

Smartlab: Next-gen virtual science laboratory (student level)



Group Members

Abdi Fatah Yusuf Mohamed (01-131222-002)

Rumman Saeed (01-131222-055)

Supervisor: Engineer Sulman Zafar

Final Year Project submitted to the Department of Software Engineering,
Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment
for the award of degree in Bachelor of Software Engineering

May 2026

FYP Completion Certificate

Student Name: Abdifatah Yusuf Mohamed Enrollment No: 01-131222-002

Student Name: Rumman Saeed Enrollment No: 01-131222-055

Programme of Study: Bachelor of Software Engineering

Project Title: Smartlab: Next-gen virtual science laboratory (student level)

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at _____ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: _____

Date: _____ Name: _____

Certificate of Originality

This is certify that the intellectual contents of the project **Smartlab: Next-gen virtual science laboratory (student level)** are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: **Rumman Saeed**

Signature: _____ Date: _____

Name of the Student: **Abdi Fatah Yusuf Mohamed**

Signature: _____ Date: _____

Smartlab: Next-gen virtual science laboratory (student level)

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	SDG 9	Industry, Innovation, and Infrastructure ✓
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities ✓
SDG 3	Good Health and Well Being	SDG 11	Sustainable Cities and Communities
SDG 4	Quality Education ✓	SDG 12	Responsible Consumption and Production ✓
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals



Range of Complex Problem Solving			
	Attribute	Complex Problem	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	✓
4	Familiarity of issues	Involve infrequently encountered issues	✓
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	
7	Consequences	Have significant consequences in a range of contexts.	✓
8	Interdependence	Are high level problems including many component parts or sub-problems	✓
Range of Complex Problem Activities			
	Attribute	Complex Activities	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	✓
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	✓
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	✓
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Abstract

Hands on laboratories are very critical in the study of Physics and Chemistry concepts of science. Nevertheless, most schools encounter issues like lack of resources in the laboratories, lack of equipments, safety issues and lack of accessibility to laboratory physical set-ups. As a solution to the problems stated above, this project introduces SmartLab: AI-Powered Virtual Lab, a web-based solution, that aims to offer students an interactive and intelligent virtual laboratory experience that complies with the FBlSE curriculum. The SmartLab system allows students to conduct virtual experiments in an easy to use interface through which students can choose experiments, change the experiment parameters, simulate the results of the experiment and create reports about the experiment. The system has been equipped with a machine learning model that offers the predictive-feedback and helps a learner to interpret the expected outcomes of experiments. This has been a brilliant feature that boosts conceptual learning as it directs the students when running experiments and it minimizes the errors encountered when choosing parameters. The system is built on the basis of recent web technologies such as Next.js, React and TypeScript, PostgreSQL, and Prisma ORM that guarantee scalability, maintainability, and effective data management. The modular design couples the frontend elements, backend services as well as database functions and enables unproblematic communication between modules of the system and future-proofing of expansion into multiple experiments. Moreover, the platform has role-based access control between students, teachers and administrators, which allows monitoring of the experiments, tracking performance and management of the system in a systematic way. SmartLab offers a convenient, risk-free and economical alternative to a conventional lab space by using simulative experimentation in combination with assistance of artificial intelligence. The proposed system on the whole enhances the teaching of science by increasing student engagement, remote learning, and also offers a smart digital lab to conduct Physics and Chemistry practical experiments.

Keywords: Virtual Laboratory, Artificial Intelligence in Education, Physics Simulations, Chemistry Simulations, Machine Learning, SmartLab, Educational Technology, Interactive Learning Systems.

Dedication

In honor of our beloved parents, who have always stood by our side whenever we needed them. Their endless love, prayers, sacrifices, encouragement, and unwavering belief in us have been the greatest source of strength throughout our journey. They continuously motivated us to overcome every obstacle and inspired us to move forward with confidence and determination. Whatever we have achieved today would not have been possible without their support and dedication. We are truly grateful for all that they have done for us, and this project is a small token of appreciation and gratitude for their countless sacrifices and unconditional care.

We would also like to express our heartfelt gratitude to our respected supervisor, whose valuable guidance, continuous support, constructive feedback, and encouragement played a vital role in the successful completion of this project. Their expertise, patience, and motivation helped us improve our work and overcome various challenges during the development process. We sincerely appreciate the time and effort they invested in mentoring us and guiding us toward achieving our goals.

Acknowledgments

We wish to our supervisor a heartfelt thanks to such invaluable advice, positive feedback, and constant support that they gave to us during the creation of SmartLab. They are engineer and have a background in educational technology, which played a crucial role in transforming this project into an actionable solution to science in education.

We owe our success in helping us to complete a project of this caliber and magnitude to the faculty of the Software Engineering department who gave us the background knowledge and skills required to achieve this success. Web development, database management, and software design principles courses were also indispensable when it came to developing this virtual laboratory platform.

A significant portion of this project was also made feasible by outstanding open-source software and platforms, such as Next.js, React, Three.js which is a 3D graphics platform, Prisma ORM, PostgreSQL, and Tailwind CSS. We are grateful especially to have the resources of Vercel and Neon that made it easy to have the resources and deploy the cloud and hosting of the database without challenges. There was a wealth of documentation and community backing to these technologies invaluable in the development process.

And last but surely not least, we would like to say our sincere thanks to our family and friends who have given their undying support, patience, and moral back up to this effort. Their faith in us enabled us to be encouraged in our difficult developmental stages.

Table of Contents

Table of Contents

FYP Completion Certificate	ii
Certificate of Originality	iii
Sustainable Development Goals	iv
Abstract	vi
Dedication	vii
Acknowledgments	viii
Table of Contents	ix
List of Figures	xii
List of Tables	xiii
Chapter 01	1
Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Main Contributions	3
1.4 Report Organisation.....	3
Chapter 02	5
Background Study / Literature Review	5
2.1 Limitations of Traditional Laboratory-Based	5
Practical Learning	5
2.2 Detailed Overview of SmartLab (Virtual Lab +	6
Academic Workflow).....	6
2.3 Review of Existing Virtual Lab Tools and Digital	7
Learning Platforms.....	7
2.4 Insights from Related Research (Simulation-Based Learning).....	8
2.5 Strengths and Weaknesses of Existing Systems.....	9
2.6 Gaps in the Current Market / Problem Area.....	10
2.7 How SmartLab Fills These Gaps	10

2.8 Chapter Conclusion	12
Chapter 03	13
System Requirements.....	13
3.1 Use Case Diagram.....	13
3.2 Development View	14
3.3 Data Model	15
3.4 Interface Requirements	16
3.5 Functional Requirements	18
3.6 Nonfunctional Requirements	41
3.7 Database Requirements	44
3.8 Project Feasibility Analysis	46
3.9 Chapter Conclusion	49
Chapter 04	50
System Design	50
4.1 Design Approach	50
4.2 Design Constraints.....	53
4.3 System Architecture	54
4.4 Logical Design	55
4.5 Dynamic View	56
4.6 Component Design	62
4.7 Data Models.....	63
4.8 User-Interface Design.....	64
4.9 Chapter Conclusion	71
Chapter 05	73
System Implementation	73
5.1 Introduction.....	73
5.2 Tools and Technologies	73
5.3 Development Process	75
5.4 Implementation Details of Key Features	77
5.5 Implementation Results.....	78

5.6 Chapter Conclusion	79
Chapter 06	81
System Testing & Evaluation	81
6.1 Introduction.....	81
6.2 Test Strategy	81
6.3 Component Testing	83
6.4 Unit Testing.....	84
6.5 Integrated Testing.....	85
6.6 System Testing	85
6.7 Test Cases.....	86
Chapter 07	101
Conclusion.....	101
7.1 Contributions	101
7.2 Reflections.....	102
7.3 Future Work.....	103
REFERENCES.....	105
APPENDIX A	106
APPENDIX B	111

List of Figures

Figure 1 Use Case Diagram	13
Figure 2 WBS	14
Figure 3 ER Diagram	15
Figure 4 Logical View.....	55
Figure 5 Activity Diagram (student experiment flow).....	56
Figure 6 Activity Diagram (AI Tutor interaction).....	57
Figure 7 Activity Diagram (Teacher Review and Feedback).....	58
Figure 8 Activity Diagram (Admin Experiment contennt management).....	59
Figure 9 Activity Diagram (school/user management workflow)	60
Figure 10 Activity Diagram (Simulation Engine Hybrid Execution)	61
Figure 11 Component Diagram.....	62
Figure 12 Package Diagram.....	62
Figure 13 Data Model	63
Figure 14 Landing page interface of the SmartLab virtual laboratory platform.....	64
Figure 15 Student login interface for accessing the SmartLab system.....	64
Figure 16 Teacher login interface for accessing the SmartLab system.....	65
Figure 17 Shows Creating Student Account	65
Figure 18 Showing Student Dashboard	66
Figure 19 Showing Students Experiments Catalog	66
Figure 20 Showing ScrewGuage Simulation	67
Figure 21 Showing Melting Point Of Napthalene Simulation.....	67
Figure 22 Showing Law of Refraction Simulation	68
Figure 23 AI Tutor Section	68
Figure 24 Teacher Dashboard	68
Figure 25 Teachers Classes Overview	69
Figure 26 Teachers Students Overview	69
Figure 27 Admin DashBoard	69
Figure 28 Total Experiments Overview	70
Figure 29 Teachers AI Tutor	70
Figure 30 Total Users Registered.....	70
Figure 31 School Admin Dashboard.....	71
Figure 32 School Classes Overview	71

List of Tables

Table 1 Differences and Similarities Between Traditional Lab Learning and SmartLab Virtual Lab System	11
Table 2 Use Case (Student Registration)	18
Table 3 Use Case (Student Login)	20
Table 4 Use Case (Manage Users)	21
Table 5 Use Case (Create and Manage Classes)	22
Table 6 Use Case (Manage Experiment Library)	23
Table 7 Use Case (Assign Experiment)	25
Table 8 Use Case (Browse and Launch Experiment)	26
Table 9 Use Case (Run Simulation).....	28
Table 10 Use Case (Adjust Exp parameter).....	30
Table 11 Use Case (AI Tutor)	31
Table 12 Use Case (view performance).....	33
Table 13 Use Case (Save Attempt)	34
Table 14 Use Case (view experiment history)	36
Table 15 Use Case (class exp progress).....	37
Table 16 Use Case (review attempt)	38
Table 17 Use Case (add feedback).....	40
Table 18 Legal Compliance	47
Table 19 Potential Concerns and Mitigation Strategies	48
Table 20 implementation details of key features	77
Table 21 component testing	83
Table 22 auth & role based test cases	86
Table 23 Authentication Validation and Access Control Test Cases.....	87
Table 24 Authorization, Session Management, and Logout Test Cases	88
Table 25 Student Dashboard and Assignment Module Test Cases	89
Table 26 Teacher Dashboard, Content Management, and Reporting Test Cases.....	92
Table 27 School Administrator Management and Access Control Test Cases	94
Table 28 Administrator Dashboard and User Management Test Cases	95
Table 29 Virtual Lab Simulation and Experiment Execution Test Cases	96
Table 30 Database Operations and Data Integrity Test Cases	97
Table 31 Non-Functional Testing and System Performance Test Cases.....	99

Chapter 01

Introduction

1.1 Motivation

Science does not just consist of reading chapters and learning formulas. At the SSC and HSSC level (Classes 9 to 12) students learn better concepts where they are able to actually conduct experiments, make readings and view how the results vary due to change of conditions. Activities also assist the students to gain confidence since it teaches them on how to operate the instruments, follow the right procedure and how to make the correct record of the results in the right manner.

But in practice most schools and colleges are not able to offer regular lab sessions. The equipment is sometimes insufficient, the laboratory is not always available to all classes, and when safety and lack of time are taken into consideration, it is hard to remember about proper performing of practical work. There are instances when students receive single or limited opportunities to do a practical preceding the exam, insufficient to make them understand well.

Owing to these problems, it is common to have students prepare practical exams by simply reading notes, video taping, or by observing an experiment being conducted by someone. This generates a learning gap:

students might be familiar with the theory, but not assured of procedure, measurements and practical knowledge. It is also a problem with teachers as they cannot easily monitor the extent to which each student comprehends a practical, which parts are not understood, and at which point they feel the most lost.

Our project SmartLab will solve this issue by offering a web-based system to place two things under the same roof:

For example, a virtual lab, in which students will be able to conduct practicals in the form of interactive simulations, and structured academic system, where teachers are

able to assign assignments, distribute notices/resources, and keep track of student learning.

SmartLab will facilitate hands-on learning in cases where hands-on access to physical labs is non-existent. The student would have an opportunity to perform a practical on a working example, learn the steps to perform and are aware of the principle the experiment is set to do. It also comes with a tiny guiding option (an AI tutor/help) to help students when they are at a loss, yet the overall learning process is teacher-led and curriculum-based.

1.2 Objectives

The main objectives of the project are:

Who: SSC and HSSC (Classes 912): To use virtual lab experience to enable students to carry out practical experiments that are safe and can be practiced on a large scale in terms of repetition through simulation. This serves to make students both learn through doing as well as reading.

- To create a role-based system with Students, Teachers, School Admins and System Admins, so that each user has the appropriate dashboard and permissions, and sensitive information is not the access of the wrong role.
- To assist teachers deal with the learning tasks by designing assignments, writing notices and uploading resources in a neat manner. This minimizes paperwork and simplifies the management of classes.
- To curb access and completion of the learning activities (assignments, notices, resources, lab practicals) of the students via a simple and transparent interface, that is, to enable them to know what to do next without being confused.

Section: To store data appropriately with a backend and database to ensure authentication, records, and academic operations are sound and consistent, even in cases when numerous users are working within the system simultaneously.

- To offer partial intelligent guidance with an AI tutor/help service, primarily to respond to frequent inquiries and provision to students during their learning, when an educator is not readily accessible.

1.3 Main Contributions

This project will add to the whole platform upon which none of the part of virtual lab simulation is distinct of learning management. Rather, SmartLab unites it all in a way that facilitates more structure and ease of provision in practical learning among SSC and HSSC students.

The most important input is that the students will be in a position to conduct practicals even not in the physical laboratory. They are able to practice an experiment, take steps and gain confidence prior to actual lab sessions or exams. This also favors students, who have varying learning speeds as they can retake without feeling time pressured.

The role based management system that aids the teachers and school administrators is another contribution. Instead of having to spread their tools all over, teachers can share academic material, manage tasks, and keep track of the learning process in a single location. It is also necessary to remember that the system will be used to record the activities, which will be helpful to evaluate and improve.

On the whole, this work contributes to simplifying the learning of science, making it less dangerous and more homogenous. It also offers a platform that can be enhanced in the long run by adding additional experiments and enhancing reporting and evaluation capabilities.

1.4 Report Organisation

The present report has the following chapters:

Chapter 1 (Introduction):

Introduces the drive behind SmartLab, the project purpose, the key contributions of the project, and describes the report structure in general. It reasons why virtual practical learning is valuable, and what is the issue that the project is solutions-focused.

Chapter 2 (Background Study / Literature Review)

A detailed description of the background study (and literature review) will be presented in this chapter

Talks about virtual laboratories, learning through simulation and web-based education. It also examines existing tools and findings that are related to the activities of digital science practical learning and learning management.

Chapter 3 (System Requirements):

Specifies the non-functional and functional requirements of SmartLab. It entails user roles requirements, critical workflows, UI requirements in addition to database requirements to sustain the system.

Chapter 4 (System Design):

Description of the entire system architecture, major module design, database design and user interface design. This chapter demonstrates how the system is made and how the various components relate with each other.

Chapter 5 (System Implementation):

Describes the technologies used and how the system was implemented. It contains information about the frontend, back end API, database integration, and how the simulation was to be implemented.

Chapter 6 (System Testing and Evaluation):

Introduces the test strategy, level of testing and the test cases of key modules. It checks on the system in regard to correctness, usability, and performance.

Chapter 7 (Conclusion):

Concludes the total project findings, addresses strengths and weaknesses as well as any future recommendations on how the project can be enhanced to take SmartLab to the next level.

Chapter 02

Background Study / Literature Review

This chapter introduces the background and related work related to the SmartLab Virtual Laboratory and Learning Management Platform. The reason is to describe the necessity of having a virtual lab with the SSSC/HSSC (Classes 912), what problems are potential with the traditional practical teaching and what digital solutions (virtual labs, simulations, learning platforms) are already available.

The chapter starts by discussing the limitations of traditional physical laboratories (Section 2.1), then gives an overview of SmartLab (Section 2.2). It then examines some existing virtual lab tools and simulation based learning research (Sections 2.3–2.4). Lastly, the strengths and weaknesses of the systems in place are pointed out (Section 2.5), gaps identified (Section 2.6) and how SmartLab fills in those gaps (Section 2.7). In Section 2.8, a succinct conclusion is presented.

2.1 Limitations of Traditional Laboratory-Based Practical Learning

Practical work is necessary in SSC and HSSC education, but physical laboratories have numerous constraints in the real world, which limit the quality and frequency of experiments.

To begin with, not all schools are equipped with all the equipment of all experiments. Still, despite the availability of a lab, there can be outdated equipment, damaged equipment, or inadequate equipment to facilitate large classes. This has many times compelled teachers to carry out experiments as demonstrations rather than allowing students to carry them out individually.

Second, there is limited time in the labs. Practicals tend to be brief often and preparation of equipment, laying out of apparatus, and safety are time consuming. This means that

students are likely to have a single opportunity to do a practical, which does not suffice in learning by repetition.

Third, safety is an issue. There are experiments that use electricity, chemicals, heat or delicate instruments. Some practical activities might be shunned in schools because of risk, absence of safety equipment, or untrained lab personnel. This decreases exposure of students to actual procedures.

Lastly, conventional on-the-job learning cannot be easily monitored and appraised. Teachers are able to check notebooks and viva responses and can hardly follow all the students step-by-step performance, his/her mistakes, progress. This results in a necessity to have an easier and more traceable system of practice-based learning support.

2.2 Detailed Overview of SmartLab (Virtual Lab + Academic Workflow)

SmartLab will be suitable in facilitating both practical learning and academic management. Rather than offering simulations alone, SmartLab surrounds the simulations with a full framework such that the entire learning process is coordinated.

Student Learning Features

SmartLab provides students with:

- A virtual laboratory where practicals can be carried out through interactive simulations.
- Guided step-by-step procedure to help students go through experiments in the proper order.
- Simulation readings / outputs that assist the students in the comprehension of the way the values vary with varying inputs.
- View of assignments, notices, and learning materials on their student dashboard.

- A small AI tutor / help option to assist students in case they have a question in the course of learning (remained light and friendly, not take over the teachers).

This will make the learning more active. Students are not reading, but doing, watching and practicing.

Teacher Management & School Management Features.

Together with the student module, SmartLab offers teacher and school staff tools:

- Teachers are able to design and administer assignments, publish notices and upload materials.
- Teachers can access report and record (where available) to see student activity and performance.
- Role-based access will make sure that only teachers, students, and admins can see what they should.

In brief, SmartLab is not simply a simulation site; it is an organized environment of realistic delivery of learning.

2.3 Review of Existing Virtual Lab Tools and Digital Learning Platforms

There are a number of existing virtual lab and simulation-based learning solutions on the market and in academia. Common categories include:

- General Virtual Lab Platforms (physics/chemistry/biology web based simulations)

Standalone Simulations (single practical simulations with simple controls)

- Learning Management Systems (LMS) which deal with content, but lack interactive lab simulation.

The following solutions are beneficial, yet most are limited to the use of SSC/HSSC:

Other platforms are more university-level lab-centric and do not correlate with local lists of SSC/HSSC practical.

- There are simulations that are graphically excellent, but lack the desirable learning process, procedure steps, and assessment aids.

Many systems either offer content management without simulations, or simulations without class/role-based management.

Teachers might not be able to easily allocate a practical track student performance and review in a single platform.

This motivates the need for a system that blends both sides: simulations + academic workflow.

2.4 Insights from Related Research (Simulation-Based Learning)

Some key aspects identified from research into digital education and simulation-based learning include:

- Learners retain information better through interaction with the system and getting feedback than when learning from theories alone.
- Virtual labs could increase engagement since students have the ability to input various numbers, repeat procedures, and learn from their mistakes without the fear of destroying costly equipment.
- Simulation-based learning is most useful in situations where real-life laboratories cannot be accessed, though this approach requires a systematic process to avoid turning it into a mere demonstration exercise.

