



**TRIBHUVAN UNIVERSITY
FACULTY OF HUMANITIES AND SOCIAL SCIENCES
LALITPUR ENGINEERING COLLEGE**

LABXPLORERX: INTERACTIVE LEARNING ENVIRONMENT

**BY
SUSHANT BRAMHACHARYA (LEC077BCA08)**

**A FINAL PROJECT REPORT
SUBMITTED TO THE DEPARTMENT OF COMPUTER APPLICATION
IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELORS IN COMPUTER APPLICATION**

**DEPARTMENT OF COMPUTER APPLICATION
LALITPUR, NEPAL**

SEPTEMBER, 2024



Tribhuvan University
Faculty of Humanities and Social Sciences

LABXPLORERX: INTERACTIVE LEARNING ENVIRONMENT

Submitted to
Department of Computer Application
Lalitpur Engineering College

**In partial fulfillment of the requirement for the degree of Bachelors in Computer
Application**

Submitted by
Sushant Bramhacharya (LEC077BCA08)
SEPTEMBER, 2024

Under the Supervision of
Er. Bibat Thokar

DECLARATION

I declare that the work hereby submitted for Bachelors in Computer Application at the Department of Computer Application , Lalitpur Engineering College entitled "**LabXplorerX: Interactive Learning Environment**" is my own work and has not been previously submitted by me at any university for any academic award. I authorize the Department of Computer Application , Lalitpur Engineering College to lend this project work to other institutions or individuals for the purpose of scholarly research.

Sushant Bramhacharya (LEC077BCA08)

September, 2024



Tribhuvan University
Faculty of Humanities and Social Sciences
Lalitpur Engineering College

LETTER OF APPROVAL

This is to certify that this project prepared by Sushant Bramhacharya entitled "**LabXplorerX: Interactive Learning Environment**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

..... Er. Bibat Thokar Project Supervisor Department of Computer Application Lalitpur Engineering College Er. Bibat Thokar BCA Program Coordinator Department of Computer Application Lalitpur Engineering College
..... Internal Examiner Er. Praches Acharya Head of Department Department of Computer Engineering Lalitpur Engineering College External Examiner Aanand KC Ratna Rajyalaxmi Campus Pradarshani Marg, Kathmandu

DEPARTMENTAL ACCEPTANCE

The project work entitled “**LabXplorerX: Interactive Learning Environment**”, submitted by **Sushant Bramhacharya (LEC077BCA08)** in partial fulfillment of the requirement for the award of the degree of “**Bachelors of Computer Application**” has been accepted as a genuine record of work independently carried out by the student in the department.

Er. Bibat Thokar

BCA Coordinator

Department of Computer Application ,
Lalitpur Engineering College ,
Faculty of Humanities and Social Sciences ,
Tribhuvan University, Nepal.

September, 2024

ACKNOWLEDGMENT

This project work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First of all, I would like to express my sincere gratitude to my Project Supervisor and BCA Program Coordinator, **Er. Bibat Thokar**, of **Lalitpur Engineering College** for providing invaluable guidance, insightful comments, meticulous suggestions, and encouragement throughout the duration of this project work.

Furthermore, we would like to extend our gratitude to the entire faculty of the Department of Computer Application . Their dedication to fostering creativity, critical thinking, and technical proficiency has been useful in our project's development. The support and guidance received from our teachers have empowered us to transform our vision into a reality.

I am also grateful to my classmates and friends for offering me advice and moral support. To my family, thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams. I am especially grateful to my parents, who supported me emotionally, believed in me and wanted the best for me.

Sushant Bramhacharya (LEC077BCA08)

September, 2024

ABSTRACT

LabXplorerX is an innovative web application designed to revolutionize science education through an interactive and immersive learning experience. Built with modern technologies like React, Express, and Postgres, the platform offers a seamless, responsive interface that caters to students' needs. At its core are real-time simulations powered by Phaser.js and Unity 3D, enabling students to engage with scientific concepts in a visually dynamic and interactive way. LabXplorerX also features a vast library of science-related articles, quizzes, and assessments, encouraging both individual learning and evaluation. The platform introduces Capsules, which serve as modular learning units combining simulations, quizzes, comments, and curated resources, providing a comprehensive learning environment. Users can add their favorite capsules to a personalized list for easy access, promoting a customized learning experience. Through integrated collaboration tools, including real-time comments on simulations and capsules, LabXplorerX fosters a community of inquisitive learners. With a robust system architecture, supported by detailed feasibility studies, and a secure authentication module, the platform is designed for scalability, reliability, and security, making it an effective and engaging tool for modern science education.

Keywords: *Interactive, Collaboration, Simulation*

TABLE OF CONTENTS

DECLARATION.....	iii
LETTER OF APPROVAL.....	iv
DEPARTMENTAL ACCEPTANCE.....	v
ACKNOWLEDGMENT.....	vi
ABSTRACT	vii
TABLE OF CONTENTS	viii
LIST OF FIGURES	x
LIST OF TABLES	xii
LIST OF ABBREVIATIONS	xiii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Objectives	1
1.4 Scope	2
1.5 Limitation	2
1.6 Development methodology	2
1.7 Report Organisation	3
2 BACKGROUND AND LITERATURE REVIEW.....	4
2.1 Background Study	4
2.2 Literature Review.....	4
3 SYSTEM ANALYSIS AND DESIGN	8
3.1 System Analysis	8
3.1.1 Requirement Analysis	8
3.2 Feasibility Analysis	11
3.2.1 Economical Feasibility	11
3.2.2 Operational Feasibility	12
3.2.3 Technical Feasibility	12
3.3 Structured System Modelling	13

3.3.1	Process Modeling	14
3.3.2	Data Modelling(ER-Diagram)	15
3.4	Structured System Design	17
3.4.1	Architecture Design	17
3.4.2	Database Schema Design	18
3.4.3	Interface Design.....	19
3.4.4	Physical DFD	23
3.5	Algorithms	24
3.5.1	Algorithm of AtomSimulator	24
3.5.2	Algorithm of CodeEditor	24
3.5.3	Algorithm of OhmsLawSimulator	25
3.5.4	Algorithm of GravitySim	25
3.5.5	Algorithm for SolarSystemSimulator	26
4	IMPLEMENTATION.....	28
4.1	Tools Used	28
4.1.1	Implementation Details of Modules	31
4.1.2	Unit Testing Test Cases	34
4.1.3	Test Cases for System Testing.....	35
5	CONCLUSION AND ANALYSIS	37
5.1	Conclusion	37
5.1.1	Screenshots of Outcomes	39
5.2	Future Recommendations	52
APPENDIX A		
A.1	Project Schedule	53
A.2	Running the Project.....	53
A.3	Basic Scene in Phaser.js	55
A.4	Postgres Database Tables.....	56
A.5	Supervisor Consultation Form	58
REFERENCES		59

LIST OF FIGURES

Figure 1.1 Report Organisation	3
Figure 3.1 Use Case Diagram	10
Figure 3.2 Process Model: Logical DFD	14
Figure 3.3 ER Diagram of System Data	16
Figure 3.4 Three Tier Architecture of System	17
Figure 3.5 Schema Design	18
Figure 3.6 Home Screen UI Design	20
Figure 3.7 Capsule Category UI Design	20
Figure 3.8 Capsules Menu Design	21
Figure 3.9 Capsule Screen Design	21
Figure 3.10 Profile Design	21
Figure 3.11 Admin Dashboard Design	22
Figure 3.12 Capsule Edit/Add Design	22
Figure 3.13 Physical Data Flow Diagram	23
Figure 4.1 API Integration Module	32
Figure 4.2 Registration Module	33
Figure 4.3 Authentication Module	33
Figure 5.1 Home Screen	39
Figure 5.2 Login	39
Figure 5.3 Register	40
Figure 5.4 About Page	40
Figure 5.5 Learning Areas	41
Figure 5.6 Simulations	41
Figure 5.7 Capsules	42
Figure 5.8 Quizzes	42
Figure 5.9 Admin Panel	43
Figure 5.10 Add Simulations	44
Figure 5.11 Profile	44
Figure 5.12 CRUD Quizzes	45
Figure 5.13 Search Results	46

Figure 5.14 Search Results	46
Figure 5.15 Edit Capsule	47
Figure 5.16 Add Capsule	48
Figure 5.17 After Addition of Capsule	49
Figure 5.18 Gravity Simulator	50
Figure 5.19 Ohm's Law Simulator	50
Figure 5.20 Atom Simulator	51
Figure 5.21 Solar System Simulator	51
Figure 5.22 JavaScript Editor	52
Figure A.23 Gantt Chart of Schedule	53
Figure A.24 Table of User	56
Figure A.25 Table of Capsule	56
Figure A.26 Table of Capsule Simulations	56
Figure A.27 Table of Comments	56
Figure A.28 Table of Favourites	57
Figure A.29 Table of Quiz Options	57
Figure A.30 Table of Quizes	57
Figure A.31 Table of Simulations	57
Figure A.32 Supervisor Consultation Form	58

LIST OF TABLES

Table 4.1	Express Endpoint Testing: Capsules GET Methods	34
Table 4.2	Express Endpoint Testing: Capsules GET Methods	34
Table 4.3	Express Endpoint Testing: User Methods	35
Table 4.4	System/Application Testing: General Functionalities.....	35
Table 4.5	System/Application Testing: Authentication	36
Table 4.6	System/Application Testing: Admin Functionalities.....	36
Table 4.7	System/Application Testing: User Functionalities	36

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
CSS	Cascading Style Sheet
DFD	Data Flow Diagram
DOM	Document Object Model
ER	Entity-Relationship
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
I/O	Input Output
IT	Information Technology
JS	JavaScript
JSON	JavaScript Object Notation
JSX	JavaScript XML
JWT	JSON Web Token
OS	Operating System
RAD	Rapid Application Development
REST	Representational State Transfer
SQL	Structured Query Language
STEM	Science Technology Engineering Mathematics
UI	User Interface
UX	User Experience
WebGL	Web Graphics Library
XML	Extensible Markup Language

1 INTRODUCTION

1.1 Introduction

LabXplorerX is revolutionizing science education by providing an innovative interactive learning platform designed to transcend traditional learning methods. Specifically tailored for students and educators in STEM fields, LabXplorerX aims to bridge gaps in practical science education by offering interactive simulations across diverse disciplines. This cutting-edge platform serves as a dedicated arena where scientific concepts can be engaged with deeply, virtual simulations can be conducted, can be visualized, and seamless collaboration can be achieved within the academics. LabXplorerX addresses critical gaps in science education by providing a dedicated platform specifically designed for simulations tailored to students in grades 7, 8, and 9. Unlike general educational platforms that lack interactive simulation components, LabXplorerX offers specialized modules such as Basic Electronics Simulations, Basic Chemistry Simulations, Basic Astronomy Simulations, and a Basic Online Coding Environment with animations. This targeted approach allows students to gain hands-on experience and apply theoretical knowledge in practical settings, thereby enhancing their understanding and retention of scientific concepts.

1.2 Problem Statement

LabXplorerX addresses critical gaps in science education by providing a dedicated platform specifically designed for virtual laboratory simulations for students of grades 7, 8, and 9. Unlike general educational platforms that lack interactive simulation components, LabXplorerX offers tailored modules such as Basic Electronics Simulations, Basic Chemistry Simulations, Basic Astronomy Simulations, and a Basic Online Coding Environment with animations. This specialized approach enables students to gain hands-on experience and apply theoretical knowledge in practical settings, enhancing their understanding and retention of scientific concepts.

1.3 Objectives

- Create an interactive learning platform for students from Grade 7,8,9 that enhances STEM education through inter-active simulations aweb various disciplines.

1.4 Scope

- The platform should provide a virtual space for students and educators to conduct interactive simulations and promote simulating learning.
- LabXplorerX facilitates collaborative learning through comments, enabling students to share insights and ask questions.
- The platform should be user-friendly and accessible, making it easy for students to engage in.
- LabXplorerX includes learning capsules where students can gain in-depth knowledge and understanding of various scientific concepts.

1.5 Limitation

- The creation of simulations is restricted to developers, as users and super users do not have the capability to create new simulations.
- The platform lacks mobile responsiveness for simulations, which limits accessibility and usability of simulations on mobile devices.

1.6 Development methodology

For the development of LabXplorerX, the Rapid Application Development (RAD) methodology is employed. This approach emphasizes iterative development and continuous user feedback rather than rigid planning. By engaging a diverse range of stakeholders, including friends, family, and esteemed faculty members from the Department of Computer Application, valuable insights on usability and functionality are gathered. This engagement allows for practical feedback from potential end-users and expert advice on educational and technological standards. As a result, LabXplorerX evolves in response to real user needs and academic requirements, ensuring the development of a more effective and user-centric platform for science education.

This iterative process ensures that LabXplorerX evolves in response to real user needs and academic requirements, resulting in a more effective and user-centric platform for science education.

1.7 Report Organisation

The material in this project report is organised into Six chapters. After this introductory chapter introduces the problem topic this project tries to address, chapter 2 contains the literature review of vital and relevant publications, pointing toward a notable project related infomations. Chapter 3 describes the Designs and Analysis of the System for the implementation of this project and models and methods. Chapter 4 provides an overview of Implementation tools, modules used and testing performed in certain unit. Chapter 5 Lesson Learn with outcomes including future recommendations. After Main Report contains have Appendix A that contains Gantt Chart and Supervisor Consultation form. Last one contains Referneces.

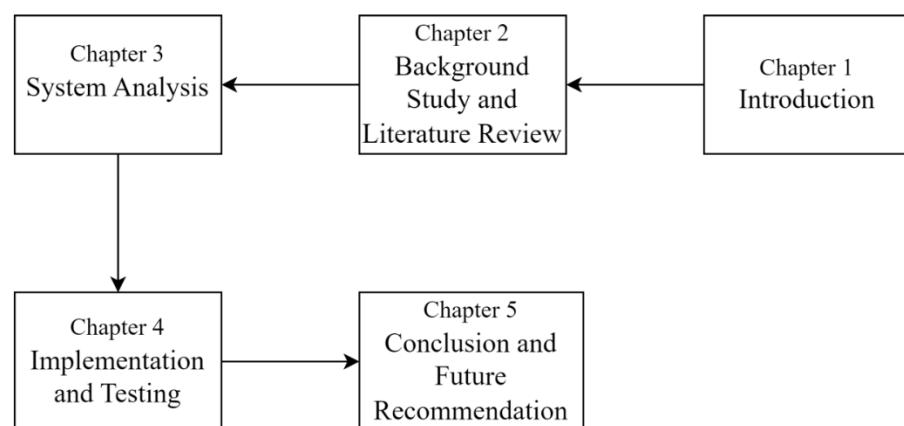


Figure 1.1: Report Organisation

2 BACKGROUND AND LITERATURE REVIEW

2.1 Background Study

Traditional science education, dependent on textbooks, lectures, and physical labs, often faces limitations due to resource constraints, high costs, and safety issues. These challenges are especially evident in under-resourced schools, where practical scientific experiences are limited. LabXplorerX addresses these issues by offering an innovative platform with interactive simulations in chemistry, physics, electronics, and astronomy. It provides a virtual space for conducting experiments, visualizing data, and engaging with scientific concepts, aiming to enhance traditional learning methods.

Additionally, LabXplorerX includes learning capsules, which function like blogs where users can access quizzes, share PDFs, and resources. This feature supports a more comprehensive and interactive learning experience, bridging gaps in practical science education and setting a new standard for STEM learning.

2.2 Literature Review

Teacher perception of OLabs pedagogy

OLabs, as name says, offers a robust web-based platform encompassing simulations, animations, tutorials, and assessments, designed to enhance interactive and accessible learning experiences outside traditional laboratory settings. Emphasizing student-centered learning, inquiry-based approaches foster essential skills such as scientific thinking, evidence-based reasoning, and creative problem-solving, which are fundamental for knowledge creation and retention [1].

How Khan Academy is changing the rules of education

This paper briefly describes how can an online learning platform change the way our education system works and improve on it.

- Khan Academy offers free, online instructional videos covering various subjects, allowing students to learn at their own pace and revisit concepts.
- The platform uses analytics to provide real-time feedback, enhancing personalized

learning experiences for both teachers and students.

- Khan Academy promotes a flipped classroom model where students watch videos at home and engage in problem-solving and discussions in class, fostering deeper understanding and collaboration.
- It democratizes education by providing high-quality instruction globally, irrespective of geographic location or socioeconomic status.
- The platform challenges traditional educational paradigms and suggests new possibilities for delivering effective education in the digital era.

Khan Academy being one of the main motivation for online learning and educating. [2]

PhET: Interactive simulations for teaching and learning physics

Perkins et al. (2006) introduce PhET, a collection of interactive simulations designed to enhance the teaching and learning of physics. These simulations aim to make abstract concepts more accessible and understandable through dynamic visualizations and interactive models. The authors emphasize the effectiveness of PhET in promoting conceptual understanding by allowing students to manipulate variables and observe real-time outcomes, thereby bridging the gap between theoretical concepts and practical application. They discuss the development process, which involves collaboration between physicists, educators, and software developers to ensure accuracy and educational efficacy. The article highlights PhET's versatility in catering to diverse learning styles and educational settings, promoting active learning and engagement [3].

An Introduction to HTML5 Game Development with Phaser.js

It provides a comprehensive guide to creating 2D games using the Phaser.js framework. It covers setting up a development environment, understanding fundamental game concepts, and managing game states and assets. The book teaches how to implement physics and collision detection, create animations and visual effects, design user interfaces, and integrate audio. It emphasizes practical, project-based learning, guiding readers through real game

development scenarios. Additionally, it offers debugging, optimization techniques, and deployment strategies for various platforms, making it an essential resource for both beginners and experienced developers looking to master HTML5 game development with Phaser.js [4].

