Indonesian Journal of Electrical Engineering and Computer Science
Vol. 42, No. 2, May 2026, pp. 337~348
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v42.i2.pp337-348

Design of a cost-effective online experimental platform for electrical experiments using a Raspberry Pi-based system

Abdelkrim Benali[1], Somia Benali[1,2], Benameur Hemidi[1]
[1]Department of Electrical Engineering, Institute of Technology, Nour Bachir El-Bayadh University Center, El Bayadh, Algeria
[2]Department of Commercial Sciences, Faculty of Economic, Commercial and Management Sciences,
Ahmed Zabana Relizane University, Relizane, Algeria

Article Info

Article history:

Received Dec 20, 2025
Revised Mar 15, 2026
Accepted May 26, 2026

Keywords:

Electronics education
Flask–Django architecture
IoT
Online experimental platform
Online learning platform
Raspberry Pi

ABSTRACT

Following the COVID-19 pandemic, online learning platforms have become vital for supporting distance education. This work presents LABTEC, an online Experimental Platform for electronics education that enables students to manipulate real hardware through a learning management system (LMS). The platform allows remote execution of experiments with electronic circuits and instruments, such as oscilloscopes, providing hands-on practice over the Internet in real time. The main contributions of this work are threefold: (i) a hybrid Flask–Django server architecture, where flask manages instrument-level control and Django provides secure and scalable web services; (ii) the use of a Raspberry Pi gateway as a cost-efficient and versatile hardware interface; and (iii) an open-source remote laboratory framework experimentally validated to support real-time interaction with average end-to-end latency below 50 ms, stable multi-user access, and low resource utilization. Experimental results demonstrate reliable operation under concurrent user scenarios, achieving consistent measurement visualization and control with reduced deployment cost compared to proprietary and institution-centric remote laboratory platforms. Performance evaluation shows a control latency below 50 ms for closed-loop tasks, a success rate above 98% under multi-user access, and average CPU and RAM usage of 35% and 420 MB on Raspberry Pi 4B during peak load. These results demonstrate that the system is responsive, reliable, and suitable for concurrent experiments. Although validated with a single instrument type, the proposed approach offers a scalable and replicable solution that can significantly enhance electronics education and lower laboratory infrastructure costs.

Corresponding Author:

Abdelkrim Benali
Department of Electrical Engineering, Institute of Technology, Nour Bachir El-Bayadh University Center
El Bayadh, Algeria
Email: benaliabd@gmail.com

1. INTRODUCTION

Remote laboratories have gained increasing attention in engineering education as a sustainable response to long-standing challenges such as limited laboratory availability, high equipment costs, and growing student enrollment. While the COVID-19 pandemic accelerated their initial adoption, recent studies indicate that remote and hybrid laboratories are now widely recognized as a permanent complement to conventional hands-on teaching rather than a temporary emergency measure [1], [2]. In fact, more than 60% of engineering institutions worldwide currently employ some form of remote or virtual laboratory to enhance accessibility and flexibility in practical education [3].

Despite the potential of these approaches to provide authentic educational services, their large-scale implementation remains constrained by organizational, human, and particularly technical challenges. While delivering online lectures is relatively straightforward, enabling learners to remotely manipulate real technological devices within an educational framework is considerably more complex [3]. In technical and engineering sciences, this difficulty is especially pronounced, as practical work is a core component of skill acquisition. Beyond virtual simulations, effective distance education requires learners to interact with physical equipment and perform real measurements remotely, commonly referred to as remote experimentation or tele-laboratories [4], [5]. These challenges are further amplified in universities within developing countries, where financial constraints and rapidly increasing student populations often lead to insufficient laboratory equipment. Remote laboratories offer a practical solution by allowing a single experimental setup to be shared among multiple learners, thereby reducing equipment costs while improving access to hands-on experimentation [6], [7].

Advances in information and communication technologies, along with the adoption of low-cost single-board computers, have facilitated the development of remote laboratories. In particular, the Raspberry Pi has become widely used in engineering and internet of things (IoT) education due to its affordability, versatility, and strong open-source ecosystem. Recent Raspberry Pi–based remote laboratory solutions enable scalable electronics and IoT experimentation with reduced infrastructure requirements [8], [9], highlighting the growing need for secure, scalable, and open architectures that support real-time interaction with physical hardware in resource-constrained educational environments.

Educational embedded-systems laboratories provide scalable, web-accessible environments for hands-on learning using platforms such as Arduino, Raspberry Pi, and TensorFlow Lite, supporting applications in AI robotics, IoT systems, and remote circuit testing [10]. Several studies have explored remote laboratories for IoT and microcontroller programming. Anhelo *et al.* [11] developed a web-based laboratory enabling remote programming of Arduino-compatible NodeMCU boards via Wi-Fi and webcam monitoring. De Zarate *et al.* [12] proposed the WebLabPRO architecture to support multiple remote laboratories targeting Arduino and FPGA boards. El-Hasan *et al.* [13] introduced a remote IoT laboratory integrating Arduino devices, sensors, and cameras, although limited measurement capabilities restrict experimental diversity. Other approaches rely on Raspberry Pi as a lightweight server for remote microcontroller programming and Industry 4.0 education [14]-[16]. Gómez-Alonso *et al.* [17] presented an open-source platform enabling users to upload and execute C code on a remote PIC microcontroller board for IoT-oriented experiments.

More recent studies have expanded remote laboratory infrastructures by integrating cloud computing, IoT technologies, and FPGA platforms to improve scalability and real-time interactivity. Alhammami [18] authors proposed an IoT and cloud-based remote FPGA laboratory for motor control applications, demonstrating strong educational relevance and student engagement. Similarly, Magyari and Chen introduced an IoT enabled multi-user FPGA framework that addresses single-user limitations while improving real-time interaction and resource sharing [19]. Low-cost Raspberry Pi-based platforms, such as RaspyLab, have also been developed to facilitate remote physical computing experiments with minimal hardware requirements [20]. In parallel, broader research efforts toward open and scalable IoT testbeds highlight a growing trend toward federated and democratized remote experimentation infrastructures [21]. Furthermore, emerging AI-augmented IoT frameworks emphasize intelligent monitoring, adaptive control, and low-latency operation as key directions for next-generation remote laboratories [22], [23].

