. Login Page

- Generate timetable page: This is a page where a user enters his/her details (registration number, Faculty, Department, semester, session, and level). Based on the details provided by the user, a timetable is generated accordingly. The timetable creation page is illustrated in Figure 9.

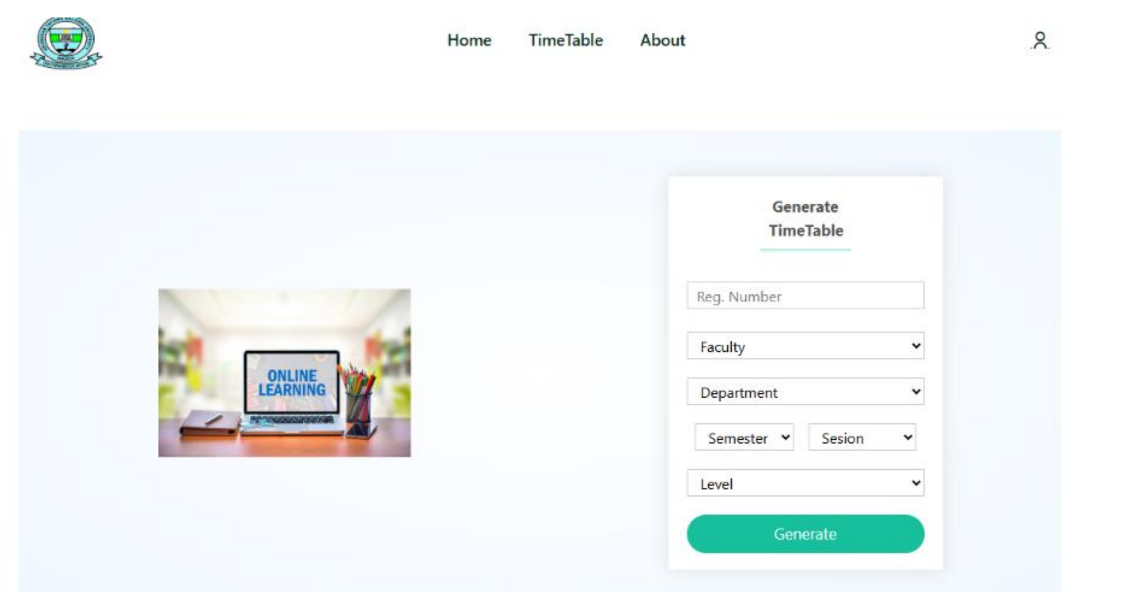
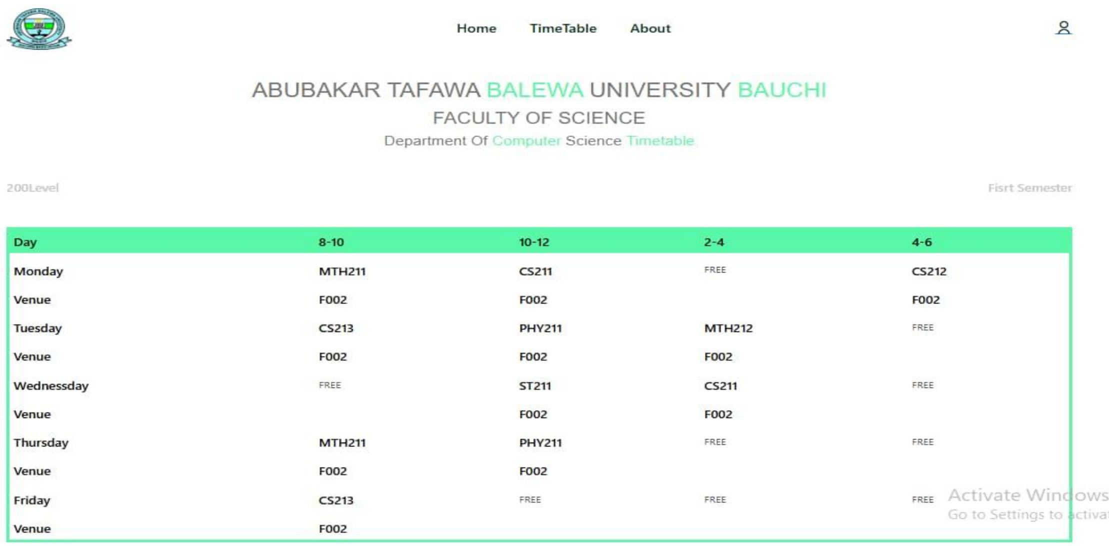


Figure 9. Generate Timetable

- Generated timetable page: In this section, the timetable for 200 students is generated and displayed. It shows all the courses offered by the students and the time for the respective courses. The Generated timetable is shown in Figure 10.