- Educators could greatly profit from having the system provide organizational assistance in the form of assignment, resource allocation, and reporting, as it would enhance class delivery and not serve as a standalone tool.

This information informs the development direction of SmartLab as interactive simulations combined with an academic workflow.

2.5 Strengths and Weaknesses of Existing Systems

Strengths

- Availability of hands-on learning without having to use real-life devices
- Safe learning environment due to the ability to conduct dangerous or costly experiments
- Ability to repeat exercises to build confidence among learners
- Good for distance education and test preparation
- Most simulation tools have great visuals and control options

Weakness

- Many simulations don't correlate with the SSC/HSSC practical list
- Lack of a learning process and procedure guidance in many simulations
- Neglect of role-based management (student, teacher, and administrator distinctions)
- Inadequate tracking and report generation capabilities
- Some simulation tools cost money, need fast internet, and powerful hardware
- Lack of integration between "teaching workflow" and "simulation activities"

All of the above disadvantages indicate that current solutions are valuable but cannot replace the entire school-level practical learning system.

2.6 Gaps in the Current Market / Problem Area

From the analysis of the available systems and their shortcomings, the following areas of opportunity have been highlighted:

- Need for integrated platform: Few platforms offer both LMS and Virtual lab together.
- Not aligned with SSC and HSSC syllabus: Most platforms lack practical content which meets local SSC/HSSC curriculum standards from Class 9 onwards.
- No support for teacher intervention: Most platforms fail to offer an intuitive interface for assigning practice tasks and resource sharing by teachers.
- No tracking and reporting: Progress tracking of students in practice is lacking in almost all of the platforms analyzed.
- Lack of scalability and management functionalities: Platforms lacking role based access, administration, and management features will be hard for schools to manage.

The above findings clearly suggest a need for developing a new platform for learning, teaching, and practicals.

2.7 How SmartLab Fills These Gaps

SmartLab has been developed taking into account the limitations of current systems:

- It allows users to take practical lessons on procedures through virtual labs.
- It comes with multiple user types (Student, Teacher, School Admin, System Admin), thus making the use of the application realistic in schools.
- The academic workflows have been considered for the development process.

- The application has a proper backend and database architecture enabling safe data storage of users, classes, and learning progress.
- The platform is scalable since it enables the inclusion of further practicals and reporting features.

Table 1 Differences and Similarities Between Traditional Lab Learning and SmartLab Virtual Lab System

The comparison between conventional laboratory methods and the SmartLab Virtual Lab System is presented in **Table 1**. The table highlights the major differences in accessibility, cost, flexibility, and learning experience between both approaches.

Characteristics	Traditional Physical Lab	SmartLab Virtual Lab Platform
Accessibility	Limited by lab availability and schedule	Accessible anytime via web
Equipment Requirement	Requires physical apparatus and instruments	No physical equipment required
Safety	Risk of accidents (electricity, chemicals, heat)	Safe simulated environment
Practical Repetition	Limited; usually 1–2 attempts	Multiple attempts possible
Teacher Monitoring	Difficult to observe every student step	Better tracking via system records (where enabled)
Learning Approach	Hands-on but limited time	Interactive + repeatable, supports practice
Cost	Equipment purchase and maintenance needed	Lower running cost after setup
Curriculum Alignment	Depends on school resources	Can be designed to match SSC/HSSC practicals
Scalability	Hard to scale for large classes	Scales more easily with users and devices
Support	Teacher available only in class Teacher available only in class	Includes learning resources + limited AI guidance

2.8 Chapter Conclusion

This chapter has examined the background and related work regarding the SmartLab project, including the drawbacks of conventional laboratories for learning and the importance of using virtual laboratories for science education. From the literature, it is evident that virtual simulation can play a significant role in providing students with adequate support in their learning process.

Nonetheless, the review indicates that most of the currently available tools can only be considered effective in one of two aspects: simulations or learning management. Most of the existing virtual simulation systems have several limitations, such as the lack of integration with the practical requirements of SSC/HSSC schools, lack of role-based access, and poor reporting/tracking features.

From this analysis, the need for developing the proposed solution becomes apparent. SmartLab is designed to offer comprehensive support for both interactive practical learning and academic workflow. This chapter sets the stage for discussing the subsequent chapters of this thesis.

Chapter 03

System Requirements

3.1 Use Case Diagram

The overall functionality and interaction between users and the SmartLab system are illustrated in **Figure 1**. The use case diagram demonstrates how students, teachers, administrators, and the AI tutor interact with the system to perform experiments, track progress, manage users, and generate feedback and reports.

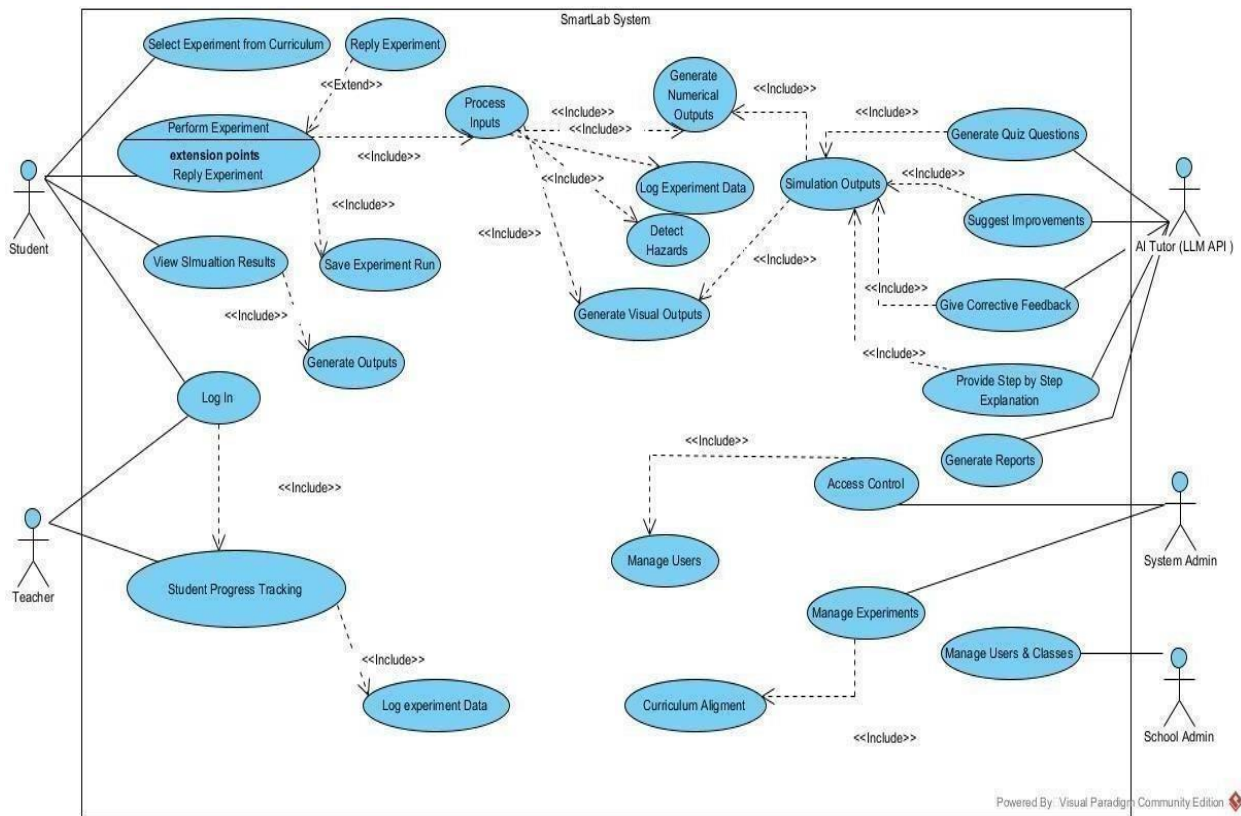


Figure 1 Use Case Diagram

3.2 Development View

Work Break Down Structure:

The complete workflow and development phases of the SmartLab project are illustrated in Figure 2. The diagram presents the major stages including initiation, requirement analysis, design, development, testing, and deployment activities.

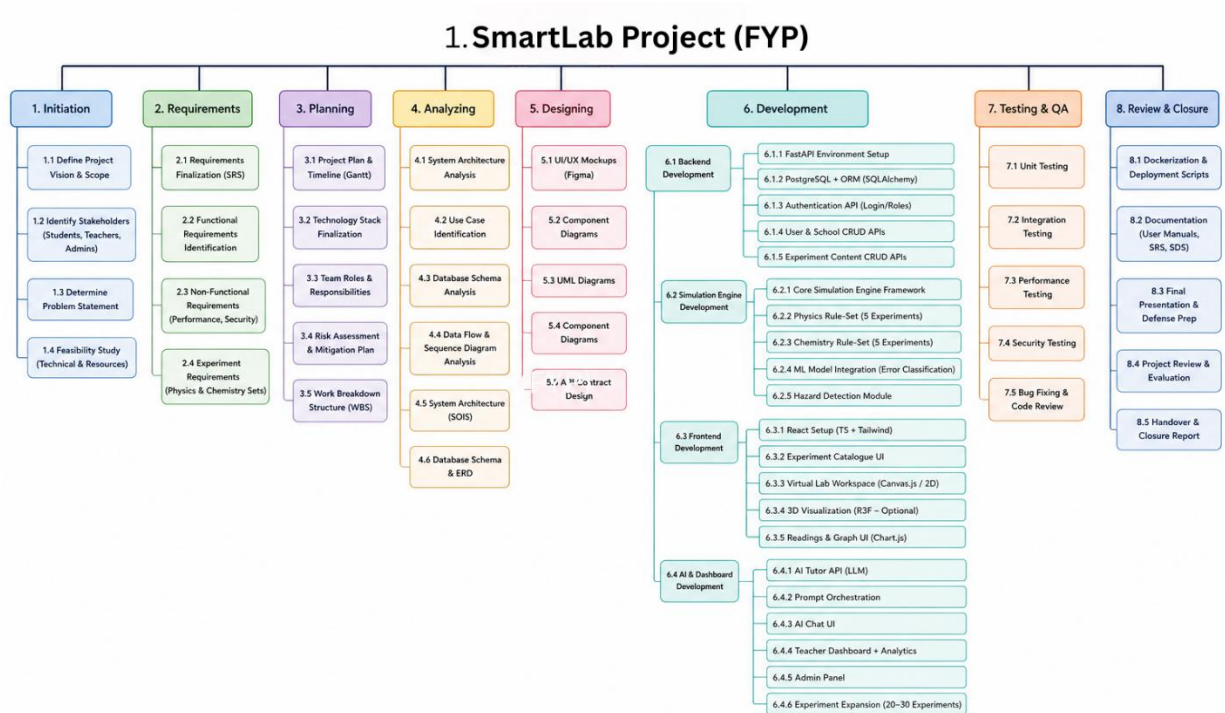


Figure 2: SmartLab Project Workflow Structure

Figure 2 WBS

3.3 Data Model

ER Diagram

Figure 3 shows the ER diagram of the proposed virtual laboratory database. The USER entity stores common user information, while STUDENT_PROFILE and TEACHER_PROFILE extend user-specific details. The EXPERIMENT entity manages experiment metadata, and EXPERIMENT_RUN records each student's experiment execution and outcomes.

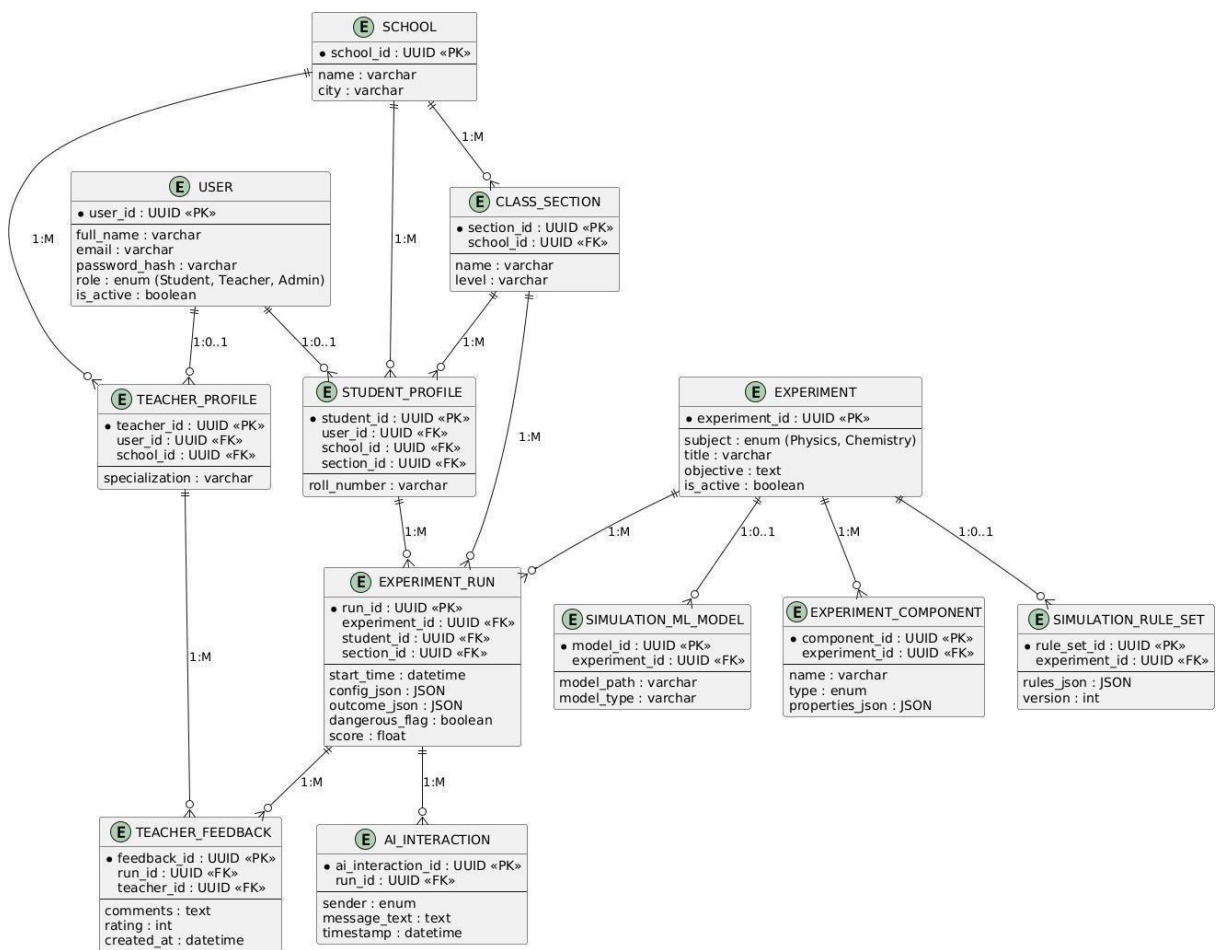


Figure 3 ER Diagram

3.4 Interface Requirements

3.4.1 User Requirements

SmartLab offers web interfaces that are reactive to users such as students, teachers, and administrators.

Interface for Students

- Login/Registration
- Experiment Dashboard
- Virtual Lab Interface:
 1. Left Side: Instruments and Parameters
 2. Center: Simulation Area (2D/3D)
 3. Right Side: AI Tutor, Readings, Warnings

Interface is designed with clarity, tooltips, animations, and accessibility in mind.

Interface for Teachers

- Class and Student Performance Dashboard
- Experiment Logs and Individual Attempt Views
- Graphical Data Analysis Tools and Reports

Interface for Administrators and School Administrators

- School Management
- Teacher Management
- Class and Student Management
- Experiment Management
- System Configurations
- AI Panel UI Standards
- Standardized Layout, Sidebar Navigation Menu, Action Buttons, etc.
- Contrasting Visual Design
- The detailed interface mockups are stored separately in Figma.

3.4.2 Hardware Interfaces

SmartLab is executed on regular computers without any additional lab equipment.

- Student Computers: Laptop or desktop computer where students perform virtual experiments and communicate with the AI Tutor.
- Teacher Computers: Laptop or desktop computer where teachers access dashboards, logs, and analytics.
- Server Computer: Cloud VM or PaaS where the FastAPI backend is executed, PostgreSQL is installed, and AI is integrated.

No IoT devices or sensors are needed for the initial version of SmartLab.

3.4.3 Software Interfaces

SmartLab interfaces with various software modules that provide virtual lab and artificial intelligence features.

- FastAPI Backend: Primary application module responsible for authentication, simulation logic, AI logic orchestration, and database connectivity.
- PostgreSQL Database: Database that holds all users' information, experiments, logging data, and analytics information (read/write operations).
- LLM Provider API: Remote HTTPS server to which requests for the generation of AI Tutor answers are made.
- (Optionally) E-mail Service: To send e-mails for resetting passwords or notifications.
- (Optionally) Logging/Monitoring Service: To log any issues or monitor app performance.
- Browser Environment: Environment where the browser application runs (including 2D/3D animation).

3.4.4 Communications Interfaces

SmartLab implements secure communication methods for enabling data transfer between system components.

- REST APIs (HTTPS/JSON): The main method of data transfer for CRUD, authentication, experiments, and dashboarding.
- WebSockets (if necessary): Can be used for streaming simulation results or AI responses.
- AI API calls (HTTPS): For secure interaction with the LLM service using authenticated requests.

Security Requirements

- TLS/HTTPS connection (TLS 1.2+) is required for all interactions.
- JWT tokens are used for authentication and sessions.
- Ai & Email API keys should be kept secure in the environment.

3.5 Functional Requirements

Student Registration

Table 2 Use Case (Student Registration)

The functional flow for student registration is presented in Table 2. The use case explains how a student creates an account in the virtual laboratory system by entering the required registration details.

Use Case ID:	UID-001
Use Case Name:	Student Registration
Actor(s):	Student

Pre-Conditions:	<ul style="list-style-type: none"> • Student has access to the SmartLab web URL. • Registration rules (if any) are defined.
Priority:	High
Basic Flow:	Student opens the registration page, enters details, and submits the form to create an account.
Actor Actions	System Response
1. Student opens registration page.	System displays registration form (name, email/phone, class, etc.).
2. Student enters details and submits.	System validates all fields and checks for existing account.
3. Student confirms email/OTP (if used).	System creates student account and shows success message.
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Student enters an already registered email/phone.	System shows: “Account already exists, please log in.”
A2. Student invalid or enters incomplete details.	System highlights errors and asks the student to correct them.

Student Login

Table 3 Use Case (Student Login)

The authentication process for students is described in Table 3. The use case outlines how registered students access the virtual laboratory system using their login credentials.

Use Case ID:	UID-002
Use Case Name:	Student Login
Actor(s):	Student
Pre-Conditions:	<ul style="list-style-type: none"> • Student is already registered in SmartLab. • Account must be active.
Priority:	High
Basic Flow:	Student opens the login page, enters credentials, and accesses the SmartLab dashboard.
Actor Actions	System Response
1. Student opens login page and enters credentials.	System verifies credentials and role.
2. Student clicks “Login”.	System creates a session/JWT and redirects to Student Dashboard.
Alternative Course of Action (if any)	
Actor Action	System Response

A1. Student enters incorrect password.	System shows error and limits login attempts.
A2. Student tries to log in with inactive/blocked account.	System displays: “Account inactive, contact your teacher.”

Manage Users

Table 4 Use Case (Manage Users)

User management operations in the virtual laboratory system are presented in Table 4. The use case explains how the System Admin or School Admin manages students, teachers, and school administrators by creating, updating, or deactivating user accounts.

Use Case ID:	UID-003
Use Case Name:	Manage Users(Students/Teachers/SchoolAdmins)
Actor(s):	System Admin / School Admin
Pre-Conditions:	<ul style="list-style-type: none"> • Admin is logged in.
Priority:	High
Basic Flow:	Admin creates, updates, or deactivates users in the system.
Actor Actions	System Response
1. Admin opens “Users” page.	System shows list of users with filters by role & school.

2. Admin clicks “Create User”.	System shows user creation form (role, name, email, class).
3. Admin submits form.	System creates user and sends credentials or activation link.
4. Admin edits or deactivates existing users.	System updates records and enforces access accordingly.
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Admin enters an email/phone that already exists.	System shows validation error.
A2. Admin attempts to delete a System Admin account.	System blocks action with “Not permitted” message.