Despite significant progress in remote laboratory technologies, existing solutions remain largely centered on proprietary and resource-intensive platforms operated by well-established institutions, which often impose high financial and infrastructural barriers and limit accessibility for smaller universities and developing regions. To address the resulting limitations in openness, flexibility, and scalability, this work presents LABTEC, a secure, scalable, and adaptable remote laboratory platform for MCU–IoT experimentation. The proposed system introduces a hybrid server architecture in which flask manages measurement instruments -such as oscilloscopes lacking embedded web servers- while Django supports the web platform, providing robust security, ORM-based database management, and long-term scalability through its structured MVC-inspired framework. A Raspberry Pi is employed as a versatile and cost-effective hardware interface, enabling seamless integration with microcontrollers, sensors, and peripheral devices. The platform is fully implemented and validated through a complete laboratory station, demonstrating reliable real-time experimentation with enhanced flexibility, security, and user interaction.

The main contributions of this work are threefold: the design of a hybrid Flask-Django architecture for instrument control and secure web management, the integration of a Raspberry Pi interface to provide a cost-effective bridge for diverse MCU-IoT hardware, and the forward-looking inclusion of FPGA support as a distinctive feature extending remote experimentation beyond conventional microcontrollers. This work advances remote laboratories by enabling real-time signal visualization.

Although limited in measuring current and voltage, this can be addressed with IoT-based sensors, ensuring a reproducible, open, and sustainable framework for academic and industrial use.

To guide the reader, the structure of this paper is organized into four main sections. Section 2 details the materials and methods, beginning with the functional specifications document, followed by the unified modeling language (UML) design, and the implementation phase, which encompasses both hardware and software platform tools. Section 3 presents and discusses the results, including the developed software application and the hardware platform. Finally, section 4 provides the conclusion, highlighting the main contributions of this work and outlining potential directions for future research.

2. METHOD

In this section, we will detail the different phases of our project implementation. We will describe the design and modeling of the system using UML, highlighting the various hardware and software packages to be implemented. Then, we will present the realization of the chosen architecture, justifying the selection of the different tools used.

2.1. Functional specifications document

The main objective of our project is to design and develop a online experimental platform while utilizing the existing measurement equipment in our university center laboratory. The system consists of a server that controls the measuring instruments and configures matrix board for practical work. It also includes a database that stores various information necessary for conducting practical experiments. The server is connected to the university center's network via the internet. The measuring instruments are connected to the server through the local LAN network. Users (Laboratory technician, teacher, or student) must authenticate themselves to access the different functionalities provided by the system. The main tasks of each user are described later in the modeling section. Additionally, the expected system functionalities are:

- There are three types of users: teacher, student, and laboratory technician. Each user has a dedicated interface, and they can log into the system simultaneously.
- Similarities with a real laboratory: the system must include all the operations that students and teachers perform in a real laboratory, from execution to evaluation and grading.
- Schedule: each practical work must be managed by a schedule established by the teacher and displayed to the students concerned. The student will carry out the practical work during the reserved time slot assigned to them.
- Notifications: in addition to notifications appearing on the students' interface, when a new practical work is added by the teacher, an email is automatically sent to each student containing the details of the practical work and its schedule.
- Secure information: the website contains all the information about teachers and students, and this information must be stored securely [6].
- Speed and flexibility: like any project, the application must ensure optimal speed and flexibility.
- User interface/user experience (UI/UX): the interaction between the student and the interface must be simple and fast. Additionally, the design should be attractive and ergonomic to prevent navigation issues between different web pages and ensure a seamless practical work process. These specifications must be taken into account in any project [3]-[7].

2.2. Unified modeling language (UML) design

After consulting and studying the specifications, we developed the final design of the system based on the UML model. UML provides a standardized framework to visualize, specify, and document the abstractions of a software system, and it remains one of the most widely used modeling approaches in system design [24]-[26]. Recent studies also emphasize its evolving role in automated code generation and integration with modern development workflows [25].

To design and document our system, we used three key UML diagrams: use case, class and sequence diagrams.

2.2.1. Use case diagram

This diagram outlines the functional interactions between the system and its three main actors: teacher, student, and laboratory technician. The teacher uploads practical work protocols, grades reports, and publishes results. The student conducts experiment and submits reports. The laboratory technician ensures equipment maintenance and physical setup. All actors must register before accessing their tasks.

2.2.2. Class diagram

The class diagram defines the system's main components, including user registration, authentication, scheduling and interaction with hardware through the "Practical Work Card". Other key classes include practical work, measurements, and reports. Only the hardware card is implemented physically; the rest are software-based.

