**1.INTRODUCTION**

In the rapidly evolving landscape of education, technology plays a pivotal role in enhancing learning experiences and providing more personalized feedback to students. This project, the "E-Learning Platform & Personalized Student Performance Tracker," aims to address the need for a streamlined, accessible, and data-driven approach to online learning. The platform is designed to serve as a comprehensive tool for students to engage with course materials, test their knowledge through interactive quizzes, and receive immediate, detailed insights into their academic performance.

The core objective of this web-based application is twofold: first, to provide a user-friendly interface for accessing structured educational content organized by academic years, semesters, and specific subjects; and second, to implement a robust system for tracking and visualizing student progress on a granular level. By offering features such as secure user authentication, a clear navigational structure, dynamic quiz generation, and personalized performance summaries, the platform seeks to empower students to take control of their learning journey, identify areas for improvement, and monitor their growth over time.

**1.1 PURPOSE**

* The goal is to bridge the gap between theoretical knowledge and practical application and provide a flexible, scalable, and advanced learning experience.
* This project specifically aims to develop a proof-of-concept module for an e-learning platform. This module will focus on a critical feature, such as an interactive code editor with real-time feedback or a personalizedlearning path recommendation system, to demonstrate the platform's potential for enhancing practical skills and engagement. The successful implementation of this module will lay the groundwork for future expansion into a comprehensive e-learning solution.

**1.2 SCOPE**

* The e-learning platform is designed for long-term sustainability, with the flexibility to evolve and integrate new technologies, ensuring continuous relevance and effectiveness in education.
* **User Interface (UI) Design:** Development of intuitive and user-friendly interfaces for the chosen module, ensuring ease of navigation and interaction.
* **Backend Logic and Database Integration:** Implementation of the necessary backend logic to support the module's functionality, including data storage and retrieval. This will involve designing and integrating a database to manage relevant information.
* **Basic Analytics and Reporting:** Implementation of core analytics to track student performance, engagement metrics, and course effectiveness, providing valuable insights for both students and educators.
* **Extensible Data Models:** The data schema designed for the chosen module will be forward-compatible, allowing for the addition of new data fields and relationships without requiring significant refactoring as the platform grows and new features are introduced.

**2.LITERATURE SURVEY**

**2.1 Yogesh Sharma-International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 04 | Apr 2021**

Research highlights the importance of integrated platforms that manage content, security, and evaluation effectively. This study proposes a cloud-based e-learning management system with enhanced features like content protection, delivery management, and cross-browser support to improve user experience.

**2.2 Fetty Poerwita Sary-Indonesian Journal of Learning and Instructione-ISSN 2614-5677Volume 4, Issue 1, April 2021**

The research shows it faces challenges in promoting student engagement and independent learning. Studies emphasize the need for effective content delivery and instructor training to improve outcomes. This study explores these aspects, highlighting that enhancing teaching methods and student motivation is crucial for maximizing e-learning benefits.

**2.3 Dipti D. Patil-International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019**.

Adaptive learning systems utilize personalization techniques to tailor content based on individual performance, improving engagement and knowledge retention. Research emphasizes the importance of continuous assessment and dynamic content adjustments to meet learners' unique needs.

**2.4 Luis Alfaro:IEEE World Conference on Engineering Education(EDUNINE)-2021**

Research highlights the importance of e-learning in supporting remote education.Studies emphasize the need to web-based tools and virtual reality, to enhance teaching effectiveness. This study builds on previous research by reviewing various e-learning technologies, their applications, and emerging trends to improve learning opportunities.

**2.5 Ava Chikurteva: International Scientific Conference Electronics - ET2020, September 16 - 18, 2020, Sozopol, Bulgaria**

Research highlighting both its potential and challenges. Studies have analyzed popular e-learning platforms, identifying their strengths and limitations. This study builds on such findings by exploring key platforms, assessing their effectiveness, and outlining future improvements for successful e-learning implementation.

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

The current e-learning system is a hybrid e-learning model that combines traditional classroom learning with online learning methods. Traditional face-to-face learning (In-person lectures and discussions.), Synchronous e-learning (Real-time online classes using Zoom, Microsoft Teams, etc.), Asynchronous e-learning.Generic content that doesn’t cater specifically to students.

**3.1.1 Disadvantage of Existing System**

* Lack of Personalized Tracking
* Scalability issues
* Limited interactivity
* Performance tracking is not integrated with individual courses.

**3.2 PROBLEM STATEMENT**

* The current e-learning platforms lack personalization and real**-**world application focus.
* Existing systems often provide genericcontent that does not cater to individual learning needs or bridge the gap between theoreticalknowledge and practicalimplementation.
* Additionally, there is no integratedperformance tracking system to monitor student progress and provide personalized feedback to help students improve.

**3.3 PROPOSED SYSTEM**

The proposed e-learning platform offers personalized learning paths, adapting content based on student performance to enhance understanding. It includes real-worldapplications through quizzes and assignments reinforcing theoretical concepts. An advanced performance tracker monitors progress across courses, providing detailed feedback for continuous improvement. The platform is also flexible and scalable, allowing easy integration of new courses, quizzes, and features.

**3.2.1 Advantage of Proposed System**

* Personalized Tracking integrated with individual courses.
* No Scalability issues.
* High interactivity, Performance tracking is integrated with individual courses.

**4.SYSTEM REQUIREMENTS AND SPECIFICATIONS**

**4.1 FUNCTIONAL REQUIREMENTS**

* **User Authentication :** Secure login and registration with hashed passwords to ensure only verified students and admins can access protected content.
* **Course Management :** Organize and display courses, subjects, and materials by year and semester to provide students with structured and academic content.
* **Progress Tracking :** Organize and display courses, subjects, and materials by year and semester to help students and instructors monitor progress in real-time.
* **Personalized Dashboard :** Organize and display courses, subjects, and materials by year and semester tailored to each student's activity and performance trends.
* **Assessment Tools   :** Provide unit-specific quizzes with instant result calculation along with retry options to enhancelearning.
* **Database Integration :** Use SQLite to store user data, courses, and performance records ensuring efficient retrieval and updates with minimal resource usage.

**4.2 NON-FUNCTIONAL REQUIREMENTS**

* **Scalability :** Handle increasing users and courses without performance loss.
* **Performance :** Handle increasing users and courses without performance loss.
* **Usability :** Provide an intuitive and user-friendly interface.
* **Compatibility :** Support multiple devices and browsers.
* **Maintainability :** Use modular code for easy updates and debugging.

**4.3 SOFTWARE REQUIREMENTS**

* **Operating System :** Windows 7 and above
* **Programming Language :** Python 3.13.1
* **Frontend Framework :** React js 19.0.0.
* **Back-end Framework :** Flask Framework
* **Database :** DB sqlite browser
* **Front-end Technologies :** HTML, CSS, JavaScript

**4.4 HARDWARE REQUIREMENTS**

* **Processor :** i5 and above, 2.5 GHz
* **RAM :** 8GB and above
* **Hard Disk :** 100 GB min

**4.5 TECHNOLOGIES USED**

**Python**

Python served as the core programming language for the backend development of this e-learning platform, primarily through the Flask microframework. Flask's simplicity and flexibility allowed for rapid development of the web application's structure, including defining URL routes, handling HTTP requests, and managing user sessions. Python's clear syntax was utilized for all server-side logic: processing user authenticationinteracting with the SQLite database and implementing the business logic for quiz submission, grading, and performance aggregation. Furthermore, Python's integration with the Jinja2 templating engine enabled dynamic generation of HTML content, allowing personalized data to be seamlessly presented to users on the frontend. The entire backend, from request handling to database interaction and dynamic content serving, was orchestrated using Python.

The backend of this e-learning platform was architected using Python version 3.9 a choice that provided significant advantages throughout the development lifecycle. This modern Python release offered enhanced features such as f-strings for streamlined string formatting and optimized dictionary operations, which contributed to cleaner and more efficient code for handling data structures like quiz results and user profiles. The stability and extensive testing of this Python version, along with its compatibility with the contemporary releases of the Flask framework and the sqlite3 standard library, ensured a reliable foundation. This allowed development efforts to be concentrated on building the platform's core e-learning functionalities—from user authentication using hashlib to complex data retrieval and storage for course content and personalized student performance tracking—rather than addressing potential version incompatibilities or relying on outdated language constructs.

**Frontend Technologies and User Interface**

The user interface of the e-learning platform was constructed using standard web technologies: HTML, CSS, and implicitly, JavaScript for any client-side interactivity managed by the browser. HTML formed the structural backbone of each page, defining elements such as forms for login and quiz submission, lists for course navigation, and tables for displaying performance data. Jinja2, a templating engine integrated with Flask, played a crucial role on the server side to dynamically generate this HTML. Jinja2 allowed for the insertion of Python-processed data directly into the templates, enabling personalized content like user greetings, specific course materials based on selections, and dynamically populated quiz questions and results. This approach ensured that the HTML structure was both flexible and responsive to user interactions and data changes.