Create & Manage Classes

Table 5 Use Case (Create and Manage Classes)

The process of creating and managing classes within the virtual laboratory system is described in Table 5. The use case explains how a School Admin creates classes and assigns teachers to them.

Use Case ID:	UID-004
Use Case Name:	Create & Manage Classes
Actor(s):	School Admin
Pre-Conditions:	<ul style="list-style-type: none"> • School Admin is logged in. • School profile is active.

Priority:	Medium
Basic Flow:	School Admin creates classes and assigns teachers.
Actor Actions	System Response
1. School Admin opens “Classes” page.	System displays all classes under the school.
2. Admin clicks “Create Class”.	System shows form for class name, year, and teacher assignment.
3. Admin submits form.	System creates the class and links it to the selected teacher.
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Class name already exists.	System shows: “Class already exists.”
A2. Teacher not assigned.	System prompts: “Please assign a teacher.”

3.5.1 Experiment Management

Manage Experiment Library

Table 6 Use Case (Manage Experiment Library)

Experiment library management operations are presented in Table 6. The use case explains how administrators add, configure, and manage virtual experiments in the system.

Use Case ID:	UID-005
Use Case Name:	Manage Experiment Library
Actor(s):	Admin
Pre-Conditions:	<ul style="list-style-type: none"> Admin is authenticated and authorized.
Priority:	High
Basic Flow:	Admin adds, configures, or disables experiments in the library.
Actor Actions	System Response
1. Admin opens “Experiments Management” page.	System displays list of existing experiments.
2. Admin clicks “Add Experiment”.	System shows form for title, class, subject, objectives, theory, safety, etc.
3. Admin defines simulation parameters and rules (formulas).	System validates fields and allows attaching rule/ML config files.
4. Admin saves experiment.	System stores experiment definition and marks it as active.
Alternative Course of Action (if any)	
Actor Action	System Response

A1. Admin submits form with missing mandatory fields.	System highlights fields and prevents save.
A2. Admin enters experiment that already exists (same subject + title).	System shows: “Experiment already exists.”
A3. Admin disables an existing experiment.	System hides experiment from Student & Teacher views.

Assign Experiments to Class

Table 7 Use Case (Assign Experiment)

Table 7 presents the process of assigning experiments to students through classes. Teachers or School Admins can allocate virtual lab activities for learning purposes.

Use Case ID:	UID-006
Use Case Name:	Assign Experiments to Class
Actor(s):	Teacher / School Admin
Pre-Conditions:	<ul style="list-style-type: none"> • Teacher is logged in. • Teacher is linked to one or more classes.
Priority:	Medium
Basic Flow:	Teacher assigns experiments to a class as homework or lab activity.
Actor Actions	System Response
1. Teacher opens “Assignments” page.	System shows available classes and experiments.

2. Teacher selects a class.	System loads that class's experiment assignment list.
3. Teacher selects an experiment and clicks "Assign".	System assigns the experiment to all students in that class.
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Experiment already assigned.	System shows: "Experiment already assigned to this class."
A2. Teacher selects a class with no students.	System displays: "No students in this class."

3.5.2 Virtual Lab Execution

Browse & Launch Experiment

Table 8 Use Case (Browse and Launch Experiment)

Table 8 presents the process of browsing and launching experiments in the virtual laboratory system. The use case explains how students access available experiments and start virtual laboratory sessions.

Use Case ID:	UID-007
Use Case Name:	Browse & Launch Experiment
Actor(s):	Student

Pre-Conditions:	<ul style="list-style-type: none"> • Student is logged in. • Experiments for their class/subject are configured in the system.
Priority:	High
Basic Flow:	Student browses available experiments and launches one to open the Virtual Lab Workspace.
Actor Actions	System Response
1. Student opens “Experiments” page.	System shows list of experiments filtered by class & subject.
2. Student filters by subject (Physics/Chemistry).	System updates experiment list.
3. Student selects an experiment.	System displays experiment details (objective, theory, safety).
4. Student clicks “Start Experiment”.	System loads Virtual Lab Workspace for that experiment.

Alternative Course of Action (if any)	
Actor Action	System Response
A1. Student opens the list but no experiments are configured for their class.	System displays: “No experiments configured for this class.”
A2. Student selects an experiment that has been temporarily disabled by admin.	System shows informational message and hides “Start” button.

3.5.3 Configure & Run Simulation

Table 9 Use Case (Run Simulation)

The simulation execution process is described in Table 9. This use case explains how students configure laboratory components, adjust parameters, and run simulations to observe experiment results.

Use Case ID:	UID-008
Use Case Name:	Configure & Run Simulation
Actor(s):	Student
Pre-Conditions:	<ul style="list-style-type: none"> • Student has launched an experiment (UID-002 complete). • Virtual Lab Workspace is loaded.

Priority:	High
Basic Flow:	Student configures lab setup and runs the simulation to observe results.
Actor Actions	System Response
1. Student drags lab components (e.g., power supply, resistor, beaker) onto workspace.	System renders components with animations and updates internal configuration.
2. Student connects components (wires, pouring chemicals, etc.).	System shows visual connections (e.g., wires snapping, liquids transferring).
3. Student adjusts parameters (e.g., voltage, resistance, concentration).	System updates configuration values in real time.
4. Student clicks “Run Simulation”.	System sends configuration to Hybrid Simulation Engine.
5. Student watches the experiment result.	System displays numeric readings, graphs, and visual effects in 2D/3D scene.
Alternative Course of Action (Dangerous or Invalid Setup)	
Actor Action	System Response
A1. Student configures a setup that is detected as unsafe/dangerous by rules.	System shows warning dialog describing risk (e.g., short circuit, explosion).

A2. Student clicks “Cancel” on the warning dialog.	System cancels run and highlights problematic components in red.
A3. Student clicks “Proceed Anyway” on the warning dialog.	System simulates dangerous outcome (e.g., sparks, explosion effect, component burn) and logs the event.

3.5.4 Adjust Experiment Parameters (UID-014)

Table 10 Use Case (Adjust Exp parameter)

Table 10 illustrates the process of adjusting experiment parameters before simulation execution. The use case explains how students modify experiment values such as voltage, resistance, or concentration within safe limits.

Use Case ID:	UID-009
Use Case Name:	Adjust Experiment Parameters
Actor(s):	Student
Pre-Conditions:	<ul style="list-style-type: none"> • Experiment is loaded in Virtual Lab. • Simulation is not currently running.
Priority:	Medium
Basic Flow:	Student edits parameters (numeric fields, sliders) before running simulation.

Actor Actions	System Response
1. Student clicks a parameter (e.g., voltage).	System highlights editable field or slider.
2. Student enters/adjusts the value.	System validates the input (range, units).
3. Student applies the change.	System updates configuration and recalculates predictions (if enabled).
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Value out of allowed range.	System shows warning and resets to nearest safe value.
A2. Invalid unit/format entered.	System displays validation error.

3.5.5 AI Tutor Module

AI Tutor Guidance

Table 11 Use Case (AI Tutor)

The AI Tutor interaction process is presented in Table 11. This use case explains how students communicate with the AI Tutor to receive guidance, explanations, and recommendations during virtual laboratory activities.

Use Case ID:	UID-010
Use Case Name:	AI Tutor Guidance
Actor(s):	Student

Pre-Conditions:	<ul style="list-style-type: none"> • Experiment is running or configured (UID-003). • AI Tutor service is available.
Priority:	High
Basic Flow:	Student asks the AI Tutor questions related to the experiment and receives guidance.
Actor Actions	System Response
1. Student types a question in AI chat (e.g., “Why is current zero?”).	System collects context: experiment details, current setup, simulation results.
2. Student sends the message.	Backend sends structured prompt to LLM, including simulation engine analysis.
3. AI Tutor generates response.	System displays explanation in chat and optionally highlights relevant components or values.
4. Student follows recommended correction.	System updates lab configuration and encourages student if improvement is observed.
Alternative Course of Action (if any)	
Actor Action	System Response

A1. Student sends a question but AI API is not reachable.	System shows: “Tutor is currently unavailable, please try again later.”
A2. Student asks a question outside curriculum scope.	Tutor provides generic explanation or refuses with a safe/bounded answer.

3.5.6 Performance & AI Insights

Table 12 Use Case (view performance)

Table 12 describes the performance monitoring and AI insights functionality. The use case explains how students review experiment performance, analytics reports, and AI-generated improvement suggestions.

Use Case ID:	UID-011
Use Case Name:	View Performance & AI Insights
Actor(s):	Student
Pre-Conditions:	<ul style="list-style-type: none"> • Student has completed multiple experiments. • Analytics module online.
Priority:	Low
Basic Flow:	Student views summary of mistakes, improvements, and experiment scores.
Actor Actions	System Response
1. Student opens “Performance” page.	System loads analytics report.
2. Student selects timeframe (weekly/monthly).	System recalculates graph.

3. Student views insight tips from AI Tutor.	System shows improvement suggestions.
Alternative Course of Action (if any)	
Actor Action	System Response
A1. No data available.	System displays: “No performance data yet.”

3.5.7 Experiment Attempt & History

Save Attempt

Table 13 Use Case (Save Attempt)

The process of saving and reviewing experiment attempts is illustrated in Table 13. This use case explains how students save simulation attempts and later review experiment configurations, results, and tutor feedback.

Use Case ID:	UID-012
Use Case Name:	Save & Review Experiment Attempt
Actor(s):	Student
Pre-Conditions:	<ul style="list-style-type: none"> • Student has completed at least one simulation run.
Priority:	Medium

Basic Flow:	Student saves an experiment attempt and later reviews the saved attempts.
Actor Actions	System Response
1. Student clicks “Save Attempt”.	System records configuration, results, AI interactions, and timestamp.
2. Student later opens “My Experiments / History”.	System displays list of attempts with key metrics (score, time).
3. Student selects a past attempt.	System shows summary report (setup, graphs, tutor feedback).
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Student opens “My Experiments / History” but has not saved any attempts.	System displays: “You have not saved any attempts yet.”

View Experiment History

Table 14 Use Case (view experiment history)

Table 14 presents the experiment history functionality. The use case explains how students access and review previously completed experiment attempts and download performance reports.

Use Case ID:	UID-013
Use Case Name:	View Experiment History
Actor(s):	Student
Pre-Conditions:	Student has at least one saved attempt.
Priority:	Medium
Basic Flow:	Student opens history section to view all past attempts.
Actor Actions	System Response
1. Student opens “History”.	System displays all attempts with timestamps.
2. Student selects an attempt.	System loads summary: setup, results, mistakes.
3. Student downloads report (optional).	System generates PDF/HTML report.
Alternative Course of Action (if any)	
Actor Action	System Response

A1. No attempts available.	System displays: “You have not saved any attempts yet.”
----------------------------	---

3.5.8 Teacher Monitoring & Feedback

Class Progress Monitoring

Table 15 Use Case (class exp progress)

The class experiment progress monitoring process is described in Table 15. This use case explains how teachers monitor student participation, experiment completion rates, and overall class performance.

Use Case ID:	UID-014
Use Case Name:	View Class Experiment Progress
Actor(s):	Teacher
Pre-Conditions:	<ul style="list-style-type: none"> Teacher is logged in and linked to one or more classes.
Priority:	High
Basic Flow:	Teacher views class-level and student-level experiment completion status.
Actor Actions	System Response
1. Teacher opens Teacher Dashboard.	System displays list of classes and quick stats.

2. Teacher selects a class.	System shows students and their completion % for each experiment.
3. Teacher clicks on a student entry.	System displays detailed view: attempts, scores, common mistakes, AI usage.
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Teacher opens a class that has no experiments assigned.	System displays info message and suggests assigning experiments.

Review Student Attempt

Table 16 Use Case (review attempt)

Table 16 illustrates the process of reviewing student experiment attempts. The use case explains how teachers analyze student experiment configurations, results, and simulation activities.

Use Case ID:	UID-015
Use Case Name:	Review Student Attempt
Actor(s):	Teacher
Pre-Conditions:	<ul style="list-style-type: none"> • Student has at least one saved attempt
Priority:	Medium

Basic Flow:	Teacher opens and reviews a specific student attempt and optionally adds feedback.
Actor Actions	System Response
1. Teacher selects a specific attempt.	System loads attempt details (config, timeline, AI interaction log).
2. Teacher views configuration & results.	System shows graphs, dangerous events, and most common mistakes.
3. Teacher optionally adds feedback comments.	System stores teacher feedback linked to that attempt.
Alternative Course of Action (if any)	
Actor Action	System Response
A1. Teacher opens attempts list but no attempts are available for that student.	System displays: “No attempts found for this student.”

Add Feedback

Table 17 Use Case (add feedback)

The feedback management process is presented in Table 17. This use case explains how teachers provide comments and guidance on student experiment attempts within the virtual laboratory system.

Use Case ID:	UID-016
Use Case Name:	Add Feedback to Experiment Attempt
Actor(s):	Teacher
Pre-Conditions:	<ul style="list-style-type: none"> • Student has at least one completed attempt. • Teacher is logged in.
Priority:	Medium
Basic Flow:	Teacher writes comments on student's experiment or mistakes.
Actor Actions	System Response
1. Teacher opens student attempt.	System displays detailed experiment report.
2. Teacher enters feedback text.	System validates input.
3. Teacher submits feedback.	System stores it and notifies student.
Alternative Course of Action (if any)	
Actor Action	System Response

A1. Teacher leaves feedback field empty.	System shows: “Feedback cannot be empty.”
--	---

3.6 Nonfunctional Requirements

These define how SmartLab should perform and behave.

3.6.1 Performance Requirements

- The system will handle up to 30–50 students conducting simulations simultaneously without any delays.
- Simulations conducted for typical experiments will be returned within 800ms.
- The AI Tutor will generate responses to questions within 3-8 seconds based on LLM latency.
- Student dashboards containing experiment logs up to 200 experiments will take less than 3 seconds to load.
- Teachers’ analytics aggregating data from up to about 500 students will take less than 5 seconds to load.
- The 2D/3D Virtual Lab should have at least 30 FPS performance even on midrange laptops.

3.6.2 Safety Requirements

- All dangerous results that can result from any experiment should be labeled as virtual simulations.

Confirmation dialogs should appear before performing any dangerous or invalid settings to explain the possible outcome of such virtual simulation.

- The system will not offer instructions on how to perform dangerous real world experiments.
- The confirmation dialog box must appear before performing any critical action.

- Experiments must be saved automatically throughout their process to avoid losing data.

3.6.3 Security Requirements

- Any communication between frontend, backend, and third parties must employ HTTPS/TLS (TLS 1.2 or higher).
- A user's password should be hashed through bcrypt/argon2.
- The solution should utilize JWT authentication with correct expiry and refresh policies.
- RBAC should guarantee that each category (students, educators, school admins, and system admins) can only gain access to what is allowed based on role assignments.
- The information about students needs to abide by minimal-data approach and should not show up in teacher analytics if not needed.
- A key used for AI or deployment purposes should be stored securely and must never be visible in the frontend application code.

3.6.4 Software Quality Attributes

Usability

- Interface shall be intuitive for students 14–18, with user-friendly graphics/icons and labeling.
- Virtual Lab workspace shall simulate realistic equipment layout and behavior.

Maintainability

- Code base shall utilize modular architecture, with distinction between front-end, backend, simulation engine, and AI orchestration layers.
- All major components must have accompanying documentation and/or commenting.

Scalability

- Back-end and database must scale horizontally/vertically as user numbers grow.

- APIs to AI and Simulation Engine services should be able to queue or process requests concurrently.

Reliability

- System shall log critical errors and recover after any back-end error gracefully.
- System shall not corrupt/discard any experiment data in case of normal operation failure.

Portability

- System should be deployable in standard cloud Linux environment and developer workstations.
- Providing docker images will help with portability.

Accessibility

- System interfaces will utilize some minimal standards of accessibility (e.g., contrast ratio, keyboard navigation).

3.6.5 Business Rules

System Admins and School Admins alone can create and edit experiments and simulation rules.

- Students can view only those experiments that have been assigned to their class unless otherwise authorized.
- The AI Tutor's response must conform to the FBISE syllabus and not be in contradiction with any theory.
- All dangerous or wrong experiment attempts should be recorded for analysis by teachers.
- Experiments can be made compulsory or optional for each individual class.
- Data on student performance cannot be exchanged between different schools.

3.6.6 Other Requirements

- **DATABASE:** PostgreSQL having suitable indexing, schema constraints, and migration.

- **LOCALISATION:** Version currently in English, and further development for labels/tutor responses in Urdu/Roman Urdu.
- **Legal/Ethics:** Abide by the privacy policy of the school level; maintain essential data only.
- **HOSTING:** Cloud hosting of system required, and database backups must be set up.
- **Deployment:** Docker based deployment of Backend + Simulation Engine + Database.

3.7 Database Requirements

What Information Should Persist

Information our system should remember forever:

Users

- Student and teacher accounts (name, email, password)
- Role (student, teacher, admin) • Their school and class

Experiment Information

- Catalog of experiments
- Metadata about each experiment (name, subject, level, steps)
- Images/video for experiment catalog

Student Usage

- Which experiments a student has attempted
- Their score and progress on each experiment
- How much time was spent in the lab

- Any feedback submitted/rating provided

Assignments

- Assignments generated by teachers for students to complete
- Due dates and completion status

Other

- School information
- Class rosters , • Any safety incidents?

Size at Launch

- Approximately 500 MB. Enough space for hundreds of students.
- Approximately 50 MB additional storage per month as students are added.
- Images/videos for experiments are uploaded to cloud storage service, not stored in database.

Database Choice

PostgreSQL — Open source SQL database built to last.

Why PostgreSQL?

- It's free and popular
- Great at relational data (students belong to classes, classes belong to schools)
- Supports JSON storage (helpful for storing experiment information)
- Offers robust security features out of the box

Why Neon?

- Free tier meets our projects needs

- Automated backups
- Can scale up as our user base grows
- Managed service, no need for us to operate a server

3.8 Project Feasibility Analysis

1. Technical Feasibility

Technology Available

- All required tools are free and well-documented (Next.js, React, Three.js, PostgreSQL)
- No proprietary software needed
- Large developer communities for support

Team Skills

- Our team knows JavaScript/TypeScript and React
- Database management with Prisma is straightforward
- 3D graphics with Three.js has extensive tutorials

Hardware Requirements

- Development: Any modern laptop works fine
- Deployment: Cloud hosting (Vercel + Neon) handles server needs
- Users only need a web browser — no special equipment

2. Operational Feasibility

User Acceptance

- Students: Familiar with web apps, intuitive interface similar to YouTube/Instagram
- Teachers: Simple dashboard for assigning work and checking progress
- Admins: Basic CRUD operations (Create, Read, Update, Delete) — standard stuff

Training Needs

- Minimal training required
- Students can start immediately with guided tutorials
- Teachers need 1-2 hours to learn the dashboard

Infrastructure at Schools

- Most schools have internet and computer labs
- Works on tablets and phones too (responsive design)
- No installation needed — just a browser

3. Legal & Ethical Feasibility

Legal Compliance

Table 18 Legal Compliance

Table 18 shows the legal and ethical feasibility of the SmartLAB project. It confirms compliance with copyright rules, user data protection, standard terms of service, and age-appropriate educational content.

Aspect	Status
Copyright	All 3D models and code are original or properly licensed
Data Protection	User passwords encrypted, no sensitive financial data stored
Terms of Service	Standard user agreement included
Age Appropriateness	Content suitable for ages 12+ (SSC/HSSC students)

Ethical Considerations

Benefits

- Access - Students can perform experiments that may not have been accessible to them otherwise (distance learning)
- Safety - No exposure to chemicals or physical harm when performing experiment virtually
- Equality - Helps bridge the gap between schools with more funds and schools with less funds
- Eco-Friendly - Eliminates physical experiment waste

Potential Concerns & Mitigations

Table 19 Potential Concerns and Mitigation Strategies

The table 19 highlights potential concerns related to the SmartLAB project and the strategies used to address them, including screen time management, data privacy, and maintaining the importance of physical lab experiments.