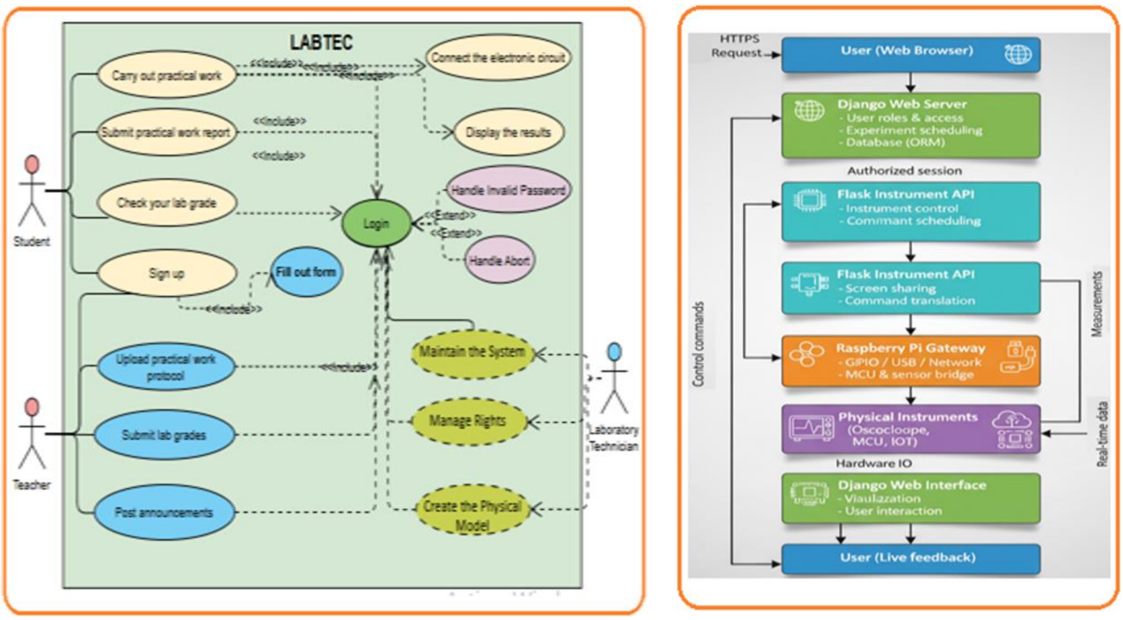
2.2.3. Sequence diagram

This diagram models the chronological flow of user interactions:

- User registration: a user completes a form and selects a role; data is stored in a database.
- Student execution: after login, the student performs the experiment, gathers results, and submits a report.
- Teacher configuration: the teacher sets up the session, uploads instructions, configures equipment, and evaluates submissions.

To describe the final design of our system, we adopted the UML and its various diagrams, which make it possible to visualize, specify, construct, and document the abstractions of a software system. UML serves as a versatile toolbox that provides standardized modeling techniques to represent both functional and structural aspects.

The roles of system actors and components are illustrated in Figure 1 using simplified UML diagrams. Figure 1(a) presents the use-case diagram, identifying the main user roles and their interactions with the system's functionalities. Figure 1(b) shows a high-level flow diagram of the LABTEC platform, depicting interactions among the Django web server, the flask instrument server, the Raspberry Pi gateway, and laboratory equipment during authentication, scheduling, and remote experimentation. In this architecture, flask provides lightweight instrument control, while Django ensures secure user management, structured data handling, and system scalability.



(a) (b)

Figure 1. Simplified unified modeling language and flow diagram
(a) case diagram and (b) high-level flow diagram

Together, these diagrams provide a clear and comprehensive overview of both the functional behavior and architectural design of the proposed system, supporting its implementation and future scalability [25], [26].

2.3. Hardware platform tools

Now, we have a clear understanding of the expected functionalities and the packages to be developed. It is evident that the system consists of both a hardware component (Practical work board, measuring equipment) and a software component (database server and web client development). To achieve this goal, the system was designed based on a target architecture that supports interaction between the lab workstation, measurement instruments, the server, and the web interface. To interact with the practical work board, we will use a Raspberry Pi as the main server, along with another server dedicated to measurement instruments. The back-end software is entirely programmed in Python, while the front-end is developed using Bootstrap, HTML, CSS, SVG, and JavaScript. We have used various technologies and tools, which are divided into two categories: hardware platform tools and software platform tools.

2.3.1. Raspberry Pi

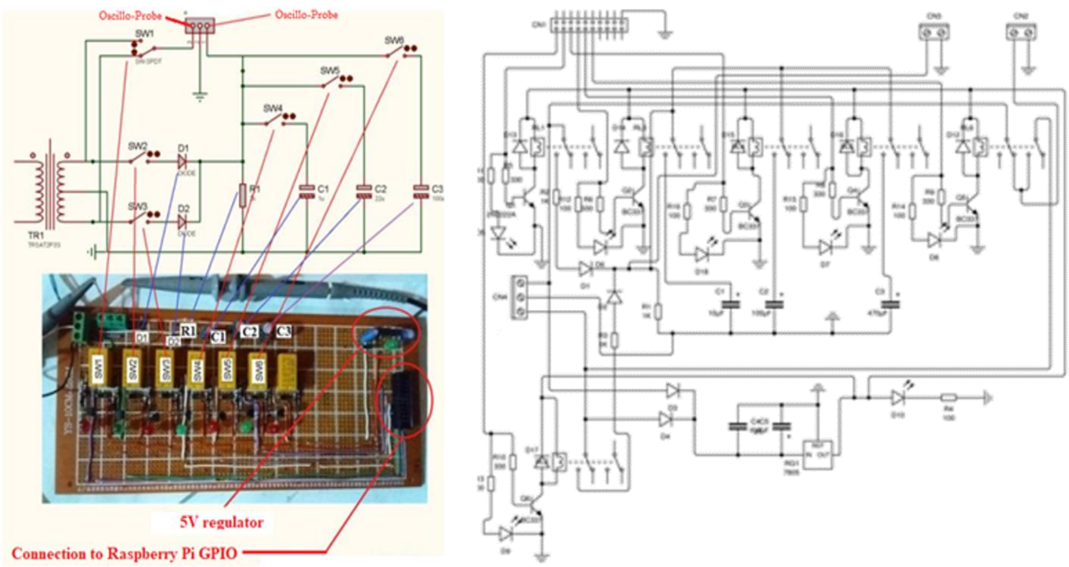
The Raspberry Pi 4 hosts the web server and serves as the central unit of the system. It is a high-performance single-board computer featuring a quad-core Cortex-A72 (ARM v8) 64-bit processor at 1.5 GHz with 8 GB RAM, Gigabit Ethernet, dual-band 2.4/5 GHz Wi-Fi (802.11ac), Bluetooth 5.0, two USB 3.0 and two USB 2.0 ports, dual micro-HDMI supporting 4K output, and a 40-pin GPIO header for hardware interfacing.