**Styling and User Experience**

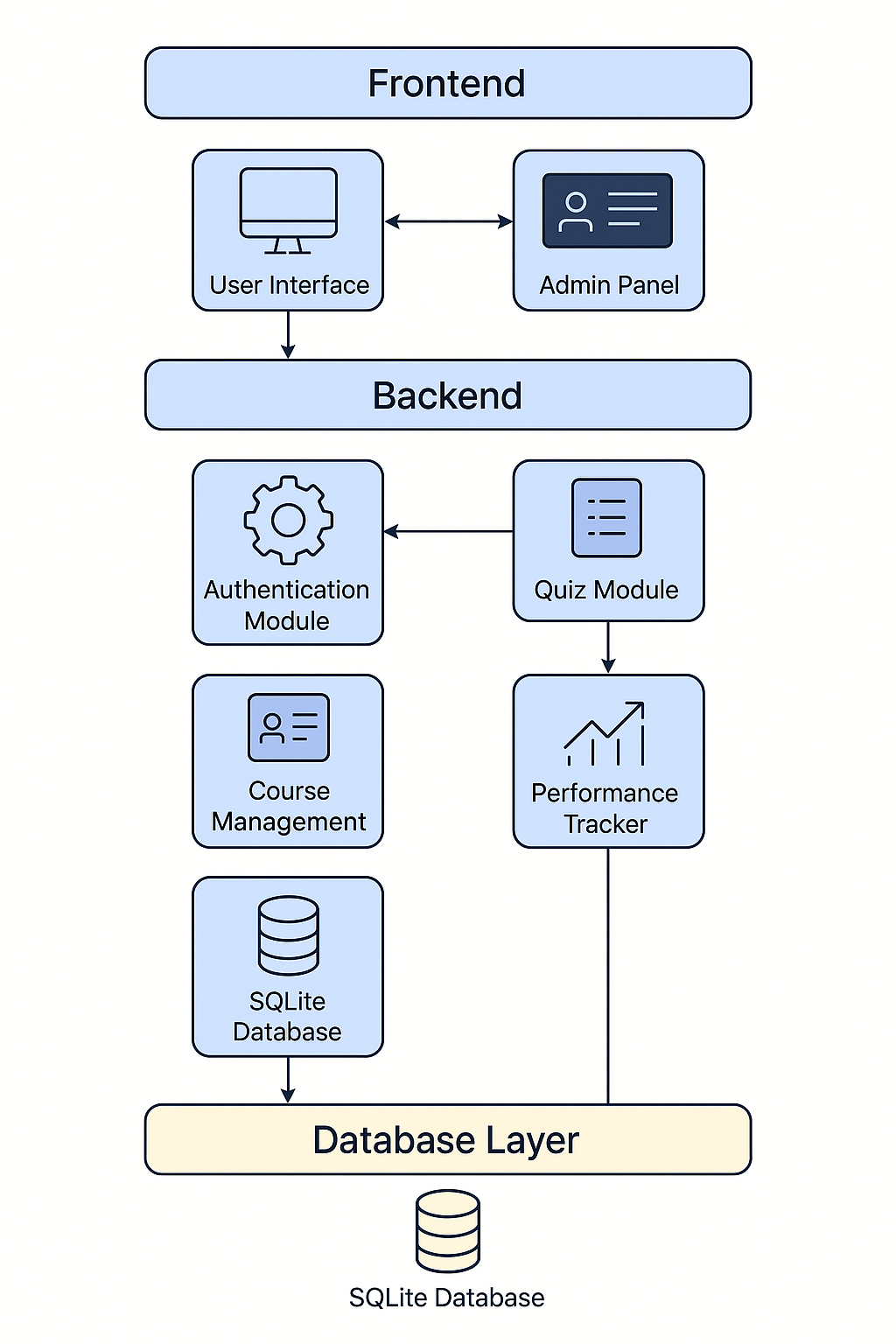
Cascading Style Sheets (CSS) were employed to define the visual presentation and user experience of the platform. A central stylesheet (style.css) was developed to ensure a consistent and professional aesthetic across all pages, encompassing layout, typography, color schemes, and the styling of interactive elements like buttons, links, and navigation menus. Specific CSS classes were applied to various HTML elements, including containers, authentication forms, dashboard links, quiz components, and performance tables, to achieve the desired look and feel. The styling also focused on usability, ensuring clear visual hierarchy, readable text, and intuitive feedback for user actions, such as hover effects on links and distinct visual cues for flash messages (success, error, warning), contributing to an overall user-friendly interface.

* **Frontend:** HTML, CSS, JavaScript.
* **Backend:** Python with Flask.
* **Database:** SQLite.
* **Authentication:** Password hashing using SHA-256.
* **Templating:** Jinja2.
* **Deployment:** Flask’s built-in development.
* **Tools:** SQLite DB Browser.

**5. SYSTEM DESIGN**

**5.1 ARCHITECTURE**

The e-learning platform follows a three-tier architecture consisting of the Frontend, Backend, and Database. The input begins with the student interacting with the Frontend, where they log in, view available courses, take quizzes, and view their performance. The Frontend sends these requests to the Backend (Flask), which processes the input by validating credentials, fetching course data, calculating quiz scores, and tracking performance. The Backend interacts with the Database (SQLite) to store and retrieve data, such as user information, course details, quiz questions, and performance records. Finally, the Backend sends the processed data back to the Frontend, which displays the output to the student in the form of a dashboard, quiz results, or performance reports. This seamless flow ensures a smooth and efficient user experience.



**Fig 5.1 System Architecture**

**5.2 SYSTEM MODULES**

System modules are individual, self-contained components or units of a larger system that perform specific functions or tasks. In a well-structured system architecture, the system is divided into multiple modules to simplify development, improve maintainability, and enhance scalability.

* **User Authentication Module:** This module securely manages user logins and registrations. It handles password encryption and forgotten password recovery. It ensures only authorized users can access the system.
* **Course Management Module:** This module allows administrators to create, update, and organize courses. It supports adding course materials and setting access permissions. Users can browse and enroll in available courses here**.**
* **Quiz Management Module:** This module facilitates the creation and deployment of quizzes. It includes features for question banking, scoring, and time limits. Students can take quizzes and receive immediate feedback.
* **Performance Tracking Module:** This module monitors and records user progress and scores. It generates detailed reports on quiz attempts and course completion. Users can view their performance analytics to identify areas for improvement.
* **Database Module:** This module manages all data storage and retrieval operations. It ensures data integrity, security, and efficient access. It's the backbone for all other modules, storing user, course, and quiz information.
* **User interface:** This module provides the visual elements and interactive components users engage with. It ensures an intuitive and responsive experience across various devices. Its design focuses on ease of navigation and clear presentation of information.

**5.3 DATA FLOW DIAGRAM**

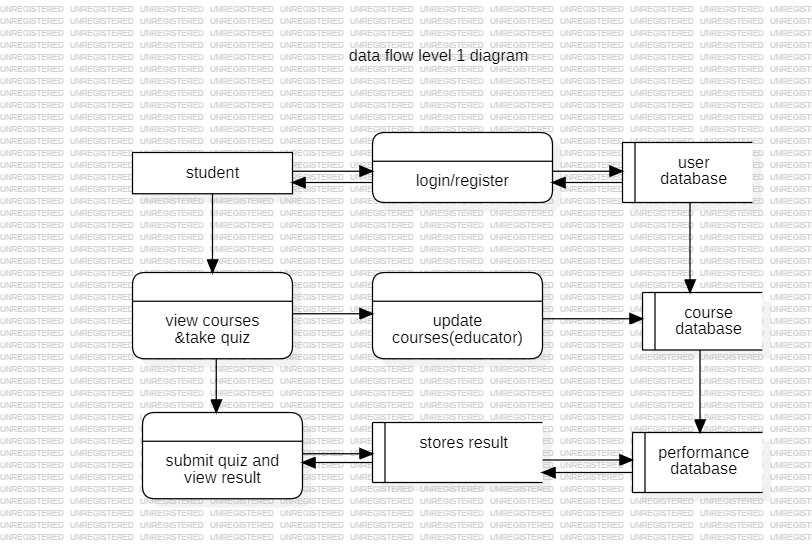
The student interacts with the e-learning platform, which serves as an interface to access educational content and services. This interaction is bidirectional, meaning the student can both send inputs and receive outputs from the platform. The E-learning platform, in turn, communicates with the DB browser database, which stores and retrieves the necessary data such as user information, course materials, and progress tracking. The database exchanges data with the platform to ensure the student has seamless access to relevant educational resources.

