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

Supervisor's Recommendation

The undersigned certify that he have read and recommend to the Department of Computer Application for acceptance, a 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 in Computer Application**".

Project Supervisor

Er. Bibat Thokar

BCA Coordinator, Lecturer

Department of Computer Application , Lalitpur Engineering College

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	..... External Examiner

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**

SUPERVISOR'S RECOMMENDATION ..... **iv**

LETTER OF APPROVAL ..... **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 : DFD 14
3.3.2 Data Modelling(ER-Diagram) 16
3.4 Structured System Design 18
3.4.1 Architecture Design 18
3.4.2 Database Schema Design 19
3.4.3 Interface Design 20
3.4.4 Physical DFD 22
3.5 Algorithms 24
3.5.1 Simple Random Sampling Algorithm 24
4 IMPLEMENTATION 25
4.1 Tools Used 25
4.1.1 Implementation Details of Modules 28
4.1.2 Unit Testing Test Cases 32
4.1.3 Test Cases for System Testing 32
5 CONCLUSION AND ANALYSIS 35
5.1 Conclusion 35
5.1.1 Screenshots of Outcomes 37
5.2 Future Recommendations 50
APPENDIX A
A.1 Project Schedule 51
A.2 Running the Project 51
A.3 Basic Scene in Phaser.js 53
A.4 Postgres Database Tables 54
A.5 Supervisor Consultation Form 56
REFERENCES 57

LIST OF FIGURES

Figure 3.1 Use Case Diagram 10
Figure 3.2 Process Model: DFD (Context Level) 14
Figure 3.3 Process Model: DFD (Level 1) 14
Figure 3.4 Process Model: DFD (Level 2) 15
Figure 3.5 ER Diagram of System Data 17
Figure 3.6 Three Tier Architecture of System 18
Figure 3.7 Schema Design 19
Figure 3.8 Physical DFD (Context Level) 22
Figure 3.9 Physical DFD (Level 1) 22
Figure 3.10 Physical DFD (Level 2) 23
Figure 4.1 API Integration Module 30
Figure 4.2 Registration Module 31
Figure 4.3 Authentication Module 31
Figure 5.1 Home Screen 37
Figure 5.2 Login 37
Figure 5.3 Register 38
Figure 5.4 About Page 38
Figure 5.5 Learning Areas 39
Figure 5.6 Simulations 39
Figure 5.7 Capsules 40
Figure 5.8 Quizzes 40
Figure 5.9 Admin Panel 41
Figure 5.10 Add Simulations 42
Figure 5.11 Profile 42
Figure 5.12 CRUD Quizzes 43
Figure 5.13 Search Results 44
Figure 5.14 Search Results 44
Figure 5.15 Edit Capsule 45
Figure 5.16 Add Capsule 46
Figure 5.17 After Addition of Capsule 47

Figure 5.18 Gravity Simulator 48
Figure 5.19 Ohm's Law Simulator 48
Figure 5.20 Atom Simulator 49
Figure 5.21 Solar System Simulator 49
Figure 5.22 JavaScript Editor 50
Figure A.23 Gantt Chart of Schedule 51
Figure A.24 Table of User 54
Figure A.25 Table of Capsule 54
Figure A.26 Table of Capsule Simulations 54
Figure A.27 Table of Comments 54
Figure A.28 Table of Favourites 55
Figure A.29 Table of Quiz Options 55
Figure A.30 Table of Quizes 55
Figure A.31 Table of Simulations 55
Figure A.32 Supervisor Consultation Form 56

LIST OF TABLES

Table 4.1 Capsule Main Functions Testing 32
Table 4.2 General System Test Cases 33
Table 4.3 Authentication Unit Test Cases 33
Table 4.4 Admin Functionalities Test Cases 34
Table 4.5 User Functionalities Test Cases 34

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

FFor 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 infromations. 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.

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 platfrom change the way our education system worka 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 Acadamy being one of the main motivation for onile 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.

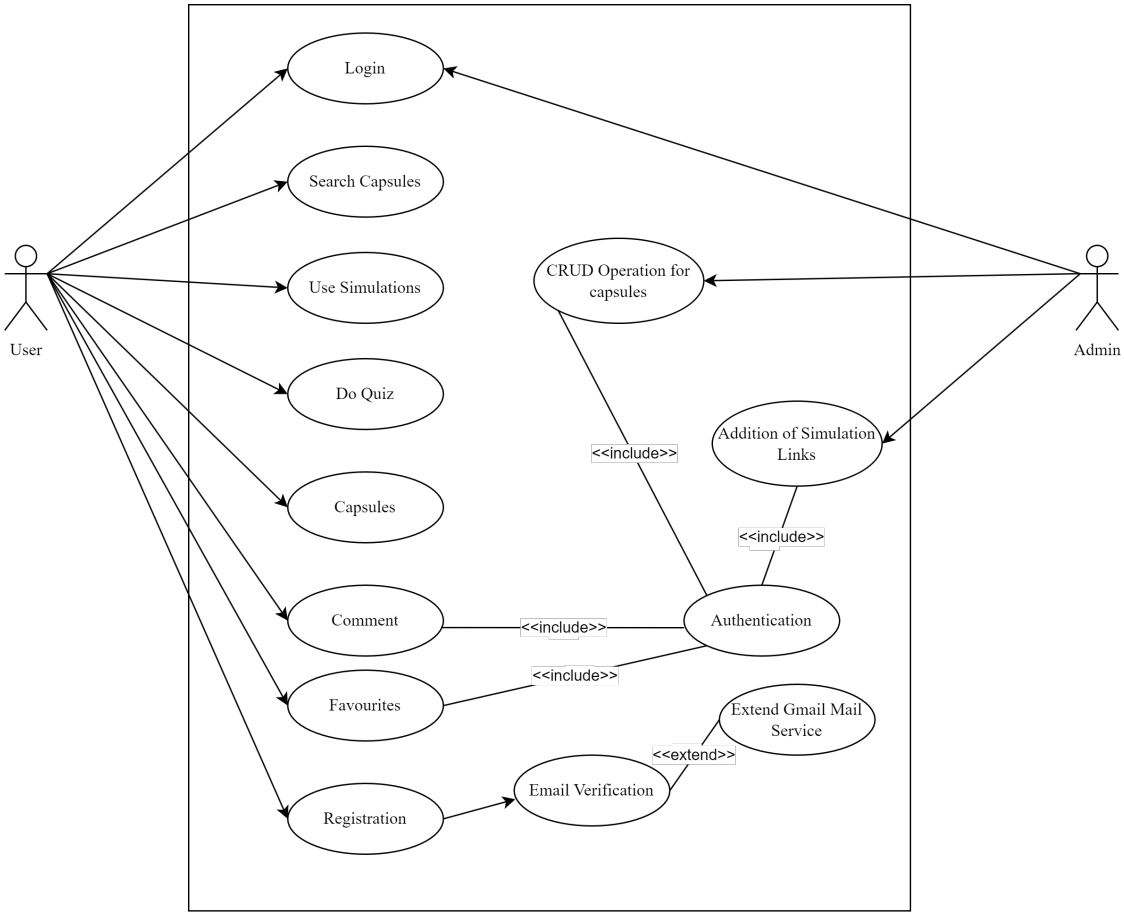


Figure 3.1: Use Case Diagram

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 : DFD

Processing Modeling visually represent the flow of processes within a system, showing how data are flowed from external entites to processes. This mainly deals with what is the system rather than how. These can be done through Data Flow Diagrams. These DFD are not the technical side rather an Representation of what the system does.

Context Level DFD

This is also known as Level 0 DFD which shows overall flow of data into system as a whole.

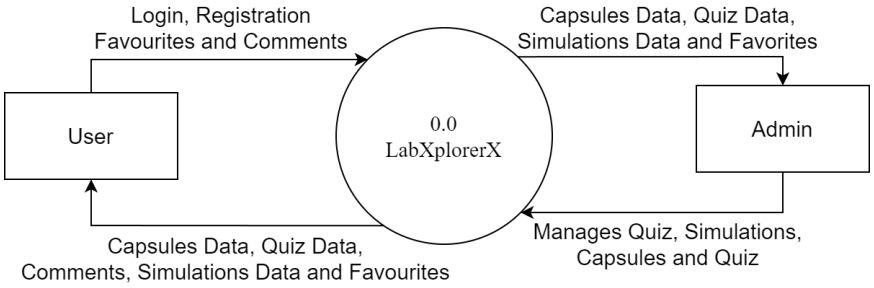


Figure 3.2: Process Model: DFD (Context Level)

Level 1 DFD

This is also known as Level 1 DFD which shows flow of data into system and its Major Processes.Here it has diffrent Process like Simulation System,User Management etc with their respective numbers 1.0, 2.0 etc.

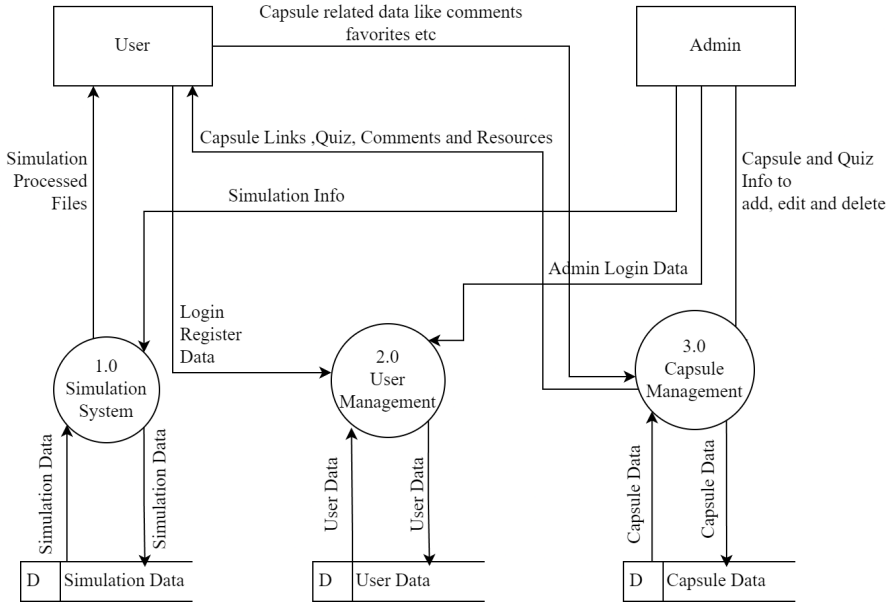


Figure 3.3: Process Model: DFD (Level 1)

Level 2 DFD

These are even more detailed structure of flow of data into processes. In Level 2, Here it has more detailed view of each Processes above. Like 1.1 View Simulation is further divided from above 1 Simulation System Process.