2.3.2. Relay switching matrix

The relay switching matrix automates and manages electrical connections by routing signals between measurement instruments and circuit components. Controlled by the main server, it allows remote users to configure and switch connections dynamically, enhancing flexibility and scalability.

The design follows two key principles: Raspberry Pi GPIO outputs act as switches for precise control, and each circuit has its own SVG design for clear visualization and interaction. This ensures efficient, scalable, and remote-friendly operation.

In this project, as shown in Figure 2, a full-wave center-tapped rectifier with a capacitor filter was designed using a center-tap transformer rated at 12 V AC and 36 VA. Figure 2(a) presents the practical work board incorporating the relay-based switching matrix, while Figure2(b) illustrates the corresponding schematic of the relay switching matrix used to configure the rectifier circuit.



(a) (b)

Figure 2. Practical work board (a) relay switching matrix design and (b) relay switching matrix scheme

2.3.3. Measurement instruments

The measuring instrument used in this practical work complies with the LXI standard. It can be controlled via a computer network (RJ45 connector) but does not have an embedded server.

The PeakTech 1240 is a next-generation digital oscilloscope featuring a high-resolution backlit color display, wide bandwidth, and high sampling rate. It includes a VGA output, large internal memory, a USB port, and LAN connectivity for network integration allowing for a wide range of applications as we have used in our project. The measurement instrument server must be equipped with a Windows operating system and oscilloscope software. This software enables real-time sharing of measurement results from the instrument, allowing remote users to access the data easily. This setup ensures seamless integration with the rest of the system, enabling efficient monitoring and analysis of measured signals.

2.4. Software platform tools

The software platform plays a crucial role in the project, and selecting the right tools was a key challenge. After extensive research, we selected Node.js to develop a web-based dashboard that enables real-time control, allowing users to remotely interact with laboratory instruments [16]. Additionally, Python was chosen as the main programming language for web server development, offering simplicity, efficiency, and a rich set of libraries to enhance functionality [12]. The importance of robust software stacks for remote laboratory frameworks has been widely emphasized in earlier works [8].

Django was chosen for web server development due to its strong security features, ORM-based database management, and structured MVC-inspired architecture, ensuring scalability and maintainability. Meanwhile, flask was used for the measurement instrument server, offering a lightweight and flexible solution. For frontend technologies, AJAX enables asynchronous updates without reloading the page, while HTML, SVG, and JavaScript ensure a dynamic and interactive user interface. Bootstrap simplifies styling and responsiveness. The project also relies on SQLite as the database management system, offering a lightweight and embedded solution for efficient data storage and retrieval.

2.4.1. Controlling the Raspberry Pi's GPIO

An efficient way to control the Raspberry Pi's GPIO through the operating system is by using the WiringPi library, developed in C. Widely used for GPIO management, especially with expansion boards, it enables direct GPIO manipulation via the terminal or system commands. WiringPi offers a range of features. It includes the GPIO utility, a command-line tool for managing GPIO pins and additional modules like PiFace and Gertboard. It is also possible to utilize Shell Scripting to control GPIO pins through scripts using the GPIO command, though this approach is less efficient than programming in C or Python. Additionally, system calls enable GPIO manipulation in other programming languages via system() calls in C/C++. The user access feature allows the GPIO command to be executed without sudo, enabling secure GPIO control for normal users.

2.4.2. Integrating Python's subprocess module

The subprocess module in Python is used to launch and manage subprocesses, allowing interaction with their input, output, and error streams while retrieving their return codes offering more flexibility and security. The recommended approach is using subprocess. call() to execute commands, with options to control input, output, and shell execution. This module is particularly useful for integrating system commands within applications, such as in a views file of a web project, enabling efficient process management as seen in Figure 3.

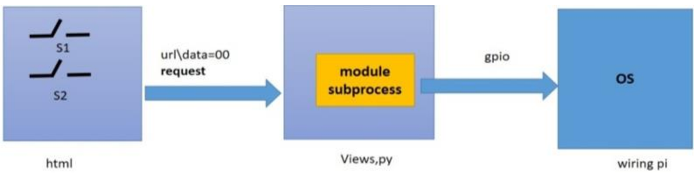
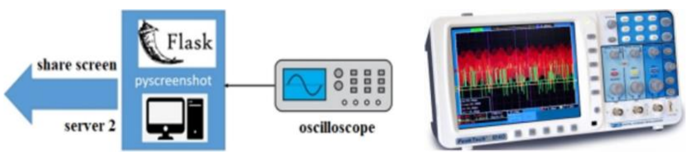


Figure 3. Integrating subprocess module in our system

2.4.3. Creating a web server for the measuring instrument (Oscilloscope PeakTech 1240)

Figure 4 illustrates the solution adopted for remote access to the PeakTech oscilloscope. Since the instrument lacks a built-in web server and operates only through proprietary Windows-based software that displays the screen, a dedicated workaround was required. As shown in Figure 4(a), a lightweight flask server running on a Windows machine connected to the oscilloscope captures the display using the pyscreenshot library and shares it over the network. Figure 4(b) shows the PeakTech oscilloscope, enabling remote users to visualize measurements in real time despite the instrument's closed architecture.



(a) (b)

Figure 4. Flask server (a) diagram for access to the oscilloscope and (b) the PeakTech oscilloscope image

2.5. Network architecture and security measures

Figure 5 summarizes the topology and network architecture of the implemented system. Static IP addresses are assigned, and the router is configured so that the main server operates on port 80, the measurement instrument server on port 5000, and port 3000 is used to display the instrument screen. The system supports both LAN and WAN access; when users connect to the router's public IP address, they are redirected to the main server.