Concern	How We Address It
Screen time	Recommended usage limits built into guidelines
Replaces real labs?	Designed to supplement, not replace physical experiments
Data privacy	Minimal data collection, clear privacy policy
Cheating	Individual progress tracking, randomized experiment parameters

Student Safety

- No toxic chemicals to spill
- No electrical shocks

- No broken glassware
- "Hazard detection" feature teaches safety awareness

3.9 Chapter Conclusion

This chapter is about the system requirements of the AI-Powered Virtual Lab platform.

It talks about how users interact with the system. The chapter uses something called use case analysis and requirement modeling to explain this. The development structure is also explained. This is about how different parts like simulation engines and AI assistance services work. These parts include dashboards and experiment repositories. They all work together to make digital experimentation possible. The data model is important too. It explains how the system stores and manages experiment results and user activity records. The system uses PostgreSQL and Prisma ORM to do this. The interface descriptions show how students and teachers use the system. They use dashboards that work well on devices. These dashboards were made using Next.js and React.

The system has to work so there are functional requirements. These requirements make sure that things like experiment execution and report submission work properly. There are also non-functional requirements. These requirements ensure that the system is reliable and secure.

All these details are important for building the AI-Powered Lab. They provide a foundation, for the technical work. This will be useful when designing the architecture, which is explained in the chapter. The AI-Powered Virtual Lab platform needs a foundation to work properly.

Chapter 04

System Design

4.1 Design Approach

The system uses PostgreSQL and Prisma ORM to do this. The interface descriptions show how students and teachers use the system. They use dashboards that work well on devices. These dashboards were made using Next.js and React.

The system has to work so there are functional requirements. These requirements make sure that things like experiment execution and report submission work properly. There are also -functional requirements. These requirements ensure that the system is reliable and secure.

All these details are important for building the AI-Powered Lab. They provide a foundation, for the technical work. This will be useful when designing the architecture, which is explained in the chapter. The AI-Powered Virtual Lab platform needs a foundation to work properly.

The system design is based on the following principles: The system uses PostgreSQL and Prisma ORM to do this. The interface descriptions show how students and teachers use the system. They use dashboards that work well on devices. These dashboards were made using Next.js and React.

The system has to work so there are functional requirements. These requirements make sure that things like experiment execution and report submission work properly. There are also -functional requirements. These requirements ensure that the system is reliable and secure.

All these details are important for building the AI-Powered Lab. They provide a foundation, for the technical work. This will be useful when designing the architecture, which is explained in the chapter. The AI-Powered Virtual Lab platform needs a foundation to work properly.

User-friendly learning environment

We have a learning environment that's easy to use.

The system lets students do experiments in a step by step way:

- * They pick an experiment

- * Then they put in the details they need

- * Next the system runs a simulation

- * After that the system makes a prediction using intelligence

- * Students then write down what they saw

- * Finally they make a report

This way of doing things is like being in a laboratory but it is all, on the computer.

Layered system architecture

The architecture is split into many different layers to improve maintainability and scalability:

- Presentation Layer (Next.js & React UI)
- Application Layer (API Routes & Services)
- Data Layer (PostgreSQL w/Prisma ORM)
- AI Prediction Layer (Machine Learning)

This approach enhances the organization of the system.

Component-based frontend structure

The frontend follows the React's pattern of being built on components:

pages/ – navigation components/ hooks/ – logic components/ – UI components store/
– state

This approach gives much flexibility for further expansion.

API-driven backend communication

The backend is built on Next.js API routes which take care of:

user authentication , experiment simulations , artificial intelligence prediction , report generation , data management tasks.

This facilitates seamless communication between the frontend and database services.

Machine learning integration

An intelligent learning algorithm is incorporated into the system to improve the understanding of experiments through: prediction of experiment outcomes
feedback based on parameters conceptual support to the learner

Security and rolebased access control

There are different levels of users that include:

Students → carry out experiments and prepare reports

Teachers → evaluate performance and read the report

Admin → manage users and the experiments carried out.

4.2 Design Constraints

Some factors that influenced the design of the SmartLab include:

1. Restrictions based on the web platform

As a browser-based application, performance may be impacted by internet connection and browser support.

2. Restrictions based on the machine learning model

Due to limitations imposed by student projects:

lightweight ML models were adopted datasets were smaller in size real-time prediction was optimized for speed

3. Curriculum matching restriction

Experiments conducted using SmartLab are aligned with FBISE Physics and Chemistry curriculum.

4. Database restrictions

SmartLab utilizes PostgreSQL and Prisma ORM and was designed for hosting on academic-level hosting services instead of enterprise-grade ones.

5. Security and privacy considerations

Authentication, password security, and access controls have been included to enhance the safety of student information.

4.3 System Architecture

4.3.1 Architectural Overview

SmartLab uses the Layered Architecture approach to achieve modularity and maintenance within its design. The following are the main layers:

1. Presentation Layer (Frontend - React)

- Presents the UI of the app for students, teachers, and admins
- Displays 2D/3D labs and AI Tutor interaction
- Does client-side validation and interactions

2. API/Application Layer (FastAPI)

- Provides REST APIs for authentication, experiments, simulation, AI, and analytics
- Implements business rules and applies role-based access

3. Simulation Layer

- Runs the Physics and Chemistry experiment code
- Supports correct, incorrect, and hazardous experimental configurations
- Calculates sensor readings and generates visuals and outcomes of experiments

4. AI Tutor Layer

- Prepares the context for the LLM
- Sends prompts and receives the structured answer from the LLM

5. Data Layer (PostgreSQL)

- Holds users, experiments, attempts, analytics, and AI Tutor logs
- Accessed via SQLAlchemy models and repo

4.4 Logical Design

Figure 4 shows the class diagram of the Virtual Laboratory Management System. The diagram illustrates the relationships between major entities such as School, User, StudentProfile, TeacherProfile, Experiment, and ExperimentRun. It also demonstrates how students perform experiments, interact with the AI Tutor, and receive teacher feedback within the system.

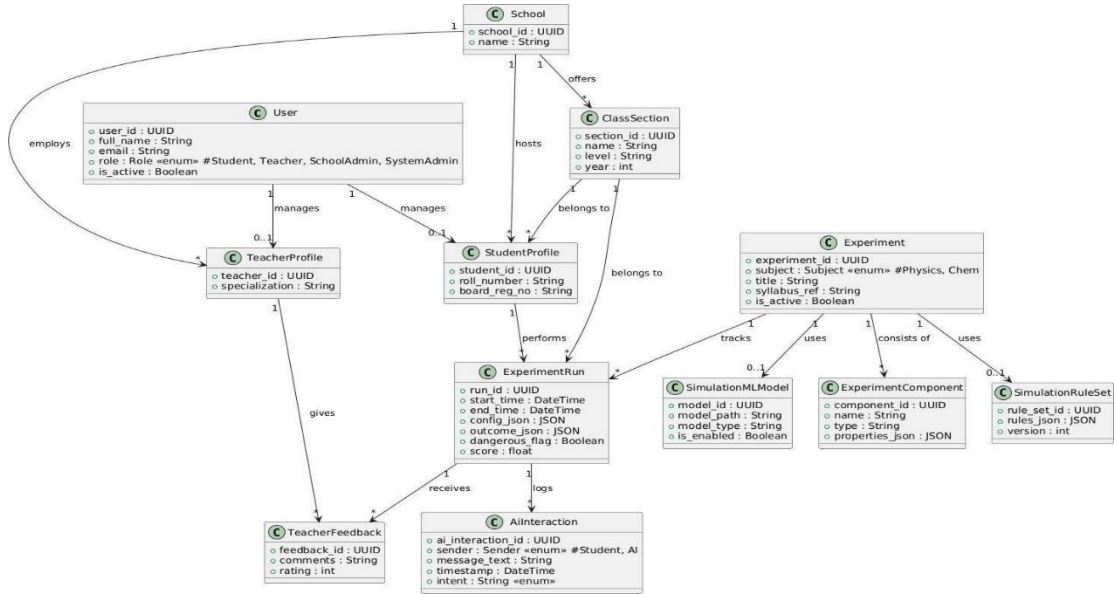


Figure 4 Logical View

4.5 Dynamic View

4.5.1 Activity Diagrams

Student Experiment Flow

Figure 5 shows the workflow of the SmartLAB virtual experiment process, including experiment setup, simulation execution, hazard detection, and result generation.

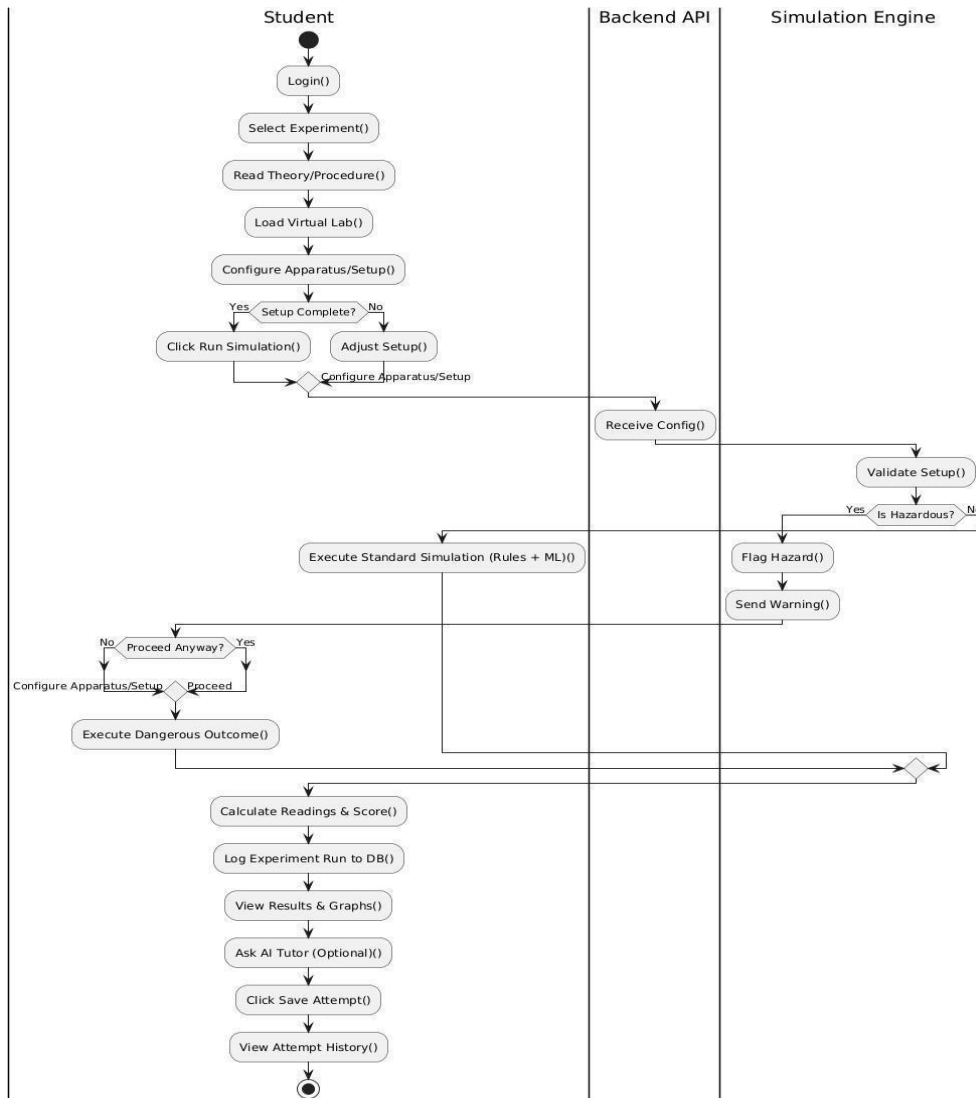


Figure 5 Activity Diagram (student experiment flow)

4.5.1.2 AI Tutor Interaction Workflow

Figure 6 illustrates the AI Tutor interaction workflow, showing how student queries are processed through the backend API and AI tutor service to generate contextual learning responses.

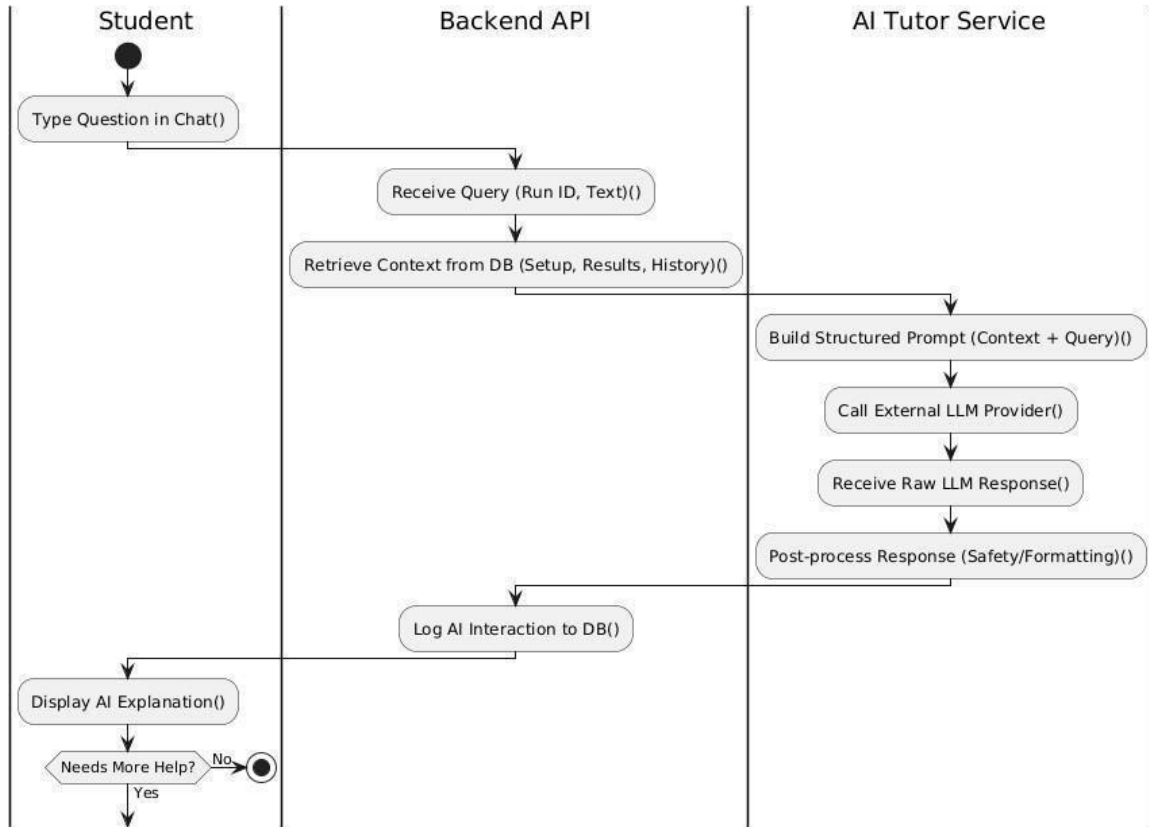


Figure 6 Activity Diagram (AI Tutor interaction)

4.5.1.3 Teacher Review and Feedback Flow

Figure 7 illustrates the teacher review workflow in SmartLAB, including class progress analysis, student attempt evaluation, and feedback submission through the backend system.

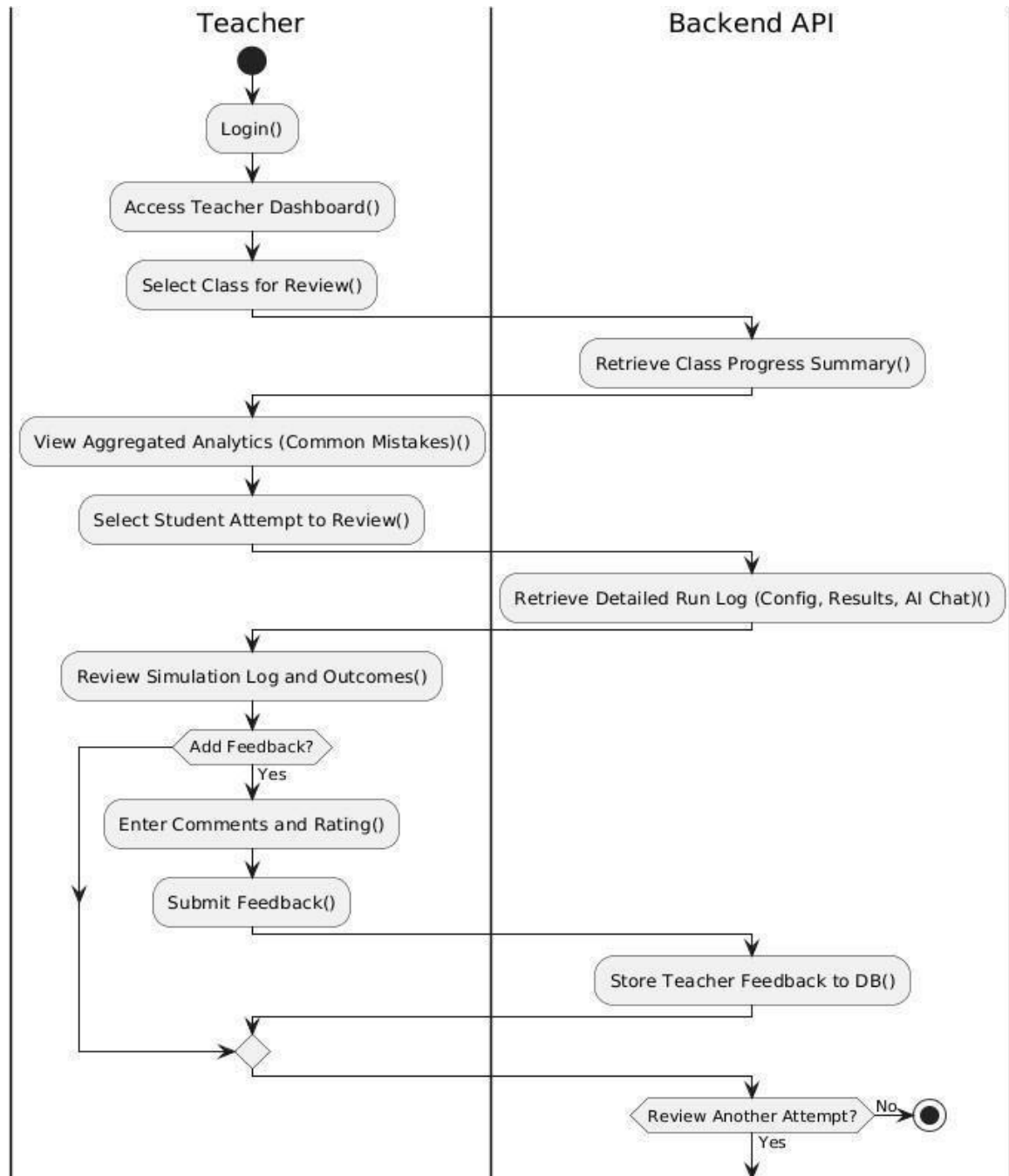


Figure 7 Activity Diagram (Teacher Review and Feedback)

4.5.1.4 Admin Experiment Content Management

Figure 8 illustrates the admin workflow and backend API interaction for creating, managing, and validating experiments in an experiment management system.



Figure 8 Activity Diagram (Admin Experiment content management)

4.5.1.5 School/User Management Workflow

Figure 9 illustrates the workflow for school administrators to manage classes and users through the backend API with validation and confirmation processes.