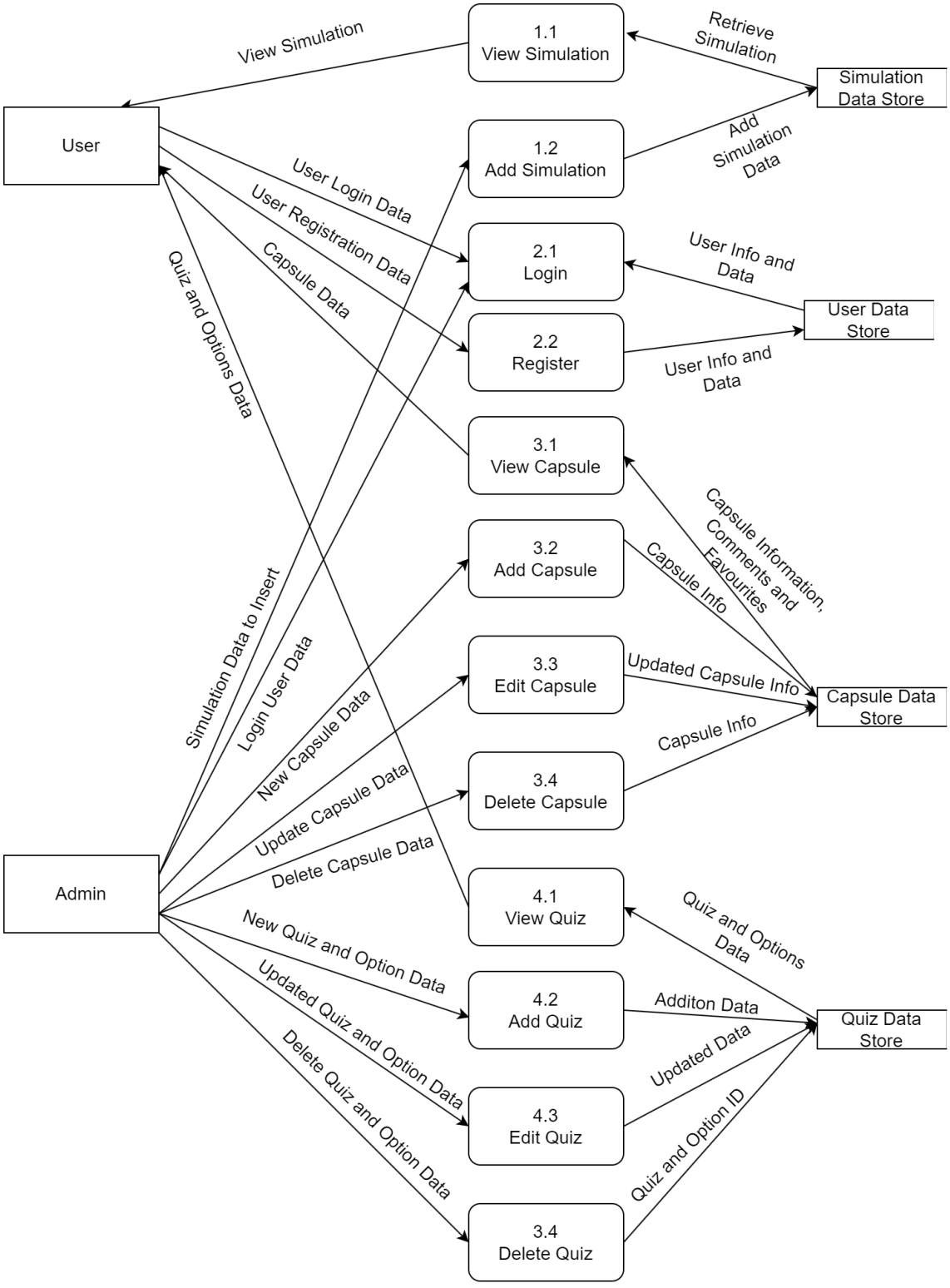


Figure 3.4: Process Model: DFD (Level 2)

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.

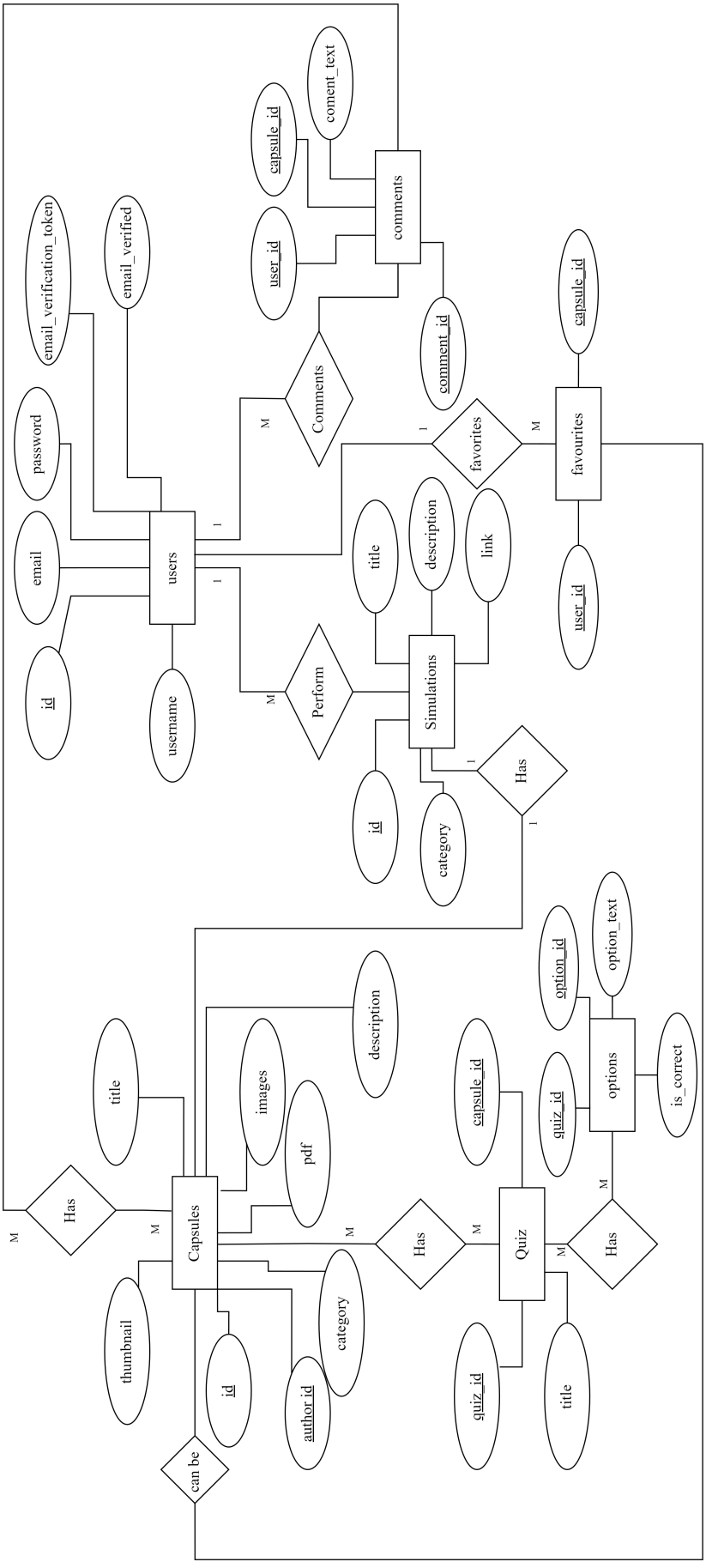


Figure 3.5: ER Diagram of System Data

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.

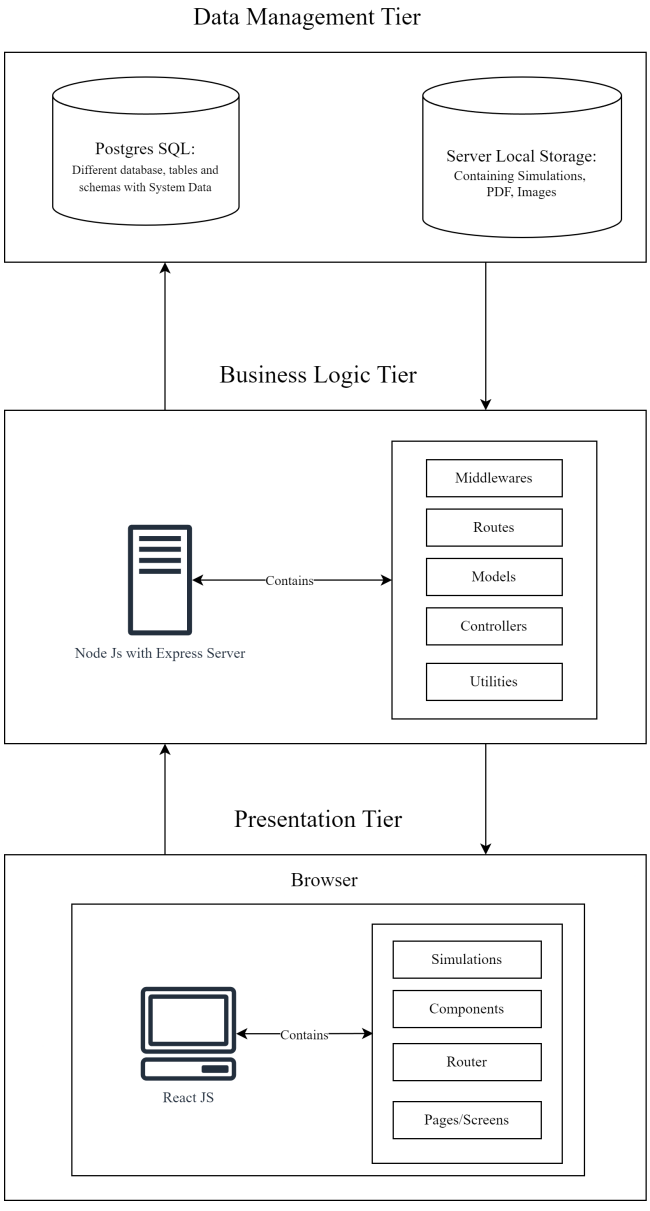


Figure 3.6: Three Tier Architecture of System

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.