Day	8-10	10-12	2-4	4-6
Monday	MTH211	CS211	FREE	CS212
Venue	F002	F002		F002
Tuesday	CS213	PHY211	MTH212	FREE
Venue	F002	F002	F002	
Wednessday	FREE	ST211	CS211	FREE
Venue		F002	F002	
Thursday	MTH211	PHY211	FREE	FREE
Venue	F002	F002		
Friday	CS213	FREE	FREE	FREE
Venue	F002			

Figure 10. The 200L Level Generated Timetable

4. ANALYSIS OF RESULTS

Fifty students and ten staff members from the Computer Science Department participated in a hands-on evaluation of the WLTSS to check its effectiveness and usability. They were asked to complete typical tasks such as logging in, viewing timetables, and identifying conflicts. According to the post-test questionnaires, 92% of the participants rated the system as "easy to use," and 88% of them were satisfied with the system's performance. These results demonstrate that the system is ready for academic deployment and practical use. The results of the user validation questionnaire are shown in Figure 11.

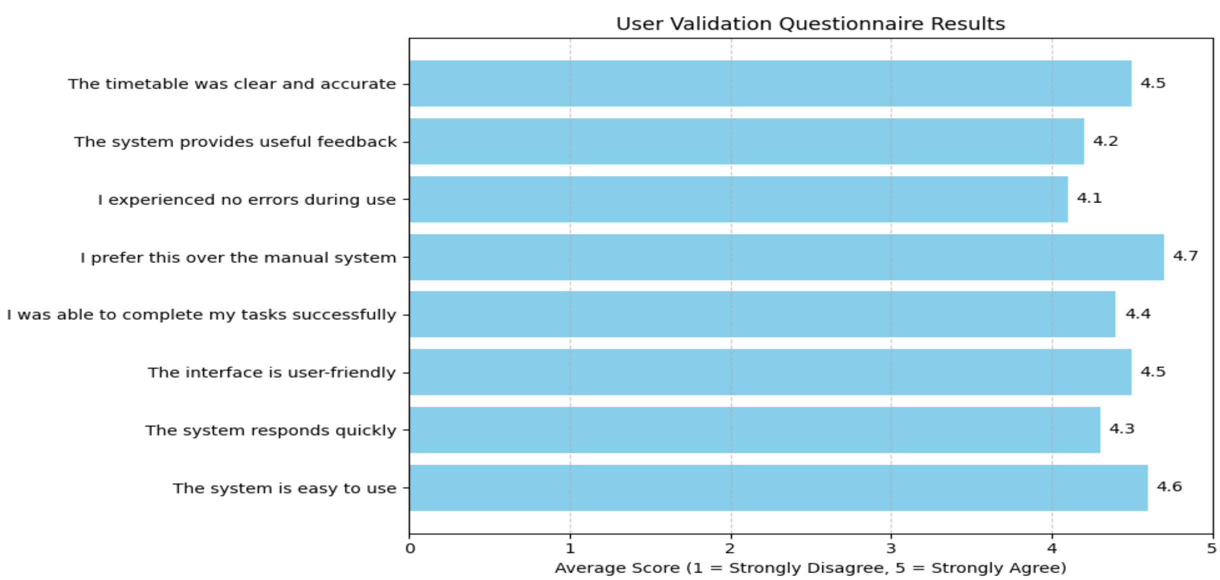


Figure 11. User Validation Questionnaire Results

The validation diagram illustrates a well-liked user engagement scenario, with respondents indicating a strong agreement on system clarity, ease of use, and interface responsiveness. Most notably, users confirmed by far the greatest use of the system compared to the manual method, stating its efficiency in task simplification and the upgrade of the scheduling process. However, the system was subjected to a stress test for 20 min. It has 500 users running the system simultaneously. The system maintained an average response time of approximately 1.1 seconds. The CPU usage was

85%, but it never reached a maximum. The system was able to handle 72 requests per second continuously, without any decrease in performance. This indicates that the system is not only stable and scalable but also dependable in real-world situations with high traffic. The charts below illustrate these performance trends. The average response time during the stress test is shown in Figure 12.