**Level 0 DFD**

****

**Fig 5.2 Level 0 DFD**

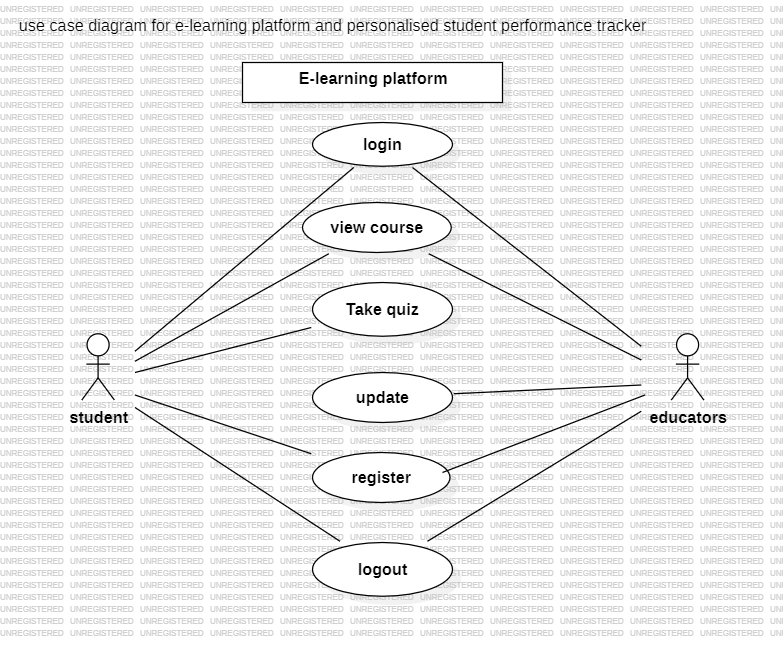
**Level 1 DFD**



**Fig 5.3 Level 1 DFD**

**5.4 UML DIAGRAMS**  
**5.4.1 Usecase Diagram**

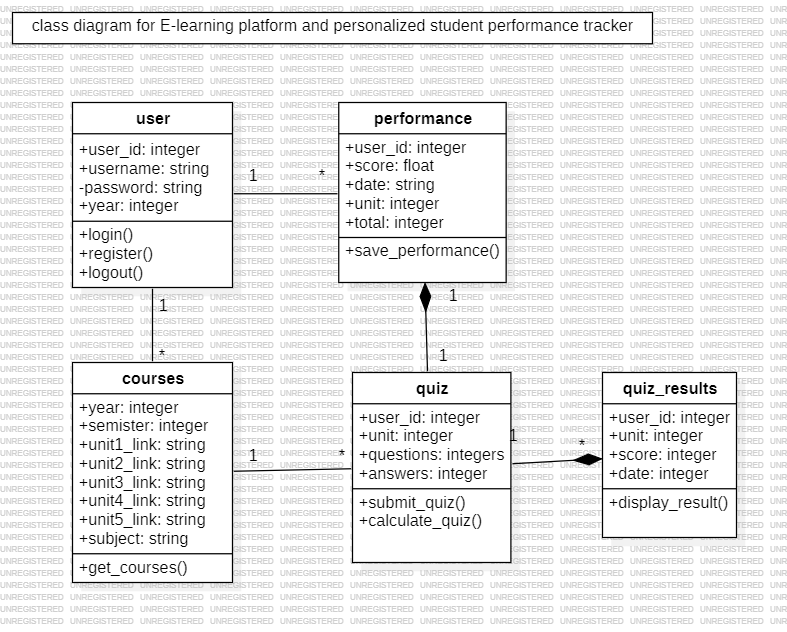
Use case diagram that represents the core functionalities of an E-learning platform from a user's perspective. It shows how a user can interact with the system through various use cases such as login, view course, take quiz, update profile, register, and logout. Each oval represents a specific action the user can perform, and the arrows indicate the flow of interaction between the user and the system. This diagram helps in understanding the system's behavior and the essential operations that need to be implemented in the E-learning application.



**Fig 5.4 Usecase diagram**

**5.4.2 Class Diagram**

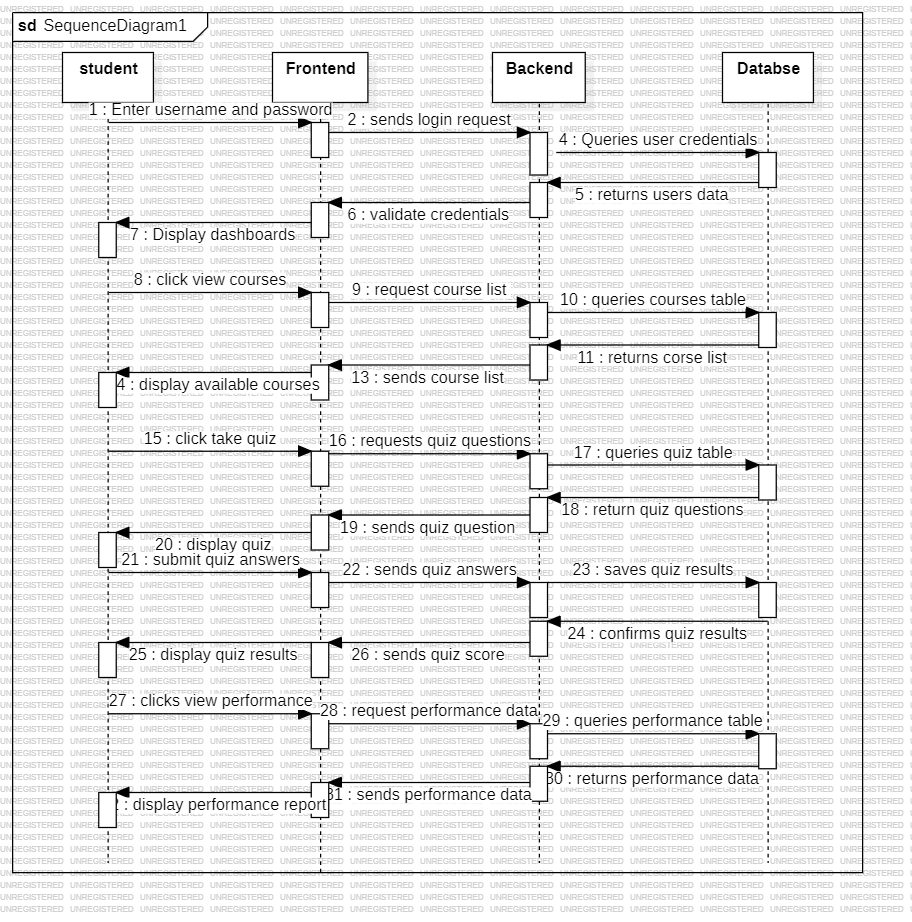
Class diagram for an E-learning platform integrated with a personalized student performance tracker. It outlines six main classes: user, courses, quiz, performance, quiz\_results, and quiz. Each class contains relevant attributes and functions to represent the structure and behavior of the system. For example, the user class handles login, registration, and logout, while the courses class provides access to learning materials. The quiz class manages the quiz-taking process and score calculation, and the performance class stores and tracks user scores. The relationships between these classes demonstrate how data flows and how users interact with quizzes, view results, and track performance. This diagram provides a clear blueprint for implementing the system's backend logic.



**Fig 5.5 Class diagram**

**5.4.3 Sequence Diagram**

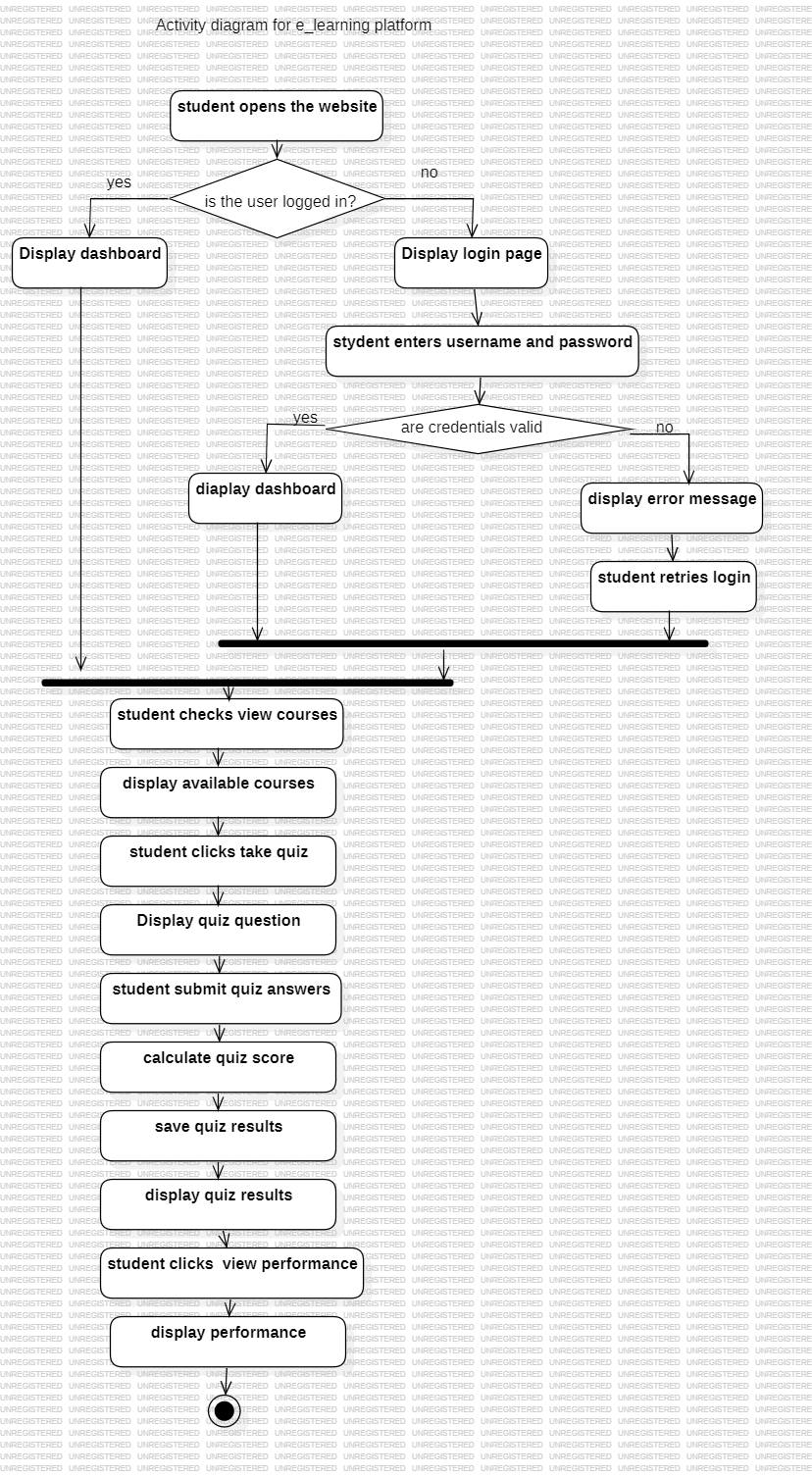
Sequence diagram that illustrates the interaction between different components of the E-learning system—namely the student, frontend, backend, and database. It shows the step-by-step flow of messages exchanged when a student logs in, views courses, takes a quiz, and checks performance. The process begins with the student entering login details, which are validated through the frontend and backend by querying the database. After successful login, the student can view available courses, attempt quizzes, and submit answers. The backend processes and saves the quiz results, which can later be retrieved and displayed as part of the performance report. This diagram effectively captures the dynamic behavior of the system and the flow of information during user activities.