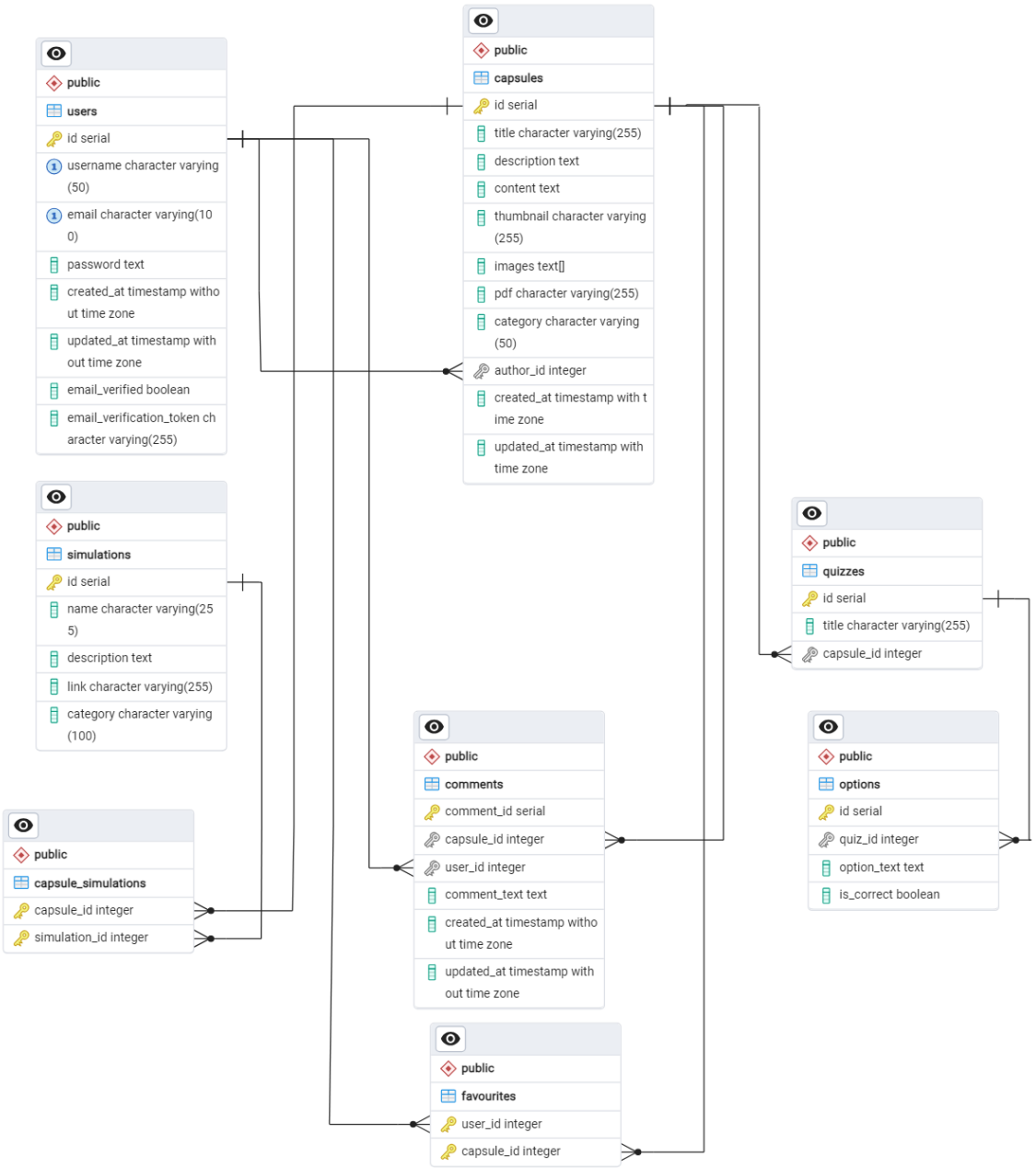


Figure 3.7: Schema Design

3.4.3 Interface Design

The interface design for this project focuses on creating a visually appealing, functional, and accessible user experience. The design includes layouts for essential screens such as the Home Screen, Menu, Capsule, Profile, and Admin Dashboard.

The main theme color chosen for the interface is Slate, a calm and professional color that provides consistency and a cohesive visual identity throughout the application. This choice ensures excellent readability and contrast with other UI elements, contributing to a clean and organized appearance.

For typography, the Poppins font is selected for its modern and clean look, which enhances readability across devices. Its uniform structure complements the minimalistic design, ensuring text remains legible and clear.

Button colors play an essential role in guiding user actions. A palette of Slate, Red, Green, and Blue is employed to differentiate between various actions. Slate is used for default or secondary actions, Red signals alerts or destructive actions, Green is for confirmation or positive actions like submissions, and Blue highlights primary actions such as saving or progressing to the next step.

Accessibility is a key consideration in the design. Buttons and interactive elements are designed to be large and clearly defined, making them easily distinguishable for users with visual impairments or motor difficulties. This promotes an inclusive experience, ensuring that all users can navigate and interact with the application comfortably.

Although the simulations are not responsive, the primary UI components have been optimized to ensure that essential elements scale properly on various devices, including desktops, tablets, and mobile screens. This responsiveness guarantees that the design maintains both usability and visual appeal across different screen sizes.

The design of buttons emphasizes usability, with large, easily clickable buttons that reduce the chance of misclicks and enhance overall user satisfaction. Their prominence ensures easy navigation, particularly on smaller devices or for users with accessibility needs.

Form design within the interface prioritizes clarity, validation, and accessibility. Each form field is accompanied by clear labels and placeholders, allowing users to understand the required input with ease. Real-time validation is implemented to provide immediate feedback, which helps users correct errors as they input data. Additionally, the form supports keyboard navigation, allowing users to move through fields efficiently using the tab key, a feature that enhances the accessibility of the form for users who rely on assistive devices.

3.4.4 Physical DFD

A Physical Data Flow Diagram (DFD) provides a detailed view of how data flows through a system, focusing on physical components and their interactions. It illustrates the actual hardware, software, and network elements involved in processing and storing data. Unlike logical DFDs, which emphasize the functions and data flows abstractly, physical DFDs depict the real-world infrastructure, including servers, databases, and user interfaces.
Below is the Context Level DFD.

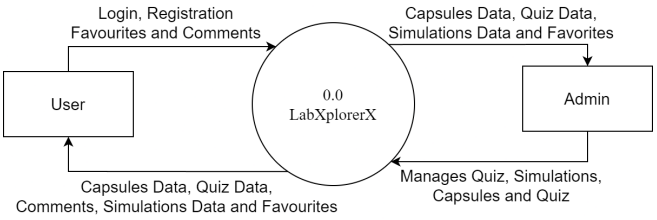


Figure 3.8: Physical DFD (Context Level)

Below is the Level 1 DFD. Which shows detailed implementation of above Context Level DFD.

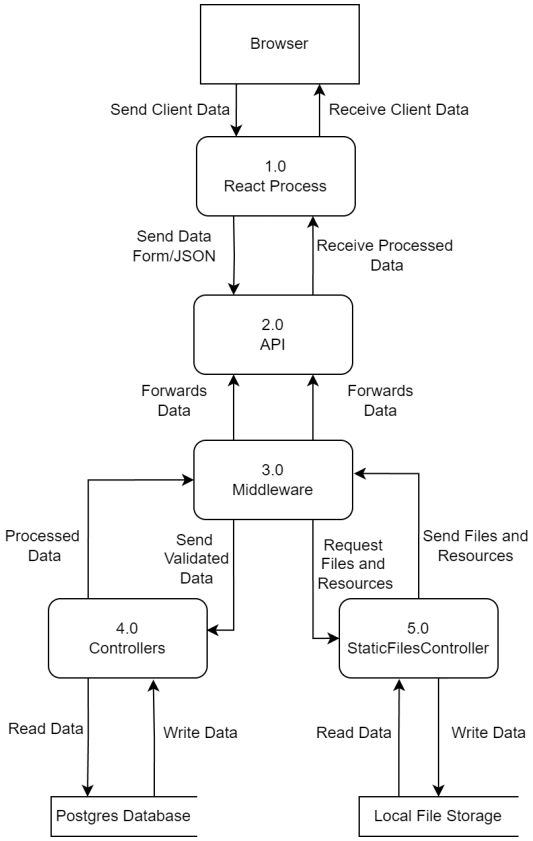


Figure 3.9: Physical DFD (Level 1)

Below is the Level 2 DFD. Which shows us More Detailed implementation of Above processes and whats behind those processes. It shows how Frontend and Backend processes the data and data being flowed in the system.