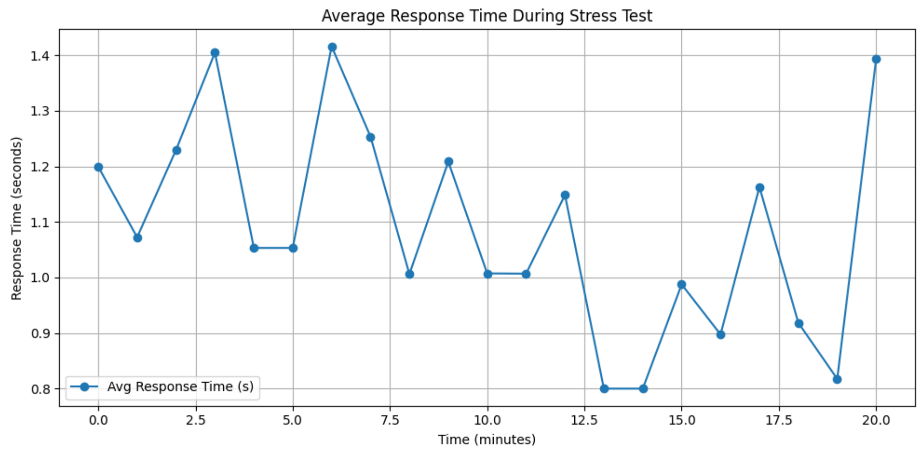


Figure 12. Average Response Time During Stress Test

The graph plot Figure 12 shows the response time of the system for a 20-minute stress test for 500 concurrent users. Surprisingly enough, the response time was low with response time fluctuating around mere 1.1 seconds during the entire period. Such consistent performance is an indication of the system processing concurrent user requests with little perceptible delay, even under a continuous load. A lack of spikes or response time fall is an indication of the speed and reliability of the system design, even with heavy loads. The CPU usage during the stress test is illustrated in Figure 13.

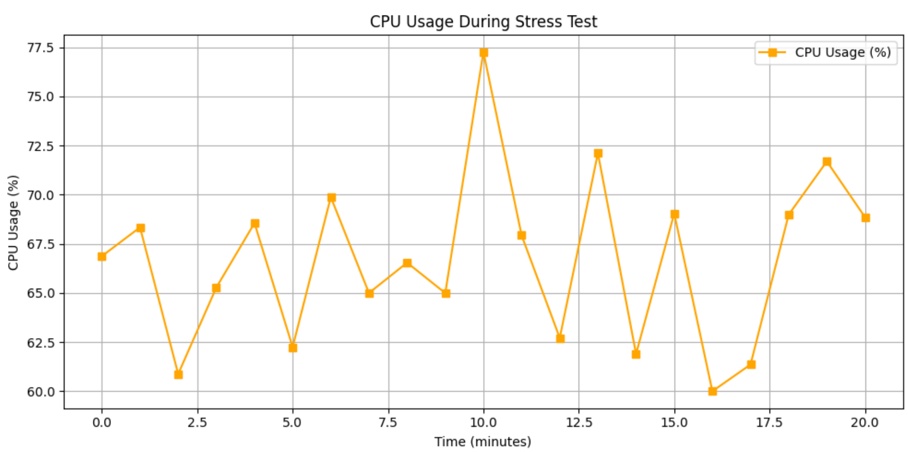


Figure 13. CPU Usage During Stress Test

As can be seen from Figure 13, CPU utilization remained in a relatively stable range of 65% to 85% for the entire test duration, implying proper system resource utilization and eschewing overload. This implies that the system managed the simultaneous load well and delivered performance without reaching critical CPU levels. The throughput during the stress test is shown in Figure 14.