The Role of Digital Simulations in STEM Education

This review focuses on the role of digital simulations in STEM education, emphasizing their benefits in teaching complex scientific concepts. The paper discusses various types of simulations, including those for physics, chemistry, and biology, and their impact on student learning and engagement. The authors argue that digital simulations can bridge the gap between theoretical knowledge and practical application, fostering a deeper understanding of STEM subjects [5].

Unity 3D: A Comprehensive Overview

Unity 3D is a widely used game development platform that offers a robust environment for creating both 2D and 3D games. The platform is known for its versatility and user-friendly interface, which makes it accessible for both beginners and experienced developers. Unity 3D provides a wide range of tools and features that facilitate the creation of interactive and immersive experiences. Key aspects of Unity 3D include:

- **Cross-Platform Development:** Unity allows developers to build games and simulations for various platforms, including Windows, macOS, iOS, Android, and consoles, with minimal adjustments.
- **Asset Management:** The platform includes a comprehensive asset management system, enabling the easy integration of 3D models, textures, animations, and audio.
- **Physics and Collision Detection:** Unity offers built-in physics engines and collision detection systems, essential for creating realistic interactions and behaviors within games.
- **Scripting and Customization:** Unity supports scripting, allowing developers to create custom game logic, interactions, and behaviors.

- **User Interface Design:** Unity provides tools for designing user interfaces, including menus, buttons, and HUD elements, which are crucial for enhancing user experience.
- **Real-Time Rendering:** The platform includes a powerful rendering engine that supports high-quality graphics and real-time visual effects, enhancing the visual appeal of games and simulations.

Unity 3D extensive documentation, active community, and support for various assets and plugins make it a preferred choice for developing interactive and educational applications [6].

3 SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

The project is following a structured approach that utilizes the Rapid Application Development (RAD) methodology. This approach segments the project into smaller, manageable components, allowing for incremental progress through iterative development. Individual modules are developed and integrated progressively, focusing on delivering and refining smaller segments. This method ensures continuous improvement and alignment with the overall goals while effectively managing the project through ongoing feedback and adjustments.

3.1.1 Requirement Analysis

Requirement analysis is a critical phase in the software development lifecycle that focuses on understanding and documenting the needs and expectations of stakeholders. This process involves gathering detailed information about what users require from a system, which includes identifying functional requirements (what the system should do), non-functional requirements (how the system should perform), and constraints (limitations or restrictions). The goal is to create a comprehensive and clear specification that guides the development team in designing and implementing the system. Effective requirement analysis ensures that the final product aligns with user needs and business objectives, reduces the risk of project failure, and facilitates efficient communication among stakeholders. By thoroughly analyzing requirements, teams can address potential issues early, prioritize features, and ensure a smoother development process.

3.1.1.1 Functional Requirements

The functional requirements of LabXplorerX are mentioned below:

- **User Profiles:** LabXplorerX allows children and teachers to create personalized profiles for managing their activity within the platform. Users can log in with unique credentials. The profile section displays only the user's own comments and interactions within the platform, along with a list of their favorited capsules. This streamlined

approach helps users easily track their contributions and revisit their most valued content.

- **Interactive Virtual Simulations:** LabXplorerX offers a range of interactive virtual simulations, including Basic Electronics, Basic Chemistry, Basic Astronomy, and an Online Coding Environment. These simulations provide immersive experiences where users can engage in hands-on activities, such as manipulating virtual equipment and conducting experiments. By integrating interactive animations and real-world scenarios, LabXplorerX facilitates experiential learning, allowing users to explore scientific principles and phenomena in a dynamic digital environment.
- **Capsule Tools:** In LabXplorerX, only administrators have the ability to create educational capsules, quizzes, and simulation links. They can design capsules with interactive quizzes, organize educational content. However, the creation and development of new simulations are restricted to developers, ensuring that complex interactive simulations are handled by technical experts while allowing administrators to manage and assign tasks within the platform.
- **Comments and Favourites for Learning Capsules:** LabXplorerX provides a comments section for each learning capsule, allowing students and teachers to leave feedback, ask questions, or share insights directly related to the content. Users can engage with one another by commenting on specific capsules. Additionally, the platform supports a "favourites" feature, enabling users to bookmark and easily revisit their preferred capsules, enhancing their personalized learning experience.
- **Quizzes and Learning Capsules:** LabXplorerX integrates quizzes and learning capsules to reinforce knowledge and assess comprehension. Quizzes are designed to evaluate understanding of concepts covered in simulations, while learning capsules provide bite-sized, focused content on specific topics. These features help consolidate learning and provide instant feedback.
- **Admin Dashboard:** The admin dashboard in LabXplorerX provides a centralized interface for managing user accounts, monitoring platform usage, and overseeing system performance. Administrators have the ability to perform full CRUD (Create, Read, Update, Delete) operations on quizzes and capsules, ensuring content is up-to-

date and relevant. They can also add simulation links to capsules, making it easier to integrate simulations into the learning experience, while maintaining control over the platform's content and functionality.

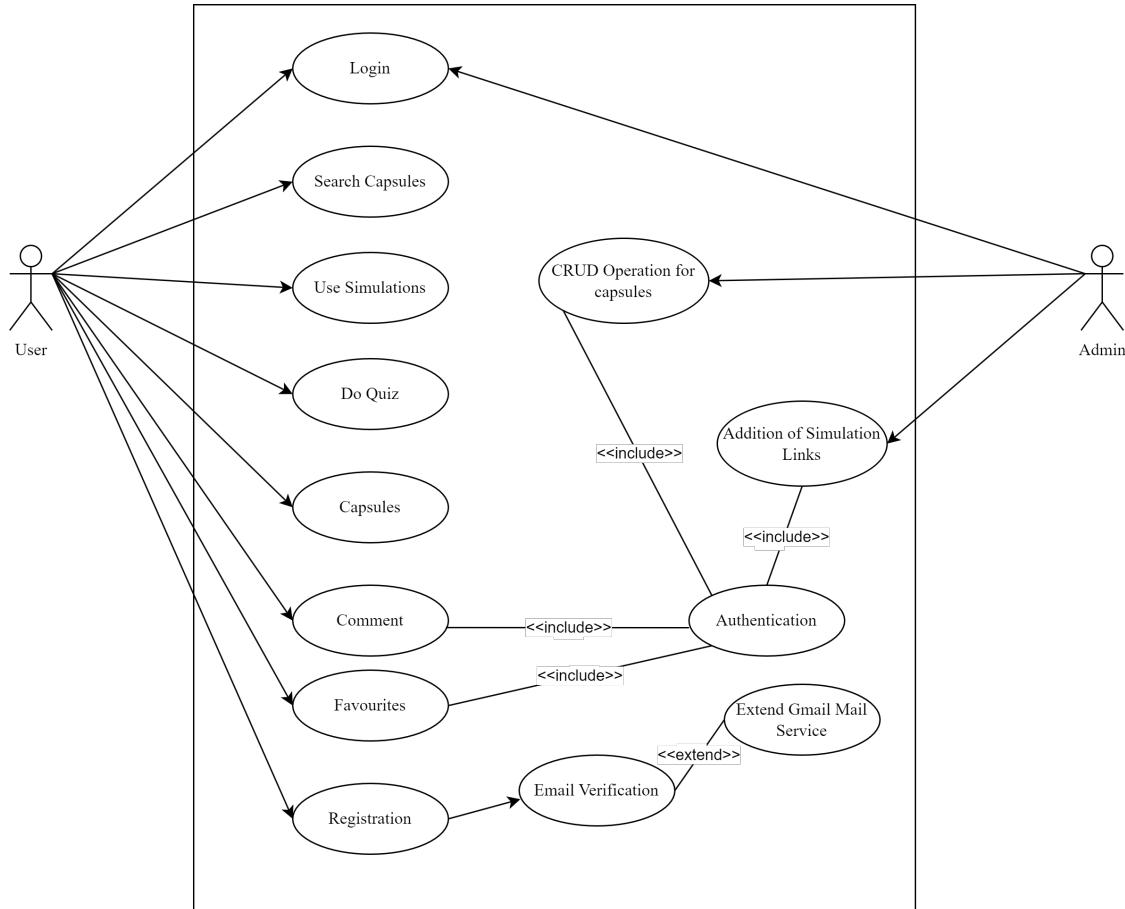


Figure 3.1: Use Case Diagram

3.1.1.2 Nonfunctional Requirements

The nonfunctional requirements of LabXplorerX are mentioned below:

- **Performance Enhancement:** The focus on performance involves optimizing the platform to handle high user loads and complex simulations efficiently. This includes minimizing reliance on external frameworks and ensuring smooth and responsive interactions.
- **Authentication Security:** Security is a paramount concern. To enhance the platform's security, advanced authentication algorithms, particularly focusing on hashing tech-

niques within the backend environment, have been implemented. This ensures that user authentication data is stored and managed in a highly secure manner.

- **Better UX Design:** User experience is central to the project's success. The emphasis on better UX design means that every aspect of the platform's interface, from navigation to interaction, will be meticulously crafted to ensure a seamless and intuitive experience. This design approach caters not only to experienced users but also to newcomers, ensuring that all users can effortlessly navigate and engage with the platform.
- **Responsive Design:** Recognizing the diverse range of devices and browsers used by users, the platform features a responsive design that ensures optimal adaptation across different screen sizes for most interface elements. This means that users can effectively access and interact with the platform whether they are using a desktop computer, tablet, or smartphone. However, it's important to note that the simulations are not responsive and are limited to desktop use. This approach guarantees a consistent and satisfying experience on various devices for general content, while simulations remain optimized for desktop environments.

3.2 Feasibility Analysis

A feasibility study is a systematic and structured analysis conducted to determine the viability and practicality of a proposed project plan. It serves as an evaluation tool to assess whether the project can be successfully implemented and if it aligns with the organization's goals and objectives. It involves gathering and analyzing relevant information to determine if the project is technically feasible, operationally feasible, economically feasible, and scheduling feasible.

3.2.1 Economical Feasibility

The development of the web application will utilize a range of free and open-source software development tools. For the frontend, React, a popular JavaScript library for building dynamic and interactive user interfaces, will be used. On the backend, Express, a minimal and flexible Node.js web application framework, will handle server-side logic and HTTP requests. PostgreSQL, an open-source relational database management system known for

its reliability and performance, will be employed for database management. Interactive simulations will be created using Phaser, a robust HTML5 game framework, while Unity, a powerful cross-platform game engine, will be used for more complex simulations and 3D elements. Additionally, funds will be allocated for economical server hosting to ensure the application remains accessible to users while managing costs effectively.

3.2.2 Operational Feasibility

LabXplorerX prioritizes operational feasibility through a user-centric design approach, emphasizing simplicity and ease of use. The system is highly interactive, enabling both students and educators to navigate effortlessly without requiring extensive technical knowledge. The user interface (UI) features a clean layout and intuitive controls, ensuring a seamless experience when accessing virtual environments and educational resources. By minimizing the need for extensive training and reducing potential barriers to adoption, LabXplorerX enhances user acceptance and engagement. The straightforward design promotes effective use of the app's features, supports educational activities, and fosters a positive user experience.

3.2.3 Technical Feasibility

Combining Express.js with React and PostgreSQL offers a robust and scalable solution for developing modern applications. Express.js, built on Node.js, provides an efficient backend framework for creating RESTful APIs and managing server-side logic. PostgreSQL, known for its reliability and advanced data management features, serves as a solid foundation for secure and efficient data storage and querying. On the frontend, React facilitates the creation of responsive and visually appealing applications across multiple platforms using a single codebase. This stack leverages the strengths of each technology: Express.js for backend scalability and API development, PostgreSQL for comprehensive data handling, and React for seamless and dynamic UI development. Supported by active communities and extensive documentation, this combination ensures ample technical support, resources, and flexibility for both deployment and maintenance, making it an ideal choice for delivering modern, interactive applications.

3.3 Structured System Modelling

Structured system modeling is a methodical approach used to design complex systems by decomposing them into manageable components and utilizing formal diagrams and tools. This approach aids in clearly defining system requirements, workflows, and interactions. By breaking down a system into its constituent parts, structured system modeling facilitates a thorough understanding of its structure and behavior. The use of formal diagrams and tools ensures that all aspects of the system are documented and analyzed systematically, which enhances clarity, communication, and accuracy throughout the design process. This methodical approach supports the creation of well-organized and efficient systems, improving overall design quality and project outcomes.

3.3.1 Process Modeling

Processing Modeling visually represent the flow of processes within a system, showing how inputs are processed into outputs. They help in understanding the system's functionality and data movement, aiding in the design and analysis of processes.

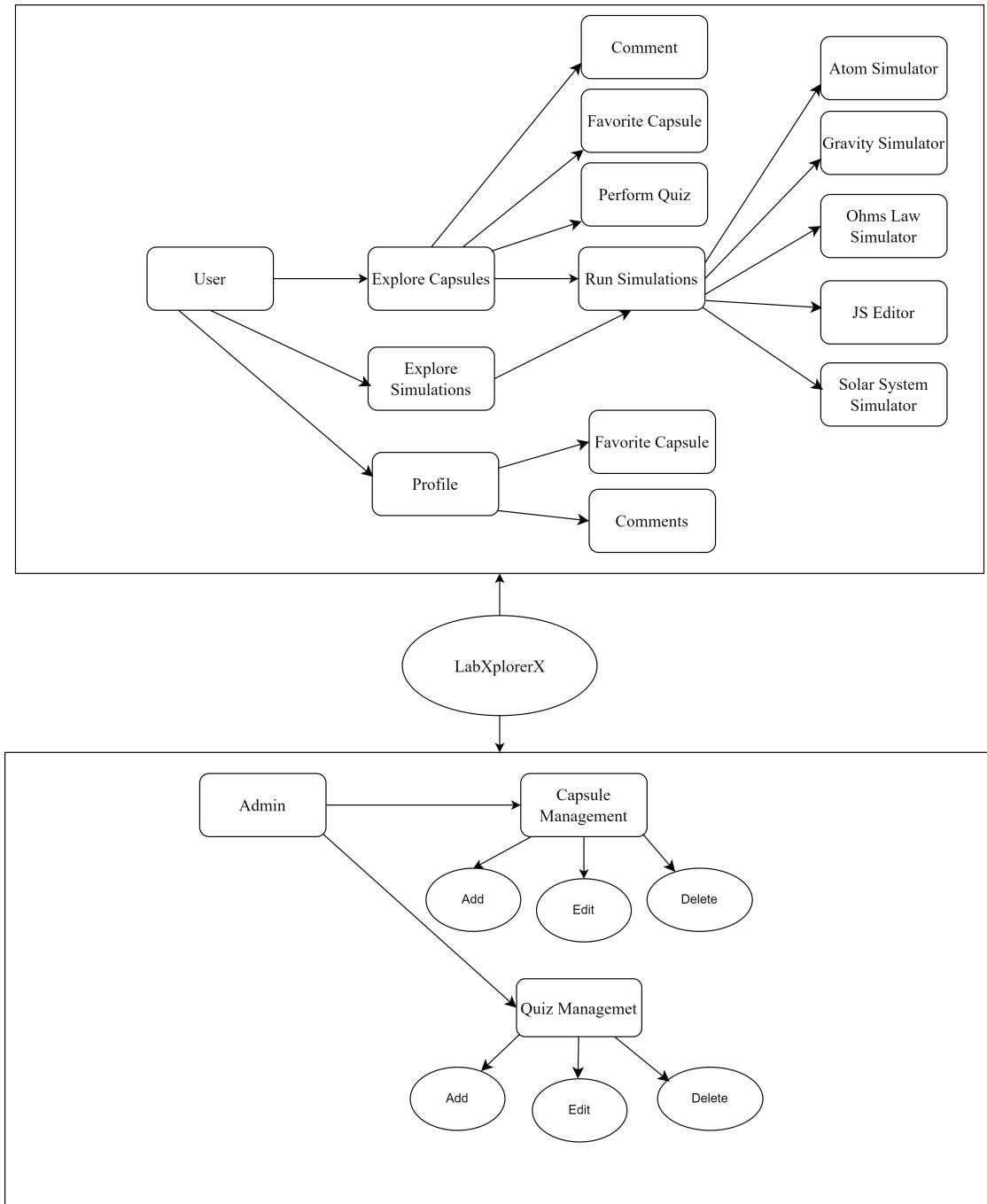


Figure 3.2: Process Model: Logical DFD

3.3.2 Data Modelling(ER-Diagram)

The Entity-Relationship (ER) Diagram is primarily used to design a database schema. The ER diagram provided below facilitates the creation of a database in SQL by clearly illustrating the entities, their attributes, and the relationships between them. This visual representation helps in structuring the database effectively, ensuring that all necessary data elements and their interconnections are accounted for. The system is organized into several key entities, each with specific attributes.

Entities Description

The Users entity represents the individuals who interact with the system, including attributes such as id, username, email, password, email_verification_token, and email_verified to manage user authentication and verification. The Capsules entity covers various educational content units, each with a unique id, and attributes like title, description, thumbnail, images, pdf, category, and author_id to provide detailed information and multimedia resources. Simulations are described by the Simulations entity, which includes id, title, description, link, and category to organize and access interactive simulations. The Comments entity enables user feedback on capsules, featuring attributes such as comment_id, comment_text, user_id, and capsule_id to link comments to their respective users and content. Quizzes, associated with capsules, are managed by the Quizzes entity, which includes quiz_id, title, category, and capsule_id for assessment purposes. The Options entity represents the various answers available for quizzes, with attributes such as option_id, option_text, is_correct, and quiz_id to identify and validate correct responses. Lastly, the Favorites entity tracks user preferences for capsules through user_id and capsule_id, enabling users to mark and easily access their favorite content.