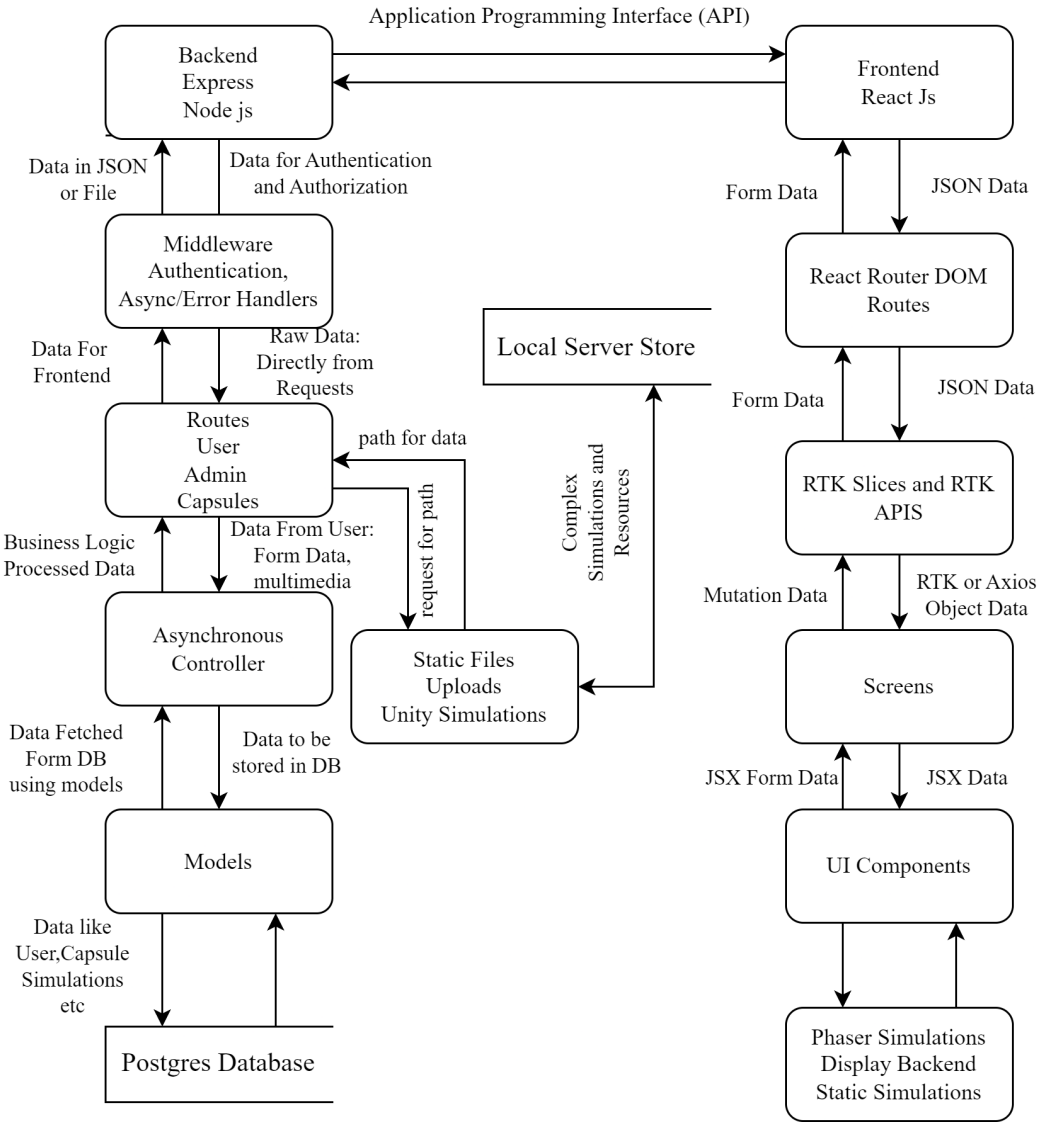


Figure 3.10: Physical DFD (Level 2)

3.5 Algorithms

3.5.1 Simple Random Sampling Algorithm

Simple Random Sampling algorithm is used in system to randomize quiz into the system. Through this algorithm we can select random set of quiz from large sample or set of data. Simple Random Sampling involves two main steps: shuffling the list and selecting the desired number of samples. Here are the details mentioned below:

Shuffling : Fisher-Yates shuffle

To achieve randomness, the list is shuffled using an algorithm such as the Fisher-Yates shuffle. This algorithm has a time complexity of $O(n)$, where n is the number of elements in the list. The Fisher-Yates shuffle works by iterating through the list once and swapping each element with another element chosen randomly from the remaining elements. This ensures each permutation of the list is equally likely.

Selection

After shuffling, the first k elements are selected where k is the number of desired samples. This step involves accessing k elements from the shuffled list, which has a time complexity of $O(k)$.

Overall Time Complexity

The total time complexity is the sum of the time complexities of shuffling and selection, which is $O(n)$ for shuffling and $O(k)$ for selection. Thus, the overall time complexity is $O(n+k)$. In practical scenarios, if k is much smaller than n, this can be simplified to $O(n)$ as the term $O(k)$ becomes negligible in comparison to $O(n)$.

Algorithm:

1. **Shuffling:**For each element in the list, from the first to the second-to-last (i.e., from index 0 to $n-2$):
    - Generate a random index j such that j is between the current index i and the last index $n-1$.
    - Swap the element at index i with the element at index j.
2. **Selection:** After shuffling, select the first k elements from the shuffled list.
3. **Output:** The selected k elements as the random sample.

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 aweb 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.

Module of AtomSimulator

The AtomSimulator module initializes the ‘selectedElement‘ with the first element from the ‘elements‘ array and sets up a Phaser game instance with appropriate dimensions and physics. It visualizes the atom by creating the nucleus and electron orbits, updates electron positions based on time, and handles user interactions to update the element and re-render the scene. The Phaser instance is destroyed upon component unmounting to free resources.

Module of CodeEditor

The CodeEditor module initializes states for ‘code‘ and ‘consoleOutput‘. It creates an iframe to execute user-provided JavaScript code and captures console messages and errors. The ‘consoleOutput‘ is updated with iframe messages, and the iframe is removed from the DOM after execution to prevent resource leaks.

Module of OhmsLawSimulator

The OhmsLawSimulator module sets default values for voltage and resistance, initializes a Phaser game instance, and creates game elements. It continuously updates the bulb brightness based on the current $I = \frac{V}{R}$, adjusts displayed values, and re-renders the simulation with user input changes. The Phaser instance is destroyed on component unmount.

Module of GravitySim

The GravitySim module sets up a Phaser game environment with gravity simulation. It creates game elements, displays initial values, and updates gravity, distance, and time during the simulation. Users can adjust gravity, reset the player, and pause or resume the simulation. The Phaser instance is destroyed on component unmount. The distance traveled is calculated using $d = \frac{1}{2}gt^2$.

Module of SolarSystemSimulator

The SolarSystemSimulator module initializes celestial bodies and the main camera. It attaches a ‘RotateAround‘ script to simulate planetary and moon orbits and a ‘FollowAtTarget‘ script to keep the camera focused on the selected object. Users can change the camera target, and the implementation involves creating bodies, configuring camera controls, and testing interactions.

Frontend API Integration Module

This diagram depicts the architecture of a web application where a React frontend communicates 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.

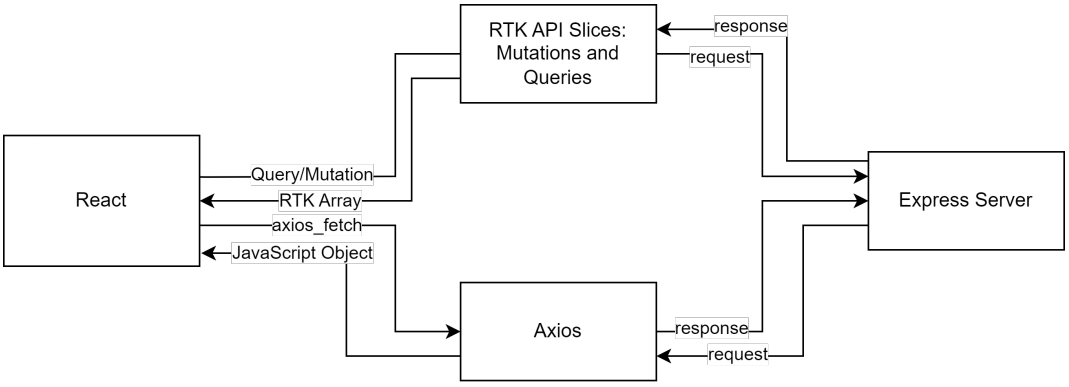


Figure 4.1: API Integration Module

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.

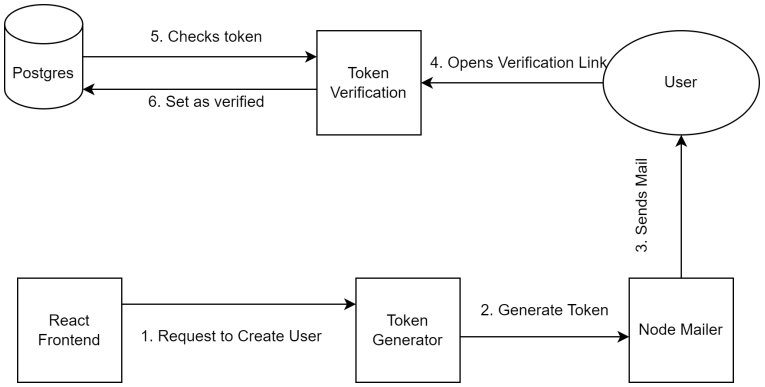


Figure 4.2: Registration Module

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.

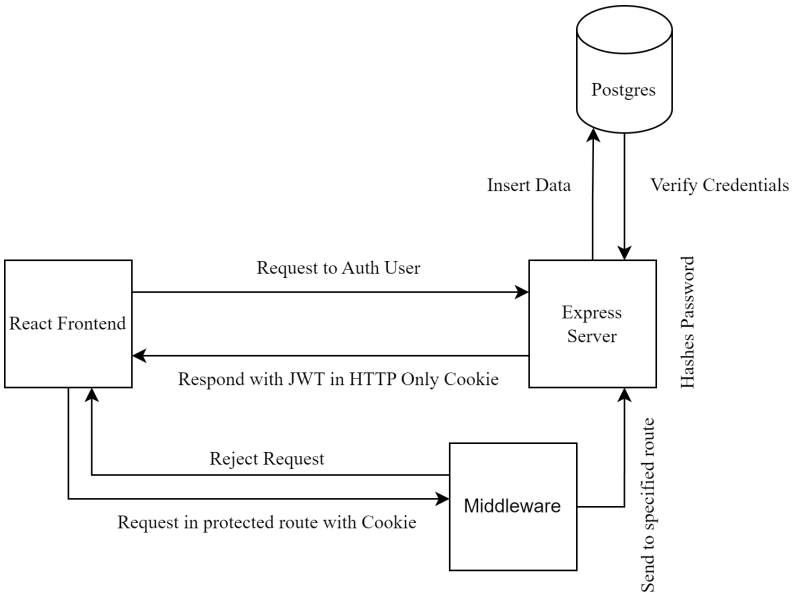


Figure 4.3: Authentication Module

4.1.2 Unit Testing Test Cases

The following tables summarize the results of the main functions tests conducted on the application. Each test case is described along with the expected and actual outcomes to verify the functionality of the application. These results ensure that the application behaves as expected across various scenarios.

Table 4.1: Capsule Main Functions Testing

Test No.	Test Case	Expected Output	Actual Output	Result
1	Insert Capsule	Capsule object with ID and details is returned, including title, description, content, and associated simulations.	Capsule object with ID and details is returned, including title, description, content, and associated simulations.	Pass
2	Read All Capsules	Array of capsule objects including `id`, `title`, `description`, and `thumbnail`, ordered by `id`.	Array of capsule objects including `id`, `title`, `description`, and `thumbnail`, ordered by `id`.	Pass
3	Read Capsule By ID	Capsule details including simulation information for the specified ID, such as title, description, content, and associated simulation details.	Capsule details including simulation information for the specified ID, such as title, description, content, and associated simulation details.	Pass
4	Update Capsule By ID	Success confirmation with updated capsule details reflecting changes in title, description, content, and linked simulations.	Success confirmation with updated capsule details reflecting changes in title, description, content, and linked simulations.	Pass
5	Delete Capsule By ID	Success confirmation indicating the capsule with the specified ID has been deleted.	Success confirmation indicating the capsule with the specified ID has been deleted.	Pass
6	Read Quiz By ID	Array of quiz details related to the specified capsule ID, including quiz questions and options.	Array of quiz details related to the specified capsule ID, including quiz questions and options.	Pass

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.