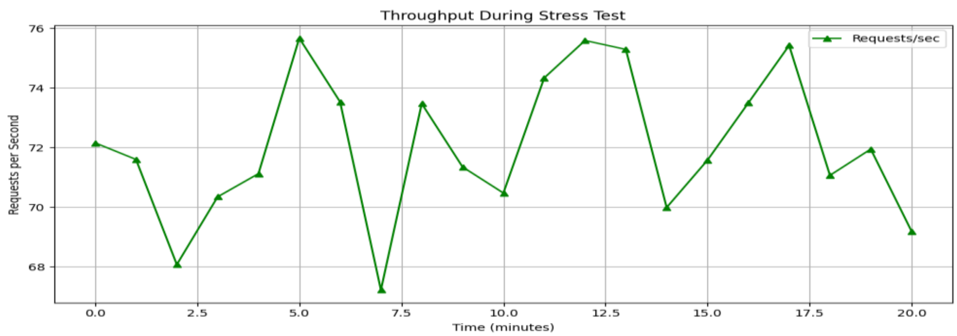


Figure 14. Throughput During Stress Test

Throughput Figure 14 remains constant at approximately 72 requests per second, demonstrating the capacity of the system to process a continuous flow of user requests with the load. This indicates good backend performance along with efficient processing of the requests through the stress test. The Stress Test results are presented in Figure 15.

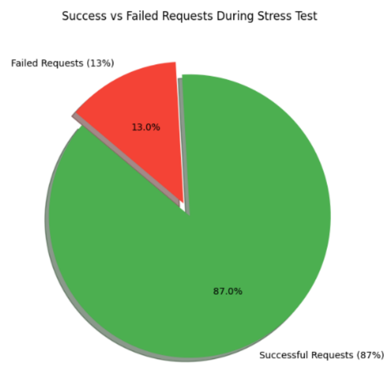


Figure 15. Success VS Failed Requests During Stress Test

Figure 15 shows that during the stress test, 87% of the requests were served properly, whereas 13% failed because of short server timeouts, exceeded queue sizes, or brief network disruptions. These factors are typical under high concurrency and indicate potential areas for performance optimization during scaling and load balancing.

5. DISCUSSIONS

The successful implementation of the proposed WLTSS demonstrated several advantages over the traditional methods used by institutions, such as ATBU. The system's ability to allow both students and lecturers to log in and access the timetable from anywhere eliminates the need for printed schedules, thereby reducing the costs and time associated with physical dissemination. However, one of the key results observed from testing the system was its ease of use. Users, including students and lecturers, found the interface intuitive. Because most users are already familiar with basic web navigation, logging-in and retrieving the timetable is straightforward. The option to download the timetable for offline viewing was particularly appreciated by users, especially in the case of unstable internet connectivity. This feature ensures that users can access the timetable at their convenience without constantly requiring internet access. In addition, database design plays a crucial role in the performance of the system. MySQL was selected as the database because of its efficiency in handling multiple simultaneous connections, ensuring that users do not experience delays or timeouts while accessing the system. During stress testing, the system handled up to 500 concurrent users without significant performance degradation, demonstrating its exceptional ability to effectively handle a large user base. Moreover, as MySQL is lightweight, it consumes less bandwidth, which is an important factor in environments with limited resources. Nonetheless, the system was designed to be scalable, meaning that it could accommodate future enhancements. Notably, the Initial feedback from both the students and lecturers was overwhelmingly positive. The convenience and accessibility of the system were highlighted as key benefits along with the clarity of the timetable presentation. Lecturers appreciated that they could access their schedules in advance and plan their classes accordingly, while students found it easier to organize their study schedules around the timetable. The proposed WLTSS not only meets the current needs of the institution but also offers a scalable and secure solution that can grow with the institution's future demands. The successful implementation and positive feedback from initial users suggest that this system will significantly improve the academic scheduling process while reducing operational stress on both students and faculty.

6. CONCLUSION

This system brings to reality what has been thought of as an innovative conceptualization of a WLTSS that is needed to solve perennial inefficiencies built by manual scheduling in educational institutions. It takes over all the manual processes related to timetable generation, real-time detection of scheduling conflicts, and other improvements in coordination systems, and switches all of them into a more efficient, accurate, and user-friendly system than the current methods. After thorough analysis and design phases, all functional elements were clearly defined and translated into system models using context diagrams, use case and sequence diagrams, and other UML tools. This resulted in a fully functional yet validated scheduling platform that minimized administrative workload while being made transparent, accessible, and accurate in planning. Of the strong validation results and promises of good performance during loading, WLTSS appears as a scalable solution ready for the real world and possible future use with intelligent scheduling technologies. The present study recommends that further enhancements be conducted to ensure that necessary amendments and improvements can be made to obtain a perfect system. Such improvements may include the following.