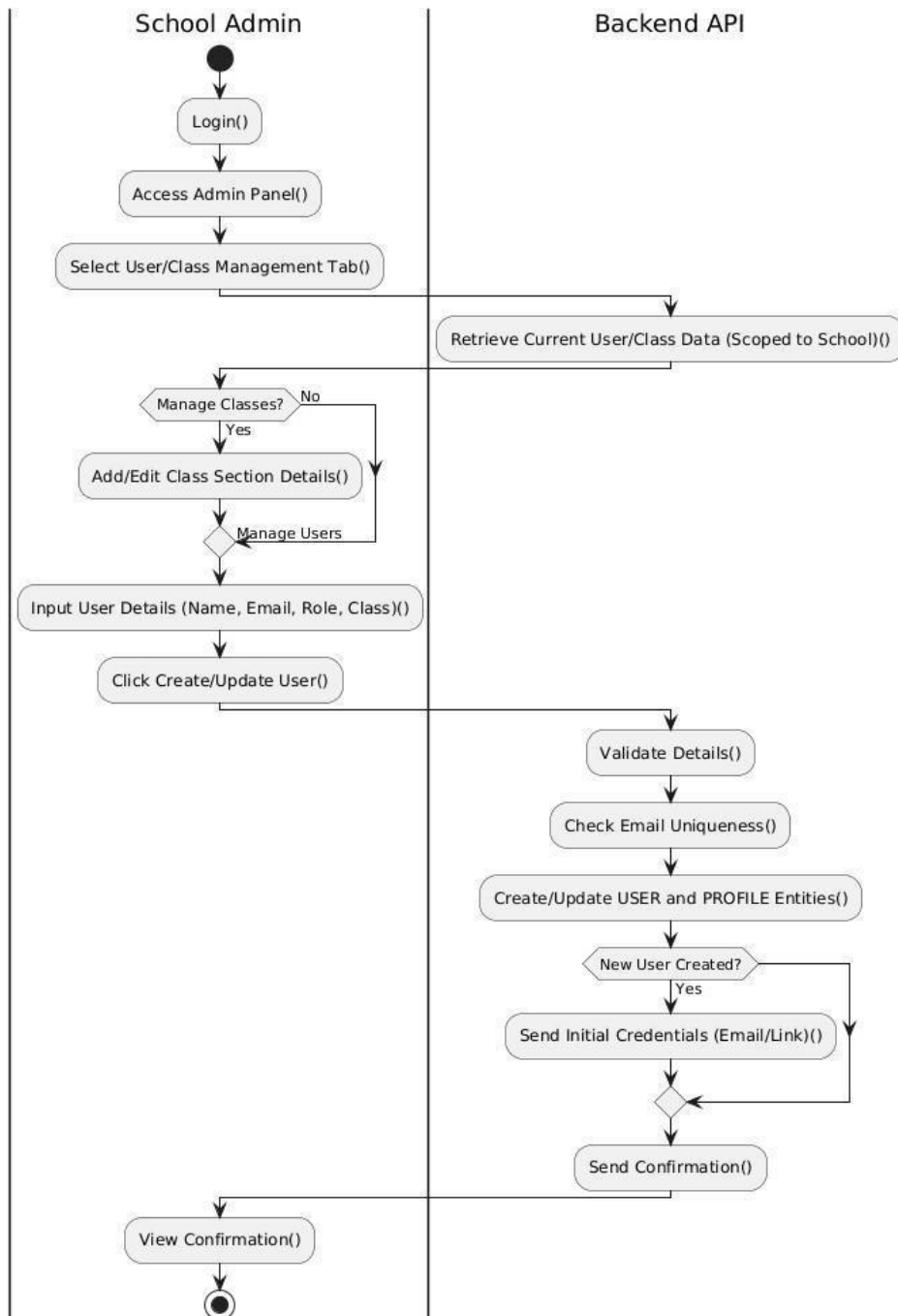


Figure 9 Activity Diagram (school/user management workflow)

4.5.1.6 Simulation Engine Hybrid Execution (Internal Flow)

Figure 10 depicts the backend API and simulation engine workflow for executing experiments, applying safety checks, performing rule-based and ML-based analysis, and returning simulation results.

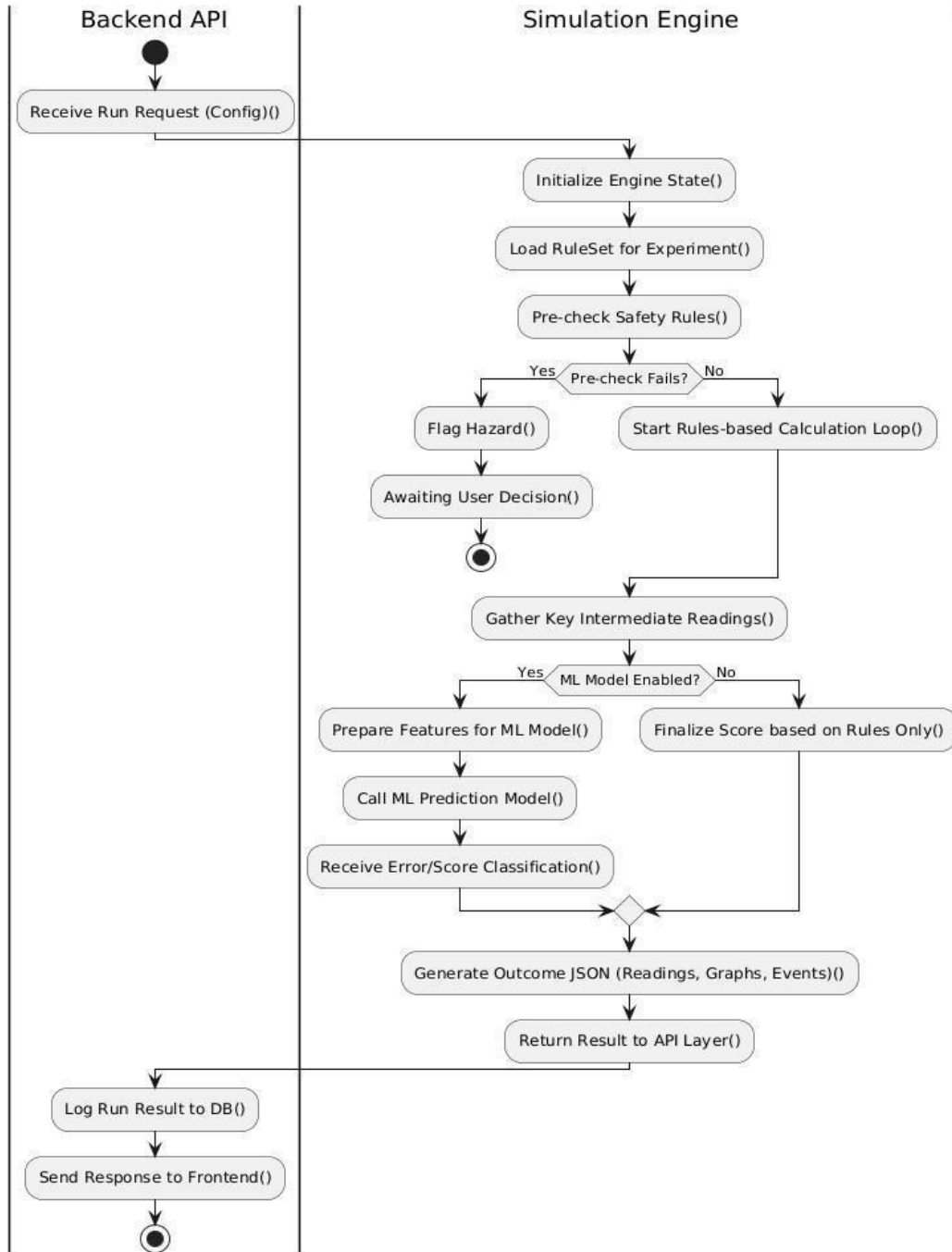


Figure 10 Activity Diagram (Simulation Engine Hybrid Execution)

4.6 Component Design

4.6.1 Component Diagram

Figure 5 illustrates the system architecture of the Virtual Laboratory Management System. The architecture consists of the presentation layer, application layer, simulation layer, and data layer, which work together to support experiment execution, AI tutoring, analytics, and data management.

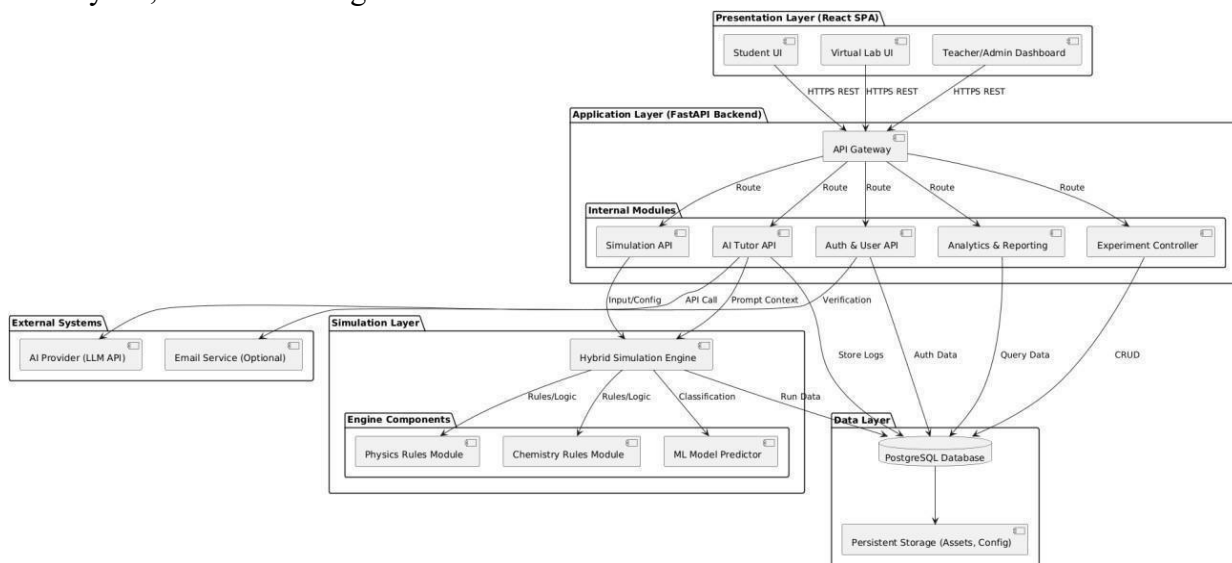


Figure 11 Component Diagram

4.6.2 Package Diagram

Figure 12 High-level system architecture illustrating the interaction between the presentation layer, application APIs, simulation engine, external services, and data access components.

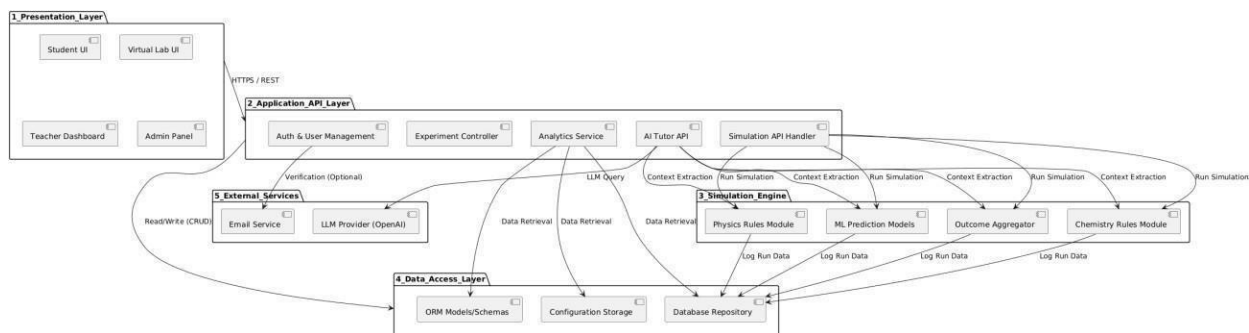


Figure 12 Package Diagram

4.7 Data Models

Figure 13 Entity Relationship Diagram (ERD) showing the database schema and relationships between schools, users, experiments, simulation models, AI interactions, and feedback modules in the virtual lab system.