General System Test Cases

The following table summarizes the results of testing various general functionalities of the system/application. Each test case is designed to verify a specific functionality, including capsule access, category display, quiz operations, and simulator interactions. The table lists the test case number, description, input used, expected results, actual results obtained, and the outcome of the test. All tests have been successfully executed with the expected results matching the actual outcomes.

Table 4.2: General System Test Cases

Test No.	Test Case	Input	Expected Results	Actual Results	Result
1	Accessing Specific Capsules	Capsule ID: 8	Shows the whole capsule, its images, PDF, and all its meta information with quiz.	Shows the whole capsule, its images, PDF, and all its meta information with quiz.	Pass
2	Capsules By Category	Category: Physics	Shows cards, thumbnail, title, description, and buttons about related category capsules.	Shows cards, thumbnail, title, description, and buttons about related category capsules.	Pass
3	Accessing Learning Areas	Learning Area View	Shows Learning Capsules with categories of each capsule.	Shows Learning Capsules with categories of each capsule.	Pass
4	Accessing Quiz	Quiz ID: 8	Shows Quiz of Specific Capsule.	Shows Quiz of Specific Capsule.	Pass
5	Submitting Quiz	Selected Quiz	Shows how many quizzes were correct.	Shows how many quizzes were correct.	Pass
6	Accessing Ohm's Law Simulator	Simulator: Ohm's Law	Shows Ohm's law simulator with V, I, and R inputs and phaser output.	Shows Ohm's law simulator with V, I, and R inputs and phaser output.	Pass
7	Accessing PDF	Open PDF Button	Shows PDF document of specific capsule.	Shows PDF document of specific capsule.	Pass
8	Accessing Images of Capsule	Click Images Button	Shows images related to capsule.	Shows images related to capsule.	Pass
9	Searching Capsules and Simulations	Search Term: Gravity	Searches and shows capsules and simulations related to the search term.	Searches and shows capsules and simulations related to the search term.	Pass

Authentication Unit Test Cases

The following table presents the results of the authentication testing for the system/application. Each test case is aimed at verifying different authentication functionalities, including user and admin login, as well as user registration. The table provides details on the test case number, the specific test being performed, the input provided, the expected results, the actual results obtained, and the outcome of each test. All tests have been executed successfully, with results meeting the expected outcomes.

Table 4.3: Authentication Unit Test Cases

SN.	Test Case	Input	Expected Results	Actual Results	Result
1	Login User	Username: 'sushantbram' Password: 'sushant123'	Logs in the User and shows profile	Logs in the User and shows profile	Pass
2	Login Admin	Username: 'Admin' Password: 'labxplorerxadmin'	Logs in Admin and shows profile	Logs in Admin and shows profile	Pass
3	Login User with incorrect credentials	Username: 'test' Password: 'test1234'	Alerts with wrong password	Alerts with wrong password	Pass
4	Register User	Username: 'Admin' Password: 'admin'	Registers User and shows profile	Registers User and shows profile	Pass

Admin Functionalities Test Cases

The table below outlines the results of the admin functionalities testing for the system/application. This testing encompasses key administrative operations such as adding, editing, and deleting capsules, as well as managing quizzes. Each test case details the specific functionality being evaluated, the input provided, the expected results, the actual results obtained, and the final outcome of each test. All tests have been completed successfully, confirming that the admin functionalities operate as expected.

Table 4.4: Admin Functionalities Test Cases

Test No.	Test Case	Input	Expected Results	Actual Results	Result
1	Add capsule	Accessing Add Capsule Form	Shows Form to add capsules and adds capsules when submitted	Shows Form to add capsules and adds capsules when submitted	Pass
2	Edit capsule	Accessing Edit Capsule Form for ID 6	Shows Form to edit capsules and updates capsule when submitted	Shows Form to edit capsules and updates capsule when submitted	Pass
3	Delete capsule	Deleting Capsule	Deletes the selected capsule and confirms deletion	Deletes the selected capsule and confirms deletion	Pass
4	Edit Quiz	Accessing Quiz Management Form for ID 6	Shows Form to add quiz, remove quiz, and update options	Shows Form to add quiz, remove quiz, and update options	Pass

User Functionalities Test Cases

The following table presents the results of the user functionalities testing for the system/application. This testing evaluates crucial features available to users, including the ability to add items to favorites, comment on capsules, and view their profiles. Each test case details the specific functionality being tested, the input used, the expected outcomes, the actual results obtained, and the final result. All tests have been successfully passed, confirming that user functionalities are working as intended.

Table 4.5: User Functionalities Test Cases

SN.	Test Case	Input	Expected Results	Actual Results	Result
1	Favourite Function	Clicking Add to Favourites button	Adds to favourite and shows in profile	Adds to favourite and shows in profile	Pass
2	Commenting	Commenting on Capsules with ”Good Capsule”	Adds Comment and shows good capsule in both capsule and author profile	Adds Comment and shows good capsule in both capsule and author profile	Pass
3	User Profile	Viewing user profile	Shows Profile having Favourite Capsule and Comments done by User	Shows Profile having Favourite Capsule and Comments done by User	Pass

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 aweb 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, quizes 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.