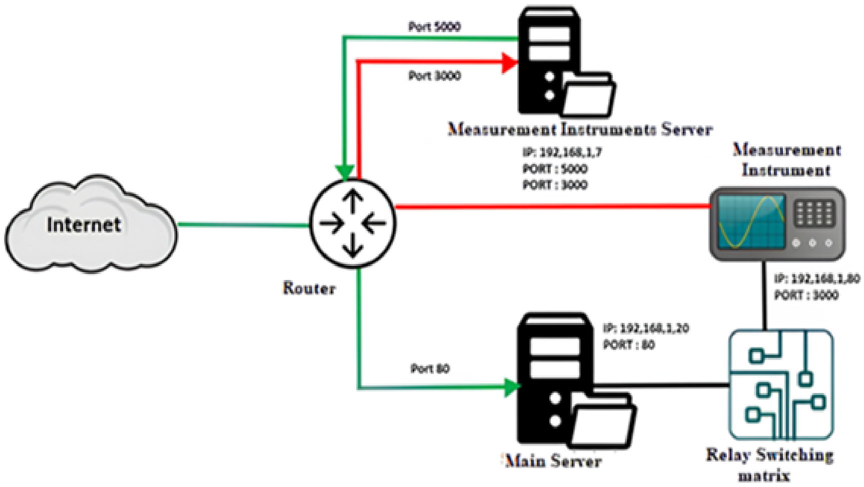


Figure 5. Architecture of the system

Security is a critical component of the online experimental platform, ensuring both technical reliability and educational value [27]. All client–server communication is encrypted with HTTPS/TLS, supported by best practices such as HSTS headers, secure cookies, and regular patching.
The platform applies layered protections:

- User authentication and session management: credentials are verified at login, sessions use secure tokens, and idle sessions expire automatically.
- Role-based access control (RBAC): permissions are dynamically assigned (student, teacher, administrator) according to the principle of least privilege.
- Automatic session timeout: unattended sessions terminate after inactivity.
- Activity logging: all actions (logins, commands, changes) are recorded for auditing and forensic analysis.
- Regular updates: software and firmware are patched against emerging vulnerabilities.

In addition to protecting system integrity, security is embedded into the learning process-students engage directly with these mechanisms, gaining practical cybersecurity skills for online experimental systems.

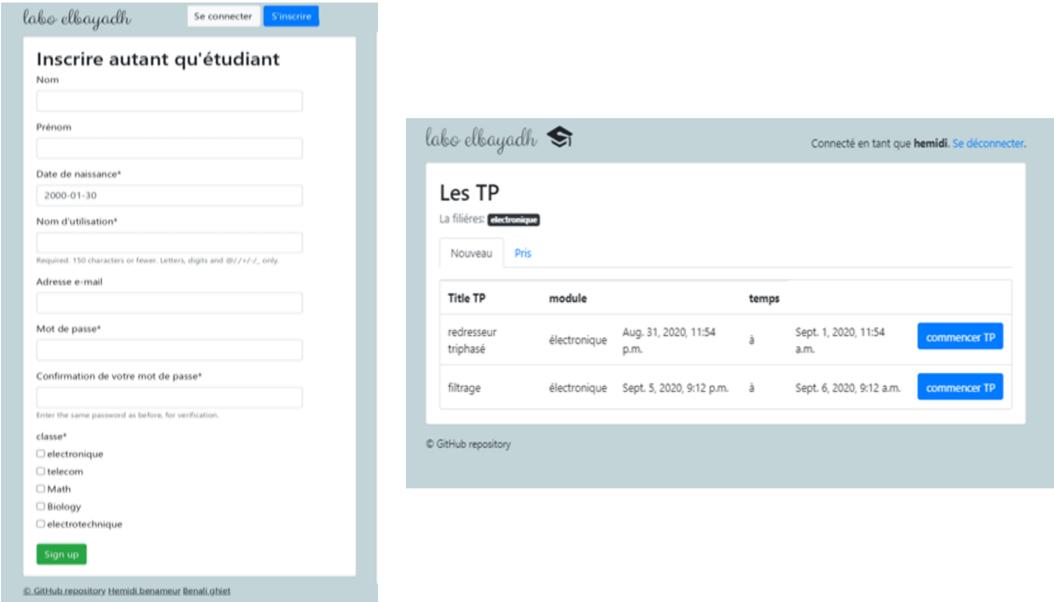
3. RESULTS AND DISCUSSIONS

3.1. Presentation of software application

The login process for the online experimental is straightforward. If you are an existing user, simply log in to access the platform. However, if you are new to LABTEC as we have named it, you will need to create an account, choosing either a student or teacher profile before proceeding.

As displayed in Figure 6 student registration interface allows users to create an account on the platform. New students can register by providing their personal information, including their last name, first name, date of birth, username, email address, and password. They must also confirm their password and select their class from available fields such as electronics, telecom, math or electrical engineering. Alternatively, as illustrate in Figure 6(a). The registered users can simply log in to access their accounts.

As shown in Figure 6(b), the student homepage displays pending practical work, along with completed tasks awaiting teacher evaluation. Students can select a practical work session and remotely interact with electronic devices in the laboratory. Once the task is completed, they submit a work report for teacher evaluation.