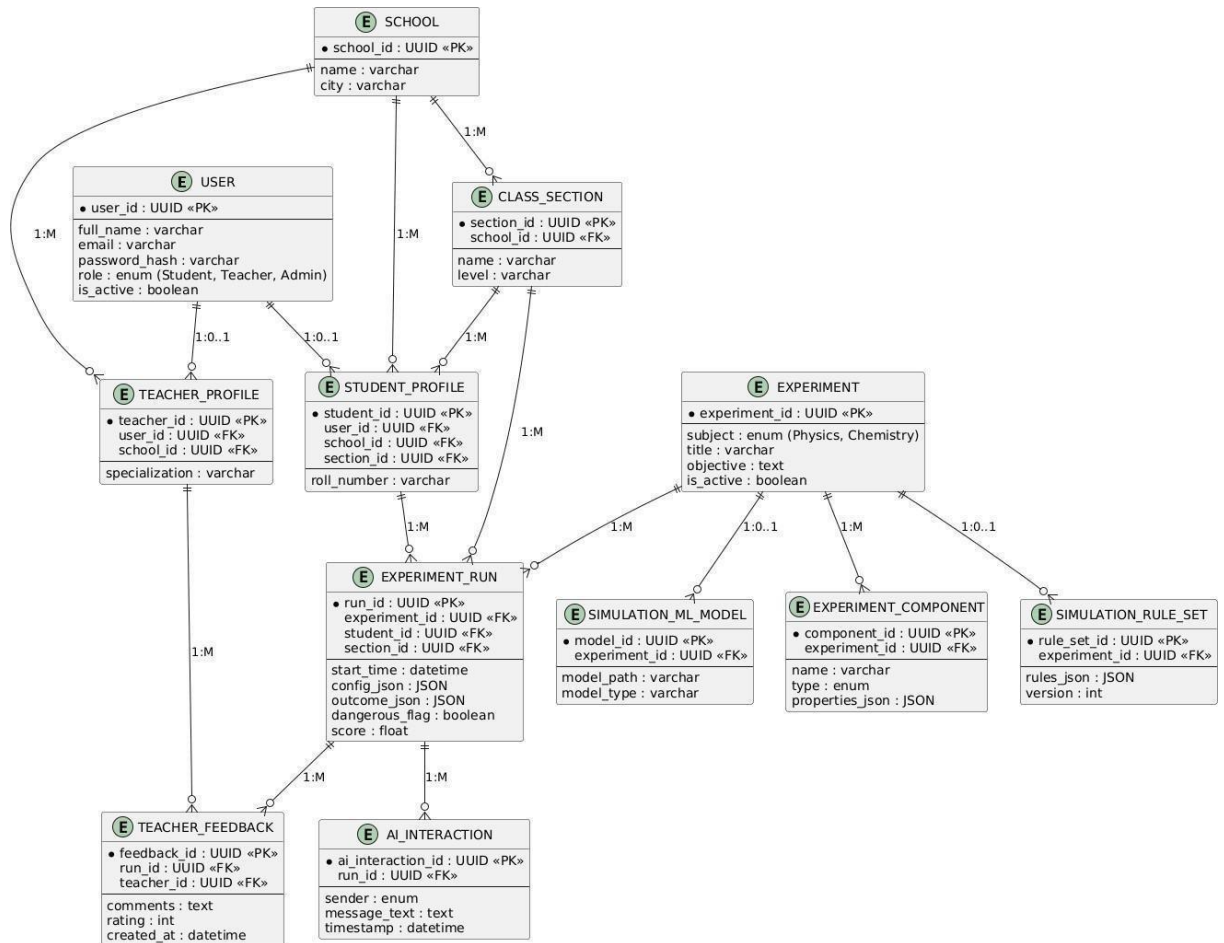


Figure 13 Data Model

4.8 User-Interface Design

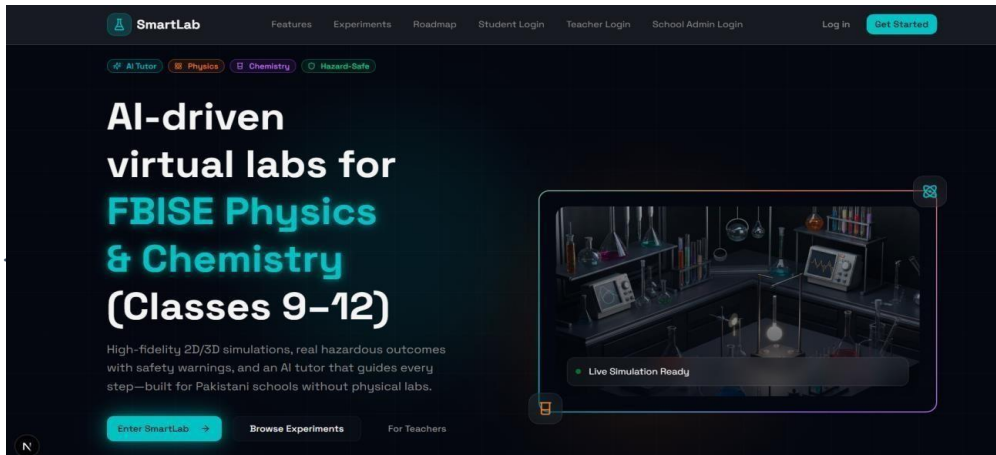


Figure 14 Landing page interface of the SmartLab virtual laboratory platform

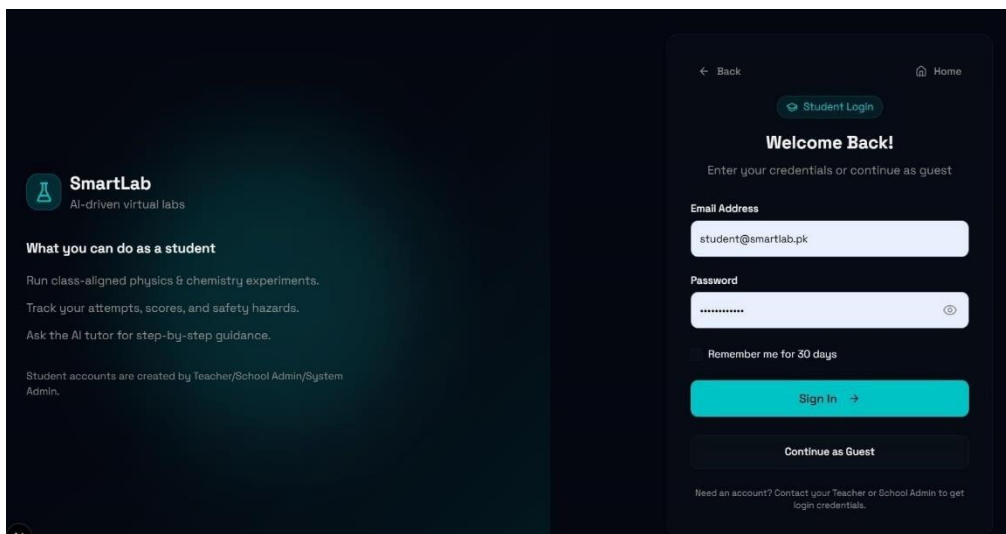


Figure 15 Student login interface for accessing the SmartLab system

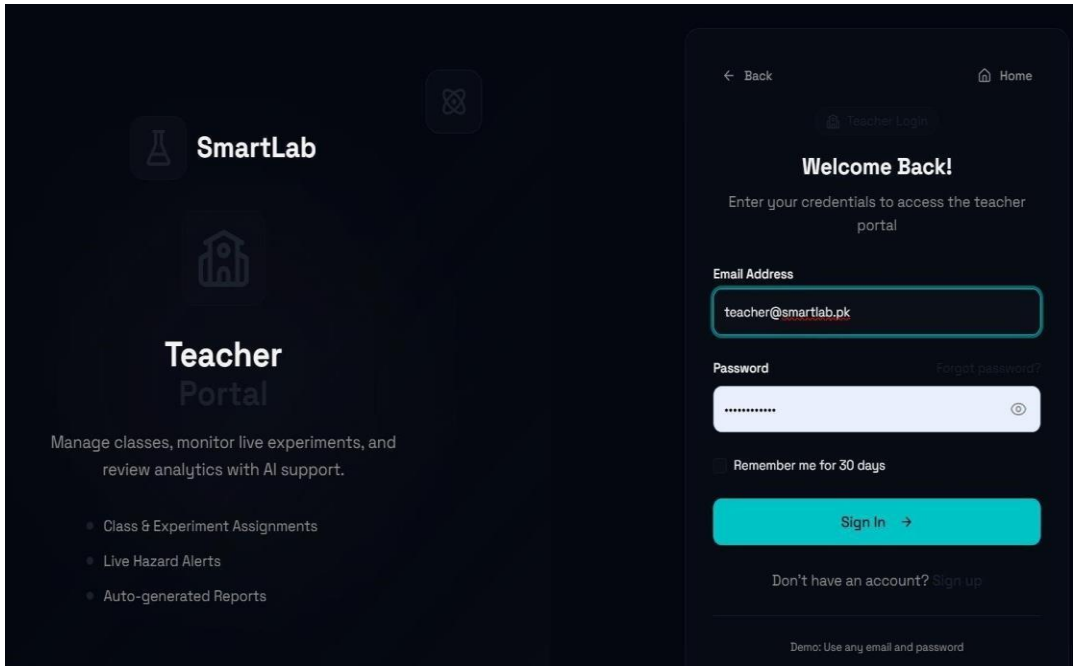


Figure 16 Teacher login interface for accessing the SmartLab system

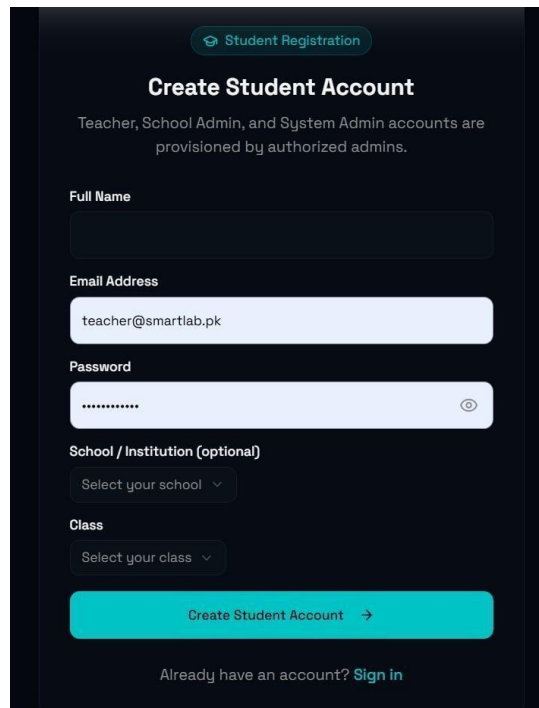


Figure 17 Shows Creating Student Account

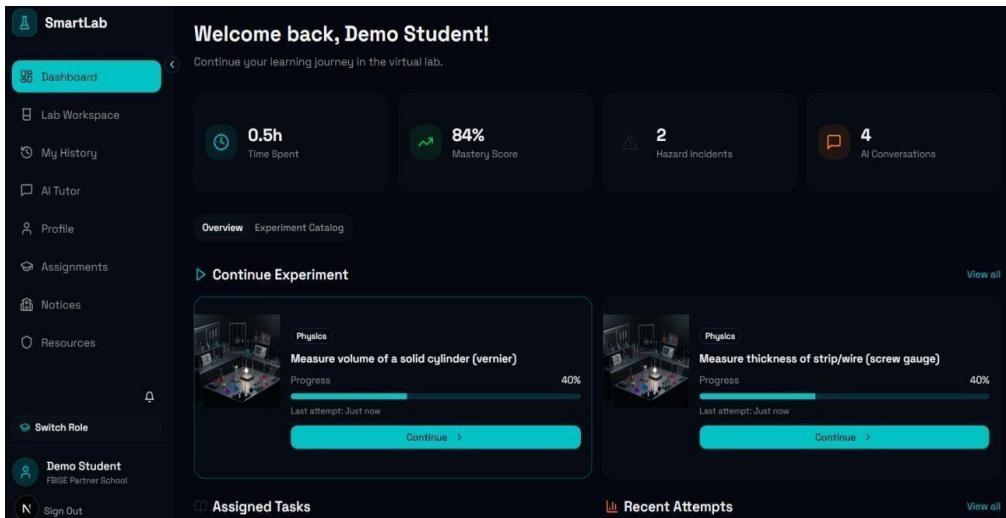


Figure 18 Showing Student Dashboard

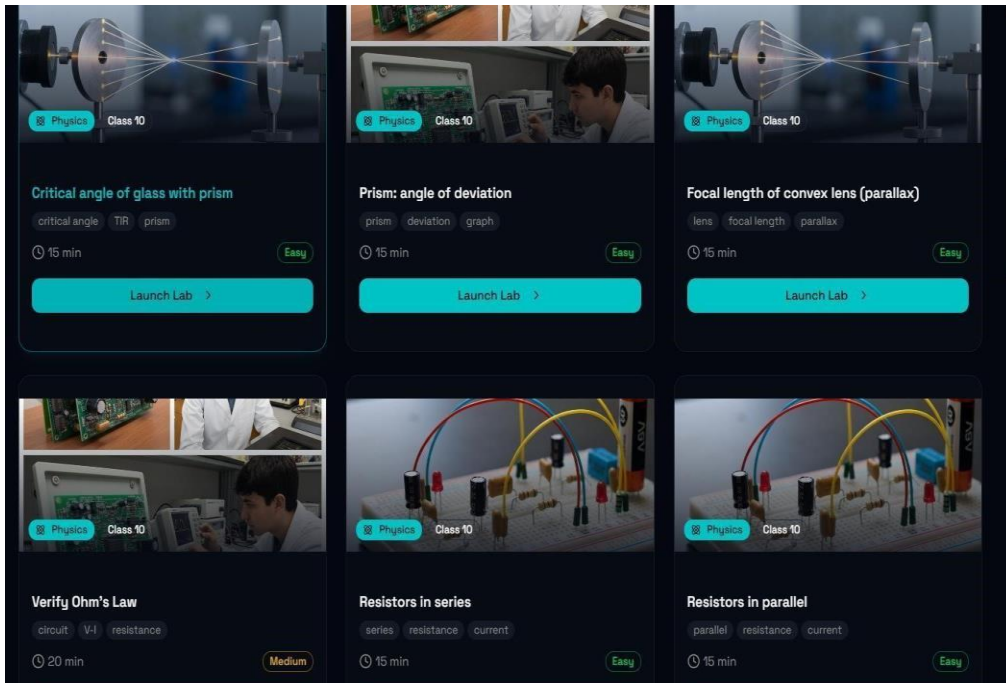


Figure 19 Showing Students Experiments Catalog

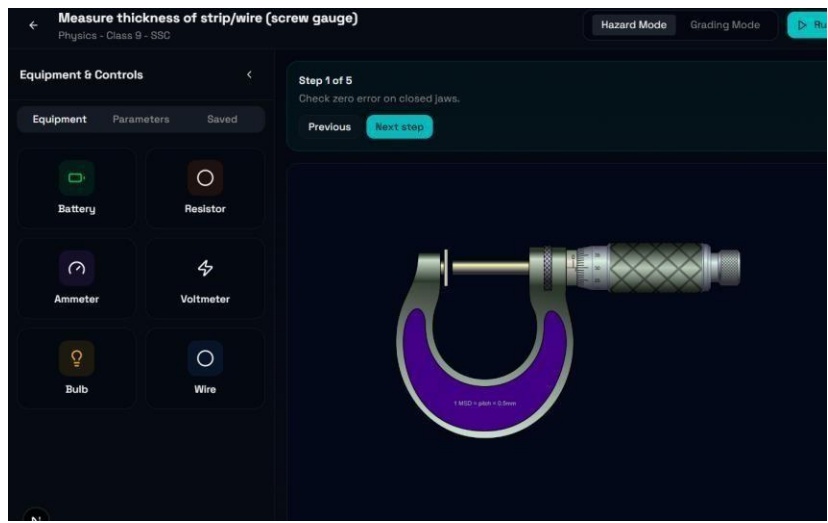


Figure 20 Showing ScrewGauge Simulation

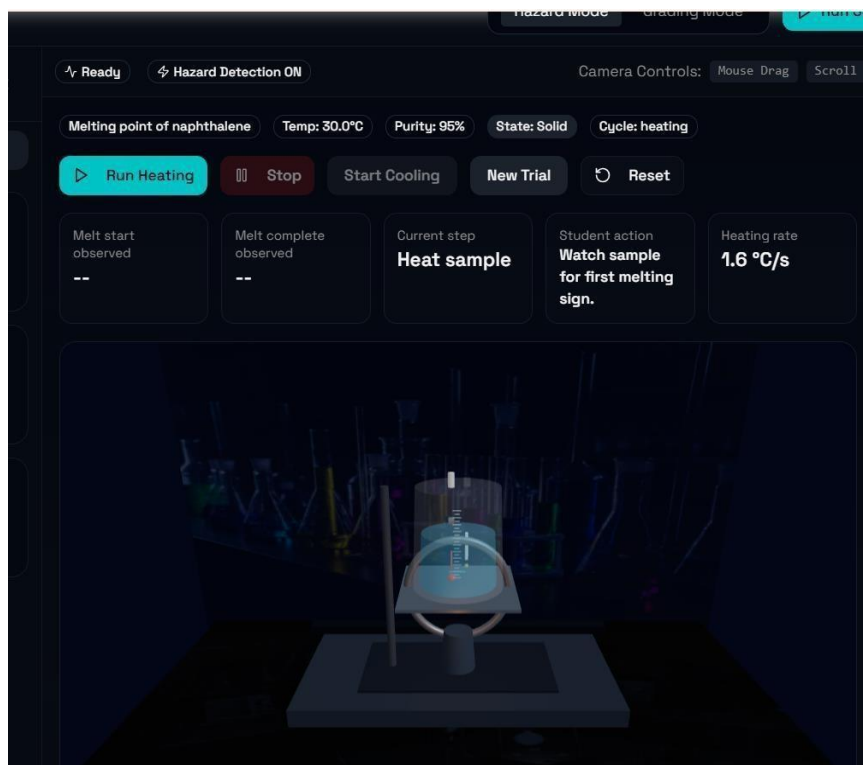


Figure 21 Showing Melting Point Of Naphthalene Simulation

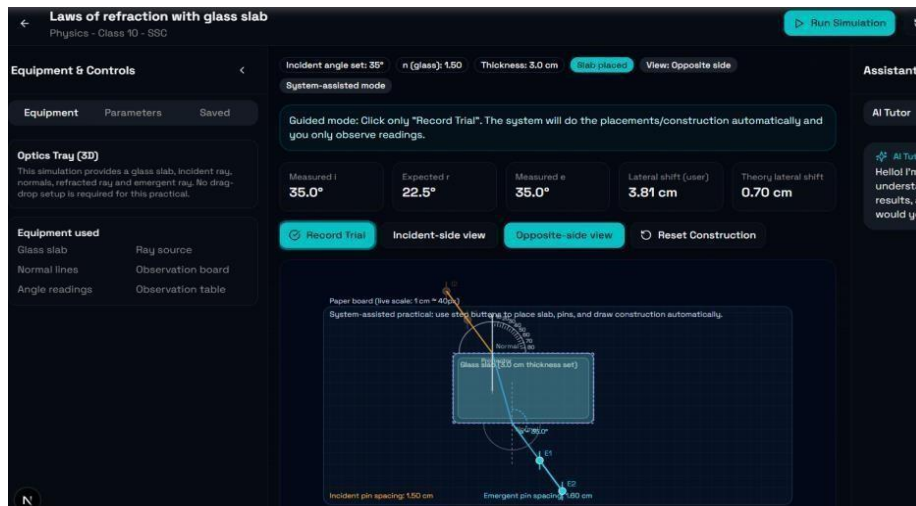


Figure 22 Showing Law of Refraction Simulation

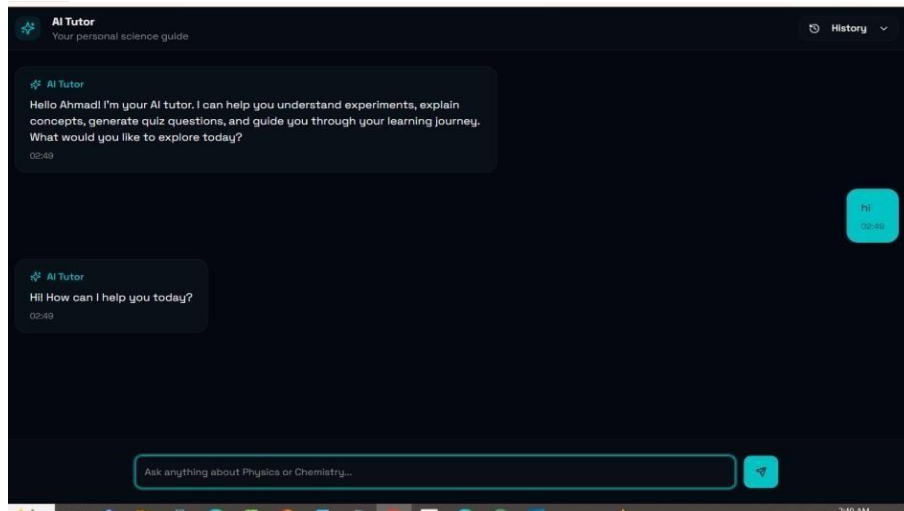


Figure 23 AI Tutor Section

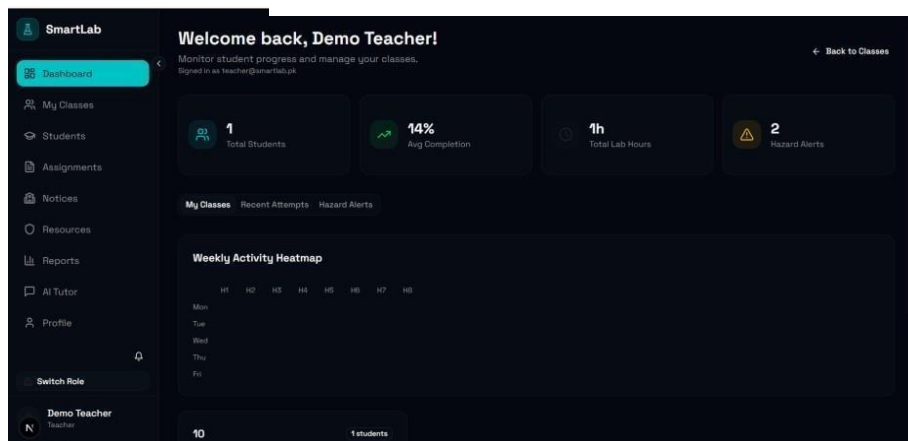


Figure 24 Teacher Dashboard

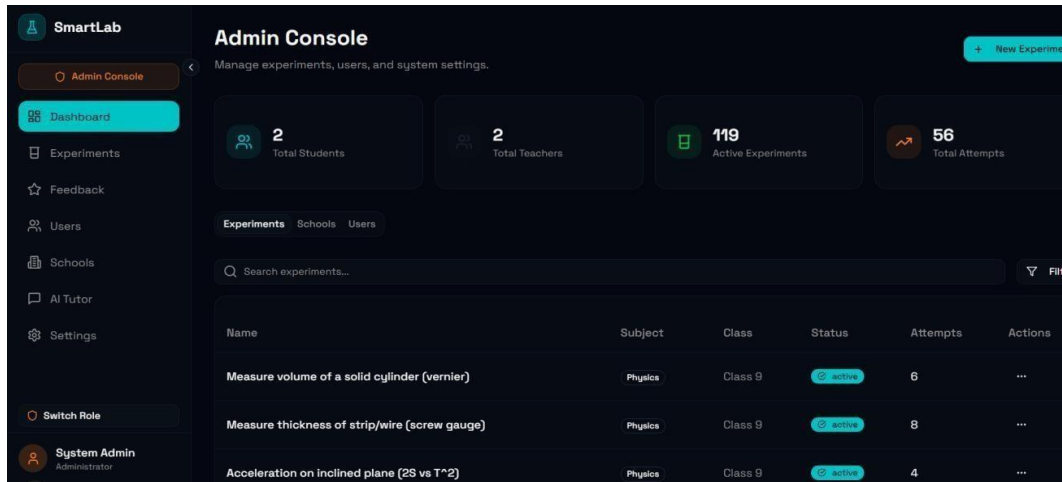


Figure 27 Admin Dashboard

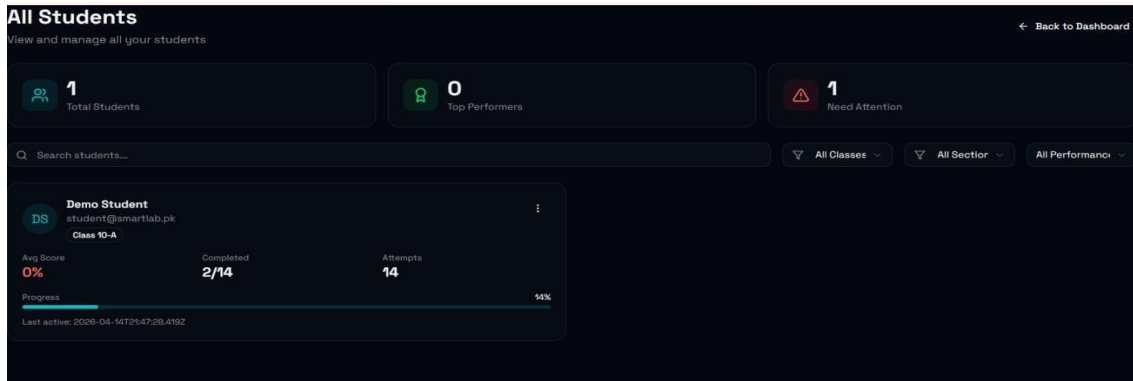


Figure 26 Teachers Students Overview

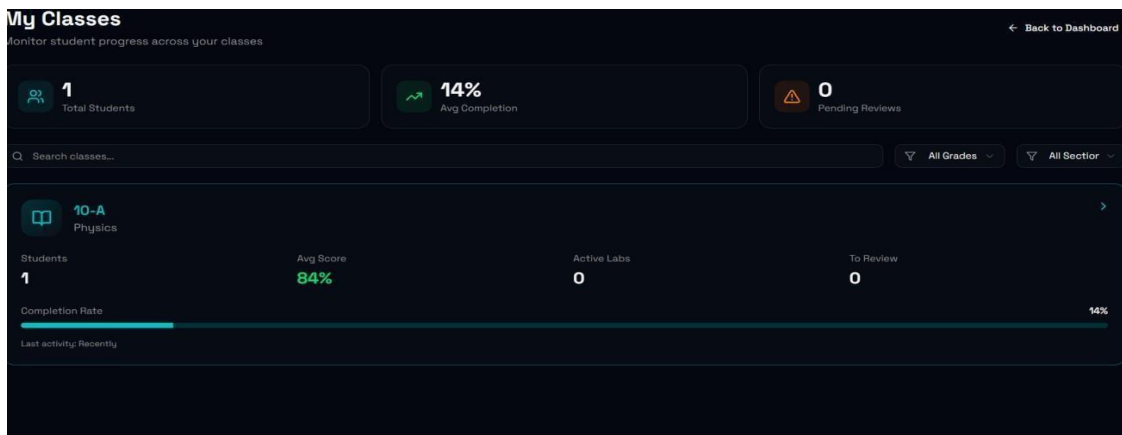


Figure 25 Teachers Classes Overview

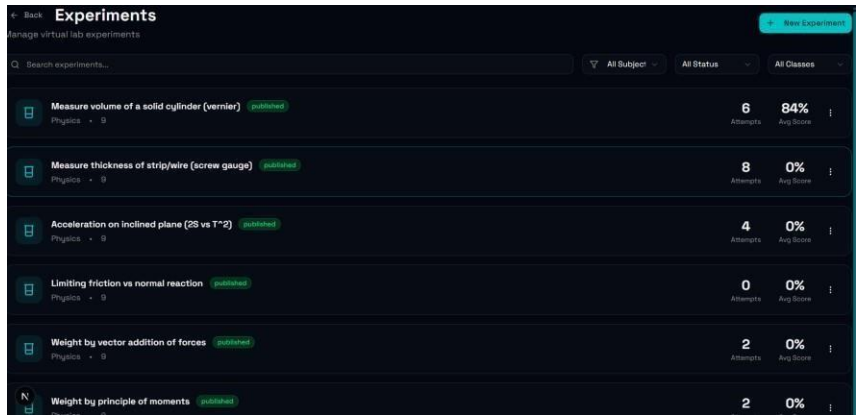


Figure 28 Total Experiments Overview

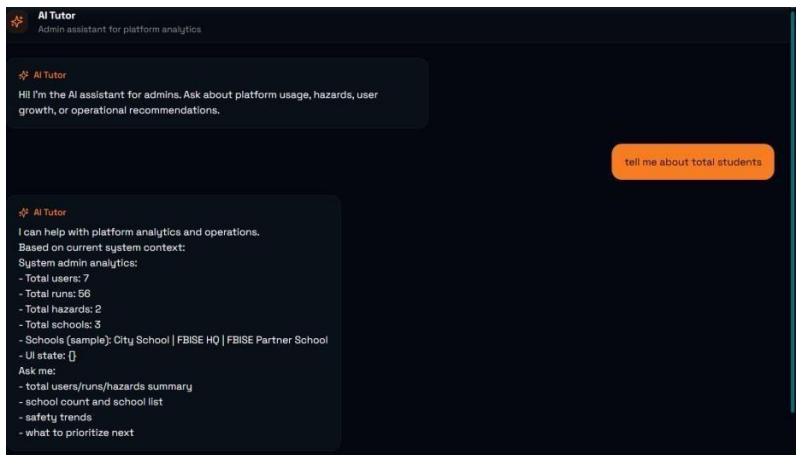


Figure 29 Teachers AI Tutor

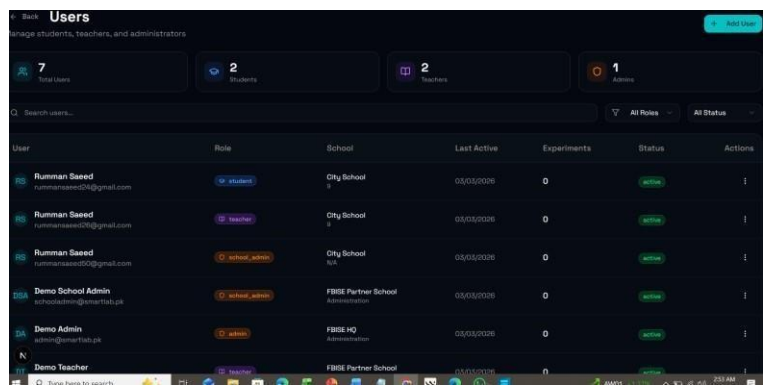


Figure 30 Total Users Registered

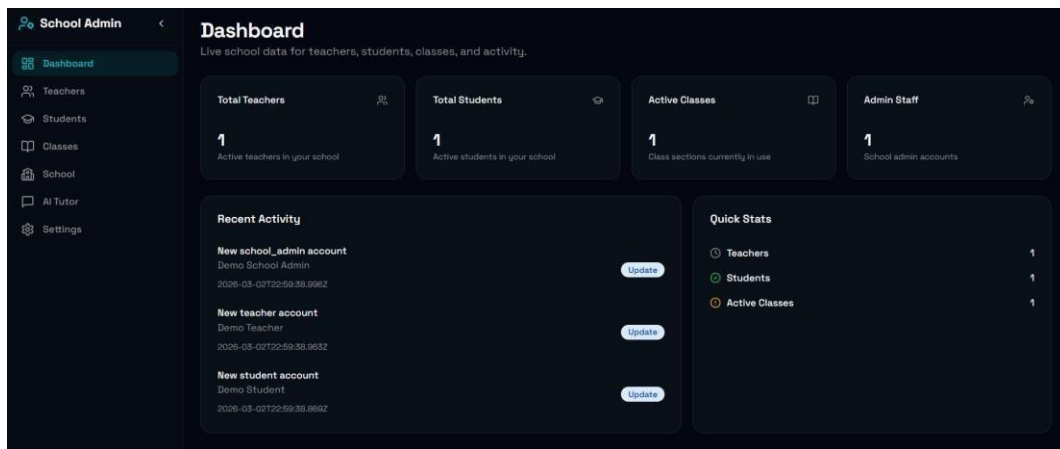


Figure 31 School Admin Dashboard

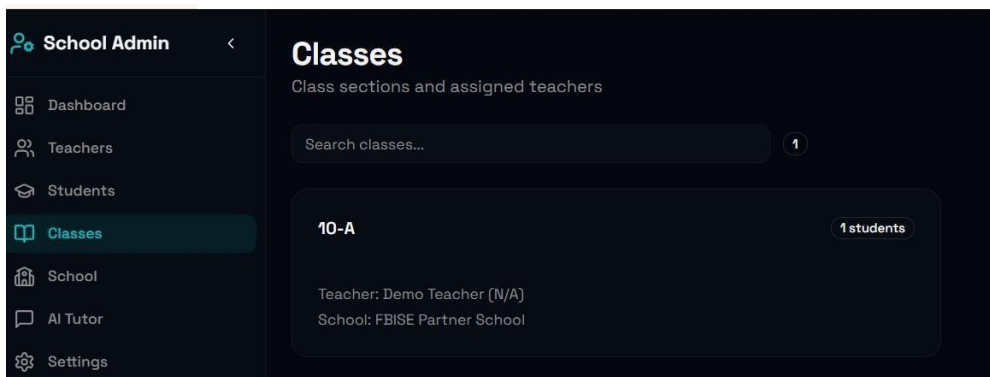


Figure 32 School Classes Overview

4.9 Chapter Conclusion

The overall design of SmartLab, an AI-Powered Virtual Lab for Physics and Chemistry Experiments, including architectural design, design strategy, and defining major system components, has been presented in this chapter as an overview of the system design. The modular and layered architecture is represented by the user interface developed with Next.js & React, the backend API, the PostgreSQL database using the Prisma ORM, and a machine learning prediction engine that supports intelligent analysis of experiments.

The logical design of the system has been discussed through architectural planning, as well as module-level organization, showing how the experiment simulation engine, AI prediction component, authentication system, and reporting module will provide the user with a complete virtual laboratory experience by showing the interaction of each

component within the overall system. System workflows, such as experiment selection, simulation execution, parameter processing, and automated prediction feedback, were developed to simulate real-world laboratory experiences within a digital environment.

Component-Level organization established a clear separation between the Presentation Layer, Application Logic Layer, and the Database Layer which allows for scalability and maintainability as enhancements are made. The addition of machine learning has significantly improved the system by allowing for prediction and providing users with intelligent support when executing experiments.

The design approach considered the FBISE curriculum for Physics and Chemistry thereby enabling students to conduct syllabus based experiments.

Chapter 05

System Implementation

5.1 Introduction

This section discusses how SmartLab has been created, with information about what tools and technologies were used, the Development Process followed and how a number of Primary Workflows were implemented. SmartLab is an interactive online web-based platform designed to be a Virtual Laboratory (Supporting Interactive Simulation) and a Role Based Learning Management System (RB LMS) for SSC and HSSC (Classes 9-12).

Each use has a Role defined within the system, as follows: Student, Teacher, School Admin and

System Admin. Students can view Assignments, Notices, Resources and Virtual Laboratory

Practical Exercises. Teachers can Create / Maintain Academic Content and Monitor Student

Progress. School Admin can assign/manage users and organize schools and School System

Administrators will monitor the systems and operations across the total operation. Additionally an "AI Tutor" or "AI Help" will be provided to assist students while completing assigned material from the Virtual Laboratory.

5.2 Tools and Technologies

When developing and implementing this system, the following technologies/tools were utilized:

Frontend (Development of Client Facing Web Application)

- Next.js (using React) alongside TypeScript was utilized to develop all dashboards, web pages, and UI components within the application

- Tailwind CSS was used for modern responsive styling and layout of all components
- Radix UI/shadcn-style components will be used (for example: tables, tabs, dialogs) as reusable UI elements within the application
- Development of Simulations (Virtual Laboratory)
- Three.js was used for rendering 3D Graphics and Simulation visuals
- @react-three/fiber was used to integrate 3D scenes into React components
- @react-three/drei was used for various 3D utilities (controls, helpers, lighting presets)

Backend (API Layer)

- Next.js's API Routes were utilized to implement backend Endpoints (including but not limited to authentication, assignments, notices, resources, runs, etc.)
- An independent FastAPI (written in Python) microservice was deployed to act as a listing service for datasets/experiments (optional and extendable in nature).

Database

- The main database utilized in this deployment is PostgreSQL
- Database schema definitions were performed using the Prisma ORM (Prisma also handles database migrations and provides database query functionality)
- Database migrations and a seed script were performed to generate the demo data.

Authentication & Authorization

- All routes were secured using role-based authorization and middleware verification and restricted as necessary

- All credential types were stored in a secure manner (i.e., passwords were stored using password hashing and thus stored in the database as a passwordHash)
- All APIs and dashboards will use Role-Based Access Control (based on roles: Student, Teacher, Admin, and School Admin).

The current and/or planned deployments for Next.js deployment will utilize Vercel hosting product as the hosting environment for the Next.js application

- Users will be able to access the PostgreSQL database as necessary.

5.3 Development Process

The Plan for Building the Platform Executed According to a Process is Excellent, as it Provides a Comprehensive Overview of the Development Process to Improve Maintenance.

Meeting the Needs of All Parties Is Essential.

- **Requirements Collection:**

Functional and non-functional requirements were defined for each user group (teacher, admin, student). The major objective was to develop a platform that supports students in completing "hands-on" solving of problems for the SSC/HSSC curriculum, while providing the necessary workflows for both teachers and administrators.

- **Design Transition to Modules:**

The system was divided into modules (Authentication, Dashboards, Assignments, Notices, Resources, Virtual Lab Work Space, Reports/Tracking) and database models were defined for each of the modules to support the associated software development. The modules were created using the Prisma schema to define and support the various modules.

- **Front-End Development:**

UI Pages (web pages) were created using Next.js App Router, following their structure of using reusable components to contain form elements, cards, navigation, panels, and data/views. Each user group has a separate dashboard and web pages that relate to them.

- **Simulation Development:**

Practical simulations were developed using React + Three.js to create the simulation workspace, which would contain the workspace for completing the simulation along with the necessary procedures and informational panels to enable students to learn how to complete the simulation work.

- **Back-End/API Development:**

API endpoints were created using Next.js API Routes for each defined user action to create the corresponding simulated student or task. Each endpoint interacts with Prisma for querying the database to validate the user performing the operation, retrieve the corresponding assignment, post the notice, and save the simulation run with the database.

- **Database Integration:**

The PostgreSQL database schema was created using Prisma. Migrations were run to create versioning of the database. A seed script was run for initializing the contents of the tables.

- **Testing and Integration:**

APIs were tested with manual calls and UI-based testing. Integration was verified by checking that UI actions (create assignment, filter content, start lab practical) correctly interact with the backend and database.

- **Deployment Setup**

The deployment workflow is being set up for Vercel. Environment variables, database connection, and Prisma migration deployment are prepared for production hosting.

5.4 Implementation Details of Key Features

Table 20 implementation details of key features

Table 20 presents the core functional features of the SmartLab system, including workflows, data structures, algorithms, and their significance. The table highlights how different modules support secure, interactive, and efficient virtual learning and experiment management.

Feature	Workflow	Data Structures	Algorithm/Logic	Importance
Role-Based Authentication	User logs in → credentials validated → role identified → redirected to correct dashboard	User model with email, passwordHash, role	Password hash verification + role-based redirect logic	Ensures secure and correct access for each user role
Student Dashboard	Student logs in → views assignments/notices/resources → opens lab practical	Assignment, Notice, Resource + links to Class	Fetch + filter APIs; UI renders lists and status	Provides a structured learning environment for students
Teacher Content Management	Teacher creates assignments/notices/resources → stored in DB → visible to students	Assignment, Notice, Resource models	CRUD operations, validation, role checks	Helps teachers manage class learning efficiently
Subject-wise Filtering	Student selects subject → API returns relevant items only	TeacherSubject enum; subject field on content	Query parameter filtering on API endpoints	Keeps content organized and reduces

				confusion for students
Virtual Lab Simulations	Student opens practical → simulation loads → equipment panel + steps guide learning	Practical config + simulation component state	Parameter updates + interactive rendering via R3F	Makes practical learning accessible without physical labs
Experiment Run Tracking	Student performs practical → run steps/measurements saved → report view for teacher	ExperimentRun, RunStep, MeasurementPoint, HazardEvent	Create/update run records and link related entities	Enables monitoring, evaluation, and performance improvement