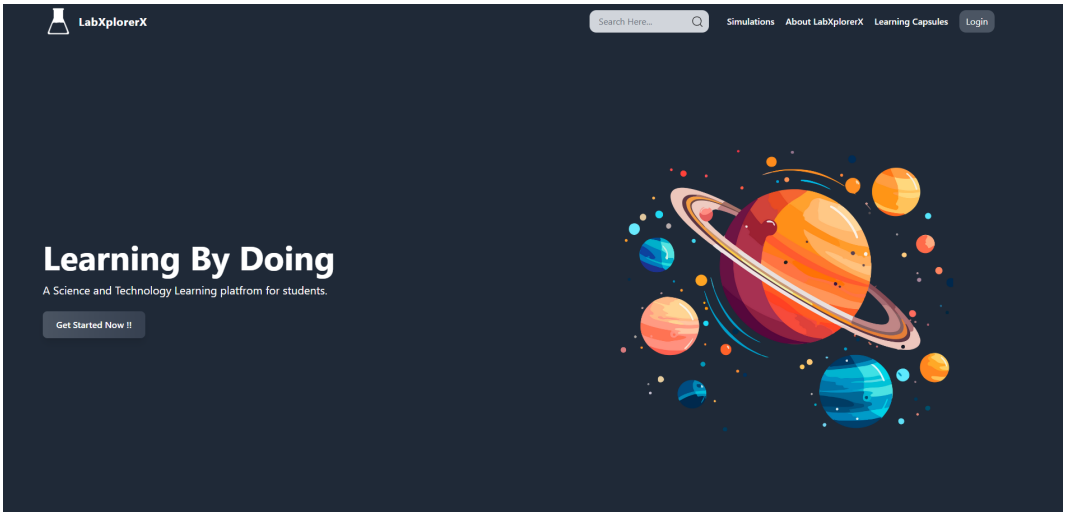


Figure 5.1: Home Screen

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

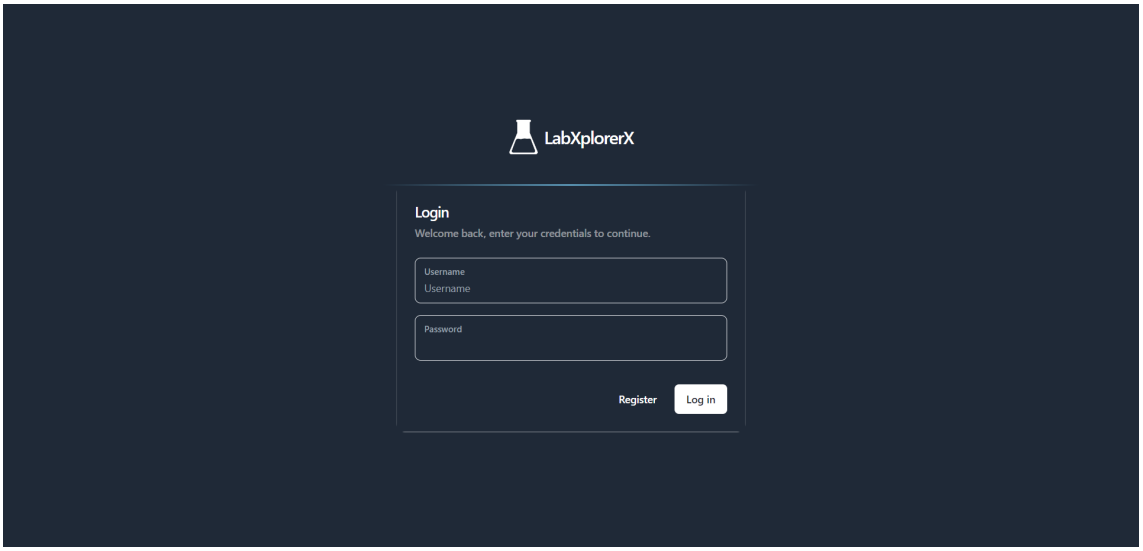


Figure 5.2: Login

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

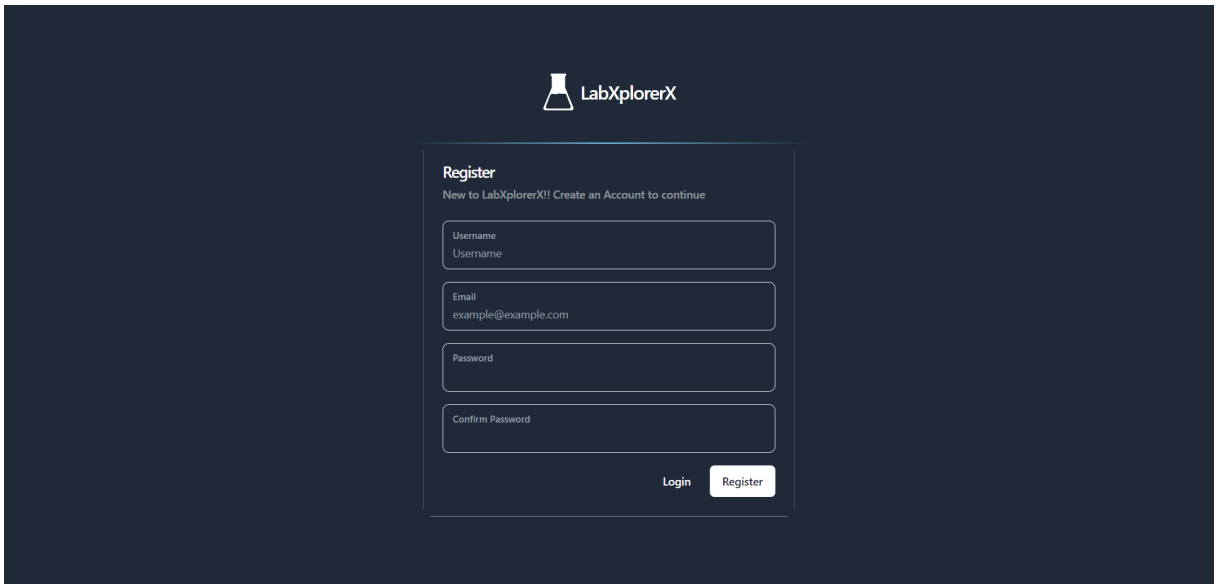


Figure 5.3: Register

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

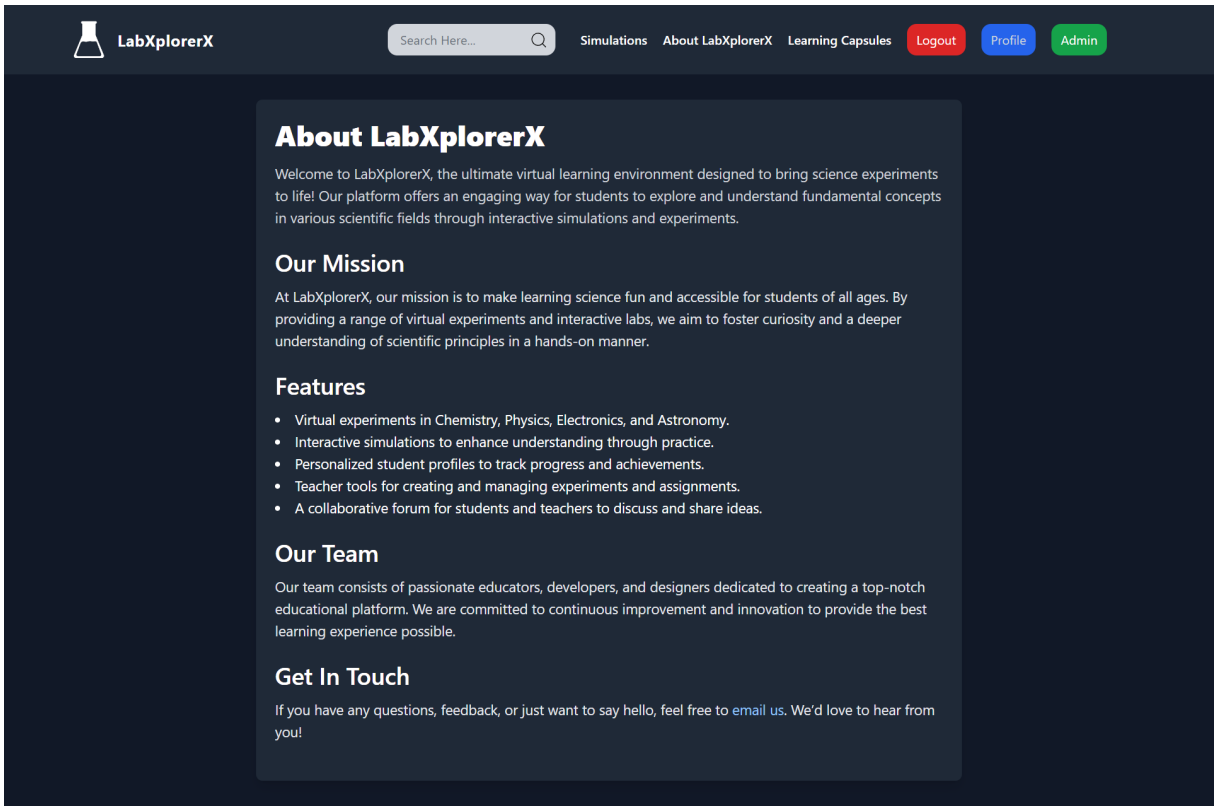


Figure 5.4: About Page

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

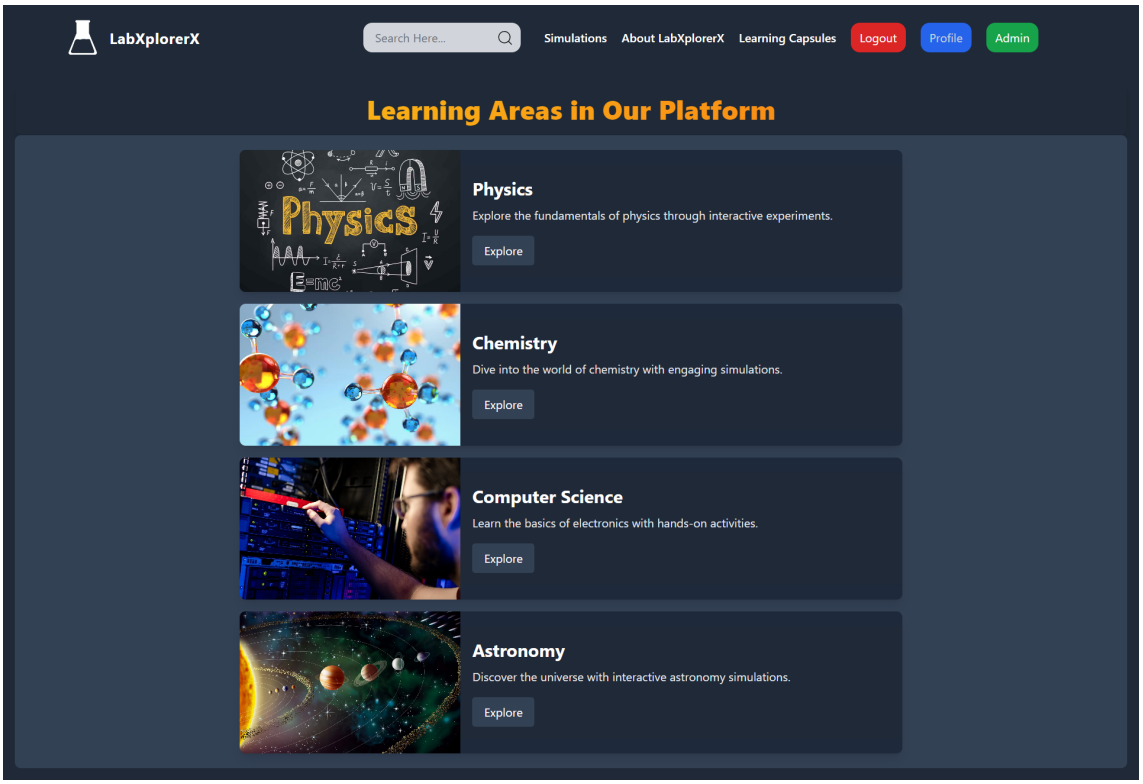


Figure 5.5: Learning Areas