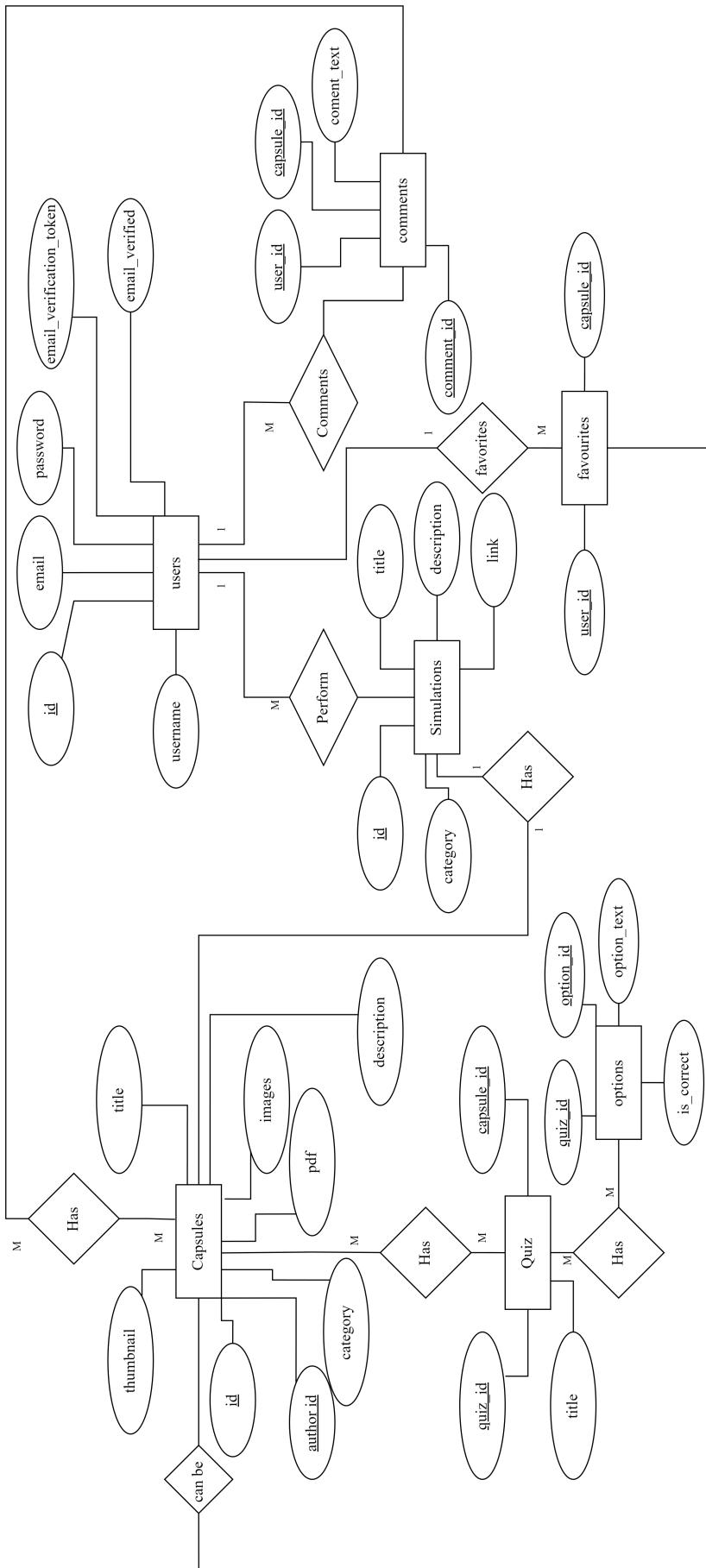


Figure 3.3: ER Diagram of System Data

3.4 Structured System Design

3.4.1 Architecture Design

The following diagram illustrates the architecture of our application. The application is structured using a three-tier architecture to ensure a clear separation of concerns and efficient functionality. The Presentation Layer, built with React.js, manages the user interface and user interactions. The Business Logic Layer, developed with Node.js and Express, handles core operations through middleware, routes, models, controllers, and utilities. Finally, the Data Management Layer uses PostgreSQL for relational database management and local server storage for handling files.

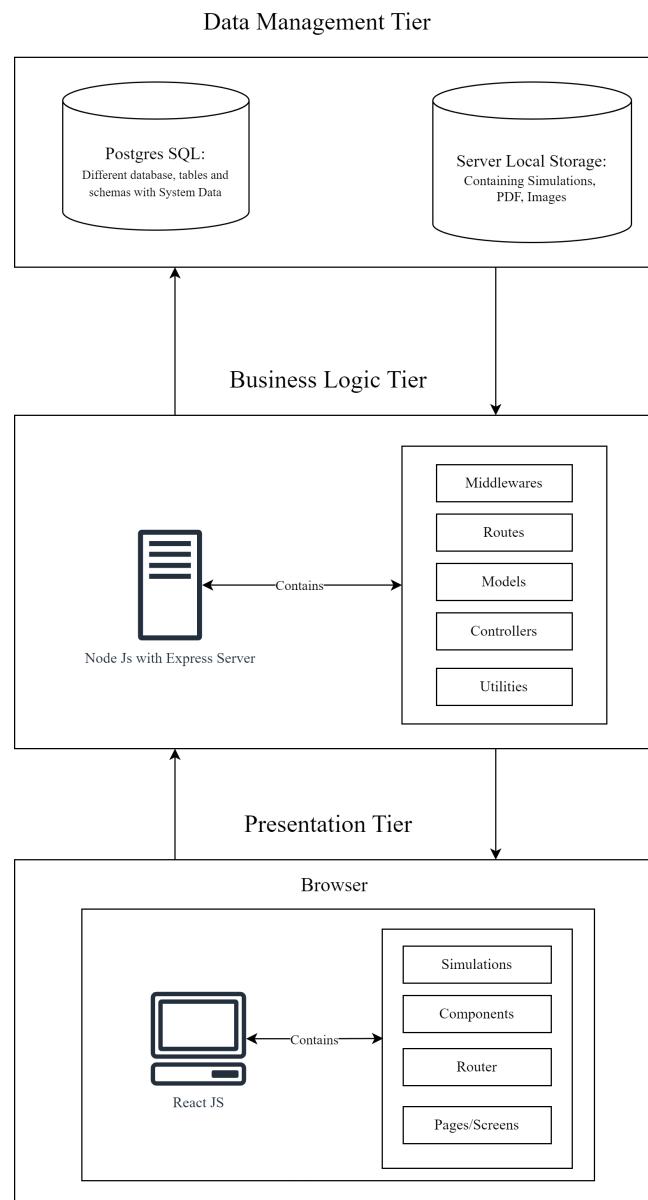


Figure 3.4: Three Tier Architecture of System

3.4.2 Database Schema Design

The schema design details the tables, their attributes, and the relationships between them, ensuring that data is stored efficiently and consistently. This design includes defining primary keys to uniquely identify records, foreign keys to establish relationships between tables, and constraints to maintain data integrity. The schema design provides a clear blueprint for creating and managing the database, supporting effective data organization and retrieval as per the application's requirements.

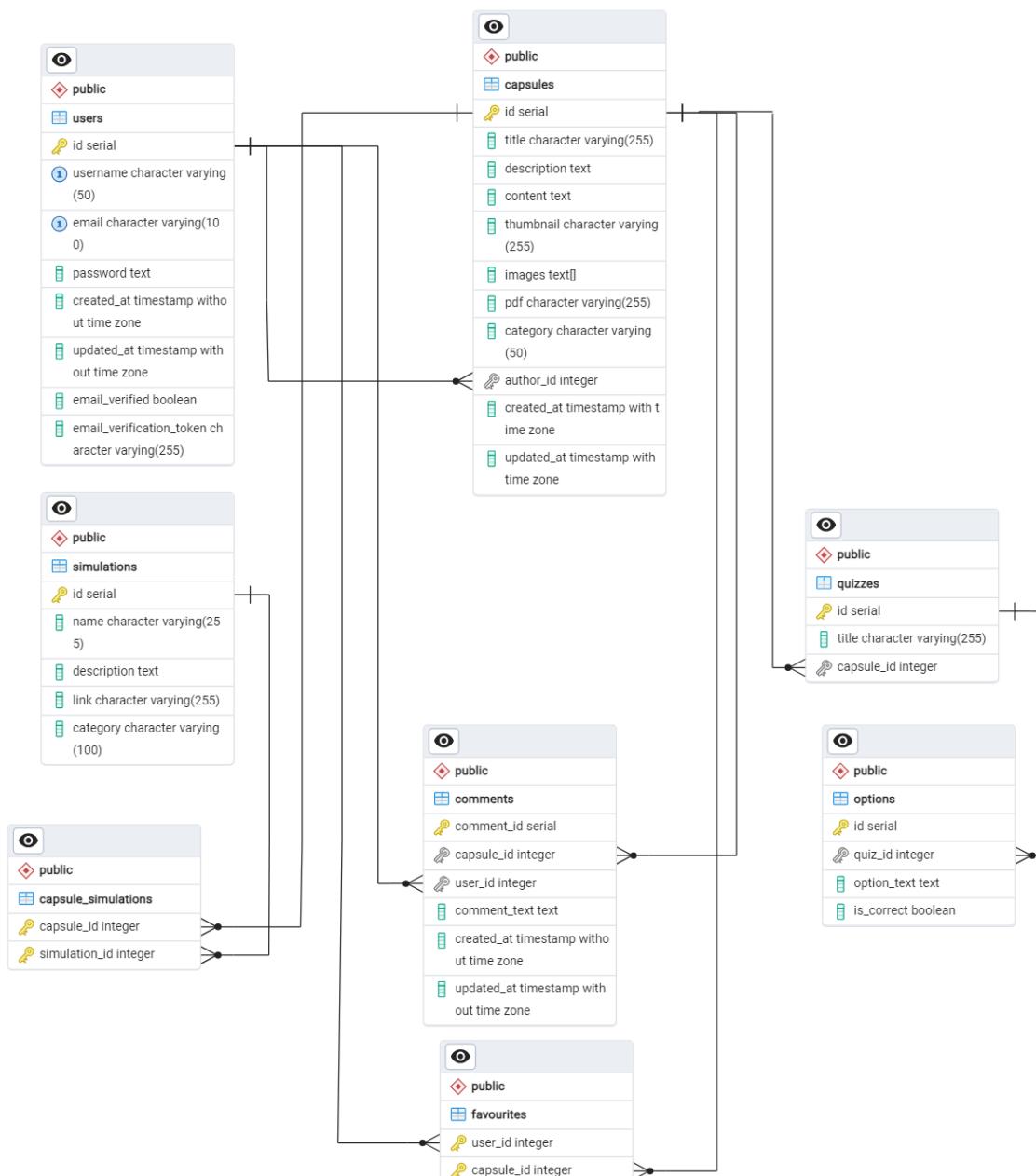


Figure 3.5: Schema Design

3.4.3 Interface Design

The interface design for this project focuses on creating a visually appealing, functional, and accessible user experience. The designs include layouts for the Home Screen, Menu, Capsule, Profile, and Admin Dashboard. Below is a detailed summary of the key design components:

- **Main Theme Color : Slate** The primary color used across the interface is Slate, chosen for its calm and professional appearance. This provides a consistent visual identity throughout the application while ensuring readability and contrast with other UI elements.
- **Font : Poppins** The Poppins font is selected for its modern and clean look. It enhances the readability of text across all devices, providing a uniform typographic structure that complements the design's minimalistic approach.
- **Button Colors:** Buttons are designed using a palette of Slate, Red, Green, and Blue. The different colors are used to signify various actions:
 - *Slate*: Default or secondary actions
 - *Red*: Alerts or destructive actions (e.g., delete)
 - *Green*: Confirmation or positive actions (e.g., submit)
 - *Blue*: Primary actions (e.g., next, save)
- **Accessibility Design:** The interface follows accessibility best practices by incorporating clearly defined and large buttons. These are designed to be easily distinguishable for users with visual impairments or motor difficulties, promoting ease of use.
- **Responsive Design:** While the UI components have been optimized for responsiveness across different devices, the simulations are not responsive. The focus is on ensuring that primary UI elements scale effectively for various screen sizes, maintaining usability and aesthetics.
- **Button Design:** Buttons throughout the interface are designed to be large and easily clickable. Their size and prominence ensure effortless navigation, reducing the chance of misclicks, and enhancing the overall user experience for a broad range of users.

- **Form Design:** The form design prioritizes clarity, validation, and accessibility. Below are the key design considerations:

- **Clear Labels and Placeholders:** All form fields are accompanied by clear labels and visible placeholders, ensuring that users can easily understand the required input.
- **Validation Feedback:** Real-time validation is implemented to give users immediate feedback.
- **Keyboard Navigation:** The form is designed to support keyboard navigation, allowing users to efficiently move through fields using the tab key. This feature improves accessibility for users who rely on keyboards or other assistive devices to interact with the form.

UI Designs

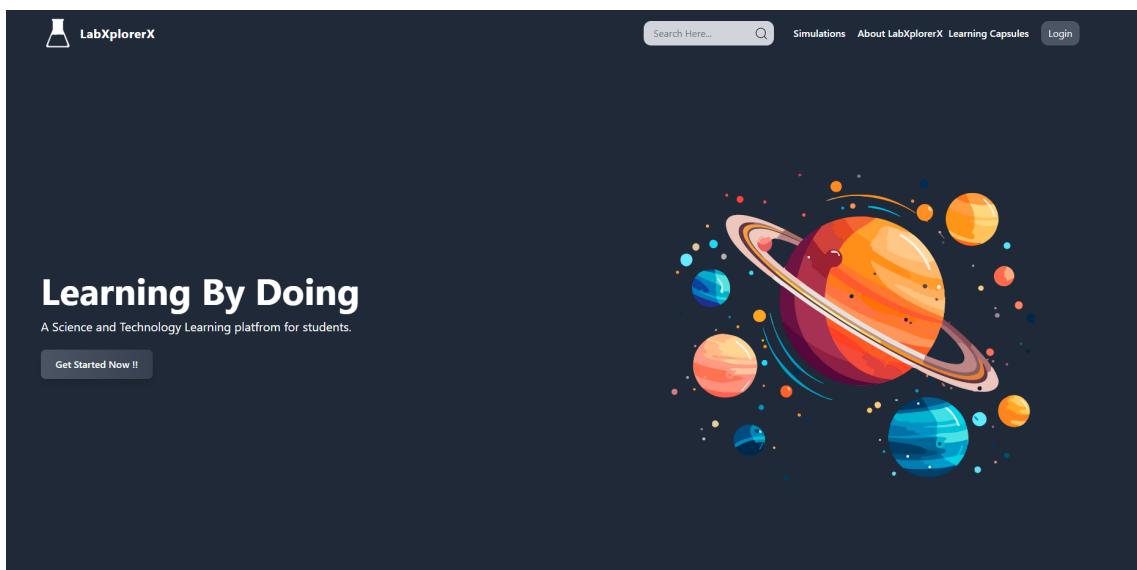


Figure 3.6: Home Screen UI Design

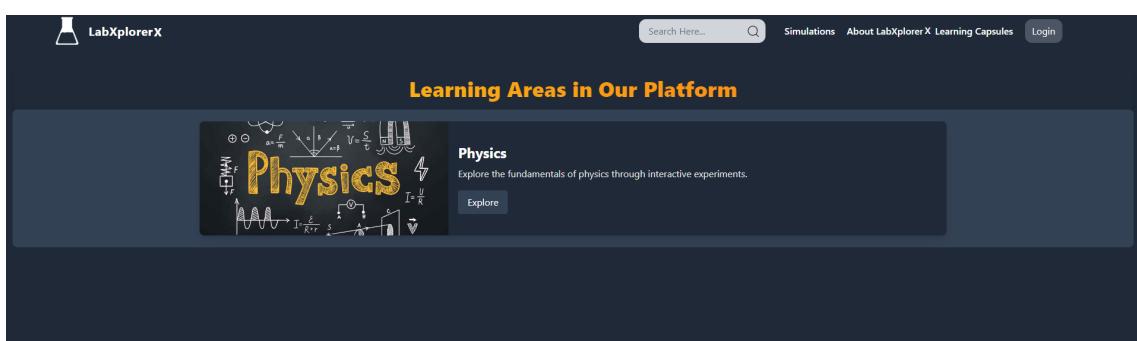


Figure 3.7: Capsule Category UI Design

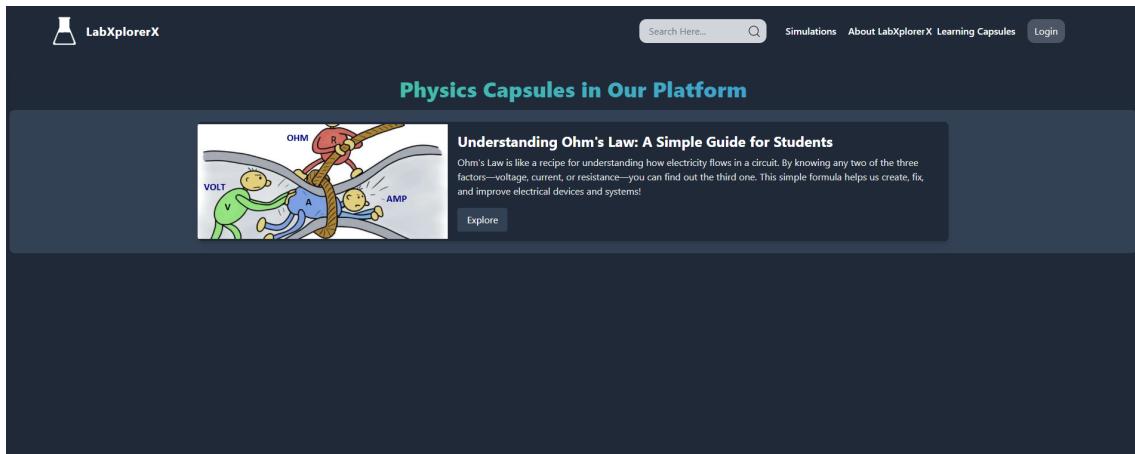


Figure 3.8: Capsules Menu Design

What is Javascript?