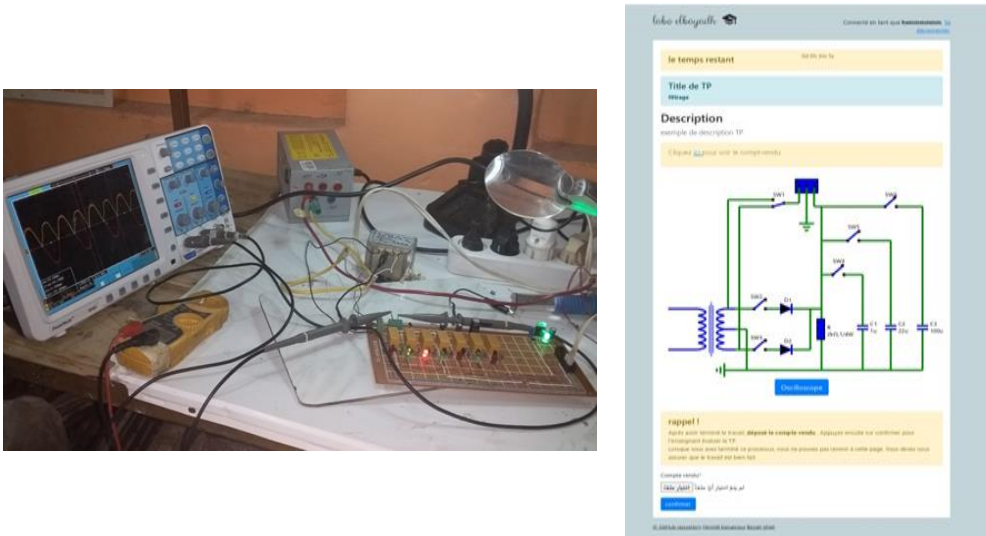
(a) (b)

Figure 6. Student registration interface (a) register student homepage and (b) pending practical work for students

The teacher has the ability to add, edit, and delete laboratory work, as well as test experiments. They can monitor the number of students who have completed each task and review individual student reports for evaluation. Additionally, the teacher can manage various settings, such as defining the minimum duration for practical tasks, scheduling time intervals, activating the calendar, and sending email notifications to students.

3.2. Presentation of hardware platform

As shown in Figure 7, the remote laboratory consists of a Raspberry Pi 4, a relay switching matrix, and an oscilloscope, all connected to the internet via LAN. As shown in Figure 7(a), the LAN-connected hardware setup enables remote access to the laboratory, while Figure 7(b) illustrates the web-based interface used to control the experiment through interactive switches. The practical work explored in this project focuses on a full-wave center-tapped rectifier circuit (D1, D2) with capacitor filters (C1 = 1µF, 22µF, and 100µF), utilizing a center-tap transformer (12V AC, 36VA).



(a) (b)

Figure 7. Remote laboratory experiment test (a) hardware equipment and (b) homepage experiment test

Switch SW1 toggles between the primary voltages of the center-tap transformer, displaying the signal on the oscilloscope. Switches SW2 and SW3 control the rectifier diodes D1 and D2, while switches SW4, SW5, and SW6 manage the filtering capacitors, either individually or simultaneously, depending on the practical task requirements. To visualize the results on the oscilloscope, the student must press the blue "Oscilloscope" button located below the circuit (see Figure 7(b)). Once the blue "Oscilloscope" button is pressed, Server 2 sends the oscilloscope image corresponding to the selected switches, as shown in Figure 8. Experimental results show that a single Raspberry Pi can support 5 to 10 concurrent users under LAN conditions, maintaining end-to-end latency below 50 ms with stable resource utilization. Under WAN access, concurrency decreases to 3–5 users, mainly due to network latency and bandwidth variability rather than server-side processing limits. Experimental results show that a single Raspberry Pi can support 5 to 10 concurrent users under LAN conditions, maintaining end-to-end latency below 50 ms with stable resource utilization. Under WAN access, concurrency decreases to 3–5 users, mainly due to network latency and bandwidth variability rather than server-side processing limits.

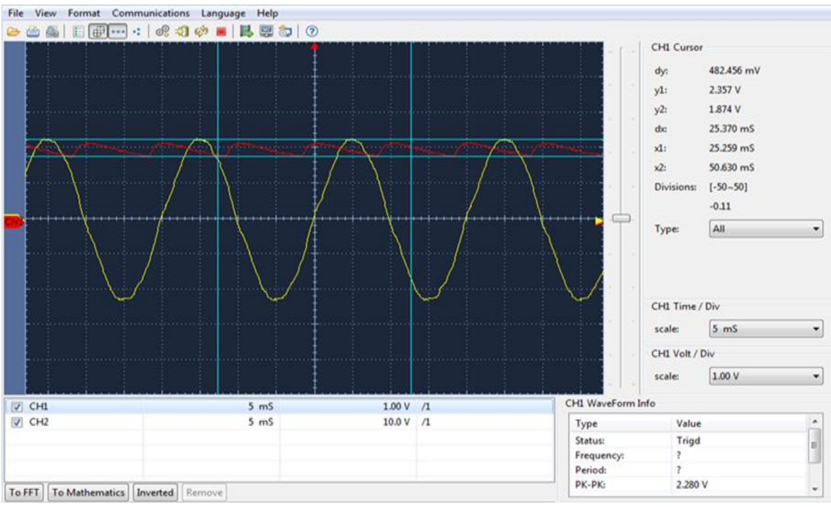


Figure 8. The oscilloscope trace when SW1, SW2, SW3, and SW4 are pressed

These results confirm that LABTEC is well suited for small to medium-scale deployments, while larger installations would benefit from distributed gateways or cloud-assisted scaling.

The obtained performance is consistent with established client-server and distributed system design principles, where decoupling control and presentation layers reduces processing overhead and improves responsiveness, as also reported in prior remote laboratory studies [17], [20]. Unlike platforms relying solely on embedded servers [17], [20], [22], LABTEC adopts a hybrid Flask–Django architecture, in which flask ensures lightweight instrument control and Django provides secure, structured, and scalable web services. This design is particularly effective for integrating low-cost oscilloscopes lacking built-in web servers. Compared with earlier works where scalability was limited or insufficiently evaluated [19], [23], the use of a Django-based framework with ORM-supported data management enhances long-term maintainability and extensibility. Moreover, unlike conceptual approaches without experimental validation [21], LABTEC is fully implemented and evaluated on real hardware, reinforcing its practical relevance. Nevertheless, the current system remains constrained by the computational capacity of a single Raspberry Pi, limiting user concurrency, multi-instrument support, and fault tolerance. The platform is also not yet integrated with learning management systems, and large-scale deployment raises additional cybersecurity considerations. LABTEC strikes an optimal balance between affordability and performance. It is significantly cheaper than commercial NI/PLC systems and only marginally more expensive than VISIR-based labs, while offering far greater capabilities. By using open-source hardware (Raspberry Pi) and software (Python, Node.js), it avoids vendor lock-in. Its hybrid architecture enables real-time multi-user interaction, directly overcoming VISIR's closed-loop limitation. With high scalability, LABTEC delivers a cost-effective, flexible solution that democratizes access to advanced remote laboratory education.