**Fig 5.6 Sequence diagram**

**5.4.4 Activity Diagram**

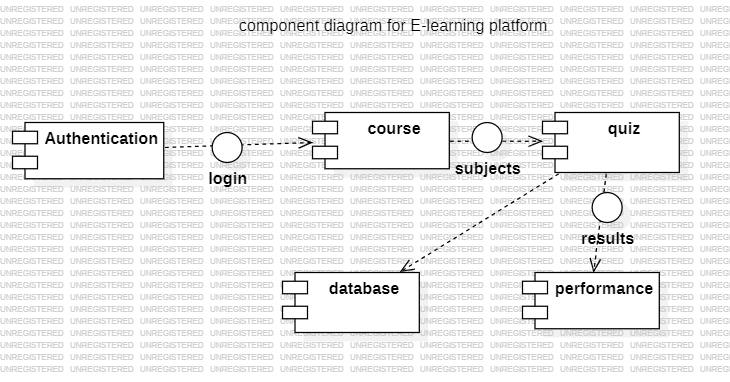
Activity diagram for the E-learning platform, depicting the flow of student interactions throughout the system. It begins with the student opening the website and proceeds based on login status. If not logged in, the student is prompted to enter credentials, which are validated before accessing the dashboard. From there, the student can view available courses, take quizzes, and submit answers. The system then calculates and displays the quiz score. Finally, students can view their overall performance. This diagram provides a clear visualization of the system’s workflow and user activity sequence.



**Fig 5.7 Activity diagram**

**5.4.5 Component Diagram**

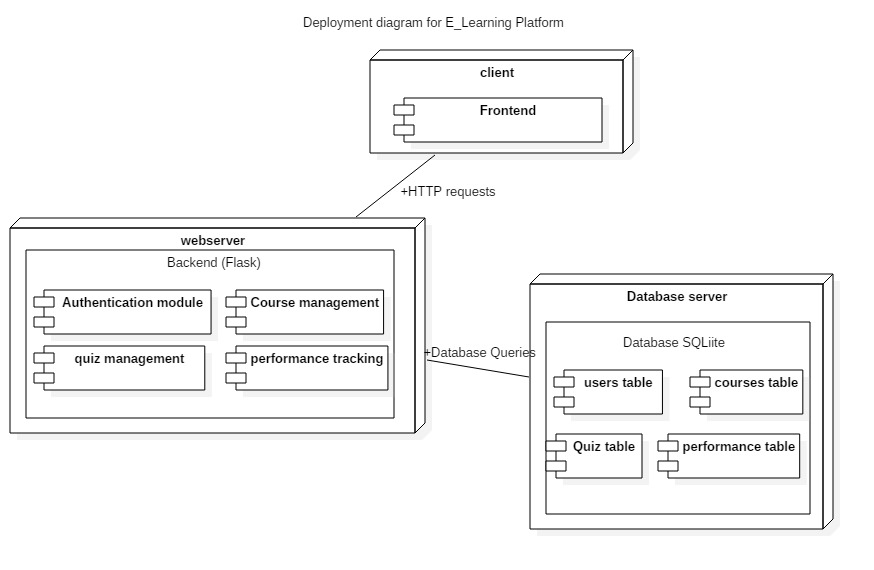
Component diagram for an E-learning platform, This diagram illustrates the high-level organization of the system, showcasing the various components and their interfaces. Key components include 'Authentication' for user verification, 'Course' for managing educational content, 'Quiz' for assessments, and 'Performance' for tracking student progress. All these components interact with a central 'Database' component, which is essential for data storage and retrieval, ensuring the system's overall functionality and data flow.



**Fig 5.8 Component diagram**

**5.4.6 Deployment Diagram**

Deployment diagram for the E-learning platform, This diagram illustrates the physical deployment of the system's components across various hardware nodes. It shows how the user interacts with the system via web browsers on their client devices, which then connect to the web server hosting the application. The web server, in turn, communicates with the application server to handle business logic, and both servers interact with the database server, where all persistent data for the e-learning platform is stored.



**Fig 5.9 Deployment diagram**

**6. IMPLEMENTATION**

**6.1 ALGORITHMS USED**

**Requirement Analysis & Planning**

* Identified platform goals: subject-wise quizzes, performance tracking, secure login.
* Planned user roles and navigation flow (Dashboard → Year → Semester → Subject → Quiz).
* **Database Design**
* Created schema with tables for Users, Courses, Quizzes, and Performance.
* Designed relationships to support course mapping and result tracking.
* **Backend Development**
* Set up Flask framework and defined route logic for each page/function.
* Implemented user authentication and session management.
* Wrote SQL queries to handle data fetch, insertion, and result processing.
* **Frontend Development**
* Built responsive UI using HTML, CSS.
* Created pages for login, dashboard, course list, quiz, and performance.
* Used base templates for layout consistency.
* **Quiz & Performance Logic**
* Developed engine to fetch, display, and evaluate quizzes.
* Stored scores and summarized latest attempts in the performance section.
* **Testing & Debugging**
* Manually tested all flows: registration, quiz attempt, score saving, logout.
* Resolved issues like database locks using SQLite.
* **UI Enhancements & Styling**
* Polished design with custom CSS for better user experience.
* Ensured consistent theme, form validation, and mobile responsiveness.

**6.2 SAMPLE CODE**

**Login**

def login\_required(f):

@functools.wraps(f)

def decorated\_function(\*args, \*\*kwargs):

if 'user\_id' not in session:

flash('Please log in to access this page.', 'warning')

session['next\_url'] = request.url

return redirect(url\_for('login'))

return f(\*args, \*\*kwargs)

return decorated\_function

**Logout**

@app.route('/logout')

def logout():

session.clear()

flash('You have been successfully logged out.', 'success')

return redirect(url\_for('login'))

**Dashboard**

@app.route('/dashboard')

def dashboard():

return render\_template('dashboard.html', username=session.get('username'))

@app.route('/year<int:year\_num>')

def year\_view(year\_num):

if not (1 <= year\_num <= 4):

flash("Invalid year specified.", "error")

return redirect(url\_for('dashboard'))

return render\_template(f'year{year\_num}.html', year=year\_num)

def semester\_view(year\_num, semester\_num):

if not (1 <= year\_num <= 4 and 1 <= semester\_num <= 2):

flash("Invalid year or semester specified.", "error")

return redirect(url\_for('year\_view', year\_num=year\_num) if 1 <= year\_num <= 4 else url\_for('dashboard’))

subjects = get\_subjects\_for\_semester(year\_num, semester\_num) # Use helper

return render\_template('semester\_list.html', subjects=subjects, year=year\_num, semester=semester\_num)

**Landing.html**  {% extends "base.html" %}

{% block title %}Welcome | {{ super() }}{% endblock %}

{# Add class to body tag to hide standard header on this page ONLY #}

{% block body\_attributes %}class="body-landing"{% endblock %}

{% block content %}

{# Landing Page Hero Section #}

<div class="landing-hero">

<div class="container"> {# Container for title/buttons #}

<h1 class="landing-title">E-learning Platform & Personalized Student Performance Tracker</h1>

<p class="landing-tagline">Your Gateway to Focused Learning and Personalized Progress.</p>

<div class="landing-actions">

<a href="{{ url\_for('login') }}" class="button button-large button-login">Login</a>

<a href="{{ url\_for('register') }}" class="button button-large button-register">Register</a>

</div>

</div> {# End Container #}

</div> {# End landing-hero #}

{# \* Features Section MOVED BACK OUTSIDE hero \* #}

<section class="features-section">

<div class="container"> {# Features uses its own container #}

<h2>Key Features</h2>

<div class="features-grid">

<div class="feature-item">

<h3>Courses</h3>

<p>Explore well-organized video lectures and reading materials.</p>

</div>

<div class="feature-item">

<h3>Personalized Tracking</h3>

<p>Monitor your progress with insightful performance dashboards.</p>

</div>

<div class="feature-item">

<h3>Accessible Anywhere</h3>

<p>Learn at your own pace, on any device, anytime.</p>

</div> </div>

</div>

</section>

{# \* END Features Section \* #}

{% endblock %}

[10:03 PM, 6/12/2025] Akhila Csed: year1.html{% extends "base.html" %}

{% block title %}Year {{ year }} Semesters | {{ super() }}{% endblock %}

{% block content %}

<div class="container"> {# Use the standard container #}

{# Flash messages handled by base.html #}

<h1>Year {{ year }} Semesters</h1> {# Standard H1 styling will apply #}

{# Apply the item-list class to the ul for styling #}

<ul class="item-list"><li>

{# Links use the generic semester\_view endpoint #}

<a href="{{ url\_for('semester\_view', year\_num=year, semester\_num=1)

}}">Semester 1</a>

</li><li>

<a href="{{ url\_for('semester\_view', year\_num=year, semester\_num=2) }}">Semester 2</a>

</li></ul>

{# Use standard back link style, potentially centered #}

<div class="list-back-link-container"> {# Optional wrapper for centering #}

<a href="{{ url\_for('dashboard') }}" class="back-link">« Back to Dashboard</a>

</div></div> {# End container #}

{% endblock %}

**7. SYSTEM TESTING**

**7.1 TEST CASES**

Test cases are crucial for validating the system's functionality and ensuring it meets the specified requirements. Each test case outlines a specific scenario, defining inputs, expected outputs, and the conditions under which a feature should operate correctly.