[Remove from Favourites](#)

JavaScript is an essential language for web development and beyond, offering a rich set of features for creating interactive, dynamic, and modern applications. It continues to evolve, with new standards and tools emerging to enhance development practices.

JavaScript is a versatile and widely-used programming language primarily known for its role in web development. Here's a comprehensive overview:

What is JavaScript?

Definition: JavaScript is a high-level, interpreted programming language that is primarily used to create interactive and dynamic content on websites. It allows developers to add functionality, manipulate web page content, and handle user interactions.

Key Features

- Client-Side Scripting:** JavaScript runs in the browser, enabling dynamic content updates without requiring a page reload. This includes tasks like form validation, animations, and interactive elements.
- Versatility:** Besides web development, JavaScript is also used on the server-side (with Node.js), in mobile app development (with frameworks like React Native), and even in desktop applications (with frameworks like Electron).
- Event-Driven:** JavaScript uses an event-driven model where actions (events) like clicks, keystrokes, or page loads trigger corresponding event handlers.
- Asynchronous Programming:** JavaScript supports asynchronous operations using callbacks, promises, and async/await, allowing for non-blocking operations and efficient API calls.

Figure 3.9: Capsule Screen Design

Welcome, Admin

Favourite Capsules

Understanding Ohm's Law: A Simple Guide for Students

[Explore](#)

What is Javascript?

[Explore](#)

Your Comments:

Great Capsule to learn
[What is Javascript?](#)

Figure 3.10: Profile Design

The screenshot shows the Admin Dashboard of the LabXplorerX platform. At the top, there is a navigation bar with the logo 'LabXplorerX', a search bar, and links for 'Simulations', 'About LabXplorerX', 'Learning Capsules', 'Logout', 'Profile', and 'Admin'. Below the navigation bar, the title 'Capsules' is displayed. A search bar labeled 'Search capsules...' and a blue button labeled 'Add Capsule' are located above the capsule cards. There are three cards: 1) 'Atom test' with an atom icon and a brief description about atoms being the basic building blocks of matter. 2) 'What is Javascript?' with a JS icon and a brief description about JavaScript being an essential language for web development. 3) 'Understanding Ohm's Law: A Simple Guide for Students' with an electrical circuit icon and a brief description about Ohm's Law. Each card has 'Edit', 'Delete', and 'Edit Quiz' buttons at the bottom.

Figure 3.11: Admin Dashboard Design

The screenshot shows the 'Edit Capsule' form. The title 'Edit Capsule' is at the top. It has two main sections: 'Category *' (set to 'Computer Science') and 'Title *' (set to 'What is Javascript?'). Below these is a 'Description *' field containing a detailed description of JavaScript. Underneath is a 'Simulators' section with a search bar and a dropdown menu set to 'None'. Finally, there is a 'Thumbnail' section with a file input field showing 'Choose File No file chosen'.

Figure 3.12: Capsule Edit/Add Design

3.4.4 Physical DFD

This Physical Data Flow Diagram (DFD) outlines the architecture of a web application with a clear separation of frontend and backend components. On the backend, Express (Node.js) handles the server-side logic, with middleware for authentication and error handling, routes for user, admin, and capsules, asynchronous controllers for managing requests, and a PostgreSQL database for storing data. Static files, including Unity simulations, are served from the backend. The frontend is built with React.js, utilizing React Router for navigation and RTK slices and APIs for state management. Screens and UI components, along with Phaser simulations, allow users to interact with backend data and display both static and dynamic simulations. The central store handles state management locally across components.

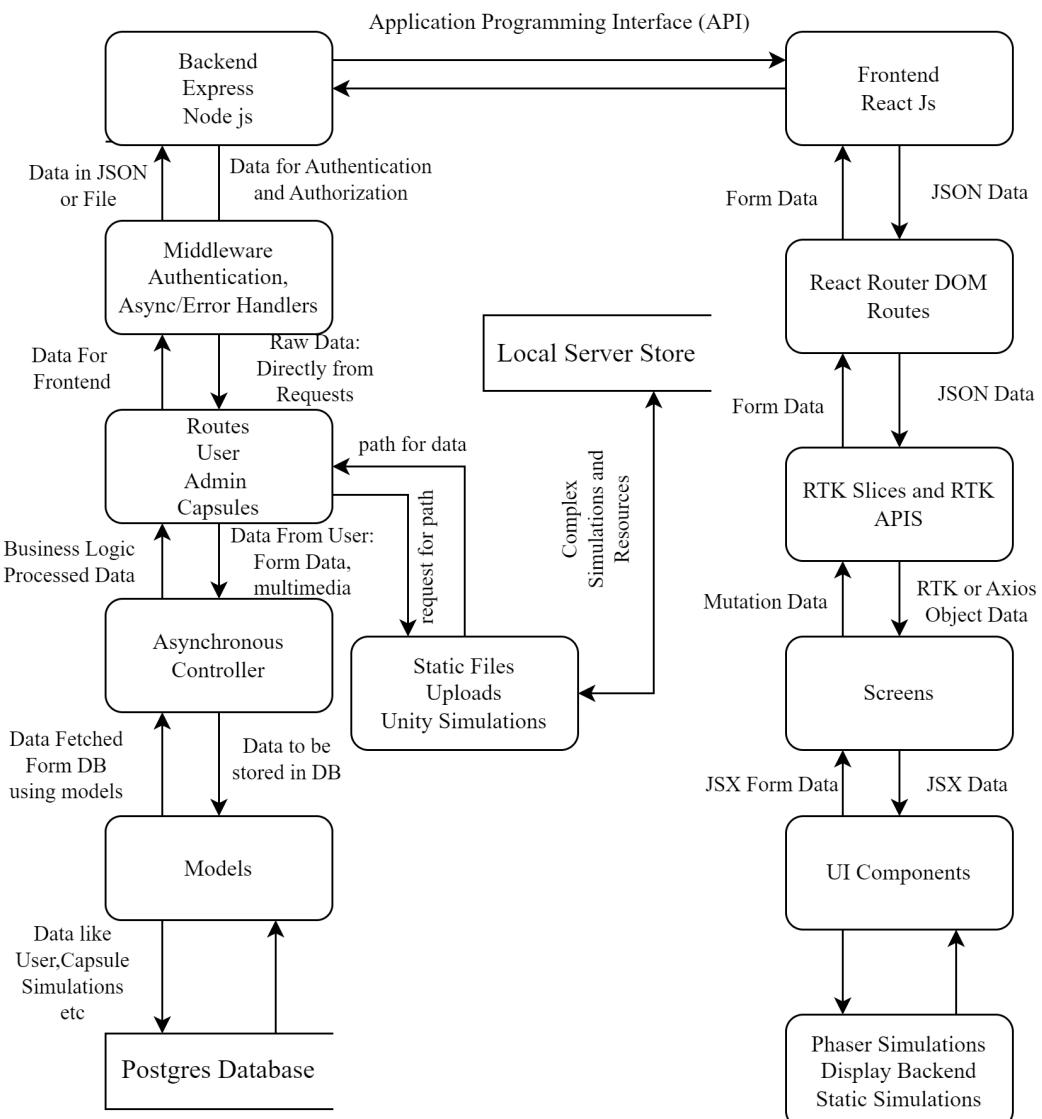


Figure 3.13: Physical Data Flow Diagram

3.5 Algorithms

3.5.1 Algorithm of AtomSimulator

Atom Sim is a virtual lab simulation tool designed to provide an interactive and educational experience in understanding atomic and molecular structures.

- **Initialization:** Initializes `selectedElement` with the first element from the `elements` array.
- **Game Setup:** Configures and initializes a Phaser game instance with specified dimensions and physics.
- **Atom Visualization:** Creates a nucleus and displays protons and neutrons; draws electron orbits and positions electrons in their respective orbits.
- **Electron Movement:** Uses a timer event to update electron positions based on their orbits and the current time.
- **User Interaction:** Updates `selectedElement` and re-renders the game scene when a new element is selected.
- **Cleanup:** Destroys the Phaser game instance when the component unmounts to free up resources.

3.5.2 Algorithm of CodeEditor

The ‘CodeEditor‘ component allows users to write, run, and view JavaScript code in a simulated environment.

- **Initialization:** Sets initial states for ‘code‘ (user input) and ‘consoleOutput‘ (output from the iframe).
- **Event Handling:** Listens for messages from the iframe to capture console output and errors.
- **Code Execution:** Creates an iframe to run the user-provided code, capturing and redirecting console messages.

- **User Interaction:** Updates the ‘consoleOutput’ based on the iframe’s console messages and renders the results.
- **Cleanup:** Removes the iframe from the DOM after execution to prevent resource leaks.

3.5.3 Algorithm of OhmsLawSimulator

The ‘OhmsLawSimulator‘ component simulates Ohm’s Law, allowing users to interactively adjust voltage and resistance to observe their effects.

- **Initialization:** Sets default values for voltage and resistance. Initializes Phaser game instance and scene references.
- **Game Setup:** Configures and starts a Phaser game with physics and rendering settings. Preloads assets and creates initial game elements like a static rectangle and a dynamic bulb.
- **Simulation Update:** Continuously updates bulb brightness based on the calculated current

$$I = \frac{V}{R} \quad (3.1)$$

- . Adjusts displayed values for voltage, resistance, and current.
- **User Interaction:** Updates Phaser scene values directly when voltage or resistance inputs change, triggering a re-render of the simulation.
- **Cleanup:** Destroys the Phaser game instance when the component unmounts to free up resources.

3.5.4 Algorithm of GravitySim

The ‘GravitySim‘ component simulates gravity effects on a player sprite, providing interactive controls for adjusting gravity and managing the simulation.

- **Initialization:** Sets up a Phaser game instance with a gravity simulation environment. Initializes references for the game, player, text elements, and timer.

- **Game Setup:** Loads assets, creates game elements (background, platforms, and player), and displays initial values for gravity, distance, and time. Pauses the simulation and player's movement initially.
- **Simulation Update:** Continuously updates gravity text, distance traveled, and elapsed time. Adjusts text position relative to the player sprite and handles the timer based on simulation state (paused or running).
- **User Interaction:** Provides buttons for increasing/decreasing gravity, resetting player position, pausing, and resuming the simulation. Updates the game state and UI accordingly.
- **Cleanup:** Destroys the Phaser game instance when the component unmounts to release resources.

Key Formula:

The distance traveled by the player due to gravity can be approximated using the formula for free-fall motion:

$$d = \frac{1}{2}gt^2 \quad (3.2)$$

where:

- d is the distance traveled (in meters),
- g is the acceleration due to gravity (in meters per second squared),
- t is the time elapsed (in seconds).

In the simulation, this formula is used to calculate and display the distance the player has fallen based on the current gravity setting and elapsed time.

3.5.5 Algorithm for SolarSystemSimulator

1. Initialization:

- Define celestial bodies (Sun, planets, moons) as GameObjects.
- Set up the main camera.

2. Setup Rotation:

- Attach RotateAround script to celestial bodies.
- Set rotation targets (e.g., planets orbit around Sun, moons around planets).
- Use `transform.RotateAround` for rotation.

3. Camera Behavior:

- Attach FollowAtTarget script to the main camera.
- Update camera to LookAt the target.

4. User Interaction:

- Attach ChangeLookAtTarget script to celestial bodies.
- On click, update camera target and adjust `fieldOfView`.

5. Implementation Steps:

- Create celestial bodies and attach scripts.
- Configure camera controls.
- Test rotation, camera focus, and user interactions.

4 IMPLEMENTATION

4.1 Tools Used

Figma

Figma is a cloud-based design and prototyping tool that empowers teams to collaborate on UI/UX design projects in real-time. It offers a user-friendly interface and powerful features that make it a popular choice among designers. With Figma, designers can create and share interactive prototypes, design components, and design systems. Its cloud-based nature allows for seamless collaboration, enabling multiple team members to work on the same design simultaneously. Figma supports version control, ensuring that design iterations can be easily tracked and managed.

React

React is a widely-used open-source JavaScript library developed by Facebook for building user interfaces, particularly single-page applications where data changes frequently. It emphasizes a component-based architecture, allowing developers to create reusable UI components that encapsulate their own structure, logic, and styling. React's use of a virtual DOM enhances performance by minimizing direct updates to the real DOM, ensuring efficient rendering. With its declarative approach, developers specify what the UI should look like based on different states, making the code more predictable and easier to debug. Additionally, React introduces JSX, a syntax extension that combines JavaScript and HTML, making it straightforward to write and understand UI components.

Postgres

PostgreSQL, often referred to simply as Postgres, is a powerful open-source relational database management system known for its reliability, robustness, and extensibility. Developed over decades and maintained by a global community of contributors, PostgreSQL offers a comprehensive set of features for managing structured data. It supports complex queries, transactions with ACID (Atomicity, Consistency, Isolation, Durability) properties, and a wide range of data types including JSON, XML, and spatial data. PostgreSQL's commitment to standards compliance and continuous improvement ensures compatibility with various programming languages and frameworks. With capabilities for scalability, data integrity, and advanced indexing, PostgreSQL is a preferred choice for applications requiring robust data management and high availability, contributing to its widespread adoption across industries from small startups to large enterprises.

Git/Github

Git is a distributed version control system that is both free and open-source, designed to handle projects of all sizes efficiently and swiftly. It simplifies collaboration by enabling multiple individuals to contribute changes that can be seamlessly merged into a single source. When using Git, the software runs locally on your computer, storing your files and their complete history. Alternatively, you can utilize online hosts like GitHub to store a copy of your files and their revision history. This central repository allows you to easily upload your changes and download updates from other developers, promoting seamless collaboration. Git facilitates automatic merging of changes, allowing multiple individuals to work on different sections of the same file and later merge their modifications without losing any work.

Node Js with Express

Node.js with Express.js is a powerful combination for building scalable and efficient web applications. Node.js provides a runtime environment that allows JavaScript to be executed server-side, leveraging its event-driven, non-blocking I/O model to handle multiple concurrent connections efficiently. Express.js, as a minimalist web framework for Node.js, simplifies the creation of APIs and routes, offering robust features such as middleware support, routing, and template engines. Together, Node.js and Express.js enable rapid development of RESTful APIs and web servers, making them well-suited for creating real-time applications, microservices, and backend systems. With a vibrant ecosystem of libraries and active community support, Node.js with Express.js remains a popular choice for developers seeking flexibility, performance, and scalability in web application development.

JavaScript

JavaScript is a programming language that is used to create interactive web pages and backend server. It is a powerful and versatile language that can be used to do a wide variety of things, including adding animation and interactivity to web pages, validating form data, processing user input, making Ajax requests to the server, and creating games and other interactive applications.

Phaser

Phaser is a powerful and popular open-source HTML5 game framework designed for creating 2D games that can run in both web browsers and mobile environments. Developed by Photon Storm, Phaser is known for its versatility and ease of use, making it a favorite among both beginner and experienced game developers. The framework supports Canvas and

WebGL rendering, automatically selecting the best option based on the device's capabilities. Phaser offers a robust set of features including physics engines (Arcade Physics, P2 Physics, and Matter.js), input handling, asset management, animations, and audio integration. Its component-based architecture allows developers to build complex games by combining reusable pieces of code, enhancing modularity and maintainability. With an active community, extensive documentation, and numerous tutorials, Phaser provides ample resources for learning and development, empowering creators to bring their game ideas to life efficiently.

Postman

Postman is a widely-used collaboration platform for API development, enabling developers to design, test, document, and monitor APIs with ease. Originally starting as a simple Chrome extension, Postman has evolved into a comprehensive tool that supports the entire API lifecycle. Its intuitive interface allows developers to construct and send HTTP requests to interact with APIs, receiving detailed responses to inspect and debug.

Unity 3D

Unity 3D is a leading game development platform renowned for its ability to create both 2D and 3D interactive experiences across a wide range of platforms, including consoles, mobile devices, and VR/AR environments. Developed by Unity Technologies, the engine offers a comprehensive suite of tools that cater to every aspect of game development, from design and prototyping to final deployment. Unity's real-time rendering capabilities, coupled with its powerful physics engine, allow developers to create highly immersive and visually stunning games. The engine's support for WebGL enables developers to deploy their games directly to the web, providing browser-based experiences without the need for plugins. WebGL in Unity leverages the engine's advanced rendering capabilities, allowing developers to create complex 3D environments that run smoothly in any modern browser. This makes Unity a versatile tool not only for traditional game development but also for creating interactive web applications.

Draw.io

Draw.io (now known as diagrams.net) is a powerful, web-based diagramming tool designed to simplify the creation of various types of visual diagrams, including flowcharts, organizational charts, network diagrams, and more. Developed to be user-friendly and accessible,

Draw.io provides a comprehensive suite of features that facilitate diagram creation and collaboration. The tool offers a wide range of pre-built templates, shapes, and connectors, allowing users to quickly construct detailed diagrams tailored to their specific needs. With its intuitive drag-and-drop interface, users can easily customize and arrange elements to convey complex information clearly and effectively. Draw.io also integrates seamlessly with cloud storage services like Google Drive and Dropbox, enabling real-time collaboration and ensuring that diagrams are easily accessible from anywhere. Its flexibility and ease of use make it an invaluable resource for professionals and teams seeking to visualize data, plan projects, or streamline workflows.