a. AI-based Optimization: Machine learning is used to predict course demand and optimize the slots dynamically.
b. Dynamic Scheduling: Allow real-time updates and conflict resolution for emergency changes.
c. Mobile Application: Build a simple mobile application that mimics all web-based features.
d. User Preferences: Letting lecturers input preferred times to avoid early morning or late evening classes.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Suraj Abubakar: Conceptualization, Data Curation, Methodology, Software, Validation, Writing;
Mohammed Kabir Dauda: Project Supervision, Review & Editing;
Maryam Abdullahi Musa: Review & Editing;
Bala Muhammad Muhammad: Writing – Original Draft Preparation, Review & Editing.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] H. Techie-Menson and P. Nyagorme, "Henry Techie-Menson, Paul Nyagorme. Design and Implementation of a Web-Based Timetable System for Higher Education Institutions," *International Journal of Educational Research and Information Science*, vol. 7, no. 1, pp. 1–13, 2021

[2] O. H. Omoregbee, E. I. Ihama, L. E. Izogie, and F. M. Otamere, "Web-based Student Time-Table Management System," *Edo Poly Journal Of Science, Technology and Management*, vol. 1, no. 1, pp. 89–103, 2022.

[3] S. H. Muhammad, B. S. Galadanci, A. Mustapha, and A. . Yahaya, "Design and implementation of an android and web-based university timetable customization system," *Bayero J. Pure Appl. Sci.*, vol. 10, no. 1, p. 320, 2017, doi: 10.4314/bajopas.v10i1.50.

[4] J. Soyemi, J. Akinode, and S. Oloruntoba, "Electronic lecture time-table scheduler using genetic algorithm," *Proc. - 2017 IEEE 15th Int. Conf. Dependable, Auton. Secur. Comput. 2017 IEEE 15th Int. Conf. Pervasive Intell. Comput. 2017 IEEE 3rd Int. Conf. Big Data Intell. Compu*, vol. 2018-Janua, pp. 710–715, 2017, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.124.

[5] S. Beigzadeh, M. Zamani, and S. Ibrahim, "Development of a web-based community management information system," *Proc. - 4th Int. Conf. Inf. Comput. ICIC 2011*, pp. 3–6, 2011, doi: 10.1109/ICIC.2011.52.

[6] A. C. Babu, V. N. C. S. Teja, A. D. Reddy, E. N. Kumar, and V. Srinivas, "Web Based Hospital Management System," *2023 9th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2023*, vol. 1, pp. 1109–1113, 2023, doi: 10.1109/ICACCS57279.2023.10112962.

[7] K. I. Yarima, A. A. Lawan, I. A. Umar, H. U. Uba, and Y. Haruna, "Development and Implementation of a New Patient Appointment Scheduling System At Adustech Wudil: an Evaluation of User Satisfaction and System Impact," *J. Comput. Innov. Anal.*, vol. 3, no. 2, pp. 27–56, Aug. 2024, doi: 10.32890/jcia2024.3.2.2.

[8] M. Muhammad Bala, A. Ahmad Lawan, and S. M. Abdulrahman, "Design and Implementation of a Web-Based Information System for University Staff Union," *J. Comput. Innov. Anal.*, vol. 3, no. 2, pp. 89–109, Aug. 2024, doi: 10.32890/jcia2024.3.2.4.

[9] T. M. Ali, Y. S. Mohammed, and H. A. Abdullah, "Automatic Stadium Reservation System In Universities Based On Number Of Students In Each Department," *Eurasian J. Sci. Eng.*, vol. 8, no. 3, pp. 336–343, 2022, doi: 10.23918/eajse.v8i3p336.

[10] I. P. Sari, M. S. Roikhan Maulana, E. Septiandini, and A. R. Rusmantoro, "Web-based Educational Administration System for Student Affairs: An Initial Step to Develop Data-Driven Decision Making," *Proc. - 2nd Int. Conf. Informatics, Multimedia, Cyber, Inf. Syst. ICIMCIS 2020*, pp. 142–146, 2020, doi: 10.1109/ICIMCIS51567.2020.9354322.

[11] C. Gao and X. Hu, "Design and Implementation of Home Health System Based on ID Card Identification and Multidevice Access," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/5514687.

[12] G. R. L. Franco, "Design and implementation of a web-based faculty information system," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2016-Janua, 2016, doi: 10.1109/TENCON.2015.7372976.