**Test case 1**

The provided table details Test Case 1 for User Login functionality. The objective of this test case was to validate the system's login mechanism using both valid and invalid credentials, ensuring that users can only access the system with correct authentication.

**Table 7.1: User Login**

|  |  |
| --- | --- |
| **Test Case 1** | **User Login** |
| Test Objective | Validate login functionality with valid and invalid credentials. |
| Test Description | Ensure users can log in only with correct credentials. |
| Requirements Verified | User table with email/password stored in hashed format. |
| Test Environment | Flask App (localhost) with SQLite. |

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action** | **Expected** | **Actual** | **Pass/Fail** |
| Valid Credentials | User is directed to dashboard. | User is directed to dashboard. | Pass |
| Invalid Credentials | Error message "Invalid username or password" is shown. | Error message "Invalid username or password" is shown. | Pass |

**Test Case 2**

Dashboard Navigation, focusing on the flow from year to semester to subject. The objective was to verify the user's ability to seamlessly navigate through academic years and semesters to access associated subjects. The results show that upon clicking "Dashboard" after login, the user was correctly directed to the year selection page. Subsequently, selecting a semester successfully displayed the corresponding list of subjects. This indicates a Pass for both actions within the web browser with Flask backend environment, confirming the proper implementation of the course and subject mapping.

**Table 7.2: Dashboard Navigation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case 2** | | **Dashboard Navigation** | | |
| Test Objective | | Verify dashboard navigation flow from year → semester → subject. | | |
| Test Description | | User selects academic year and semester to access subjects. | | |
| Requirements Verified | | Course and subject mapping in database. | | |
| Test Environment | | Web browser with Flask backend. | | |
| **Action** | **Expected** | | **Actual** | **Pass/Fail** |
| Click on “Dashboard” after login | User is navigated to year selection page. | | User is navigated to year selection page. | Pass |
| Select a semester | Corresponding subjects list is shown. | | Corresponding subjects list is shown. | Pass |

**Test Case 3**

Quiz Submission & Scoring, aiming to verify the functionality of quiz submission and automatic scoring within the E-learning platform. The test involved a user attempting and submitting a quiz, where the system was expected to calculate and display the score automatically. As per the results, the score was indeed calculated and displayed upon quiz submission.

Furthermore, attempts to skip questions correctly blocked submission and prompted the user to answer all questions, ensuring data completeness. Both actions passed successfully in the Flask web interface with SQLite DB environment, confirming the robust implementation of the quiz and scoring logic.

**Table 7.3: Quiz Submission and Scoring**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case 3** | | **Quiz Submission and Scoring** | | |
| Test Objective | | Verify quiz submission and auto-scoring. | | |
| Test Description | | User attempts and submits quiz, system calculates score. | | |
| Requirements Verified | | Quiz questions stored; scoring logic implemented | | |
| Test Environment | | Flask web interface with SQLite DB. | | |
| **Action** | **Expected** | | **Actual** | **Pass/Fail** |
| Quiz submitted | Score is automatically calculated and displayed. | | Score is automatically calculated and displayed. | Pass |
| Skipped questions | Submission is blocked; user is prompted to answer all questions first. | | Submission is blocked; user is prompted to answer all questions first. | Pass |

**7.2 RESULTS AND DISCUSSION**

**7.2.1 Landing Page**

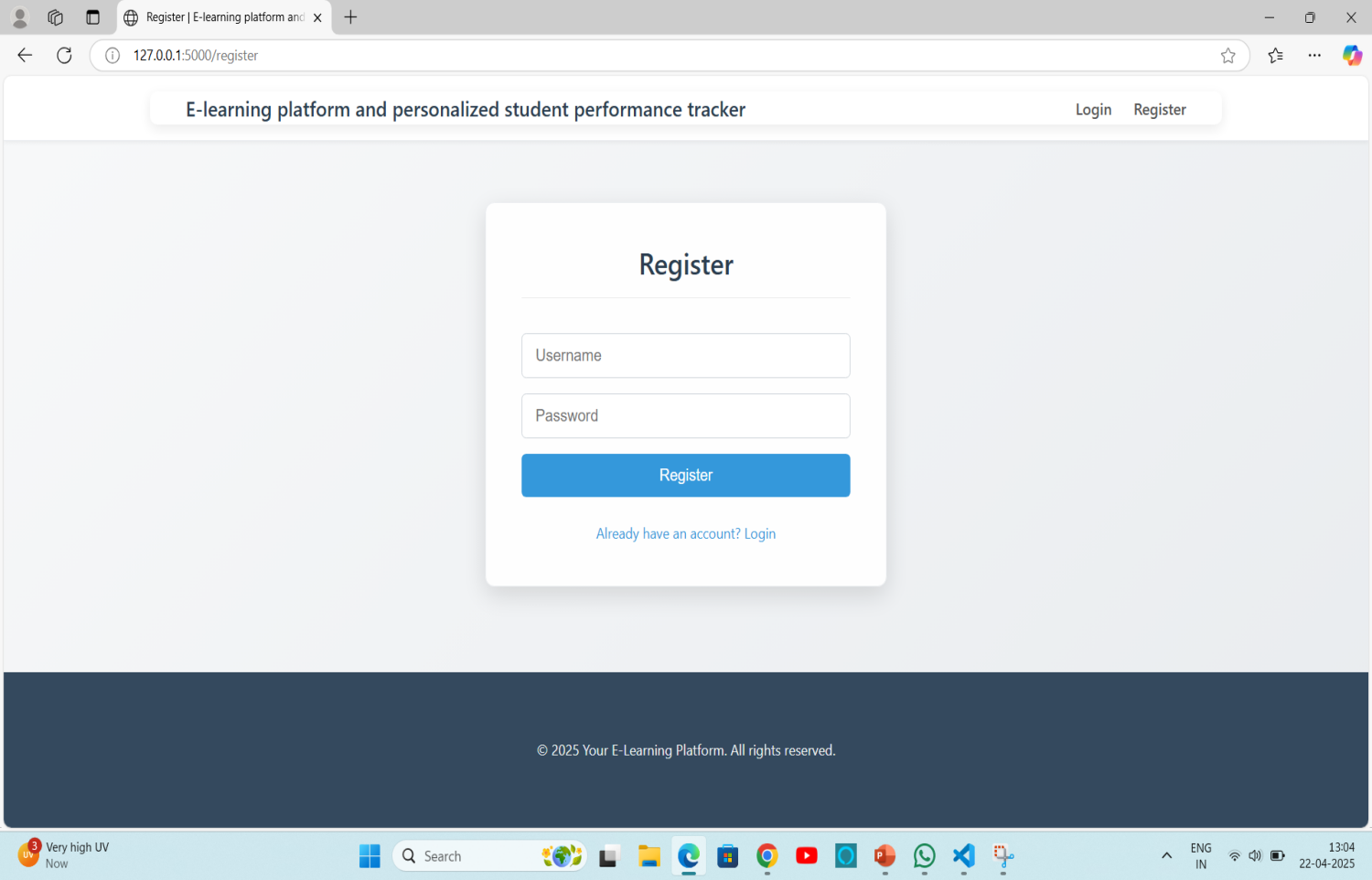
Your personalized gateway to learning, quizzes, and performance tracking — all in one platform. The Landing Page, depicted in Fig 7.1, serves as the primary entry point for users into the E-learning platform. Designed as a personalized gateway, it provides an immediate overview of the system's core functionalities, such as access to learning materials, quizzes, and individual performance tracking. This page is crucial for creating a positive first impression, offering a welcoming and intuitive interface that guides users to explore the platform's features and begin their educational journey efficiently.