Figma

Figma is a leading web-based design tool renowned for its collaborative capabilities and versatility in creating user interfaces, prototypes, and vector graphics. Developed by Figma, Inc., the platform provides a robust set of design features that support both individual and team-based projects. With real-time collaboration, multiple users can simultaneously work on the same design, offering a seamless experience for brainstorming, feedback, and iterative design processes. Figma's vector graphics editor allows for precise design work, while its prototyping tools enable users to create interactive and dynamic prototypes that simulate the user experience. The platform supports extensive plugin integrations, enhancing functionality and streamlining workflows. Its cloud-based nature ensures that designs are always up-to-date and accessible from any device, making Figma an essential tool for modern design teams seeking efficient and collaborative design solutions.

4.1.1 Implementation Details of Modules

This subsection outlines the implementation specifics for each module, detailing the core functionalities and algorithms utilized. It covers the programming languages, frameworks, and tools used in development, along with the interaction and communication between modules. Key design patterns, data management strategies, and error-handling mechanisms are discussed to ensure optimal performance. Additionally, security measures and optimizations applied during implementation are highlighted.

Frontend API Integration Module

This diagram depicts the architecture of a web application where a React frontend com-

municates with an Express server through RTK Query and Axios. The process starts with the React frontend triggering either queries or mutations, which are handled by the Redux Toolkit (RTK) API slices. These slices manage API interactions, including the request and response lifecycle. Axios is used to send HTTP requests from the frontend to the Express server. The server processes incoming requests, performs necessary operations (such as database interactions), and sends responses back to Axios. Axios then passes the response data to RTK slices, which manage state updates. This architecture ensures smooth, organized data flow between the client and server, making it easier to handle API calls, manage state, and ensure a responsive user experience.

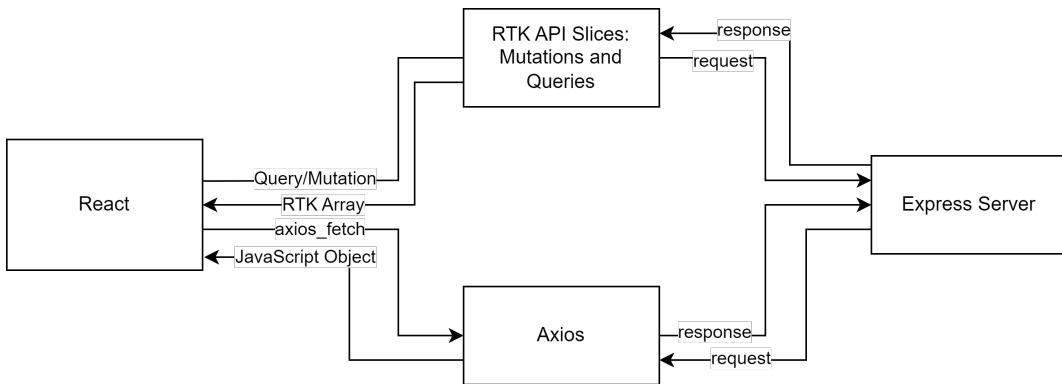


Figure 4.1: API Integration Module

User Registration Module

This diagram outlines a registration or authentication process involving several components. The user interacts with a React frontend to register or request access. A Token Generator backend service creates an authentication token, which is sent to the user via email using Node Mailer (implemented with Node.js and Nodemailer). The user submits the token back to the system, and the Token Verification service checks its validity. The Postgres database stores user information and tokens, ensuring that the submitted token matches the one in the database for verification. This flow connects the frontend, backend, email system, and database for user authentication.

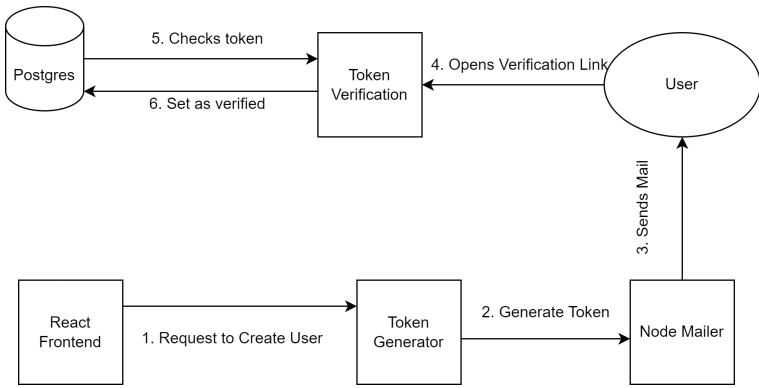


Figure 4.2: Registration Module

Authentication Module

The Authentication Module utilizes JSON Web Tokens (JWT) for secure user authentication. JWTs are compact, URL-safe tokens that encode user information, including a signature to verify the token's integrity. After a successful login, a JWT is generated and stored in an HTTP-only cookie, preventing unauthorized access via client-side scripts. The module also includes bcrypt hashing for securely storing user credentials and authentication middleware that checks the validity of the JWT on each request, ensuring only authenticated users can access protected resources.

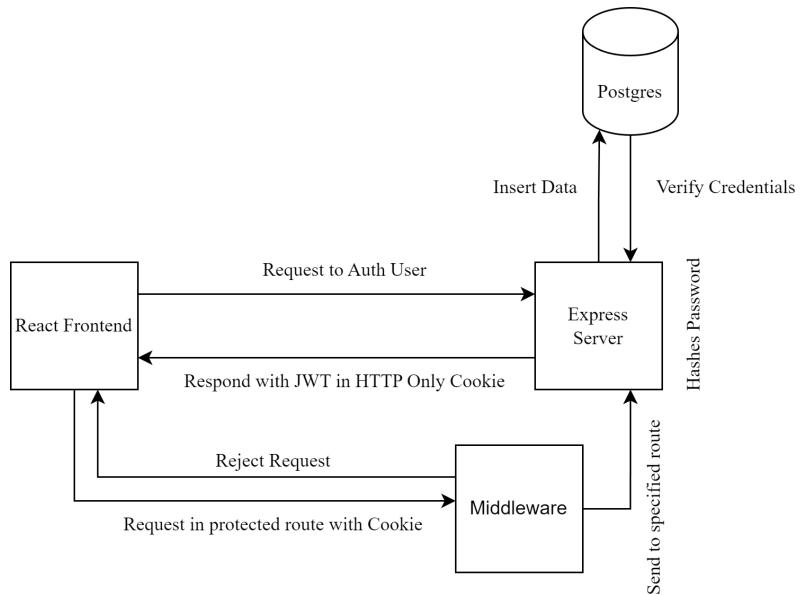


Figure 4.3: Authentication Module

4.1.2 Unit Testing Test Cases

These API unit testing are performed using Postman. API unit testing using Postman involves creating and sending requests to API endpoints to ensure they function correctly. You can write test scripts in Postman to validate responses against expected outcomes, such as status codes and response content. Postman also allows for automating tests using the Collection Runner and Newman for continuous integration and delivery.

Table 4.1: Express Endpoint Testing: Capsules GET Methods

Test No.	Test Case	Endpoint	Output
1	Getting Capsules By Category	/api/capsule/category?category=physics	Returns JSON response with Array of Objects with specific category
2	Getting Capsules By Id	/api/capsule/?capsuleId=1	Returns JSON response with Object of Capsules with specific id
3	Getting All Capsules	/api/capsule/all	Returns JSON response with Array of Objects of Capsules

Table 4.2: Express Endpoint Testing: Capsules GET Methods

Test No.	Test Case	Endpoint	Output
1	Login user/admin endpoint	/api/user/login body: Username and Password	Creates JWT token and sets an HTTP Only Cookie to the client side
2	Register api/user/register	api/user/register body: Username,Password and Password	Returns JSON response with Array of Objects with id

Table 4.3: Express Endpoint Testing: User Methods

Test No.	Test Case	Endpoint	Output
1	Admin add capsule endpoint	api/admin/add body:capsule informations and images	Sucessfully adds images in uploads folder and corresponding capsule into database

4.1.3 Test Cases for System Testing

The objective of System Testing is to conduct a comprehensive evaluation of the entire PERN application, encompassing both frontend and backend components. This testing phase aims to validate the correct and cohesive functioning of all integrated parts of the system.

Table 4.4: System/Application Testing: General Functionalities

Test No.	Test Case	Input	Output
1	Acessing Specific Capsules	/capsule/8	Shows whole capsule its images, pdf and all its meta information with quiz.
2	Capusles By Category	/capsules/physics	Shows cards, thumbnail, title, description and buttons about related category capsules.
3	Acessing Learning Areas	/learning-area	Shows Learning Capsules with categories of each capsules.
4	Acessing Quiz	/quiz/8	Shows Quiz of Specific Capsule.
5	Submitting Quiz	Selecting quiz and submitting	Shows How many quiz was right.
6	Accessing Ohms Law Simulator	/sims/ohmslaw	Shows Ohms law simulator with V,I and R inputs and phaser output.
7	Accessing PDF	In capsule screen click on Open PDF	Shows PDF document of specific capsule.
8	Accessing Images of Capsule	Clicking Images	Shows Images Realated to capsule.
9	Searching capsules and Simulations	/search/Gravity	Searches and shows capsules and simulation related to capsules.

Table 4.5: System/Application Testing: Authentication

Test No.	Test Case	Input	Output
1	Login User	/login Input: Username 'test' and Password 'test'	Logs in the User and shows profile
2	Login Admin	/login Input: Username 'Admin' and Password 'admin'	Logs in Admin and shows profile
3	Login User with incorrect credentials	/login Input: Username 'test' and Password 'test1234'	Alerts with wrong password
4	Register User	/login Input: Username 'Admin' and Password 'admin'	Logs in Admin and shows profile

Table 4.6: System/Application Testing: Admin Functionalities

Test No.	Test Case	Input	Output
1	Add capsule	/admin/add	Shows Form to add capsules and adds capsules when submitted
2	Edit capsule	/admin/edit-capsules/6	Shows Form to edit capsules and adds capsules when submitted
3	Delete capsule	Delete Capsule	Shows Form to add capsules and adds capsules when submitted
4	Edit Quiz	/admin/quiz-mgmt/6	Shows Form to add quiz,remove quiz and options

Table 4.7: System/Application Testing: User Functionalities

Test No.	Test Case	Input	Output
1	Favourite Function	Clicking Add to Favourites button	Adds to favourite and shows in profile
2	Commenting	Commenting on Capsules with "Good Capsule"	Adds Comment and shows good capsule in both capsule and author profile
3	User Profile	/profile	Shows Profile having Favourite Capsule and Comments done by User

5 CONCLUSION AND ANALYSIS

5.1 Conclusion

LabXplorerX is an innovative virtual laboratory platform tailored for enhancing science education through interactive simulations and experiments. It aims to revolutionize how students and educators engage with scientific concepts by offering a diverse range of features. LabXplorerX facilitates seamless exploration, collaboration, and learning across various scientific disciplines. This platform empowers users to conduct experiments, share insights, and leverage sophisticated algorithms to deepen their understanding. Additionally, LabXplorerX integrates advanced reporting capabilities and decision-making tools, enriching the educational experience beyond traditional classroom settings.

sectionLesson Learned / Outcome

The LabXplorerX project has made substantial progress in developing an interactive platform that fosters educational engagement. Throughout this journey, several key lessons have been learned:

- **Creating Engaging Simulations:** The development of 5 interactive simulations (Gravity Simulator, Ohm's Law Simulator, Atom Simulator, Solar System Simulator, and JavaScript Editor) has shown the value of hands-on learning tools. These simulations have proven effective in transforming abstract concepts into practical, experiential learning opportunities, greatly enhancing student understanding.
- **Content Design through Learning Capsules:** The creation of structured learning capsules highlighted the importance of organizing content into manageable, interactive modules. These capsules, while lacking visual aids, demonstrated that well-structured, interactive content is sufficient to foster student engagement and deeper understanding across various educational subjects.
- **Secure Authentication Systems:** Implementing a robust authentication system revealed the critical role of secure access management. Secure login, registration, and account management features are not only essential for protecting user data but also for building trust between students, teachers, and administrators.
- **Centralized Admin Panel:** The creation of a comprehensive admin panel underscored

the need for efficient management tools. By allowing administrators to manage capsules, quizzes and simulations, the panel ensures streamlined operations and platform sustainability.

- **Quizzes for Self-assessment:** Integrating quizzes into the learning modules provided valuable insight into the importance of continuous self-assessment. These quizzes enabled students to evaluate their grasp of the material, offering immediate feedback and enhancing the learning process.
- **User-centric Interface Design:** The development of various user interface screens (Home Screen, Login Screen, Register Screen, Simulations Screen, etc.) highlighted the significance of intuitive design. A user-friendly interface is key to an accessible and engaging learning experience, ensuring both students and teachers can easily navigate the platform.

All planned features have been successfully implemented, providing a secure, interactive, and engaging platform for learning. These lessons will guide future improvements, helping to maintain LabXplorerX as a leading educational tool.

5.1.1 Screenshots of Outcomes

Home Screen: Below is the screenshot of the Home Screen, which is the main interface of the application.

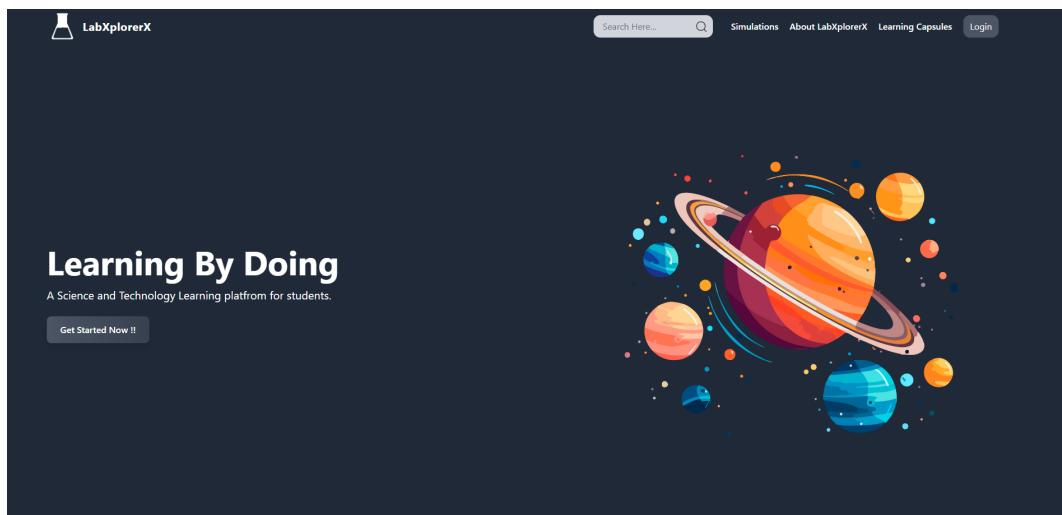


Figure 5.1: Home Screen

Login Screen: Below is the screenshot of the Login Screen, where users enter their credentials to access their accounts.

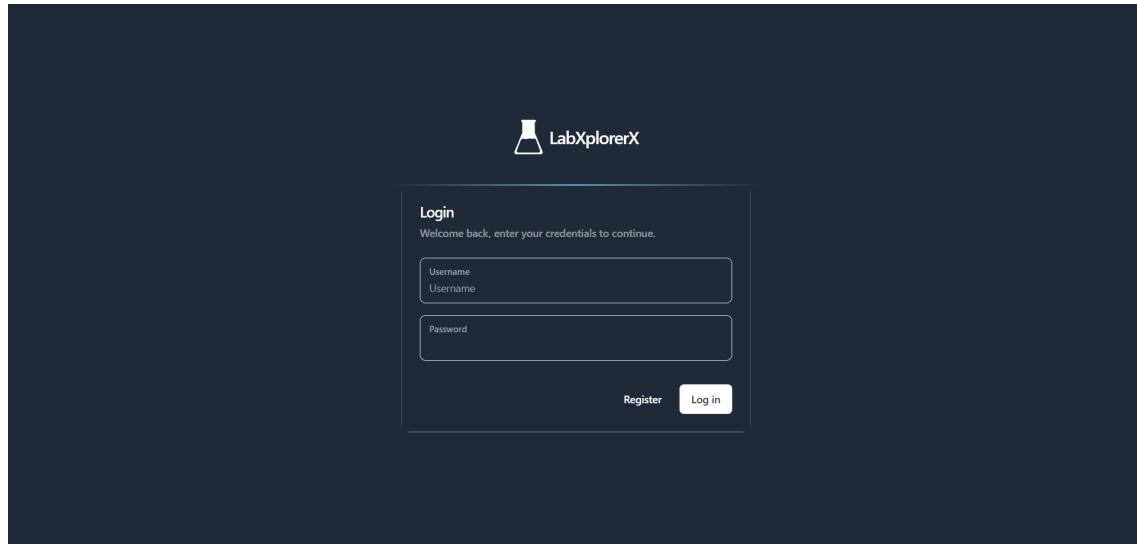


Figure 5.2: Login

Register Screen: Below is the screenshot of the Register Screen, where new users can create an account.

The screenshot shows the 'Register' page of the LabXplorerX application. At the top center is the LabXplorerX logo, which consists of a stylized flask icon followed by the text 'LabXplorerX'. Below the logo is the word 'Register' in bold. A sub-instruction 'New to LabXplorerX!! Create an Account to continue' is displayed. The registration form contains four input fields: 'Username' (placeholder 'Username'), 'Email' (placeholder 'example@example.com'), 'Password', and 'Confirm Password'. At the bottom right of the form are two buttons: 'Login' and 'Register'.

Figure 5.3: Register

About Page: Below is the screenshot of the About Page, which provides information about the application.