Future work will therefore focus on distributed and cloud-assisted architectures, support for heterogeneous instruments, learning management system (LMS) integration via standard APIs, and strengthened security mechanisms. Further extensions will include FPGA-based experiments and AI-driven monitoring to enhance adaptability, scalability, and long-term sustainability, while contributing to the digital transformation of engineering education in alignment with the United Nations Sustainable Development Goals (SDG 4 and SDG 9) and Industry 4.0 principles through a low-cost, scalable, and open smart laboratory framework.

4. CONCLUSION

This paper has demonstrated that LABTEC provides a technically validated, low-cost remote laboratory architecture capable of supporting real-time electronics experimentation with stable performance and sub-50 ms latency, which is critical for responsive closed-loop control and interactive measurement tasks. The experimental evaluation confirms that the proposed hybrid Flask–Django architecture, combined with a Raspberry Pi interface, enables reliable multi-user access, efficient resource utilization, and structured system scalability, thereby constituting a concrete contribution beyond conceptual remote lab designs. Despite these results, the current implementation is limited by the computational capacity of the Raspberry Pi and by the absence of direct current and voltage measurement modules, which restricts the number of concurrent users and the range of supported experiments. Future developments will therefore focus on integrating IoT-based electrical sensors, strengthening security mechanisms for multi-user operation, and extending hardware support to Wi-Fi-enabled microcontrollers and FPGA platforms. In particular, FPGA integration will enable high-performance and reconfigurable experiments aligned with Industry 4.0 requirements, while AI-based monitoring and multi-institution deployments will be pursued to further validate scalability, interoperability, and long-term sustainability.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Abdelkrim Benali	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	
Somia Benali		✓	✓			✓			✓	✓			✓	
Benameur Hemidi	✓		✓				✓				✓			✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] J. Ma and J. V. Nickerson, "Hands-on, simulated, and remote laboratories," *ACM Computing Surveys*, vol. 38, no. 3, p. 7, Sep. 2006, doi: 10.1145/1132960.1132961.

[2] A. Bochicchio, A. Longo, and S. Panunzio, "Remote laboratories in engineering education: trends, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 118245–118260, 2022.

[3] J. Á. Ariza, "Can in-home laboratories foster learning, self-efficacy, and motivation during the COVID-19 pandemic? -- A case study in two engineering programs." Mar. 30, 2022. [Online]. Available: http://arxiv.org/abs/2203.16465.

[4] C. Gode and M. S. Madankar, "A remote experimentation using virtual laboratory," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, IEEE, Jul. 2013, pp. 1–6. doi: 10.1109/ICCCNT.2013.6726552.

[5] H. S. Jo and R. S. Jo, "Design and development of remote laboratory system to facilitate online learning in hardware programming subjects," in *2020 13th International UNIMAS Engineering Conference (EnCon)*, IEEE, Oct. 2020, pp. 1–5. doi: 10.1109/EnCon51501.2020.9299326.