**Fig 7.1 Landing Page**

**7.2.2 Register and Login Pages**

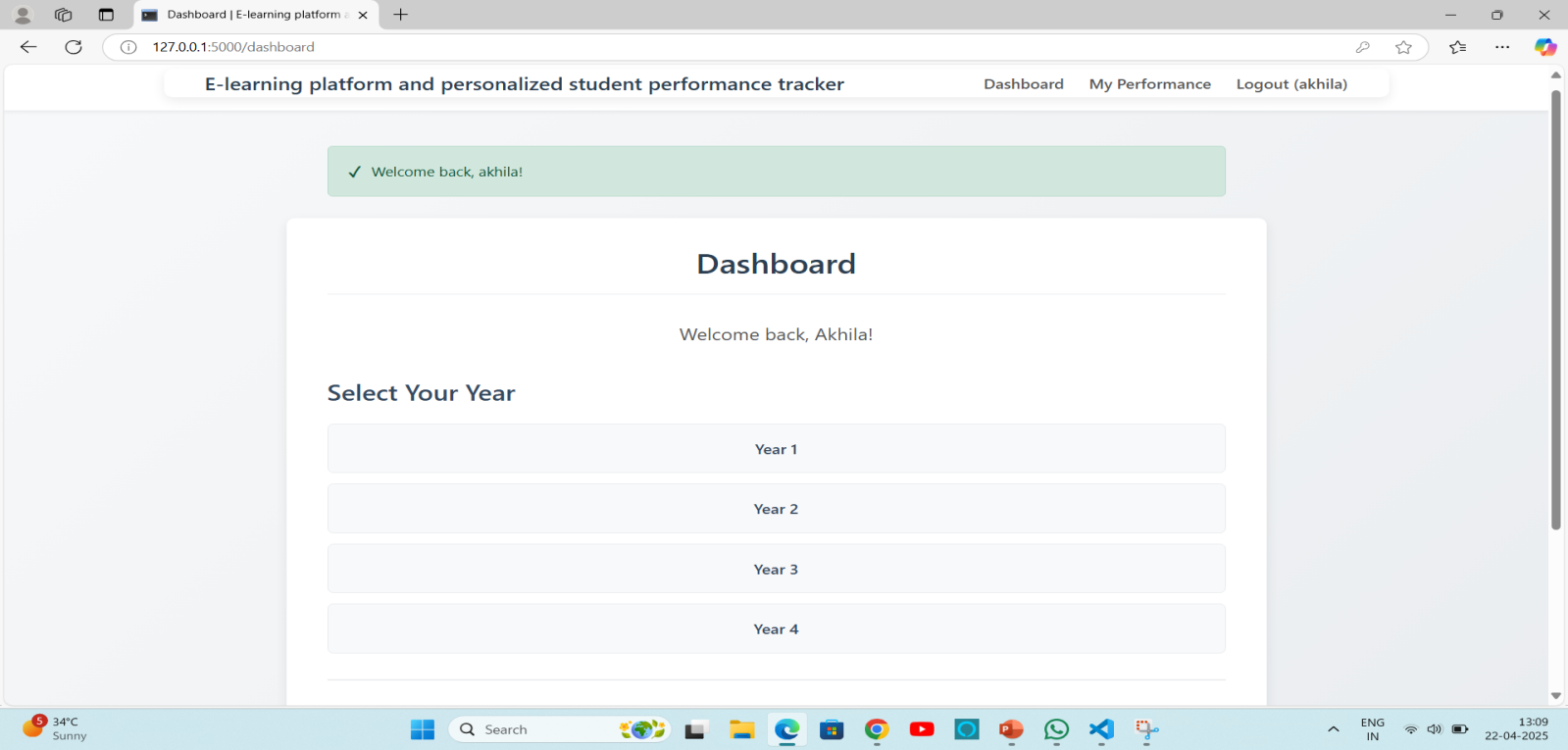
Access your learning journey — securely log in or create a new account to get started. The Register and Login Pages are fundamental to the E-learning platform, serving as the secure entry points for all users. These pages facilitate essential user authentication, allowing existing users to securely log in to access their personalized learning journey. Simultaneously, new users can easily create an account, ensuring a smooth and guided onboarding process.



**Fig 7.2 Register Page**

**7.2.3 Dashboard**

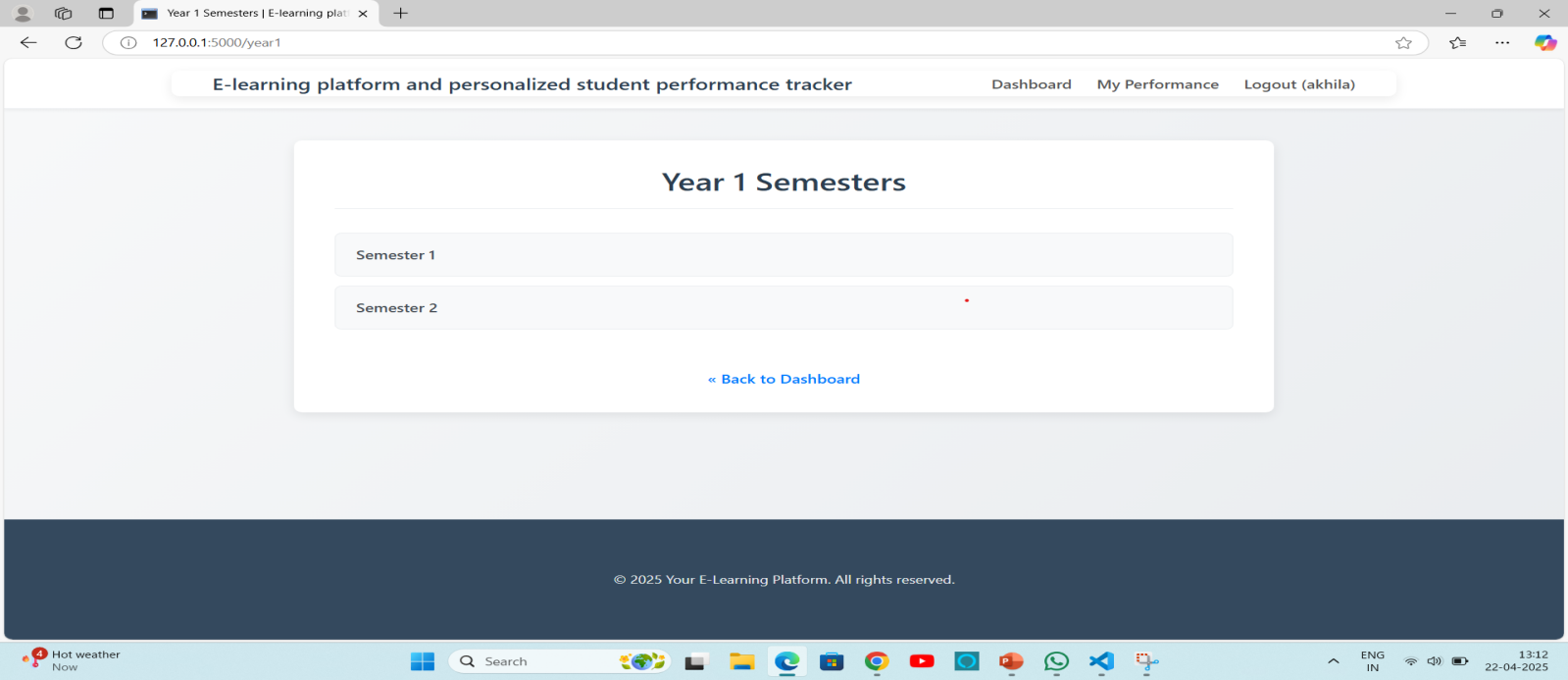
Your personalized learning space — navigate subjects, view quizzes, and monitor your performance. The Dashboard, illustrated in Fig 7.4, serves as the central, personalized learning hub for each user within the E-learning platform. It provides a comprehensive overview of the user's academic journey, allowing for easy navigation through various subjects, direct access to assigned quizzes, and efficient monitoring of individual performance. This intuitive interface is designed to consolidate all essential learning activities and progress tracking in one accessible location, enhancing the overall user experience and fostering a focused learning environment.



**Fig 7.4 Dashboard**

**7.2.4 Academic Year and Semesters**

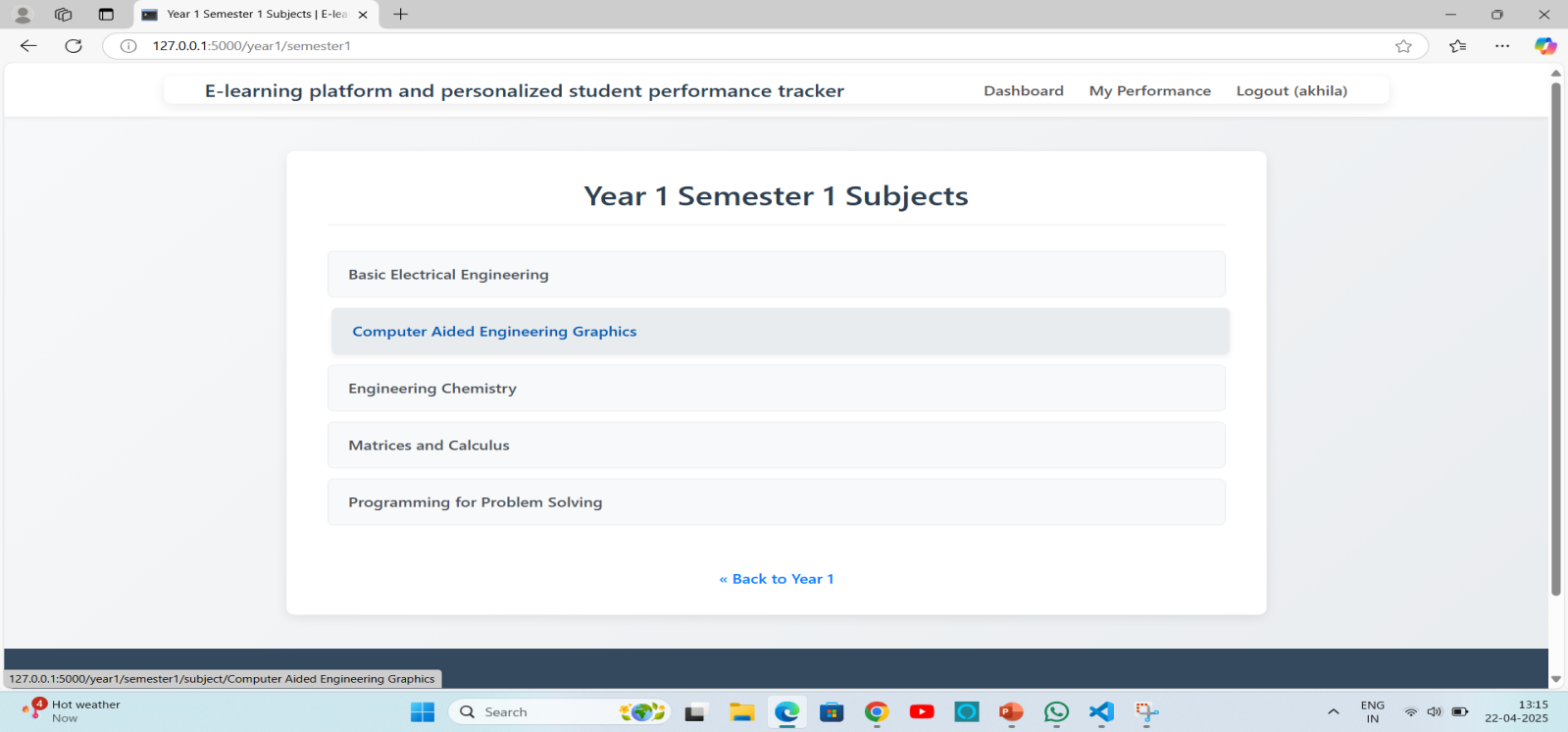
Select your academic year and semester to access relevant courses and quizzes. The Year and Semesters Page, shown in Fig 7.5, is a key navigational component that allows users to filter and access academic content relevant to their specific study period. By enabling users to select their academic year and corresponding semester, this page effectively organizes the vast amount of learning materials and quizzes. This targeted access ensures that students can efficiently find and engage with the courses and assessments pertinent to their current curriculum, streamlining their learning experience



**Fig 7.5 Year and Semesters Page**

**7.2.5 Subjects Page**

Choose a subject to begin learning, take quizzes, and track your understanding. The Subjects Page serves as the gateway for users to delve into specific academic disciplines within the E-learning platform. This page allows students to select a particular subject, which then unlocks access to its associated learning materials, integrated quizzes, and personalized tools for tracking their understanding and progress within that chosen area. By centralizing subject-specific resources, this page streamlines the learning process and encourages focused study.