The screenshot shows the 'About LabXplorerX' page. At the top is a navigation bar with the LabXplorerX logo, a search bar containing 'Search Here...', and links for 'Simulations', 'About LabXplorerX', 'Learning Capsules', 'Logout' (in red), 'Profile' (in blue), and 'Admin' (in green). The main content area has a dark background with white text. It starts with a section titled 'About LabXplorerX' with a brief welcome message. Below it is 'Our Mission' with a description of the platform's purpose. The 'Features' section lists several bullet points: 'Virtual experiments in Chemistry, Physics, Electronics, and Astronomy.', 'Interactive simulations to enhance understanding through practice.', 'Personalized student profiles to track progress and achievements.', 'Teacher tools for creating and managing experiments and assignments.', and 'A collaborative forum for students and teachers to discuss and share ideas.' The 'Our Team' section contains a paragraph about the team's dedication to creating a top-notch educational platform. Finally, the 'Get In Touch' section encourages users to email the team with questions or feedback.

Figure 5.4: About Page

Learning Areas Screen: Below is the screenshot of the Learning Areas Screen, showing the various educational categories available.

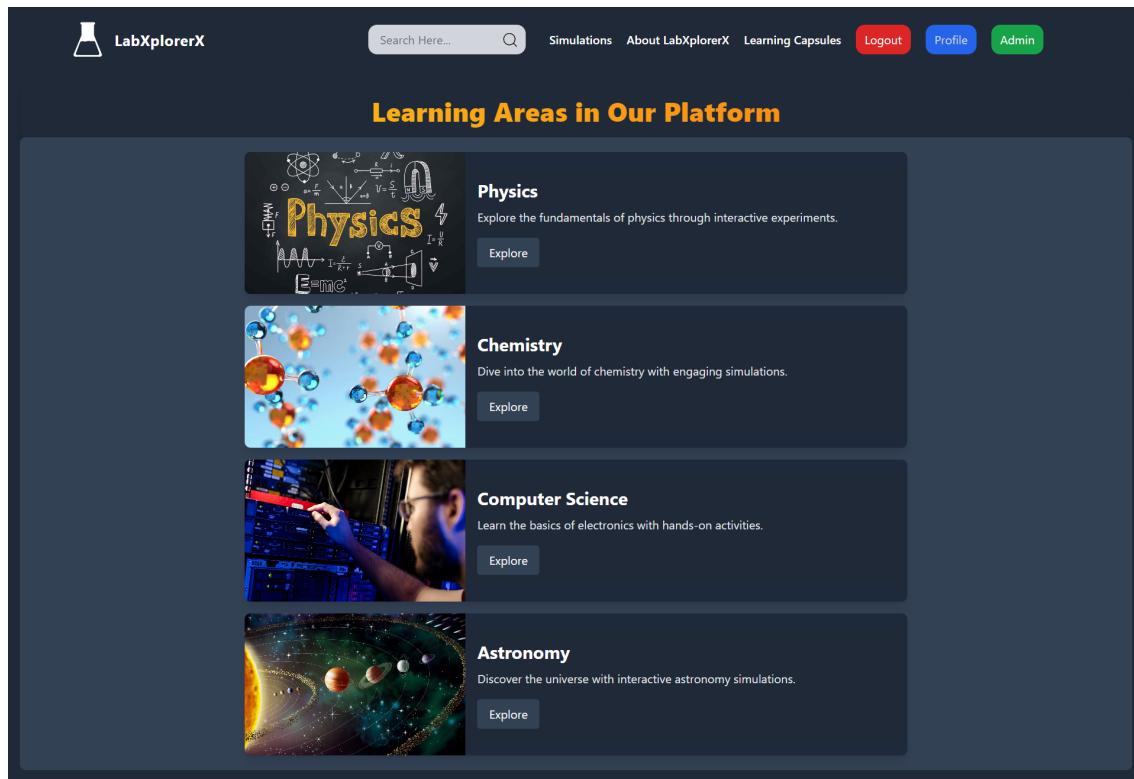


Figure 5.5: Learning Areas

Simulations Screen: Below is the screenshot of the Simulations Screen, where users can access different simulations.

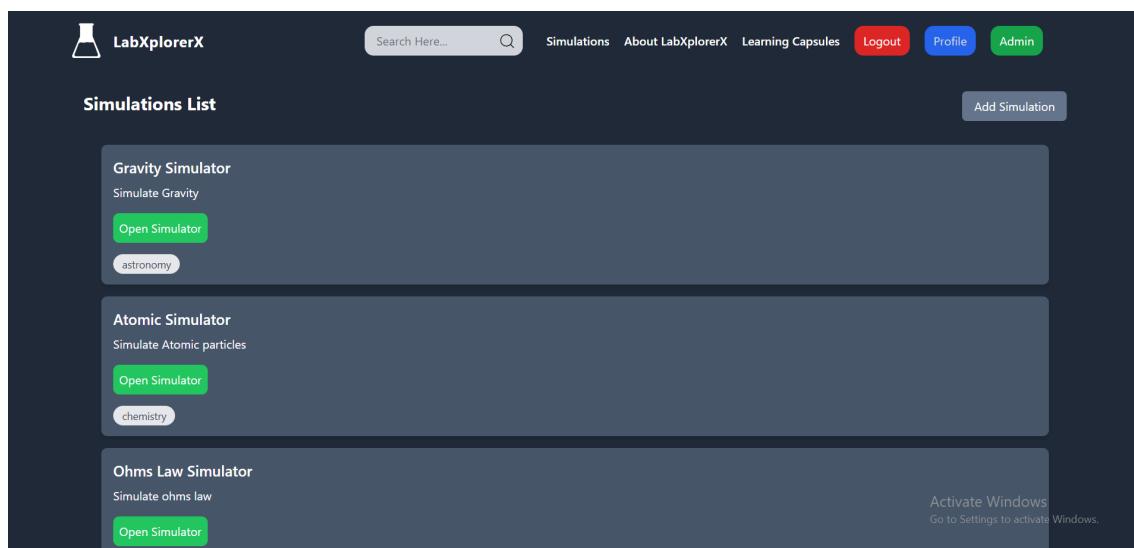


Figure 5.6: Simulations

Capsules Screen: Below is the screenshot of the Capsules Screen, which displays the available capsules in the application.

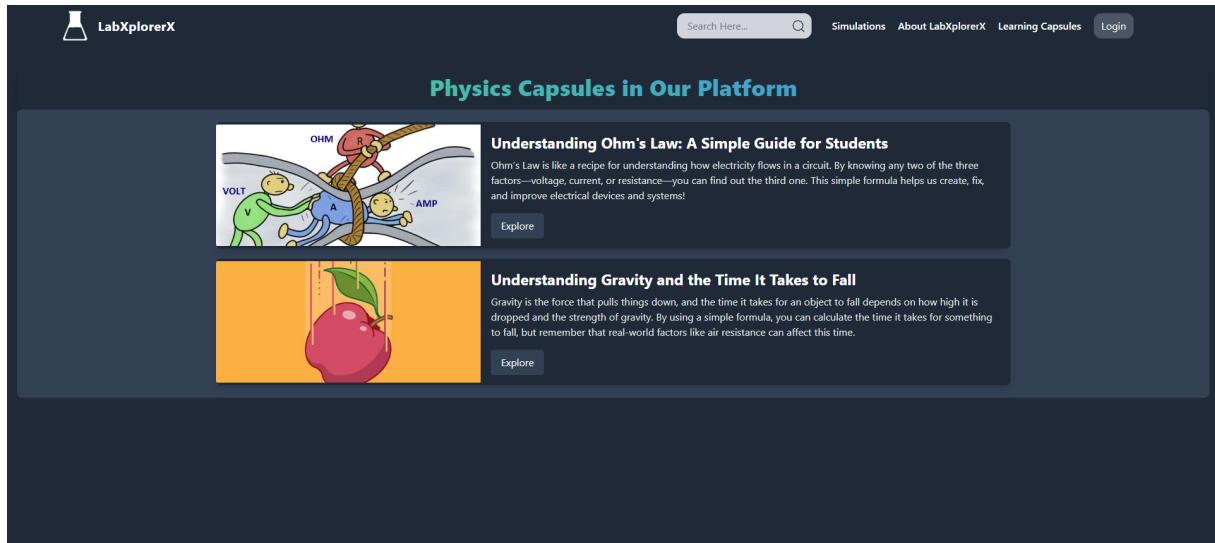


Figure 5.7: Capsules

Quizzes Screen: Below is the screenshot of the Quizzes Screen, where users can participate in various quizzes.

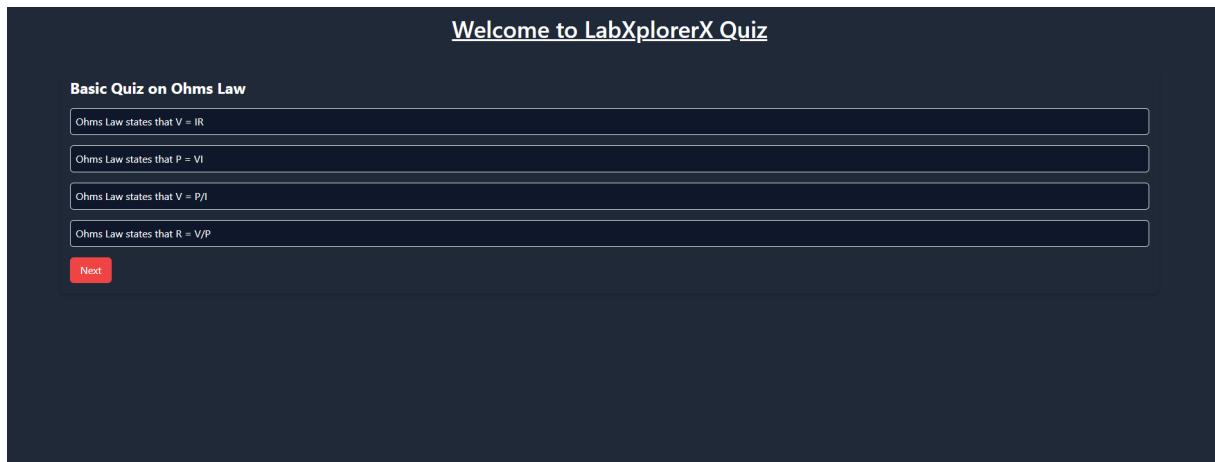


Figure 5.8: Quizzes

Admin Panel: Below is the screenshot of the Admin Panel, used by administrators to manage the application.

The screenshot shows the Admin Panel of the LabXplorerX application. At the top, there is a navigation bar with the logo 'LabXplorerX', a search bar, and links for 'Simulations', 'About LabXplorerX', 'Learning Capsules', 'Logout', 'Profile', and 'Admin'. Below the navigation bar, the page title is 'Capsules'. There is a search bar labeled 'Search capsules...' and a blue button labeled 'Add Capsule'. The main content area displays six learning capsules in a grid:

- Atom test**: An atom is the smallest unit of matter that retains the properties of an element. It is the basic building block of all matter. test. Includes three buttons: 'Edit' (blue), 'Delete' (red), and 'Edit Quiz' (green).
- What is Javascript?**: JavaScript is an essential language for web development and beyond, offering a rich set of features for creating interactive, dynamic, and modern applications. It continues to evolve, with new standards and tools emerging to enhance development practices. Includes three buttons: 'Edit' (blue), 'Delete' (red), and 'Edit Quiz' (green).
- Understanding Ohm's Law: A Simple Guide for Students**: Ohm's Law is like a recipe for understanding how electricity flows in a circuit. By knowing any two of the three factors—voltage, current, or resistance—you can find out the third one. This simple formula helps us create, fix, and improve electrical devices and systems! Includes three buttons: 'Edit' (blue), 'Delete' (red), and 'Edit Quiz' (green).
- Understanding Gravity and the Time It Takes to Fall**: Gravity is the force that pulls things down, and the time it takes for an object to fall depends on how high it is dropped and the strength of gravity. By using a simple formula, you can calculate the time it takes for something to fall, but remember that real-world factors like air resistance can affect this time. Includes three buttons: 'Edit' (blue), 'Delete' (red), and 'Edit Quiz' (green).
- Introduction to the Solar System**: The solar system is a dynamic and diverse environment with a variety of celestial objects interacting in complex ways. Studying these components helps us understand not only our own planetary system but also the broader mechanics of how solar systems form and evolve throughout the galaxy. Whether through observations, space missions, or scientific research, learning about the solar system provides valuable insights into the nature of our universe. Includes three buttons: 'Edit' (blue), 'Delete' (red), and 'Edit Quiz' (green).
- Learning Arrays in Java Script**: We will be learning arrays. Includes three buttons: 'Edit' (blue), 'Delete' (red), and 'Edit Quiz' (green).

Figure 5.9: Admin Panel

Add Simulations Screen: Below is the screenshot of the Add Simulations Screen, where administrators can add new simulations.

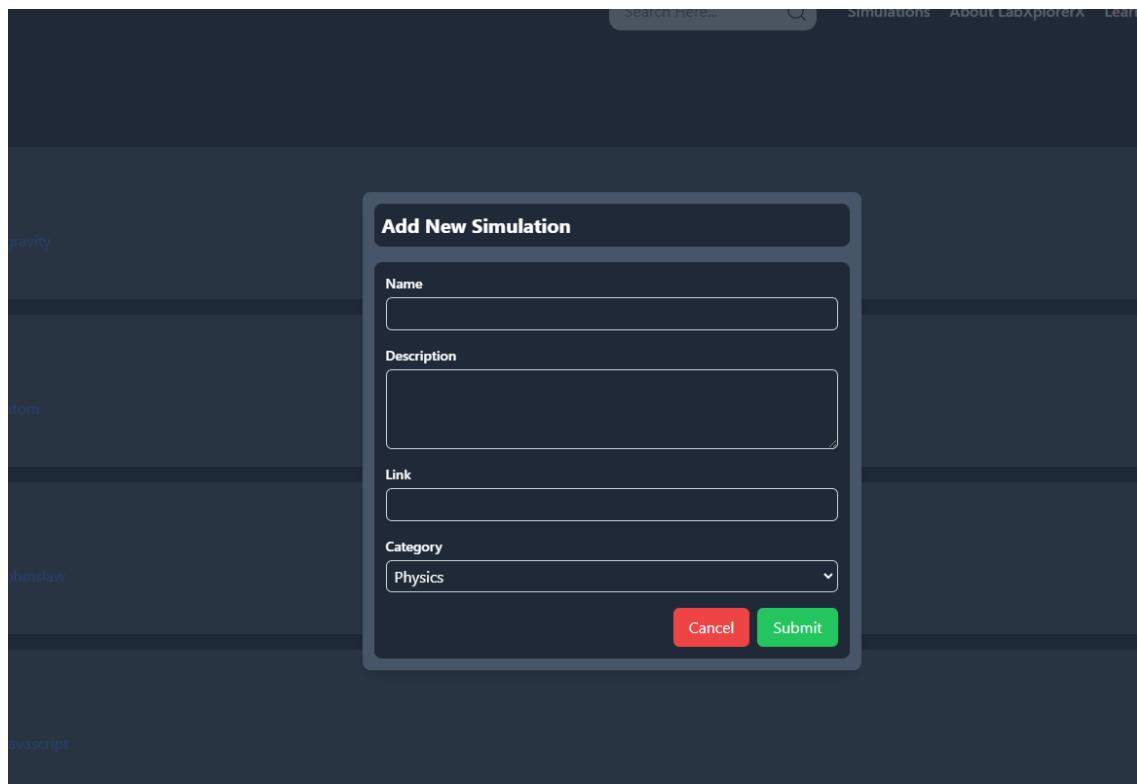


Figure 5.10: Add Simulations

Profile Screen: Below is the screenshot of the Profile Screen, where users can view and edit their profile information.

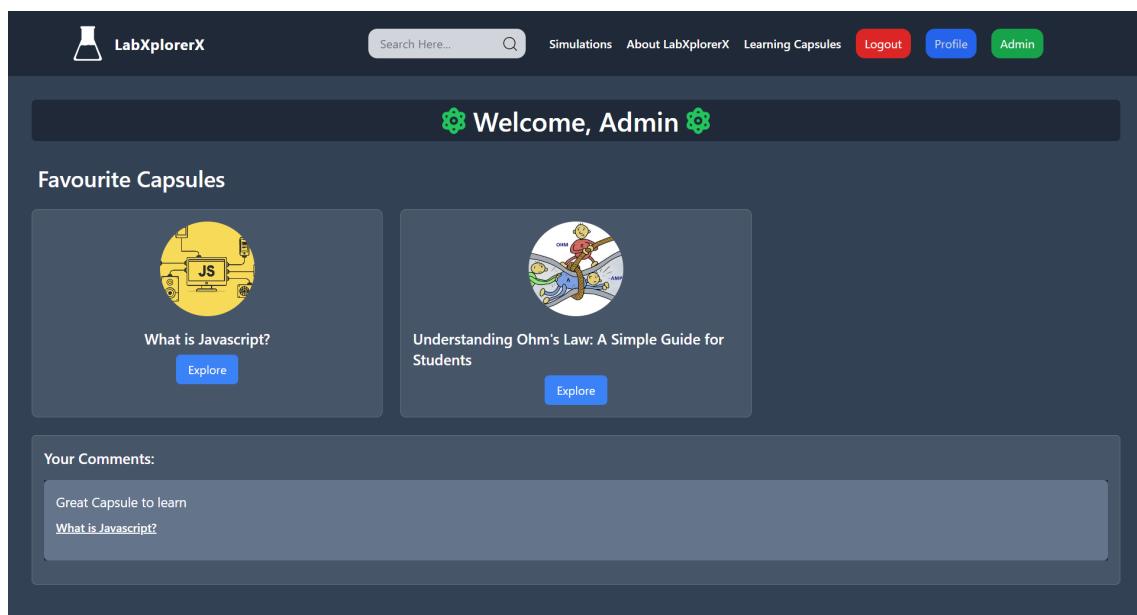


Figure 5.11: Profile

Edit Quizzes Screen: Below is the screenshot of the Edit Quizzes Screen, used for creating, updating, and deleting quizzes.