Notifications (Academic Events)	Content created/graded → notification stored → user can view updates	Notification model with type, meta, readAt	Create notification records on key actions	Improves communication and awareness for users

5.5 Implementation Results

The following were achieved post-implementation:

- All students have access to all of their assignments (along with other notices or resources) via their student portal (including filtered results by subject if required);
- Teachers may use role pages and APIs to manage academic materials through their accounts; therefore, all materials/ academic content are securely saved within a database;

- The lab has virtual laboratory practicals that may be accessed via its laboratory workspace, along with simulations which will occur interactively via 3D-type tools;
- The structure of the system's database has been organized by Prisma and migrations; in addition, seed data has allowed for a rapid demonstration setup and testing process to be completed;
- The system contains role-based access control, in order for certain pages to be prevented from being accessed by users without that role (unauthorized);

The following items were confirmed after screenshots provided/gathered, as well as successful completion of testing:

- Each dashboard opens successfully after a user logs in (e.g., role-based);
- The filtering of each type of information (assignments, notices & resources) functions as intended;
- Loading of each simulation page works, plus remains stable while interacting;
- Upon creating new content within the database, the records associated with it have been updated (as expected).

5.6 Chapter Conclusion

This chapter described the implementation of SmartLab, including the technologies used and the development workflow followed. The system successfully provides a web-based platform that supports SSC/HSSC practical learning through virtual simulations while also offering role-based academic management features for teachers and administrators.

By combining Next.js, Prisma, and PostgreSQL with Three.js-based simulations, SmartLab delivers an accessible and organized learning environment. The implementation aligns with project objectives by improving access to practical learning, supporting structured teacher workflows, and enabling scalable management for

different roles. The deployment process is being finalized so that the platform can be hosted online and accessed by users without local setup.

Chapter 06

System Testing & Evaluation

6.1 Introduction

The purpose of this chapter is to show how the SmartLab Virtual Laboratory and the Learning Management System were tested, validated and verified to ensure that they functioned properly for all working roles as well as all user workflows. These tests were run against the entire SmartLab, including the SmartLab (Next.js), the API layer (Next.js API routes), the PostgreSQL database operations (using the Prisma ORM), and the interactive 3D lab simulations. Both manual and automated testing methods were used to verify that the SmartLab system is reliable, secure, accurate, and easy to use by Students, Teachers, School Admins, and System Admins.

The following areas of focus of the project are:

- Authentication / Role-based access control
- assignment / notices / learning resource workflow
- Practical execution of student lab experiments / experiment run tracking
- Teacher reporting and monitoring functionality
- Database integrity, consistency, performance under typical load

6.2 Test Strategy

Test strategy included verification of both functional & non-functional requirements, to ensure correctness of key features associated with role-based login and protected dashboard access, CRUD operations for classes, notices, resources and assignments, student practical execution (running logging), student measuring and recording practical results, assignment submission and grading work flow, notifications and the correctness of data in the purpose-specific dashboards.

Non-Functional Testing Areas of Focus:

Usability: ease of navigation, clear feedback from the UI, proper error handling.

Performance: responsiveness of the page and response times with respect to API calls.

Reliability: ensuring database updates occur consistently. Security: ensuring protection of application routes, role based access restriction, blocking invalid access attempts.

Automated vs Manual Testing

Manual testing will be used for scenarios such as:

Interaction of user with 3D simulations (dragging/dropping equipment), experiment step completion

Identifying edge-cases in UI and user journey while using multiple roles in application.

Automated testing using Selenium/Playwright will be utilized for repetitive tasks (logging in, navigating through the application, submitting a completed form, creating/updating/deleting a resource).

Layered Test Approach

Testing will occur at the following levels:

Unit Testing Level – items being tested include (utility functions, form validation, API helper methods).

Component Testing Level – items being tested include (UI components and simulation components)

Integration Testing Level – items being tested include (UI, API and Database).

System Testing Level – items being tested include (end-to-end operations across role types).

6.3 Component Testing

Each module was tested individually to ensure correct behavior before integration.

Table 21 component testing

Table 21 presents the testing and validation results of major SmartLab system components. The table highlights the testing methods used and verifies the successful functionality, reliability, and stability of each module.

Component	Purpose	Testing Method	Outcome
Authentication Module	Login for student/teacher/admin/schooladmin, session handling	Manual + automation	Login/redirect works, invalid credentials rejected
Role Dashboards	Display role-specific features and data	Manual testing	Correct role pages accessible, others blocked
Assignments Module	Create, view, filter by subject, submit, grade	Manual + automation	CRUD works, filtering works, grading updates reflect
Notices & Resources	Publish and view notices/resources subject-wise	Manual testing	Data appears per class/subject correctly
Lab Simulation Workspace	Render simulations, step guidance, equipment panel	Manual testing	Sim loads correctly, controls respond, UI stable

Experiment Run Tracking	Save run steps, measurements, hazards, reports	Manual + DB verification	Run data stored and retrievable reliably
Teacher Reports	View student activity and run summaries	Manual testing	Correct runs display and are filterable
Database Layer (Prisma/Postgres)	Persist users, classes, runs, assignments, submissions	Manual + seed validation	Relations/integrity maintained, seed data created

6.4 Unit Testing

Unit tests emphasize testing small individual functions along with reusable code, such as:

- Password hashing and verification- The hashing format of passwords needs to be consistent and able to validate account login credentials correctly.
- Redirecting a user based on their assigned role- Users will be redirected to the correct login page in the event they try to log into a system without proper authorization.
- Filtering assignments based on a subject- When passed a subject in the API query string (i.e., '?subject=Physics'), the returned assignment records should match exactly
- Validating experiment run data- The run object should have the correct expected fields (i.e., status, duration, steps, measurements) and it must be able to reject any invalid payload being submitted to it.

- Validation on Forms- All forms (login data entry) must check for (1) required fields and (2) whether or not the data entered in said fields are valid (i.e., Email or Password).

6.5 Integrated Testing

Through integration testing, we validated that the multiple modules function properly together.

- The frontend, API, and database (Prisma/Postgres) successfully function together for the following example flow:
 - Teacher creates assignment in the database → student pulls the assignment → student submits the assignment for grading → teacher grades and saves the mark → student sees updated status.
- Lab Simulation, Run Logging, and Reporting function properly together for the following example flow:
 - Student completes a practical → system logs run steps and measurements → results of the run are stored in the database → teacher is able to report on the summary of the run.
- Access through Roles and Protected Access flow together properly since:
 - Students are prevented from accessing teacher/admin routes; teachers are routed away
 - from admin-only pages.

6.6 System Testing

Across the platform, system testing confirmed the end-to-end testing of all workflows.

Workflows Tested:

- User accesses the platform > selects login role > signs in > goes to the appropriate/current dashboard
- Teacher assigns/creates notices/resources > student sees them > performs action(s)
- Student enters lab practical > performs procedure > lab practical is saved to the student's data file > teacher sees the progress/report of the student
- Admin-level user manages users/classes in the system as well as oversees activity in the entire system In addition to testing workflows, other system testing included:
 - verification of invalid attempts to access an invalid route (security)
 - logout/session handling • concurrent access checks (multiple users retrieving the same dashboard simultaneously)

6.7 Test Cases

6.7.1 Authentication & Role-Based Access (All Roles)

Table 22 auth & role based test cases

Table 22 presents the authentication and role-based access test cases of the SmartLab system. The table describes the verification of different login interfaces, route validation, and access control mechanisms for students, teachers, administrators, and school administrators to ensure secure and role-specific authentication functionality.

UC ID	TC ID	PreCondition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-AUTH	1.1	Server running	Open /login/student	Student login page loads	Student login form rendered with Email/Password fields and "Login" button	Pass

UC-AUTH	1.2	Server running	Open /login/teacher	Teacher login page loads	Teacher login UI displayed; role specific title shown	Pass
UC-AUTH	1.3	Server running	Open /login/admin	Admin login page loads	Admin login screen displayed successfully	Pass
UC-AUTH	1.4	Server running	Open /login/school-admin	School admin login page loads	School admin login page displayed successfully	Pass
UC-AUTH	1.5	None	Open invalid route /login/abc	Show 404 / not found	“404 This page could not be found” displayed	Pass

Table 23 Authentication Validation and Access Control Test Cases

Table 23 presents the authentication validation and access control test cases of the SmartLab system. The table verifies successful role-based login, invalid credential handling, form validation, and unauthorized access protection.