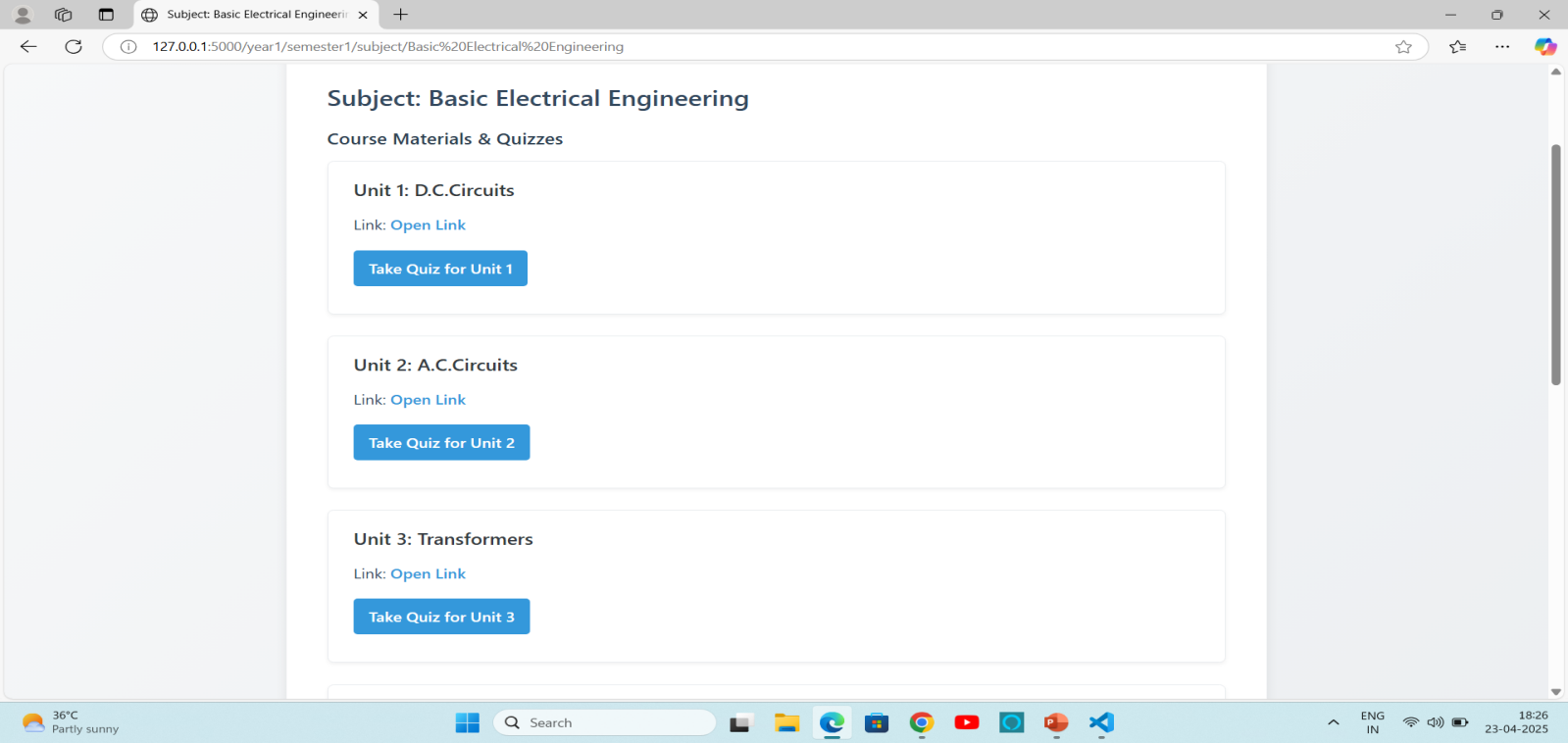


**Fig 7.6 Subjects Page**

Furthermore, its intuitive design ensures that learners can easily navigate through different topics, fostering an environment conducive to self-paced learning and reinforcing comprehension through readily available assessments. This dedicated subject view is critical for organizing content effectively and providing a clear path for students to deepen their knowledge.

**7.2.6 Course Videos**

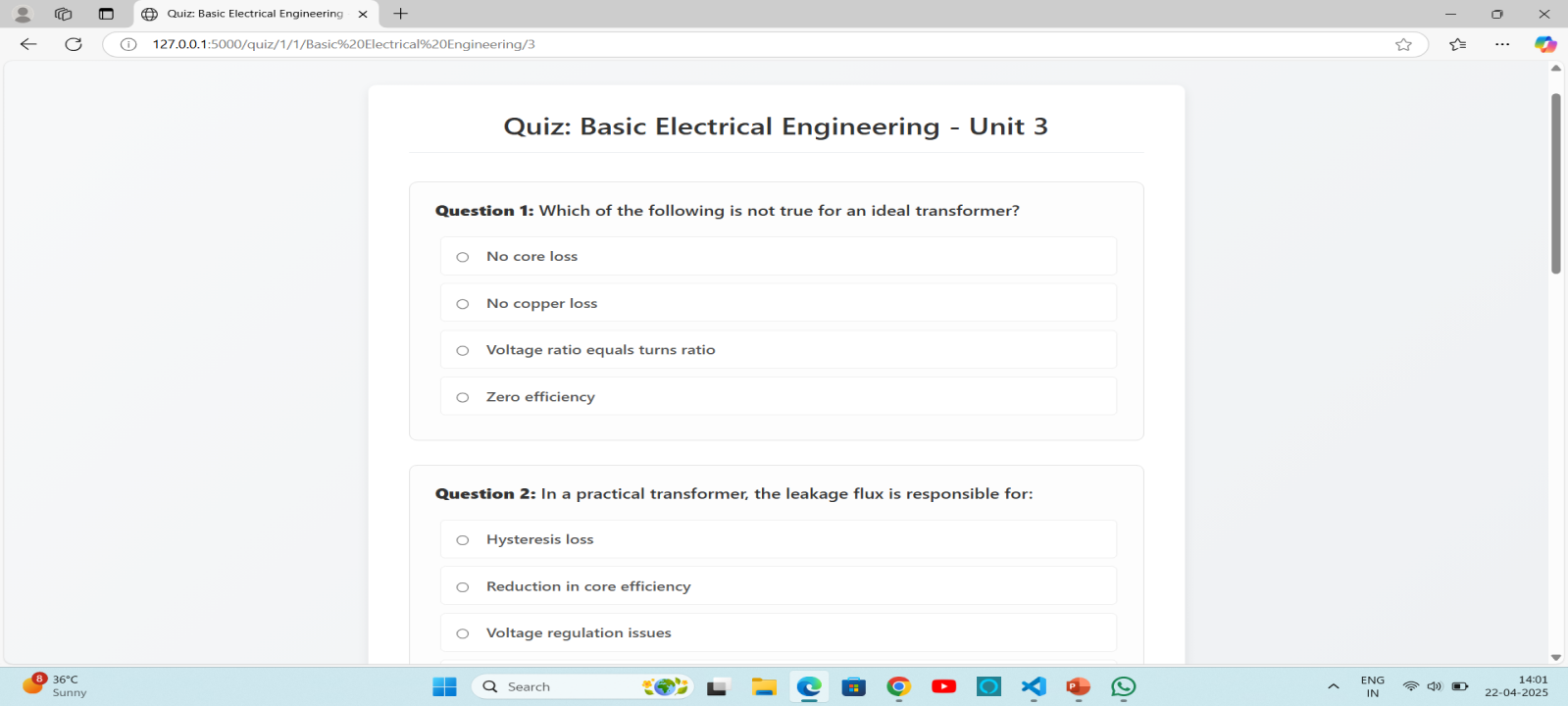
Access course videos to deepen your knowledge and test your skills. The Course Videos Page, depicted in Fig 7.7, is a vital component of the E-learning platform designed to enhance the learning experience through multimedia content. This section provides users with access to a rich library of educational videos, which serve as an engaging and effective medium to deepen their understanding of course material. These videos are integrated not only for knowledge acquisition but also to allow users to visually reinforce concepts and, where applicable, test their skills through practical demonstrations or problem-solving exercises, thereby supporting diverse learning styles.