The screenshot shows the 'Edit Quiz' interface on the LabXplorerX platform. At the top, there is a navigation bar with links for 'Simulations', 'About LabXplorerX', 'Learning Capsules', 'Logout', 'Profile', and 'Admin'. The main area is titled 'Edit Quiz'.

Question 1 *

What are the correct ways to declare a variable in JavaScript?

Option 1 *

`leta myVar = 10;`

Correct Answer Remove Option

Option 2 *

`number myVar = 10;`

Correct Answer Remove Option

Option 3 *

`int myVar = 10;`

Correct Answer Remove Option

Option 4 *

`let myVar = 10;`

Correct Answer Remove Option

Add Option

Remove Question

Question 2 *

Is java similar to javascript

Option 1 *

No

Correct Answer Remove Option

Option 2 *

Yes

Correct Answer Remove Option

Option 3 *

Maybe

Correct Answer Remove Option

Option 4 *

Ok bruh

Correct Answer Remove Option

Add Option

Remove Question

Add Question

Save Quiz

Figure 5.12: CRUD Quizzes

Search Results Screen: Below is the screenshot of the Search Results Screen, displaying results from user search queries.

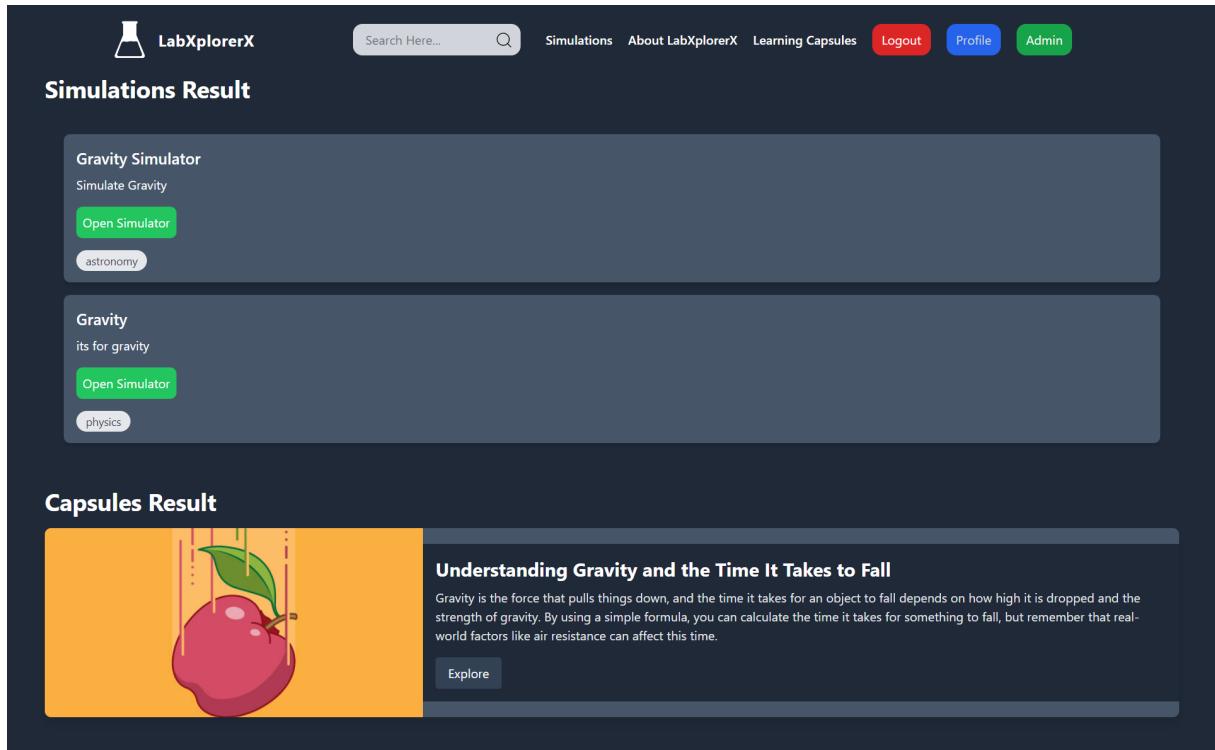


Figure 5.13: Search Results

Mail Verification Screen: Below is the screenshot of Main Verification link send to user while registering.

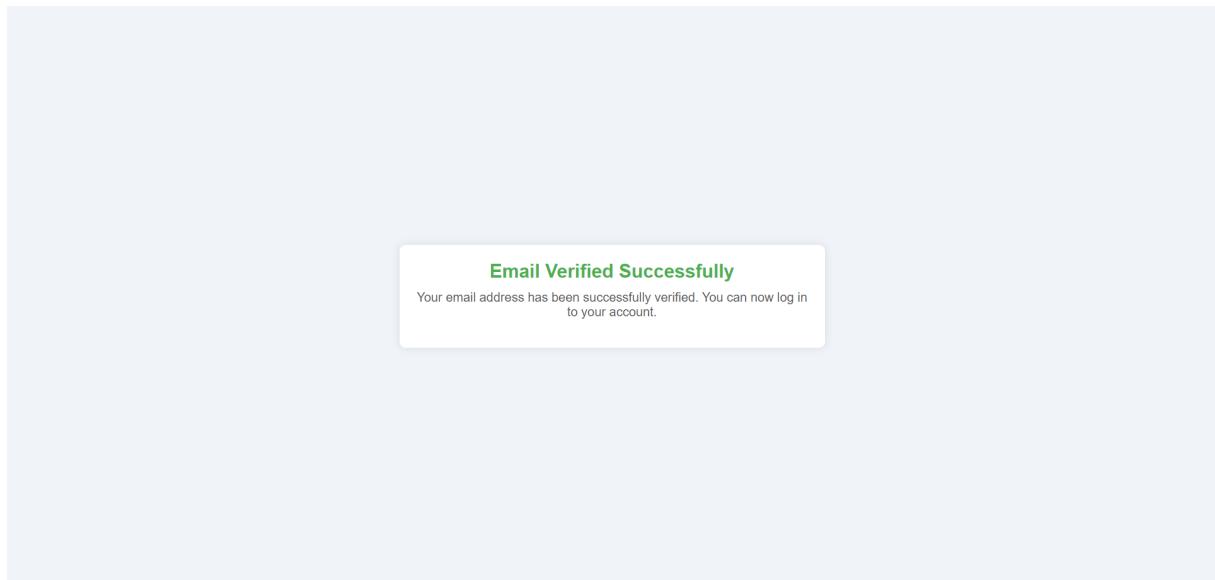


Figure 5.14: Search Results

Edit Capsule Screen: Below is the screenshot of the Edit Capsule Screen, used for modifying details of existing capsules.

 LabXplorerX

Search
Simulations
About LabXplorerX
Learning Capsules
Logout
Profile
Admin

Edit Capsule

Category *
Title *

Computer Science

What is Javascript?

Description *

JavaScript is an essential language for web development and beyond, offering a rich set of features for creating interactive, dynamic, and modern applications. It continues to evolve, with new standards and tools emerging to enhance development practices.

Simulators

Javascript Editor

Thumbnail

Choose File No file chosen

Images

Choose Files No file chosen

Include PDF Document

Choose File Lab.pdf

Lab : Use of Where clause and aggregate functions Oracle
 Objective : Learn about where clause and aggregate functions in oracle
Theory:
 A subquery is a SELECT statement nested inside another statement such as SELECT, INSERT, UPDATE, or DELETE. Typically, you can use a subquery anywhere that you use an expression.
 Initial Tables

Content *

Sans Serif H1 H2 I U B U B I U B I U

JavaScript is a versatile and widely-used programming language primarily known for its role in web development. Here's a comprehensive overview:

What is JavaScript?

Definition: JavaScript is a high-level, interpreted programming language that is primarily used to create interactive and dynamic content on websites. It allows developers to add functionality, manipulate web page content, and handle user interactions.

Key Features

- 1. **Client-Side Scripting:** JavaScript runs in the browser, enabling dynamic content updates without requiring a page reload. This includes tasks like form validation, animations, and interactive elements.
- 2. **Versatility:** Besides web development, JavaScript is also used on the server-side (with Node.js), in mobile app development (with frameworks like React Native), and even in desktop applications (with frameworks like Electron).
- 3. **Event-Driven:** JavaScript uses an event-driven model where actions (events) like clicks, keystrokes, or page loads trigger corresponding event handlers.
- 4. **Asynchronous Programming:** JavaScript supports asynchronous operations using callbacks, promises, and async/await, allowing for non-blocking operations such as API calls or reading files.

Cancel
Save

Figure 5.15: Edit Capsule

Add Capsule Screen: Below is the screenshot of the Add Capsule Screen, where administrators can add new capsules.

The screenshot shows the 'Add Capsule' page of the LabXplorerX platform. At the top, there is a navigation bar with links for 'Simulations', 'About LabXplorerX', 'Learning Capsules', 'Logout', 'Profile', and 'Admin'. The main title is 'Add Capsule'. The form fields include:

- Category ***: Computer Science
- Title ***: Learning Java Script Objects
- Description ***: We will be learning about JavaScript Objects
- Simulators**: Java, Javascript Editor
- Thumbnail**: Choose File logo.png (uploaded a yellow octagon with 'MERN')
- Images**: Choose Files (3 files uploaded: a laptop icon, a yellow hexagon with 'MERN', and a grid icon)
- Include PDF Document**: Choose File Advance java lab question.pdf (uploaded a PDF showing a simple form with fields for 'Enter Name' and 'Enter add', and a gender selection field with 'female' selected)
- Content ***: A rich text editor containing a paragraph about JavaScript objects, followed by a detailed explanation of objects, methods, inheritance, and prototypes.

At the bottom right, there are 'Cancel' and 'Save' buttons.

Figure 5.16: Add Capsule

After Addition of Capsule Screen: Below is the screenshot of the screen displayed after a new capsule has been successfully added.

The screenshot shows the LabXplorerX interface after adding a new capsule. At the top, there is a navigation bar with icons for search, simulations, about, learning capsules, logout, profile, and admin. The main content area has a dark background with a white card. The card title is "Learning Java Script Objects". Below the title is a red "Add to Favourites" button. A sub-section titled "We will be learning about JavaScript Objects" contains descriptive text and code snippets. The code examples demonstrate various operations on objects, such as creating objects with properties and methods, accessing properties, calling methods, adding new properties, modifying existing ones, and deleting properties. Below the code is a green "Open Javascript Editor" button. Further down, there is a section for uploaded documents with a "Check out document uploaded by author" link and an "Open PDF" button. The bottom part of the card features three decorative icons: a laptop, a yellow hexagon labeled "MERN", and a cluster of four overlapping squares containing icons for Node.js, MongoDB, Express.js, and React.js. At the very bottom, there is a "Comments" section with a "Latest" tab selected, a text input field for comments, and a "Submit" button.

Figure 5.17: After Addition of Capsule

Gravity Simulator: Below is the screenshot of the Gravity Simulator, demonstrating the principles of gravity.

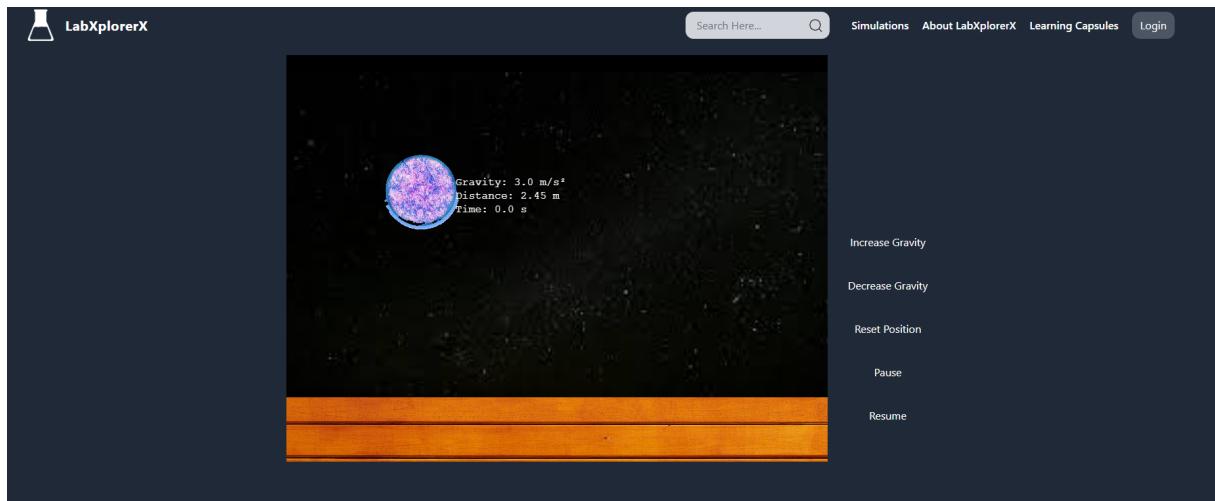


Figure 5.18: Gravity Simulator

Ohm's Law Simulator: Below is the screenshot of the Ohm's Law Simulator, illustrating Ohm's Law in an interactive format.

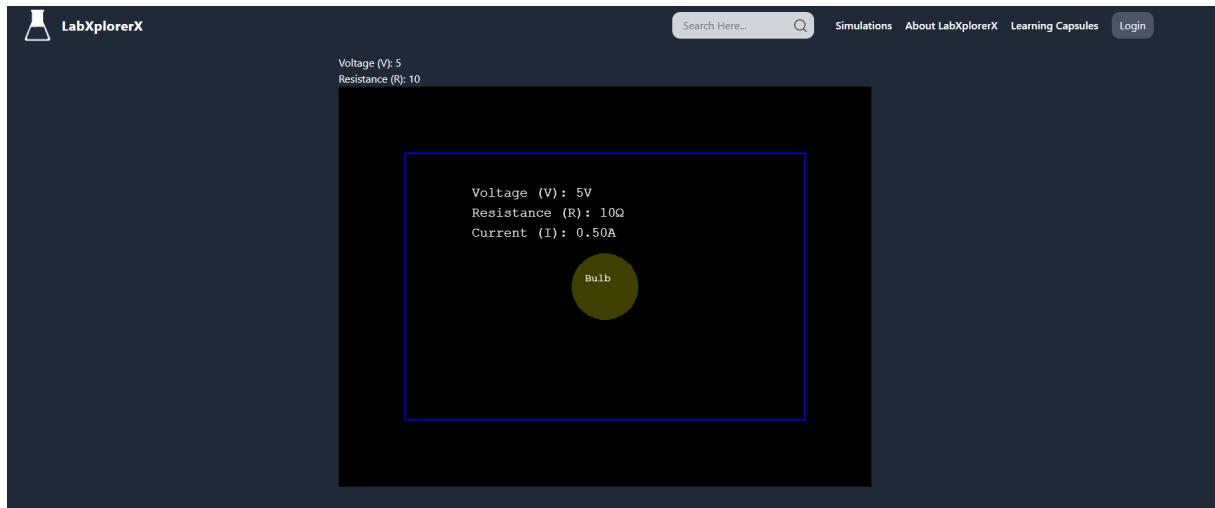


Figure 5.19: Ohm's Law Simulator

Atom Simulator: Below is the screenshot of the Atom Simulator, allowing users to explore atomic structures.

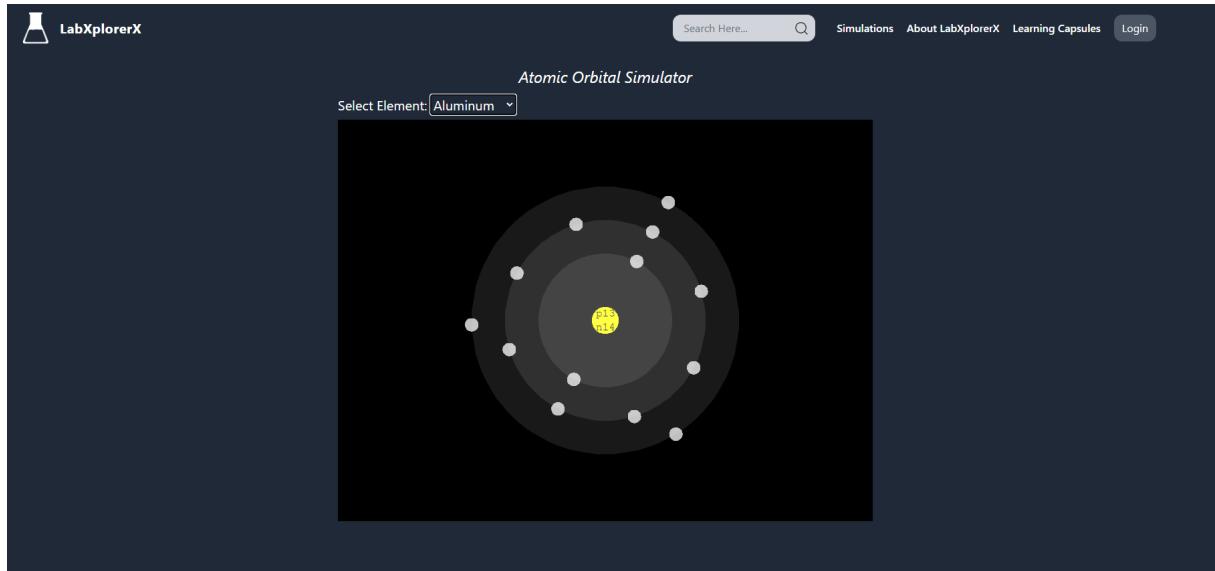


Figure 5.20: Atom Simulator

Solar System Simulator: Below is the screenshot of the Solar System Simulator, providing an interactive view of the solar system.