UC ID	TC ID	PreCondition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-AUTH	2.1	Demo student exists	Login with valid student credentials	Redirect to student dashboard	Redirected to /student/dashboard; dashboard widgets visible	Pass
UC-AUTH	2.2	Demo teacher exists	Login with valid teacher credentials	Redirect to teacher dashboard	Redirected to /teacher/dashboard; teacher sidebar visible	Pass
UC-AUTH	2.3	Demo admin exists	Login with valid admin credentials	Redirect to admin dashboard	Redirected to /admin/dashboard; admin navigation visible	Pass

UC-AUTH	2.4	Demo school admin exists	Login with valid school admin credentials	Redirect to school admin dashboard	Redirected to /schooladmin/dashboard; school admin panels visible	Pass
UC-AUTH	3.1	None	Enter wrong password and submit	Error shown, no login	Error message displayed (invalid credentials); route remains on login page	Pass
UC-AUTH	3.2	None	Submit empty email/password	Validation shown	Form shows required field validation; login not submitted	Pass
UC-AUTH	4.1	Not logged in	Open /student/dashboard directly	Redirect to /login/student	Redirect performed; student login page displayed	Pass

Table 24 Authorization, Session Management, and Logout Test Cases

Table 24 presents the authorization and session management test cases of the SmartLab system. The table verifies protected route access, role-based restrictions, logout functionality, and session security behavior for authenticated and unauthenticated users.

UC ID	TC ID	PreCondition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-AUTH	4.2	Not logged in	Open /teacher/dashboard directly	Redirect to /login/teacher	Redirect performed; teacher login displayed	Pass

UC-AUTH	4.3	Logged in as student	Open /teacher/dashboard	Access denied/redirect	Redirected to /login/teacher or blocked due to role mismatch	Pass
UC-AUTH	4.4	Logged in as teacher	Open /admin/dashboard	Access denied/redirect	Redirected to /login/admin /unauthorized prevented	Pass
UC-AUTH	5.1	Logged in	Click Logout	Session ends	Token/session cleared; redirected to login/home	Pass
UC-AUTH	5.2	Logged out	Use browser back to open dashboard	Still blocked	Dashboard not accessible; redirected back to login	Pass

6.7.2 Student Module (Dashboard, Assignments, Notices, Resources, Profile)

Table 25 Student Dashboard and Assignment Module Test Cases

Table 25 presents the student dashboard and assignment management test cases of the SmartLab system. The table verifies successful dashboard loading, student navigation, and assignment viewing functionality within the student portal.

UC ID	TC ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-STU	1.1	Student logged in	Open /student/dashboard	Dashboard loads	Dashboard page renders with cards/sections and student navigation	Pass
UC-STU	1.2	Student logged in	Open Assignments page	List loads	Assignments list displayed; due dates visible	Pass
UC-STU	1.3	Assignments list visible	Select subject filter = Physics	Only Physics items shown	List updates and shows only Physics-tagged assignments	Pass
UC-STU	1.4	Assignments list visible	Select subject filter = Chemistry	Only Chemistry items shown	List updates and shows only Chemistry-tagged assignments	Pass
UC-STU	1.5	No assignments for subject	Select a subject with zero items	Empty state shown	“No assignments found” (or similar) message displayed	Pass

UC-STU	2.1	Student logged in	Open Notices page	Notices list loads	Notices displayed with date/time and title	Pass
UC-STU	2.2	Notices visible	Filter notices by subject	Filter applied	Only selected subject notices remain visible	Pass
UC-STU	3.1	Student logged in	Open Resources page	Resources list loads	Resource cards show Title + URL link	Pass
UC-STU	3.2	Resources visible	Filter resources by subject	Filter applied	List refreshes with correct subject resources	Pass
UC-STU	3.3	Resource link present	Click open resource link	Opens resource	Browser opens resource in new tab / navigation works	Pass
UC-STU	4.1	Student logged in	Open Profile page	Profile loads	Profile details visible (name/email/class/school)	Pass

6.7.3 Teacher Module (Classes, Students, Assignments, Grading, Notices, Resources, Reports)

Table 26 Teacher Dashboard, Content Management, and Reporting Test Cases

Table 26 presents the teacher module test cases of the SmartLab system. The table verifies dashboard access, assignment and content management, grading functionality, report generation, filtering features, and role-based access restrictions for teachers.

UC ID	TC ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-TEA	1.1	Teacher logged in	Open /teacher/dashboard	Dashboard loads	Teacher dashboard renders with summary widgets	Pass
UC-TEA	1.2	Teacher logged in	Open Classes page	Classes shown	Assigned class sections displayed	Pass
UC-TEA	1.3	Teacher logged in	Open Students page	Students shown	Student list loads with names + class info	Pass
UC-TEA	2.1	Teacher logged in	Create assignment (experiment)	Created in DB	Success toast shown; assignment appears in list	Pass
UC-TEA	2.2	Teacher creating assignment	Leave required field empty	Validation error	Error shown; assignment not created	Pass
UC-TEA	2.3	Teacher logged in	Create assignment with subject=Physics	Saved correctly	Assignment created; subject badge	Pass

					shows Physics	
UC-TEA	2.4	Teacher logged in	Open Teacher assignments list	All shown	Assignments appear ordered by due date	Pass
UC-TEA	3.1	Teacher logged in	Create notice	Created and visible to students	Notice created; appears in teacher notices + student notices	Pass
UC-TEA	3.2	Teacher logged in	Create resource	Created and visible to students	Resource created; appears in student resources list	Pass
UC-TEA	4.1	Student submission exists	Open grading view (submission)	Submission details load	Student answers/run linked data displayed	Pass
UC-TEA	4.2	In grading view	Enter score + feedback and save	Grading saved	Success message; status changes to graded; score visible to student	Pass
UC-TEA	5.1	Teacher logged in	Open Reports page	Reports load	Student run summaries visible (practicalId, status, score)	Pass
UC-TEA	5.2	Reports visible	Filter report (class/student/practical if present)	Filtered results	Table updates to matching runs	Pass

UC-TEA	5.3	Teacher logged in	Attempt to access admin-only route	Blocked	Redirect to admin login / access denied	Pass
--------	-----	-------------------	------------------------------------	---------	---	------

6.7.4 School Admin Module (Teacher Management, School Overview)

Table 27 School Administrator Management and Access Control Test Cases

Table 27 presents the school administrator module test cases of the SmartLab system. The table verifies teacher management operations, data validation, dashboard accessibility, and role-based access restrictions for school administrators.

UC ID	TC ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-SADM	1.1	School admin logged in	Open /schooladmin/dashboard	Dashboard loads	Dashboard renders with school admin navigation	Pass
UC-SADM	1.2	School admin logged in	Open Teachers management page	Teacher list visible	Teachers table loads with names/emails/subjects	Pass
UC-SADM	1.3	School admin logged in	Add a teacher (valid data)	Teacher created	Success message; teacher appears in list	Pass

UC-SADM	1.4	Adding teacher	Enter duplicate email	Error shown	“Email already exists” (or similar) shown; not created	Pass
UC-SADM	1.5	School administrator logged in	Update subject teacher	Updated in DB	Teacher subject updated; reflected in UI	Pass
UC-SADM	1.6	School administrator logged in	Try admin-only operations	Blocked	Access denied / redirect	Pass

6.7.5 System Admin Module (System-Level Administration)

Table 28 Administrator Dashboard and User Management Test Cases

Table 27 presents the administrator module test cases of the SmartLab system. The table verifies dashboard functionality, user management operations, input validation, and administrative access control features.

UC ID	TC ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-ADM	1.1	Admin logged in	Open /admin/dashboard	Dashboard loads	Admin dashboard UI displayed	Pass

UC-ADM	1.2	Admin logged in	Open Users page (if present)	Users list loads	Users displayed with role labels	Pass
UC-ADM	1.3	Admin logged in	Create new user (teacher/student)	User created	Success toast; user appears in list	Pass
UC-ADM	1.4	Create user	Invalid email format	Validation error	Inline validation prevents submit	Pass
UC-ADM	1.5	Admin logged in	Delete/disable user (if feature exists)	User access revoked	Deleted/disabled status reflected; login denied	Pass

6.7.6 Virtual Lab Practical Execution (Simulations + Equipment + Measurements)

Table 29 Virtual Lab Simulation and Experiment Execution Test Cases

Table 29 presents the virtual lab simulation and experiment execution test cases of the SmartLab system. The table verifies practical access, simulation interaction, equipment handling, step navigation, experiment tracking, and error-handling functionality within the virtual laboratory environment.

UC ID	TC ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-LAB	1.1	Student logged in	Open Lab list /student/lab	Practical list shows	Practical cards display (title, subject)	Pass

UC-LAB	1.2	Lab list visible	Open a practical (e.g., Ohm's Law)	Workspace loads	Lab workspace shows simulation canvas + procedure/theory panels	Pass
UC-LAB	1.3	Workspace loaded	Adjust simulation parameter (e.g., voltage slider)	Readings update	Current/voltage readings change and UI badges update	Pass
UC-LAB	1.4	Workspace loaded	Place/select equipment from equipment panel	Equipment appears/activates	Equipment state changes; UI reflects placement	Pass
UC-LAB	1.5	Workspace loaded	Go next/previous step	Step changes	Procedure step highlight changes accordingly	Pass
UC-LAB	1.6	Student runs practical	Save/end run (if available)	Run saved	Run entry stored; appears in reports/history	Pass
UC-LAB	1.7	Network/API issue simulated	Start run while API unavailable	Proper error handling	Error toast shown; app does not crash	Pass

6.7.7 API & Database Integrity (Backend Validation)

Table 30 Database Operations and Data Integrity Test Cases

Table 30 presents the database operation and data integrity test cases of the SmartLab system. The table verifies database queries, record insertion and updates, filtering

operations, transaction handling, and relational integrity within the PostgreSQL and Prisma-based data layer.

UC ID	T C ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC - DB	1.1	DB connected	Login triggers DB query	Correct user fetched	Correct role user retrieved; session created	Pass
UC - DB	1.2	Teacher creates assignment	DB insert occurs	Record inserted	Assignment stored with correct classId, createdById, dueDate	Pass
UC - DB	1.3	Student loads assignments	DB select with filter	Correct filtered results	Only matching subject items returned by API	Pass
UC - DB	1.4	Teacher grades submission	DB update occurs	Grade saved	Submission updated with score, feedback, gradedAt fields	Pass
UC - DB	1.5	Experiment run saved	DB transaction	Run saved with relations	ExperimentRun created steps/measurements linked with	Pass

UC - DB	1.6	Delete class/user (if exists)	Cascade behavior	No orphan records	Prisma cascade/set-null rules behave correctly	Pass
---------------	-----	-------------------------------	------------------	-------------------	--	------

6.7.8 Non-Functional Test Cases (Performance, Usability, Reliability)

Table 31 Non-Functional Testing and System Performance Test Cases

Table 31 presents the non-functional testing and performance-related test cases of the SmartLab system. The table verifies system responsiveness, stability, reliability, error handling, and usability under different operational conditions. Table 30 presents the non-functional testing and performance-related test cases of the SmartLab system. The table verifies system responsiveness, stability, reliability, error handling, and usability under different operational conditions.

UC ID	TC ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-NF	1.1	Normal load	Open dashboard pages	Fast load	Pages load within acceptable time; no layout shift	Pass
UC ID	TC ID	Pre-Condition	Actions/Steps	Expected Output	Actual Output	Pass/Fail Criteria
UC-NF	1.2	Multiple users scenario	Run multiple login attempts	System stable	No crashes; correct sessions per user	Pass

UC-NF	1.3	Simulation heavy page	Open practical on low-end device	Responsive UI	3D canvas loads; controls usable; acceptable FPS	Pass
UC-NF	1.4	Invalid inputs	Submit forms with invalid data	Error messages	Clear error shown; data not stored	Pass
UC-NF	1.5	Reliability	Refresh during session	Session maintained	User stays logged in (or redirected properly if expired)	Pass

Chapter 07

Conclusion

This chapter concludes the SmartLab project by summarizing what was achieved, how the work relates to the project objectives, what we learned during development, and what improvements can be made in the future. SmartLab was designed to support **SSC/HSSC (Classes 9–12)** science practical learning by providing a **virtual laboratory** along with a **role-based academic management system**. The system aims to reduce the limitations of physical labs and make practical learning more accessible, repeatable, and organized.

7.1 Contributions

The objectives stated in Chapter 1 can be directly related to the contributions made by this project:

1) **Virtual laboratory for practical instruction**

A major contribution of SmartLab is the development of an interactive virtual lab environment where students can perform practicals through simulations. This helps kids to practice experiments in a safe and repeatable manner even when physical labs are not accessible.

2) **Platform for role-based learning management**

SmartLab offers distinct modules for Students, Teachers, School Admins, and System Administrators, therefore guaranteeing that every user only views features pertinent to them. This donation helps toward the goal of creating a regulated and safe platform with practical school use.

3) **Academic workflow aid including resources, assignments, notifications.**

By allowing teachers to share assignments, communications, and learning materials and allowing pupils to get them from their dashboards, the system helps classroom

procedures. This helps us to achieve the goal of organizing learning and minimizing dependence on dispersed communication solutions.

4) Data-driven structure with appropriate backend and database

SmartLab has a suitable backend and database design so that user records, courses, assignments, and learning activity can be kept reliably. This helps one reach the goal of creating a reliable, expandable system wherein records are kept appropriately.

5) Restricted supportive direction using AI

Including artificial intelligence tutor/help capabilities helps kids with queries is a little yet helpful contribution. This fits the goal of raising learning support while preserving the general teacherdriven character of the learning process.

7.2 Reflections

Strengths

- Repeatability and accessibility enable pupils to conduct experiments many times, hence enhancing understanding and confidence.
- Better learning structure: Combining simulations with assignments/notices/resources helps to keep learning rather than erratic.

Role-based design: Clear demarcation of student, teacher, and administrative functions improves utility and safety.

- Strong database base: The structured base for upcoming capabilities like sophisticated reporting and analytics is the Prisma/PostgreSQL schema.
- Scalability prospect: More practicals, more topics, and more monitoring capabilities can be added to the system without resetting the whole platform.

Weaknesses and defects

- Though simulations are engaging, certain practicums can be enhanced to better reflect actual laboratory conditions (more precise constraints, physics behavior, and measurement uncertainty).
- Coverage of testing: While manual testing validates accuracy, a more thorough automated end-to-end test suite would improve dependability and cut back on regression defects.

For actual public usage, more effort is needed on deployment readiness, secret management, logging, and monitoring as well as production hardening.

- Depth of assessment: Though the system can monitor actions, better automatic assessment— rubrics, scoring models, and teacher analytics—still needs development.

Influence on society and problem area

By increasing access to hands-on learning, SmartLab helps the field of education. It enables students to practice safely, lessens reliance on physical labs, and helps instructors manage practical information in a more systematic fashion. This is especially helpful in situations when resources are scarce or kids have insufficient laboratory exposure before tests.

7.3 Future Work

SmartLab might be expanded in many significant ways:

1) Enhance realism and accuracy in simulation.

Include measurement tolerance, instrument calibration, and error margins among your realistic lab behaviors.

- Enhance the design of equipment positioning and procedural limitations to produce simulations more closely related to actual lab practice.

2) Broaden reporting and automation.

- Provide school administrators and instructors detailed analytics dashboards (progress trends, common mistakes, time spent per practical).
- Incorporate more ordered evaluation, such rubric-based grading for practical performance.

3) Improve quality assurance and testing.

- For logins, dashboards, CRUD processes, and lab run saving, use full end-to-end automated testing (e.g., Playwright/ Selenium).
- Integrate performance testing for simulations and fine-tune for low-end devices.

4) Implementation and Operational Enhancement

- On a managed database and safe environment setting, completely deploy the system on a production platform.
- For greater operational stability, integrate monitoring, error tracking, rate limiting, and audit logs.

5) Increase content breadth.

- For both Physics and Chemistry, include more SSC/HSSC practicals and match them more closely to board-wise expectations.
- Teacher resources should be added to tailor practical actions, notes, and workbooks.

6) AI tutor augmentation (regulated and secure)

Improve artificial intelligence tutor using information from the present practical, student development, and teacher-established guidelines.

- Through establishing limits, documenting, and teacher supervision, guarantee safe use.

REFERENCES

- [1] Vercel, Inc., "Next.js Documentation." [Online]. Available: <https://nextjs.org/>.
- [2] Meta Platforms, Inc.'s "React Documentation" is available online: <https://react.dev/>.
- [3] Contributors of Three.js, "Documentation for Three.js." [Online]. Available: <https://threejs.org/>.
- [4] Node.js Contributors, "Node.js Documentation." [Online]. Accessible: <https://nodejs.org/en>.
- [5] Tailwind Labs Inc., "Tailwind CSS Documentation." [Online]. Available: <https://tailwindcss.com/docs>.
- [6] Lucide Contributor, "Lucide Icons Documentation." [Online]. Available: <https://lucide.dev/>.
- [7] Radix UI Contributors, "Radix UI Documentation." [Online]. Available: <https://www.radixui.com/docs/primitives/overview/introduction>.
- [8] Framer, Inc., "Framer Motion Documentation," available online at <https://www.framer.com/motion/>.
- [9] M. A. Alonso, R. G. C. L. Ferreira, and M. M. Cruz, "Virtual and remote laboratories in education: "Systematic review," Computers & Education, vol. 147, pp. 103778.

APPENDIX A

TURNITIN AI DETECTION REPORT

Adeel Syed

FYP REPORT - Roman

Document Details

Submission ID

trn:oid::3618:135410730

Submission Date

Apr 16, 2026, 8:12 AM GMT+0

Download Date

Apr 16, 2026, 8:16 AM GMT+0

File Name

FYP REPORT - Roman.pdf

File Size

3.2 MB

112 Pages

15,182 Words

100,476 Characters



Page 1 of 114 - Cover Page

Submission ID trn:oid::3618:135410730

TURNITIN AI DETECTION REPORT



22% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

-  **60 AI-generated only 22%**
Likely AI-generated text from a large-language model.
-  **0 AI-generated text that was AI-paraphrased 0%**
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI-generated as AI-generated and AI-paraphrased or likely AI-generated and AI-paraphrased writing as only AI-generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.



What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.





TURNITIN PLAGIRISM REPORT






11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **108 Not Cited or Quoted 10%**
Matches with neither in-text citation nor quotation marks
-  **4 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 5%  Internet sources
- 2%  Publications
- 10%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **108 Not Cited or Quoted 10%**
Matches with neither in-text citation nor quotation marks
- **4 Missing Quotations 0%**
Matches that are still very similar to source material
- **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 5% Internet sources
- 2% Publications
- 10% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.bahria.edu.pk	1%
2	Internet	prh.hec.gov.pk	<1%
3	Student papers	Higher Education Commission Pakistan on 2025-05-14	<1%
4	Internet	www.coursehero.com	<1%
5	Student papers	University of Greenwich on 2023-04-18	<1%
6	Student papers	Higher Education Commission Pakistan on 2022-11-19	<1%
7	Student papers	Islington College, Nepal on 2025-12-29	<1%
8	Student papers	University of Wales Institute, Cardiff on 2025-08-28	<1%
9	Student papers	Vilniaus kolegija on 2025-01-09	<1%
10	Internet	gcris.leu.edu.tr	<1%

11	Student papers	RMIT University on 2023-05-27	<1%
12	Student papers	Arab Open University on 2024-11-07	<1%
13	Student papers	Asia Pacific University College of Technology and Innovation (UCTI) on 2026-01-23	<1%
14	Student papers	University of Westminster on 2017-11-17	<1%
15	Student papers	Vilniaus kolegija on 2026-01-12	<1%
16	Student papers	The Hong Kong Polytechnic University on 2012-04-03	<1%
17	Student papers	Higher Education Commission Pakistan on 2018-01-05	<1%
18	Student papers	Higher Education Commission Pakistan on 2025-07-15	<1%
19	Student papers	University of Ulster on 2015-04-27	<1%
20	Student papers	University of Westminster on 2025-12-10	<1%
21	Student papers	Birzeit University Main Library on 2026-01-25	<1%
22	Student papers	University of Greenwich on 2017-04-10	<1%
23	Student papers	University of Northampton on 2024-07-21	<1%
24	Student papers	Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur on 2019-0...	<1%

APPENDIX B

The following are key API endpoints used in SmartLab:

- `/api/auth/login` — login user
- `/api/auth/logout` — logout user
- `/api/dashboard/student` — student statistics and progress
- `/api/dashboard/teacher` — teacher stats and class overview
- `/api/admin/experiments` — manage experiments
- `/api/admin/experiment-feedback` — view experiment ratings & feedback
- `/api/school-admin/students` — manage students
- `/api/school-admin/classes` — manage classes