**Fig 7.7 Course Videos Page**

**7.2.7 Quizzes**

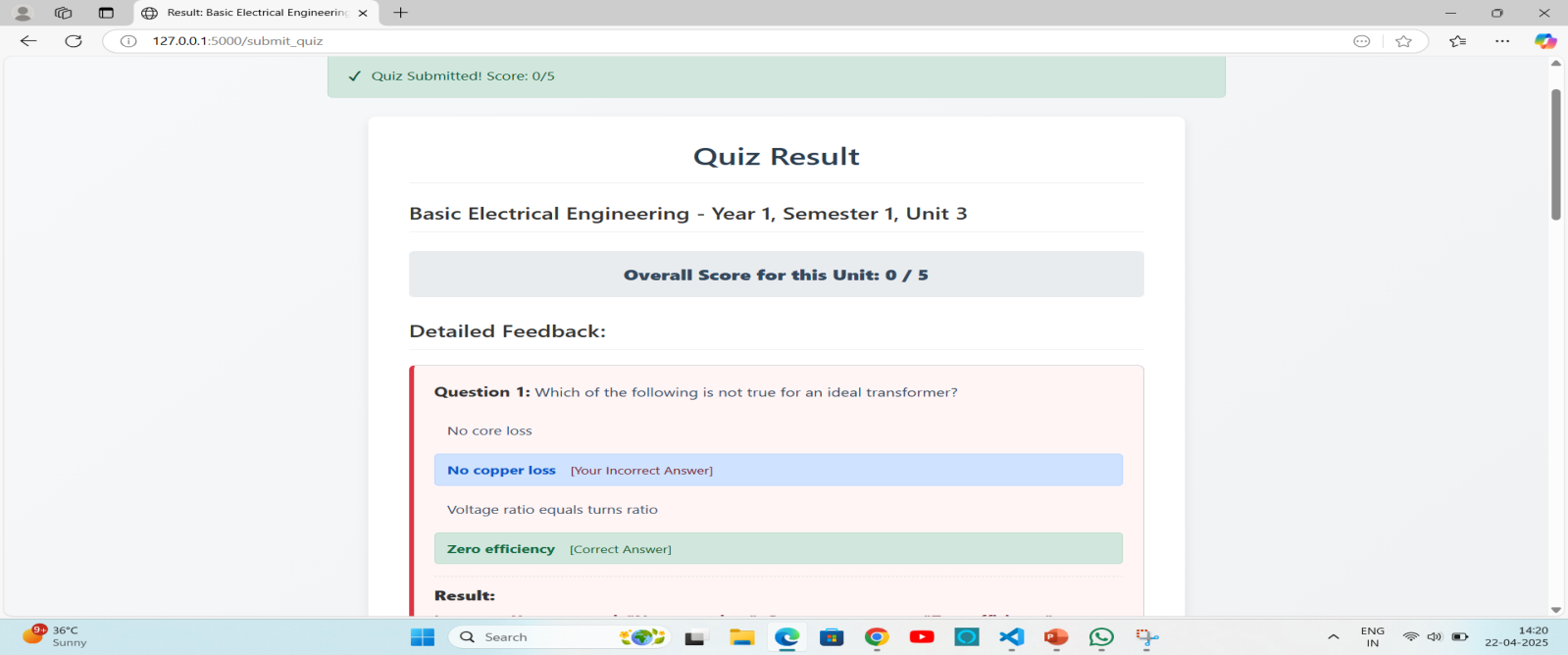
Every Question Counts — Master Every Concept. Each question is strategically weighted to ensure comprehensive understanding and tracking of progress.



**Fig 7.8 Quizzes Page**

**7.2.8 Quiz Result**

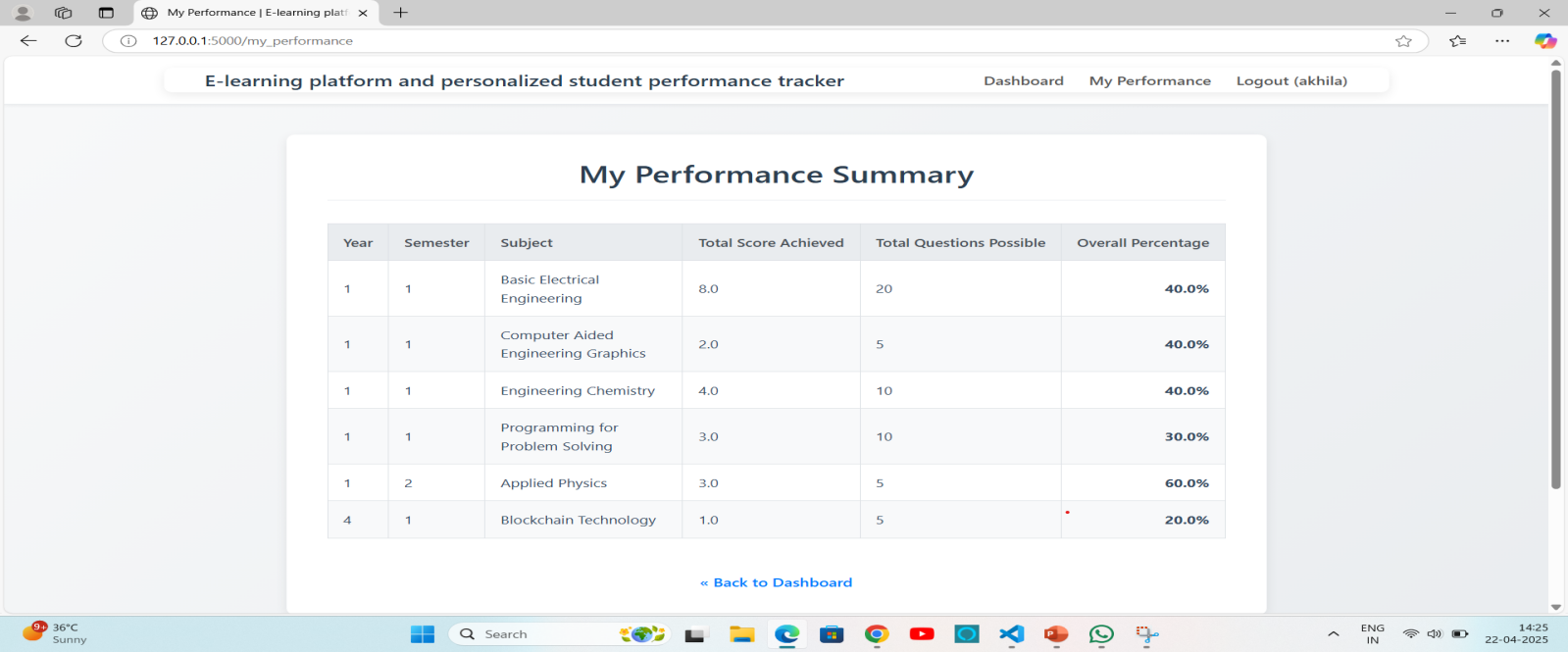
Result will be displayed based on correct answers. The 'Quiz Result' page visually presents the user's performance immediately after quiz completion. Results are calculated and displayed based on the number of correct answers, providing instant feedback on concept comprehension.



**Fig 7.9 Quiz Result Page**

**7.2.9 Performance**

Performance will be displayed based on quiz result. The **Performance** page provides users with a comprehensive and visual representation of their learning progress. This section aggregates data from all completed quizzes, allowing users to easily track their quiz results and identify areas where they excel or need further study. By consolidating this information, the Performance page serves as a crucial tool for self-assessment and continuous improvement.



**Fig 7.10 Performance Page**

**8. CONCLUSION AND FUTURE SCOPE**

**8.1 CONCLUSION**

The development of an e-learning platform with a personalized student performance tracker addresses critical gaps in current educational technology, particularly for Engineeringstudents. By integrating personalized learning paths, real-world applications, and advanced performance tracking, the platform aims to enhance the learning experience, making it more interactive, engaging, and industry-relevant. The use of modern technologies which ensures a robust, scalable, and flexible system that can adapt to future advancements. This project not only bridges the gap between theoretical knowledge and practical implementation but also empowers students to achieve their full potential through personalized feedback and adaptive learning. By leveraging web development, the platform sets a new standard for e-learning, preparing students for the demands of the ever-evolving tech industry.

**8.2 FUTURE SCOPE**

* **Mobile app integration:** Develop a cross-platform mobile app (Android/iOS) to make learning accessible anytime, anywhere.
* **Discussion Forums:** Add a forum or comment section under each course for peer-to-peer interaction and collaborative learning.
* **Third-party Integration (Zoom, Google Classroom):** Integrate with popular platforms for enhanced functionality and institutional adoption.

**9. REFERENCES**

[1] Alavi, Maryam and Dorothy E Leidner (2001). “Knowledge management and knowledge management systems: Conceptual foundations and research issues”. In: MIS quarterly, pp. 107–136.

[2] Aljawawdeh, Hamzeh (2022).“An Enriched e-Learning Model to Teach Kids in Arab Countries How to Write Code”. In: 2022 International Arab Conference on Information Technology (ACIT). IEEE, pp. 1–10.

[3] Almeida, Manuela Vaz de, Jo˜ ao JM Ferreira, and Fernando AF Ferreira (2019). “Developing a multi-criteria decision support system for evaluating knowledge transfer by higher education institutions”. In: Knowledge Management Research & Practice 17.4, pp. 358–372.

[4] Awad, Elias M et al. (2007). Knowledge management. Pearson Education India. Bonk, Curtis J and Charles R Graham (2012). The handbook of blended learning: Global perspectives, local designs.

[5] John Wiley & Sons. Donkin, Rebecca, Eva Hatje, and Nicole B Reinke (2022). “An eLearning module is comparable to face-to-face teaching in a nursing human pathophysiology subject”. In: Nurse Education Today 113, p. 105377.