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

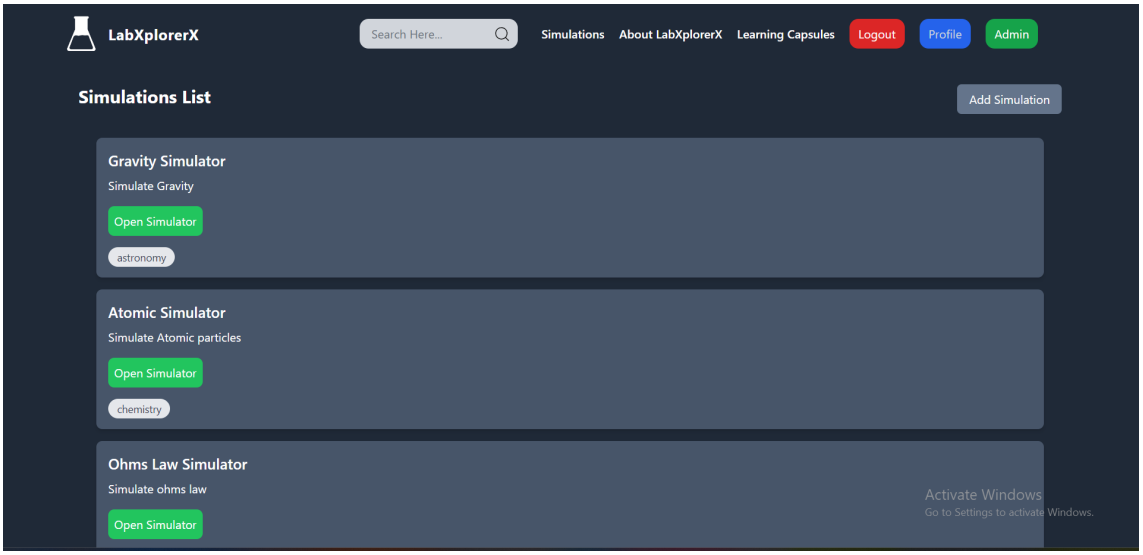


Figure 5.6: Simulations

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

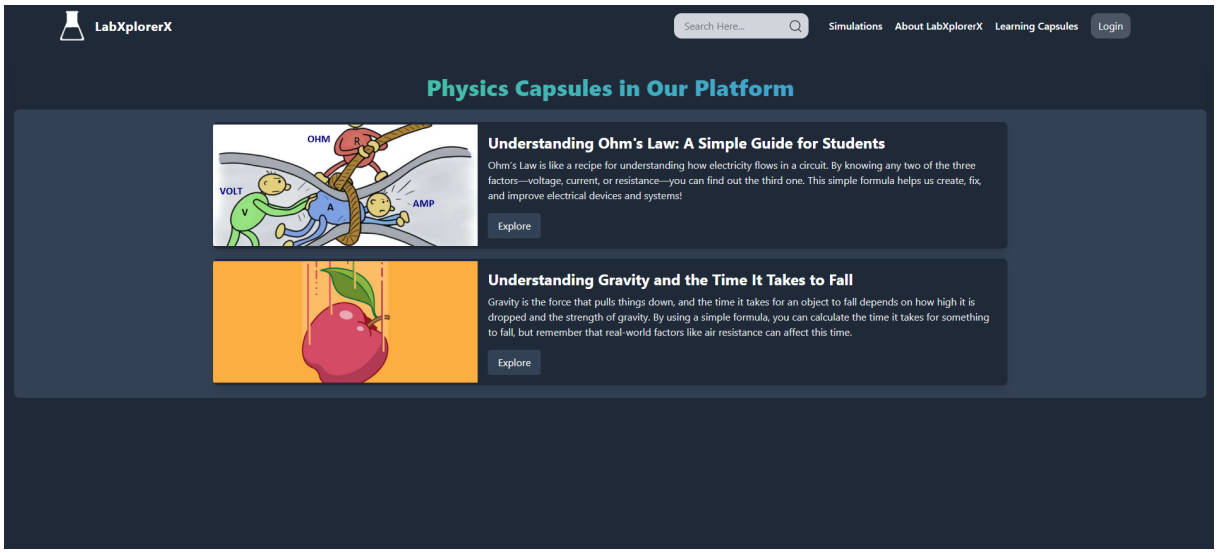


Figure 5.7: Capsules

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

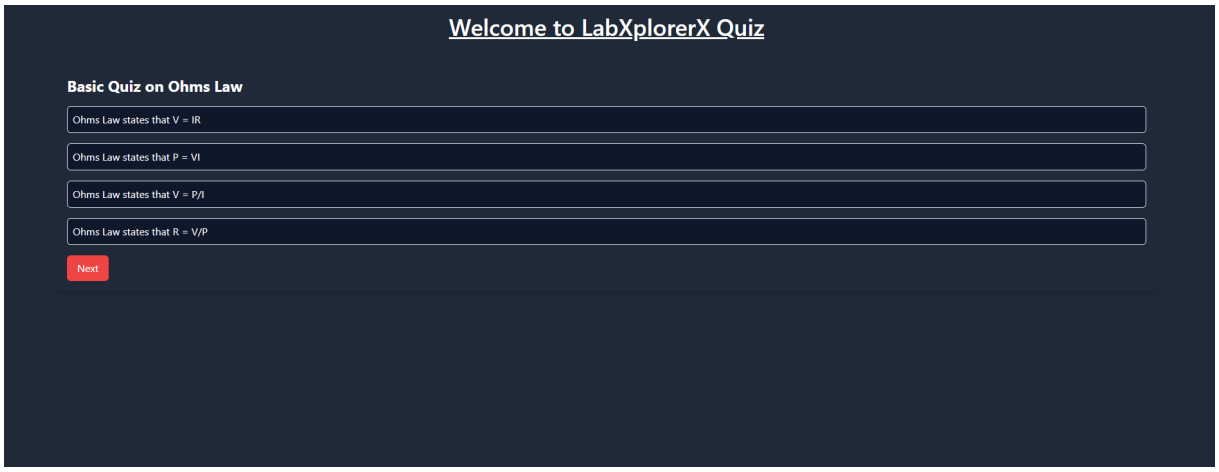


Figure 5.8: Quizzes

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

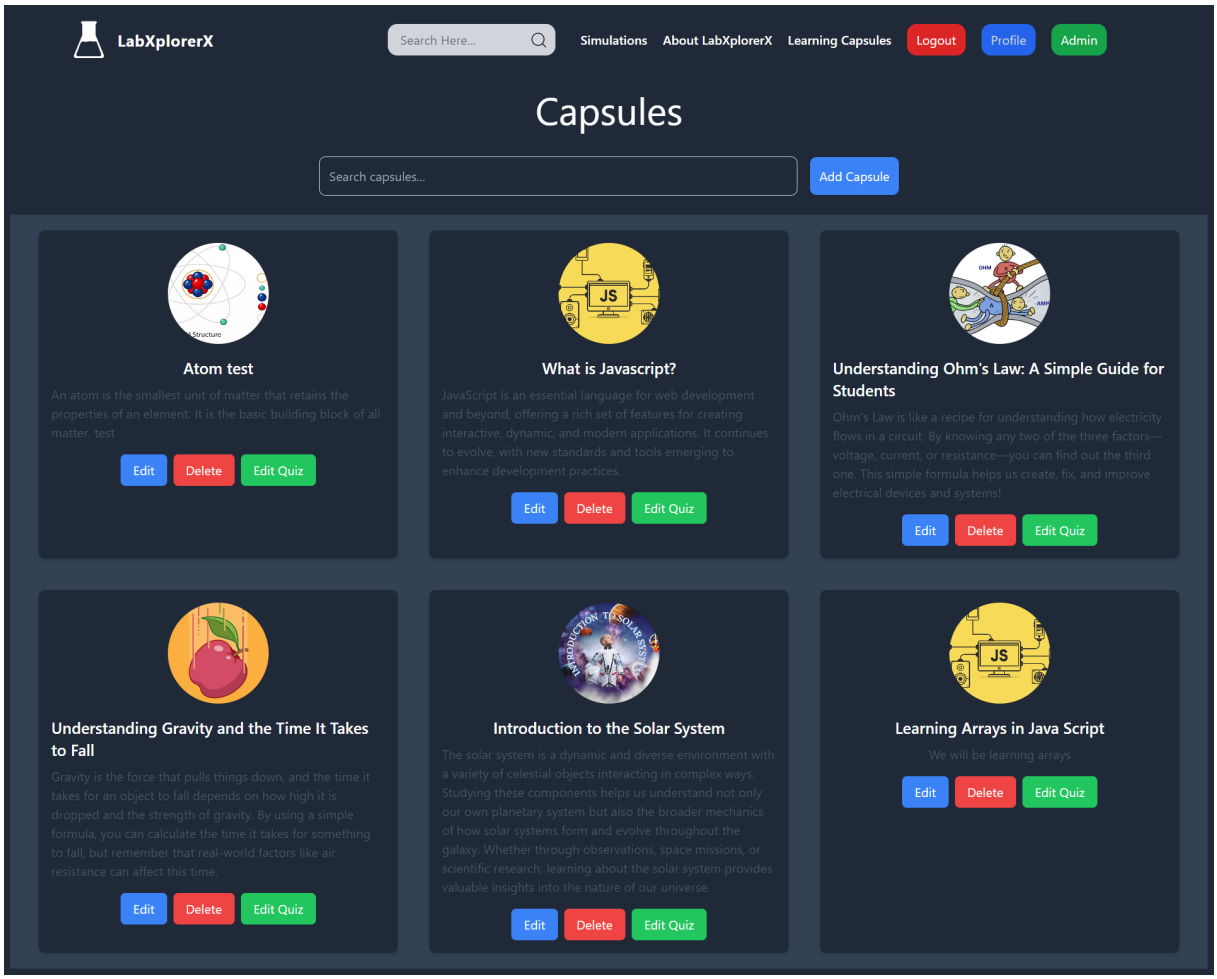


Figure 5.9: Admin Panel

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

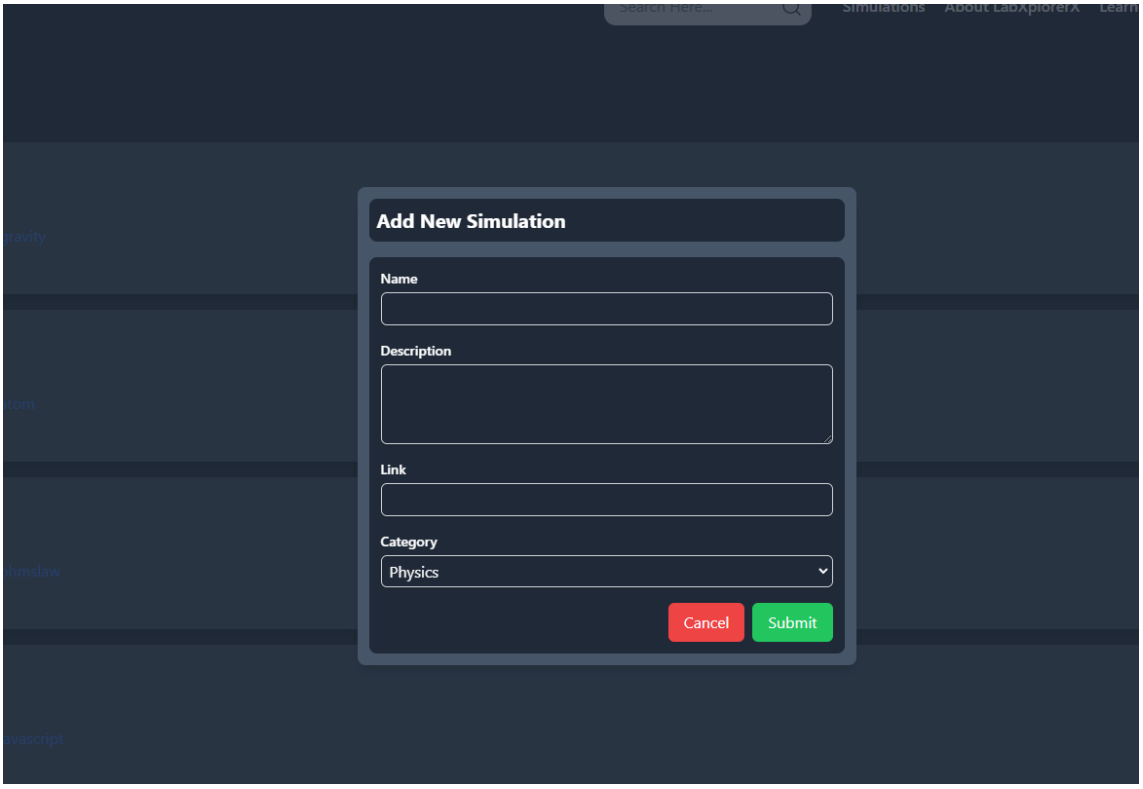