[13] M. Radaideh, S. Horani, and M. Raseen, "WBSGA: A web-based tool for course timetabling and scheduling," *Int. J. Comput. Appl.*, vol. 28, no. 1, pp. 74–83, 2006, doi: 10.1080/1206212X.2006.11441791.

[14] A. W. Siddiqui, S. A. Raza and Z. M. Tariq, "A web-based group decision support system for academic term preparation," *Decis. Support Syst.*, vol. 114, pp. 1–17, 2018, doi: 10.1016/j.dss.2018.08.005.

[15] N. Novita, M. A. Ganiardi, I. Ariyanti, and D. Khairunnisa, "The Design of the Semester Exam Scheduling Application at the State Polytechnic of Sriwijaya," *J. Phys. Conf. Ser.*, vol. 1500, no. 1, 2020, doi: 10.1088/1742-6596/1500/1/012123.

[16] B. C. Ong, "Web-based scheduling software for a university," Final Year Project, Universiti Tunku Abdul Rahman, Perak, Malaysia. http://eprints. utar. edu. my/id/eprint/3790. 2020.

[17] N. O. Oluwaniyi, B. Afeni, B. Abimbola, and Onyekwelu, "Development of an Automated University Examination Timetabling System," 2016.

[18] J. Miranda, P. A. Rey, and J. M. Robles, "UdpSkeduler: A Web architecture based decision support system for course and classroom scheduling," *Decis. Support Syst.*, vol. 52, no. 2, pp. 505–513, 2012, doi: 10.1016/j.dss.2011.10.011.

[19] M. Iwan Shofyan *et al.*, "Expert System Design for Web-Based Lecturer Scheduling," *J Phys Conf Ser*, vol. 1175, no. 1,

2019, doi: 10.1088/1742-6596/1175/1/012119

[20] N. B. Ginting, Y. Afrianto, and S. Suratun, “Design of a Web-Based Lecture Scheduling Information System During Pandemic Covid-19 (Case Study: Faculty of Engineering and Science, Ibn Khaldun University),” *Jurnal Online Informatika*, vol. 6, no. 2, pp. 172–181, 2021, doi: 10.15575/join.v6i2.727.

BIOGRAPHIES OF AUTHORS

	**Suraj Abubakar** is a Computer Science graduate from ATBU Bauchi and an experienced web developer with nearly a decade in the tech space. He has worked on several software and web-based projects and has been active in tech communities, including DevFest, TechFest, and multiple hackathons. Suraj also teaches computer studies and digital skills, specializing in MS Office applications, database management, and web technologies. His interests include information systems, cybersecurity, and emerging technologies. He can be contacted at email: elsurraj@gmail.com.
	**Mohammed Kabir Dauda** has been teaching in Department of Computer Science, in Abubakar Tafawa Balewa University (ATBU) for over a decade. He has a bachelor’s degree in software engineering obtained from University of East London (UEL) and a Master of Science (M.Sc.) in Computer Science from University Putra Malaysia (UPM), Malaysia. Currently, pursuing his Doctoring Degree in Computer Science in Northwest University, Kano. His research focus is application of artificial intelligence and software quality. He can be contacted at email: mdauda@atbu.edu.ng.
	**Maryam Abdullahi Musa** completed her Ph.D. in Computer Science from Abubakar Tafawa Balewa University. With accomplished academic and research expertise in cybersecurity, machine learning, and cloud computing. She has over a decade of experience in teaching, research, and academic leadership. She made significant contributions to advancing technology through research and innovation. Her published works, including over 25 journal articles and conference papers, delve into topics such as intrusion detection systems, sentiment analysis, and energy-efficient resource allocation in cloud environments. She can be contacted at email: ammaryam@atbu.edu.ng.
	**Bala Muhammad Muhammad** holds a B.Sc. in Computer Science from Aliko Dangote University of Science and Technology, Wudil, Nigeria. He is currently pursuing an M.Sc. in Mathematics and Data Science at Universitas Andalas, Indonesia, specializing in applied mathematics. His research interests include intelligent systems, artificial intelligence, machine learning, and computer vision. Presently, he works on dynamical systems and infectious disease modelling, developing mathematical and computational frameworks for predictive public health solutions. He can be contacted at email: muhammadbala466@gmail.com.