[6] N. Thakur, S. Pradhan, and C. Y. Han, “Investigating the impact of COVID-19 on online learning-based web behavior.” Apr. 27, 2022. doi: 10.54941/ahfe100850.
[7] N. G. L. Flores, A. S. Islind, and M. Óskarsdóttir, “Effects of the COVID-19 pandemic on learning and teaching: a case study from higher education.” May 04, 2021. [Online]. Available: http://arxiv.org/abs/2105.01432.
[8] J. Álvarez Ariza and C. Nomesqui Galvis, “RaspyControl Lab: a fully open-source and real-time remote laboratory for education in automatic control systems using Raspberry Pi and Python,” *HardwareX*, vol. 13, p. e00396, Mar. 2023, doi: 10.1016/j.ohx.2023.e00396.
[9] M. Alkhatib, Y. Qu, R. Cai, R. A. G. Unglaub, and R. Ritter, “Real time virtual laboratory solution prototype and evaluation for online engineering degree programs,” *IEEE Access*, vol. 11, pp. 130508–130516, 2023, doi: 10.1109/ACCESS.2023.3334262.
[10] H. A. Guerrero-Osuna *et al.*, “Developing a cloud and iot-integrated remote laboratory to enhance education 4.0: an approach for FPGA-based motor control,” *Applied Sciences*, vol. 14, no. 22, p. 10115, Nov. 2024, doi: 10.3390/app142210115.
[11] J. Anhelo, A. Robles, and S. Martin, “Internet of things remote laboratory for MQTT remote experimentation,” in *Proceedings of the 15th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2023)*, 2023, pp. 166–174. doi: 10.1007/978-3-031-48590-9_16.
[12] L. O. de Zarate, I. Angulo, A. Villar-Martínez, L. Rodriguez-Gil, and J. García-Zubía, “Remote laboratory for the development of customized low-power computing and IoT systems,” in *International Conference on Remote Engineering and Virtual Instrumentation*, 2023, pp. 249–260. doi: 10.1007/978-3-031-42467-0_23.
[13] T. S. El-Hasan, “Internet of Thing (IoT) based remote labs in engineering,” in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019, pp. 976–982. doi: 10.1109/CoDIT.2019.8820591.
[14] A. Fernández-Pacheco, S. Martin, and M. Castro, “Implementation of an arduino remote laboratory with Raspberry pi,” in *IEEE Global Engineering Education Conference, EDUCON*, 2019, pp. 1415–1418. doi: 10.1109/EDUCON.2019.8725030.
[15] B. Bordel, R. Alcarria, and T. Robles, “Industry 4.0 paradigm on teaching and learning engineering,” *International Journal of Engineering Education*, vol. 35, no. 4, pp. 1018-1036, 2019.
[16] A. D. Gueye and L. Yade, “Work-in-progress: realization of IoT lab benches controlled by a Raspberry Pi accessible via the web,” in *Interactive Mobile Communication, Technologies and Learning*, 2024, pp. 355–363. doi: 10.1007/978-3-031-56075-0_34.
[17] Á. Gómez-Alonso, S. Martin, and M. Castro, “Industry 4.0 remote lab based on pic microcontrollers,” *Dyna*, vol. 98, no. 4, pp. 334–334, Jul. 2023, doi: 10.6036/10925.
[18] M. Alhammami, “FPGA hardware kit for remote training platforms,” *Discover Education*, vol. 3, no. 1, p. 102, Jul. 2024, doi: 10.1007/s44217-024-00203-w.
[19] A. Magyari and Y. Chen, “FPGA remote laboratory using iot approaches,” *Electronics (Switzerland)*, vol. 10, no. 18, p. 2229, Sep. 2021, doi: 10.3390/electronics10182229.
[20] J. A. Ariza and S. G. Gil, “RaspyLab: a low-cost remote laboratory to learn programming and physical computing through python and raspberry Pi,” *Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 17, no. 2, pp. 140–149, May 2022, doi: 10.1109/RITA.2022.3166877.
[21] P. Kuriscak, P. Rossa, H. Fernandes, and J. N. Silva, “Design and implementation of a framework for remote experiments in education,” in *2022 8th International Engineering, Sciences and Technology Conference (IESTEC)*, IEEE, Oct. 2022, pp. 258–265. doi: 10.1109/IESTEC54539.2022.00046.
[22] S.-C. Sun and Y.-C. Tsai, “A modular AIoT framework for low-latency real-time robotic teleoperation in smart cities.” Oct. 13, 2025. [Online]. Available: http://arxiv.org/abs/2510.11421.
[23] E. González, J. Rivas, and H. Melgar, “Enabling remote learning and intercontinental collaboration at the tecnológico de monterrey,” NI Case Studies, National Instruments. Accessed: Apr. 03, 2026. [Online]. Available: https://www.ni.com/en/solutions/academic-research/case-studies/enabling-remote-learning-itesm.html.
[24] J. Beauden, “An analysis of the significance of UML diagrams focus on medium-sized projects,” Ladoke Akintola University of Technology. Accessed: Jan. 29, 2026. [Online]. Available: https://www.researchgate.net/publication/388683775_An_Analysis_of_the_Significance_of_UML_Diagrams_Focus_on_Medium-Sized_Projects.
[25] A. Bates, R. Vavricka, S. Carleton, R. Shao, and C. Pan, “Unified modeling language code generation from diagram images using multimodal large language models,” *Machine Learning with Applications*, vol. 20, p. 100660, Jun. 2025, doi: 10.1016/j.mlwa.2025.100660.
[26] D. Rouabhia and I. Hadjadj, “Behavioral augmentation of uml class diagrams: an empirical study of large language models for method generation.” Jun. 01, 2025. [Online]. Available: http://arxiv.org/abs/2506.00788.
[27] A. E. Pena-Molina and M. M. Larrondo-Petrie, “Safety and security considerations for online laboratory management systems,” *Journal of Cybersecurity and Privacy*, vol. 5, no. 2, p. 24, May 2025, doi: 10.3390/jcp5020024.

BIOGRAPHIES OF AUTHORS

Abdelkrim Benali born in 1966 in Algeria, is a seasoned academic and electrical engineer with a strong foundation in both industry and higher education. He earned his Engineer degree in Electrotechnics from Saida University in 1998, followed by a master’s degree in 2011, and a Ph.D. in Electrical Engineering from the National Polytechnic School of Oran, specializing in advanced electrical systems and embedded electronics for renewable energy. Dr. A. Benali began his career with extensive practical experience in electrical control, installation, and maintenance before transitioning to academia in 2012. He currently serves as a professor in the Department of Electrical Engineering at Nour Bachir University Center of El-Bayadh, Algeria. His teaching portfolio includes a wide range of undergraduate and graduate courses such as electrical devices, control systems, PLCs, and microcontrollers. His research interests focus on remote laboratory, electronics for embedded systems integrated in renewable energy sources and AI. He can be contacted at email: benaliabd@gmail.com.

Somia Benali earned her Ph.D. in Management Sciences from the University of Relizane. She specializes in Information and Communication Technologies (ICT), innovation, and e-commerce. Her research focuses on technological innovation, marketing innovation, and operational innovation in the context of digital transformation. Dr. S. Benali is particularly interested in the development of e-commerce in Algeria, analyzing its challenges, growth opportunities, and its impact on the national economy. Her work also explores the applications of artificial intelligence in e-commerce, including recommendation systems, customer service automation, predictive analytics, and intelligent customer data management. In addition to her academic expertise, Dr. Benali has strong skills in web development, enabling her to design, analyze, and implement digital platforms tailored to the evolving needs of online commerce. This combination of theoretical insight and technical proficiency allows her to contribute effectively to the digitalization and innovation efforts of businesses in Algeria. She can be contacted at email: benalisomia1710@gmail.com.

Benameur Hemidi is born on March 16, 1994 in Algeria. He obtained his master degree in electronics of embedded systems in 2020 from Nour El Bachir University Center, El Bayadh, Algeria, his research interests are AI, smart buildings, remote control systems and advanced programming systems. He can be contacted at email: benameurhemidi@gmail.com.