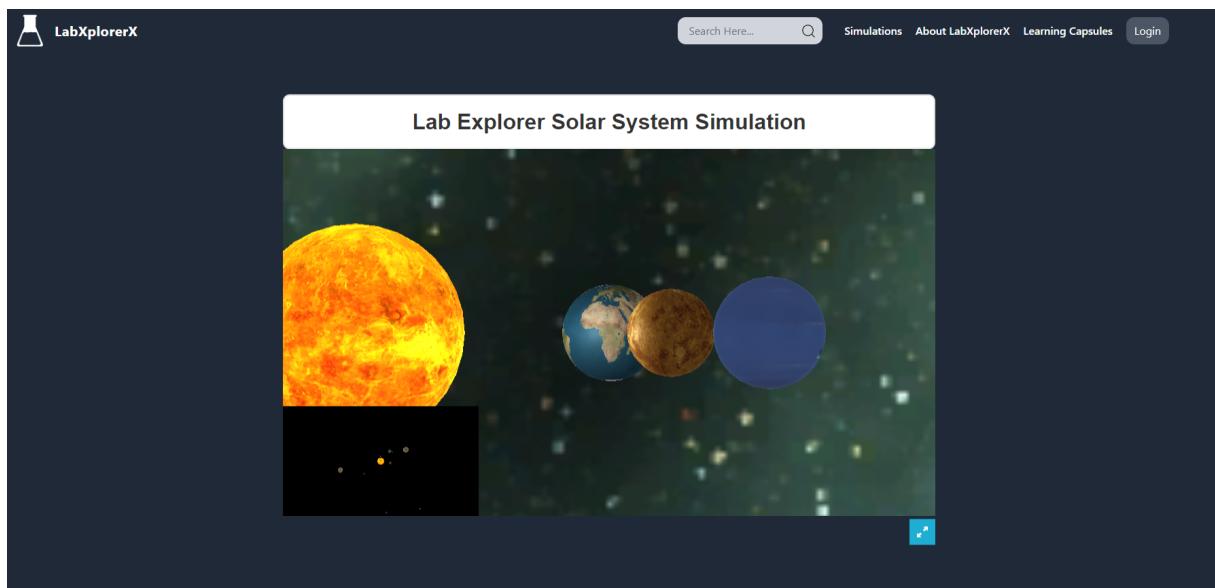


Figure 5.21: Solar System Simulator

JavaScript Editor: Below is the screenshot of the JavaScript Editor, where users can write and test JavaScript code.

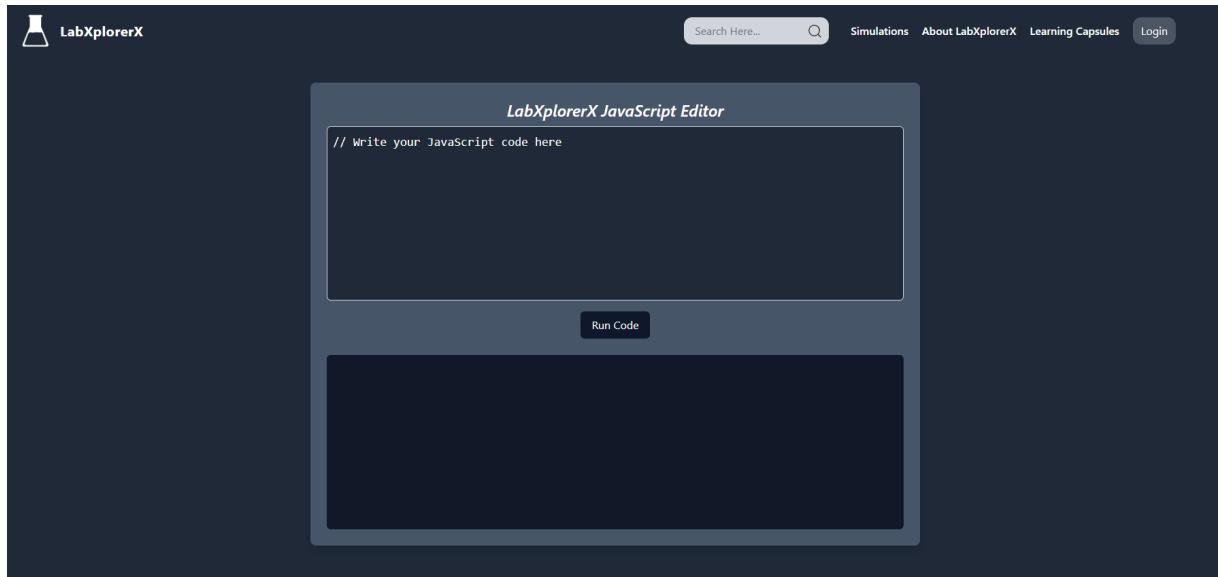


Figure 5.22: JavaScript Editor

5.2 Future Recommendations

While the implementation of LabXplorerX is complete, several enhancements could further improve the platform:

- **Additional Simulations:** Expand the range of simulations by developing new ones. This will provide a broader scope of interactive learning experiences and cover more subjects.
- **Responsive Design for Simulations:** Enhance the responsiveness of existing simulations to ensure optimal performance across different devices and screen sizes. This improvement will make the simulations more accessible and user-friendly on mobile and tablet devices.
- **Extended Profile Functions:** Introduce additional features to user profiles, such as advanced progress tracking, customizable learning paths, and enhanced account management options. This will further personalize and improve the learning experience for students.

APPENDIX A

A.1 Project Schedule

Below is the Gantt chart for the project schedule. Specific tasks are planned to be performed within the designated time frames as illustrated. This chart provides a visual representation of the project's timeline, highlighting the start and end dates for each task, as well as their dependencies. By following this schedule, the project team can effectively manage resources, track progress, and ensure timely completion of each phase.

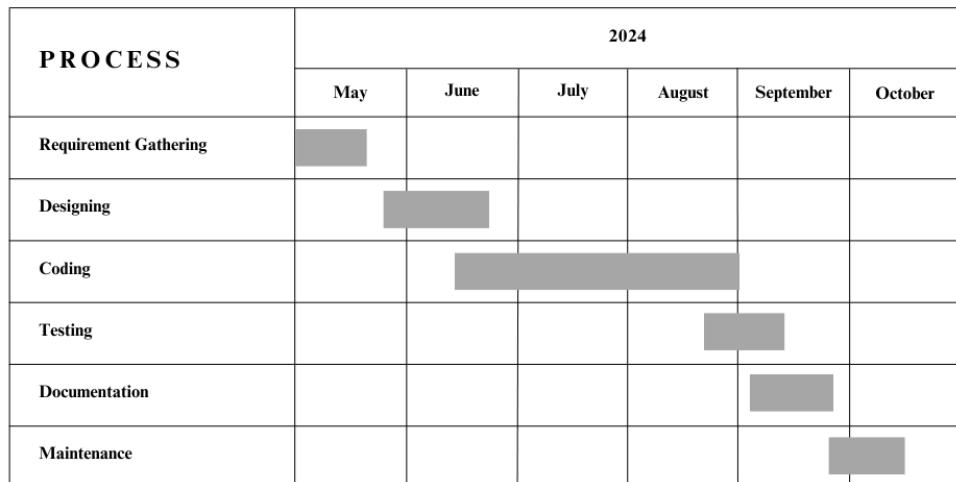


Figure A.23: Gantt Chart of Schedule

A.2 Running the Project

- Cloning the Repository

Step 1: Open a Terminal or Command Prompt.

Step 2: Navigate to the Directory Where You Want to Clone the Repository.

```
cd path/to/your/directory
```

Step 3: Clone the Repository.

```
git clone https://github.com/sushantbramhacharya/  
LabXplorer_Proj.git
```

Step 4: Navigate into the Cloned Repository.

```
cd LabXplorer_Proj
```

- **Restoring the SQL File in pgAdmin**

Step 1: Open pgAdmin.

Step 2: Connect to Your PostgreSQL Server.

- If you haven't already set up a server connection, add a new connection using your server credentials.

Step 3: Create a New Database (If Needed).

- Right-click on the “Databases” node and select “Create ↗ Database...”
- Enter a name for your database and click “Save.”

Step 4: Restore the Database from the SQL File.

- Right-click on your database and select “Restore.”
- In the “Restore” dialog, select the “File” tab.
- Click the “...” button to browse for the SQL file you want to restore.
- Select the SQL file and click “Restore.”

Step 5: Verify the Restoration.

- Expand the “Schemas” and “Tables” nodes to ensure that your database schema and tables have been correctly restored.

- **Running the Project**

Step 1: Ensure Node.js and npm Are Installed.

- Check if Node.js and npm are installed by running:

```
node -v  
npm -v
```

- If not installed, download and install Node.js from <https://nodejs.org>.

Step 2: Install Project Dependencies.

```
npm install
```

Step 3: Start the Development Server.

```
npm run dev
```

Step 4: Access the Application.

- Open your web browser and go to <http://localhost:5173> (or the port specified in your project configuration).

A.3 Basic Scene in Phaser.js

Below is a simple example of creating a basic scene using Phaser.js. This scene displays a star image and some text, and logs the pointer coordinates when clicked.

```
const config = {
    type: Phaser.AUTO,
    width: 800,
    height: 600,
    backgroundColor: '#2d2d2d',
    scene: {
        preload: preload,
        create: create,
        update: update
    }
};

const game = new Phaser.Game(config);

function preload() {
    this.load.image('star', 'https://labs.phaser.io/assets/sprites/star.png');
}
```

```

function create() {
    this.add.image(400, 300, 'star');

    this.add.text(10, 10, 'Hello Phaser!',
    { font: '24px Arial', fill: '#ffffff' });

    this.input.on('pointerdown', function (pointer) {
        console.log('Pointer down at', pointer.x,
        pointer.y);
    });
}

function update() {
    // Game loop logic
}

```

A.4 Postgres Database Tables

	<u>id [PK] integer</u>	<u>username character varying (50)</u>	<u>email character varying (100)</u>	<u>password text</u>	<u>created_at timestamp without time zone</u>	<u>updated_at timestamp without time zone</u>	<u>email_verified boolean</u>	<u>email_verification_token character varying (255)</u>
1	13	deep	deep@deep.com	\$2b\$10\$TG...	2024-07-23 21:40:52.551439	2024-07-23 21:40:52.551439	true	[null]
2	15	Admin	admin@labxplorerv.com	\$2b\$10\$O3...	2024-08-07 16:06:11.700329	2024-08-07 16:06:11.700329	true	[null]
3	31	sushant	madin66280@asaud.com	\$2b\$10\$5a...	2024-09-14 11:37:04.104393	2024-09-14 11:37:04.104393	true	[null]

Figure A.24: Table of User

	<u>id [PK] integer</u>	<u>title character varying</u>	<u>description text</u>	<u>content text</u>	<u>thumbnail character varying (255)</u>	<u>images text[]</u>	<u>pdf character varying</u>	<u>category character varying (50)</u>	<u>author_id integer</u>	<u>created_at timestamp with time zone</u>	<u>updated_at timestamp with time zone</u>
1	6	Atom test	An atom is t...	<p>Struc...	uploads\17231...	{uploads\1723131479...	uploads\...	Chemistry	15	2024-08-08 21:2...	2024-08-08 21...
2	7	What is J...	JavaScript i...	<p>JavaScript i...	uploads\17231...	{uploads\1723131928...		Cs	15	2024-08-08 21:3...	2024-08-08 21...
3	8	Understan...	Ohm's Law i...	<p>Ohm'	uploads\17231...	{uploads\1723132242...		Physics	15	2024-08-08 21:3...	2024-08-08 21...
4	9	Understan...	Gravity is th...	<h3>Understandi...	uploads\17231...	{uploads\1723132504...		Physics	15	2024-08-08 21:4...	2024-08-08 21...
5	10	Introducti...	The solar sy...	<p>The solar syst...	uploads\17231...	{uploads\1723136808...		Astronomy	15	2024-08-08 22:5...	2024-08-08 22...
6	11	Learning ...	We will be le...	<p>const arr[]=[1,...	uploads\17232...	{}		Cs	15	2024-08-09 22:2...	2024-08-09 22...
7	25	Learning ...	We will be le...	<p>JavaScript ob...	uploads\17262...	{uploads\1726246274...	uploads\...	Cs	15	2024-09-13 22:3...	2024-09-13 22...

Figure A.25: Table of Capsule

	<u>capsule_id [PK] integer</u>	<u>simulation_id [PK] integer</u>
	1	25

Figure A.26: Table of Capsule Simulations

	<u>comment_id [PK] integer</u>	<u>capsule_id integer</u>	<u>user_id integer</u>	<u>comment_text text</u>	<u>created_at timestamp without time zone</u>	<u>updated_at timestamp without time zone</u>
1	12	8	16	Hello World	2024-08-26 11:31:37.039652	2024-08-26 11:31:37.039652
2	13	6	16	How Can that be done	2024-08-26 11:31:51.737553	2024-08-26 11:31:51.737553
3	15	8	16	Nice	2024-08-26 18:18:00.020979	2024-08-26 18:18:00.020979
4	16	7	15	Great Capsule to learn	2024-09-12 12:08:51.356486	2024-09-12 12:08:51.356486

Figure A.27: Table of Comments

	user_id [PK] integer	capsule_id [PK] integer
1	15	7
2	15	8
3	16	8
4	31	8

Figure A.28: Table of Favourites

	id [PK] integer	quiz_id integer	option_text text	is_correct boolean
1		8	3 Ohms Law states that $V = IR$	true
2		9	3 Ohms Law states that $P = VI$	false
3		10	3 Ohms Law states that $V = P/I$	false
4		11	3 Ohms Law states that $R = V/P$	false
5		12	4 As resistance increases, current decreases if voltage is constant	true
6		13	4 As resistance increases, current increases if voltage is constant	false
7		14	4 As resistance decreases, current decreases if voltage is constant	false
8		15	4 As resistance decreases, current remains the same if voltage is con...	false

Figure A.29: Table of Quiz Options

	id [PK] integer	title character varying (255)	capsule_id integer
1	3	Basic Quiz on Ohms Law	8
2	4	Effects of Resistance on Current in Ohms Law	8
3	5	Gravity and Time to Fall	9
4	6	Basic Concepts of Gravity	9
5	7	What is the atomic number of Hydrogen and how many protons does it hav...	6

Figure A.30: Table of Quizes

	id [PK] integer	name character varying (255)	description text	link character varying (255)	category character varying (100)
1	4	Gravity Simulator	Simulate Gravity	http://localhost:5173/sims/gravity	astronomy
2	5	Atomic Simulator	Simulate Atomic particles	http://localhost:5173/sims/atom	chemistry
3	6	Ohms Law Simulator	Simulate ohms law	http://localhost:5173/sims/ohmslaw	physics
4	7	Javascript Editor	code javascript here	http://localhost:5173/sims/javascript	cs
5	8	Solar System Simulator	Solar System Simulator	http://localhost:5173/sims/solarsystem	physics
6	9	Gravity	its for gravity	http://localhost:5173/sims/gravity	physics

Figure A.31: Table of Simulations

A.5 Supervisor Consultation Form

Tribhuvan University
Faculty of Humanities & Social Sciences, Lalitpur Engineering College
Department of Computer Application
Student & Supervisor Consultation Form
(BCA Project-II)

Notes:

Consultation form is the "Gate Pass" to participate in presentations

At least FIVE (new) consultations (evenly distributed) before Midterm Checkpoint

At least TEN (new) consultations (evenly distributed) before FINAL Checkpoint

Project Title	LabXplorer X: Interactive learning Environment
Student Name & CRN	Sushant Bramhacharya LEC 077 BCA08
Supervisor Name	Er. Bibat Thokar

S.N.	Summary of Discussion	Date	Supervisor Signature
1	Authentication System	8/1/03/15	Thokar
2	Learning Capsules	8/1/03/13	Thokar
3	Admin Panel	8/1/04/12	Thokar
4	Creation of Simulations	8/1/04/11	Thokar
5	Quizes for capsules	8/1/04/28	Thokar
6	Creation of Gravity Simulator	8/1/04/25	Thokar
7	Creation of Atom Simulator	8/1/04/26	Thokar
8	Creation of Ohms law Circ	8/1/05/08	Thokar
9	Creation of Java Script Editor	8/1/05/09	Thokar
10	Creation of Solar System Simulator	8/1/05/10	Thokar
11	Quiz (RD) Functionality Admin	8/1/05/11	Thokar
12	User profile, Comment and Favourites	8/1/05/12	Thokar
13	Edit of Capsule and Search	8/1/05/14	Thokar
14	Registration Verification	8/1/05/28	Thokar
15	Final Touch up and Optimization	8/1/05/29	Thokar

.....

 Er. Bibat Thokar
 Program Coordinator

Figure A.32: Supervisor Consultation Form

REFERENCES

- [1] Pantina Chandrashekhar, Malini Prabhakaran, Georg Gutjahr, Raghu Raman, and Prema Nedungadi. Teacher perception of olabs pedagogy. In *Fourth International Congress on Information and Communication Technology: ICICT 2019, London, Volume 2*, pages 419–426. Springer, 2020.
- [2] Clive Thompson. How khan academy is changing the rules of education. *Wired Magazine*, 126:1–5, 2011.
- [3] Katherine Perkins, Wendy Adams, Michael Dubson, Noah Finkelstein, Sam Reid, Carl Wieman, and Ron LeMaster. Phet: Interactive simulations for teaching and learning physics. *The Physics Teacher*, 44(1):18–23, 2006.
- [4] Travis Faas. *An introduction to HTML5 game development with Phaser.js*. AK Peters/CRC Press, 2017.
- [5] Jonathan Smith and Lisa Brown. The role of digital simulations in stem education. *Journal of Educational Technology*, 15(3):45–59, 2020.
- [6] Alan Johnson. *Unity 3D Game Development by Example*. Packt Publishing, 2021.