Figure 5.10: Add Simulations

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

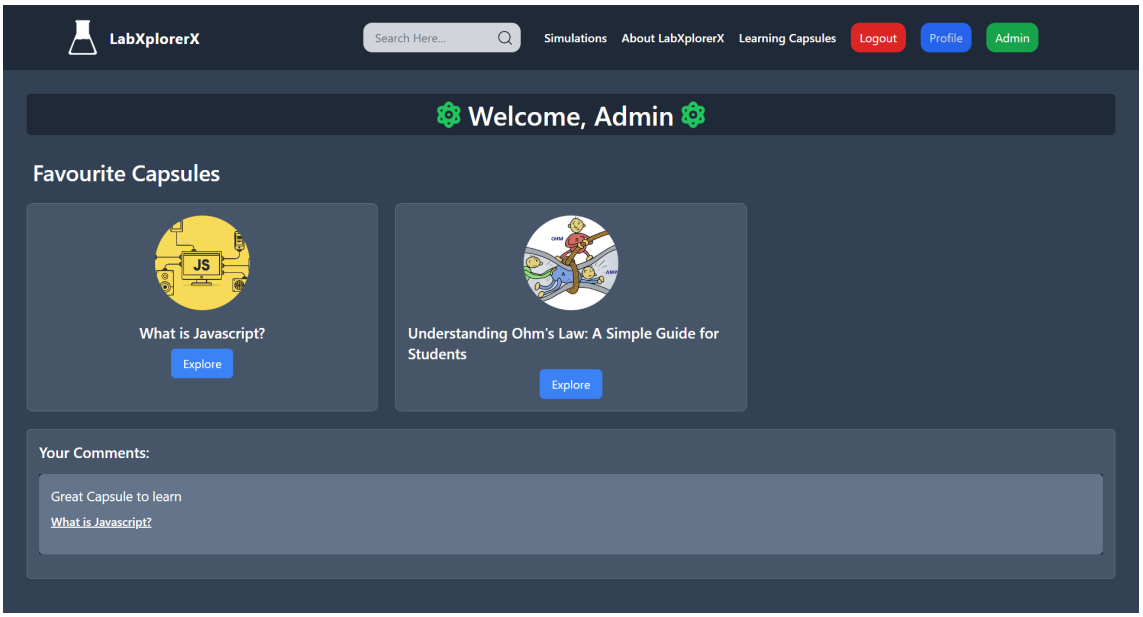


Figure 5.11: Profile

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

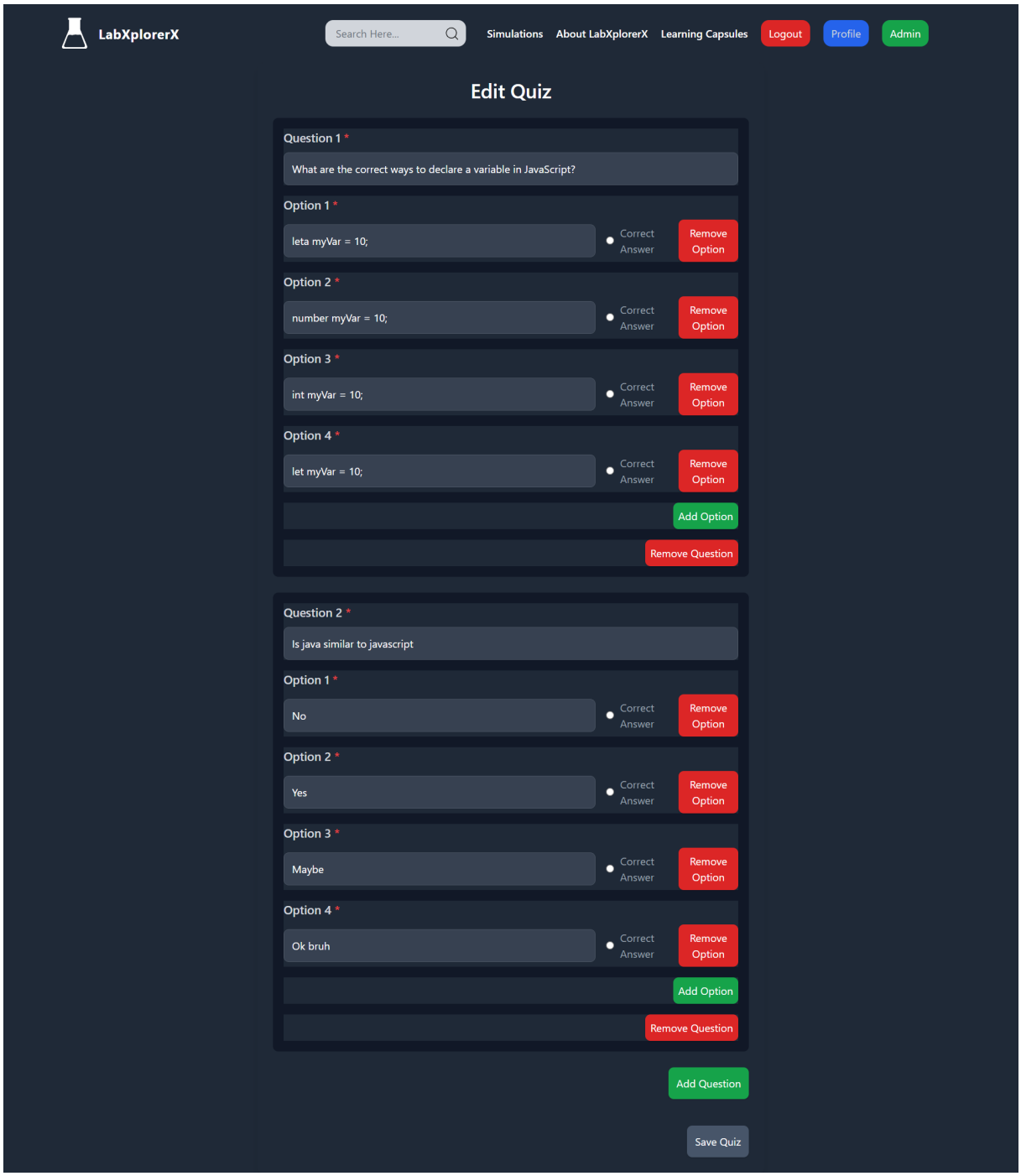


Figure 5.12: CRUD Quizzes

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

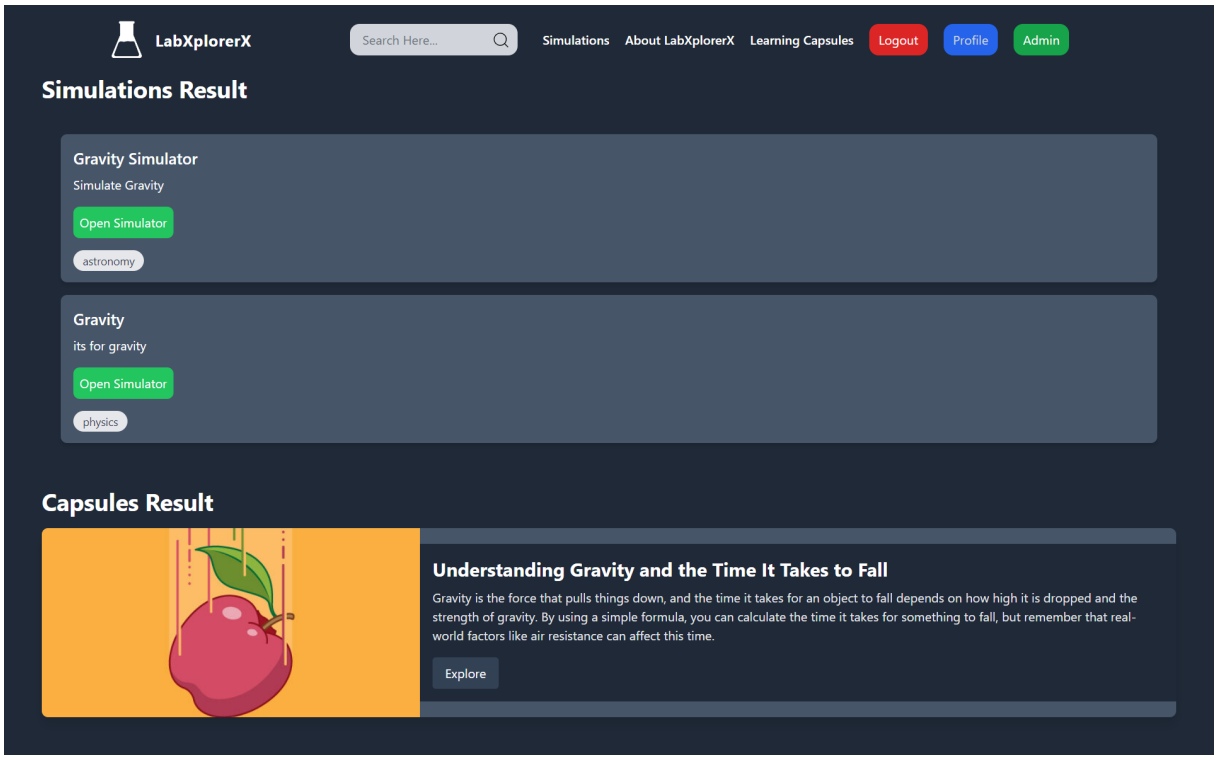


Figure 5.13: Search Results

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

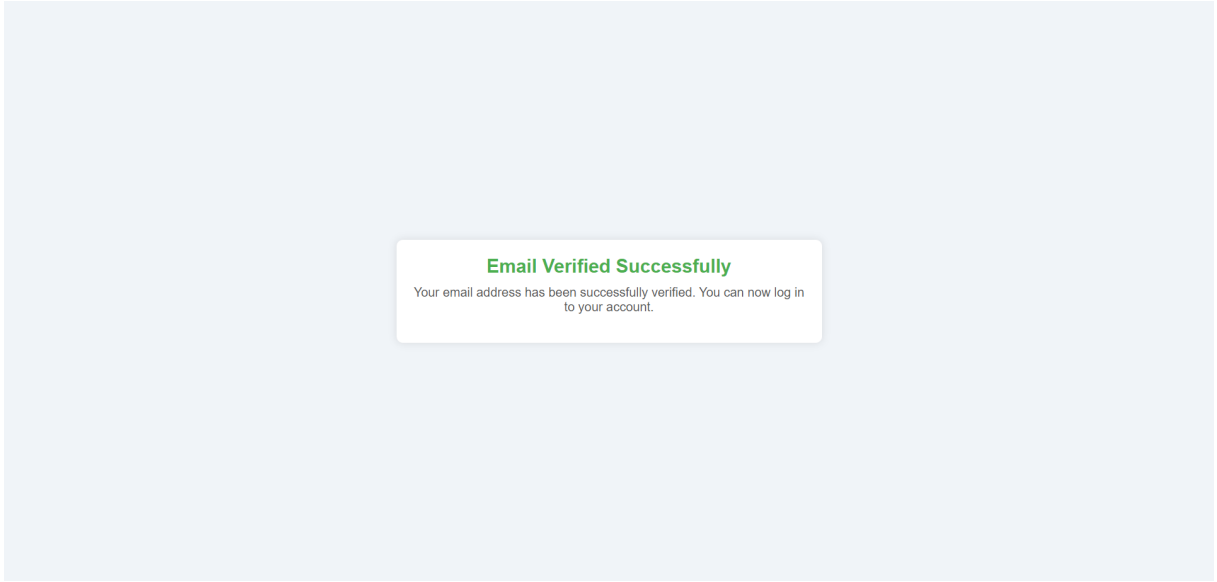


Figure 5.14: Search Results

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

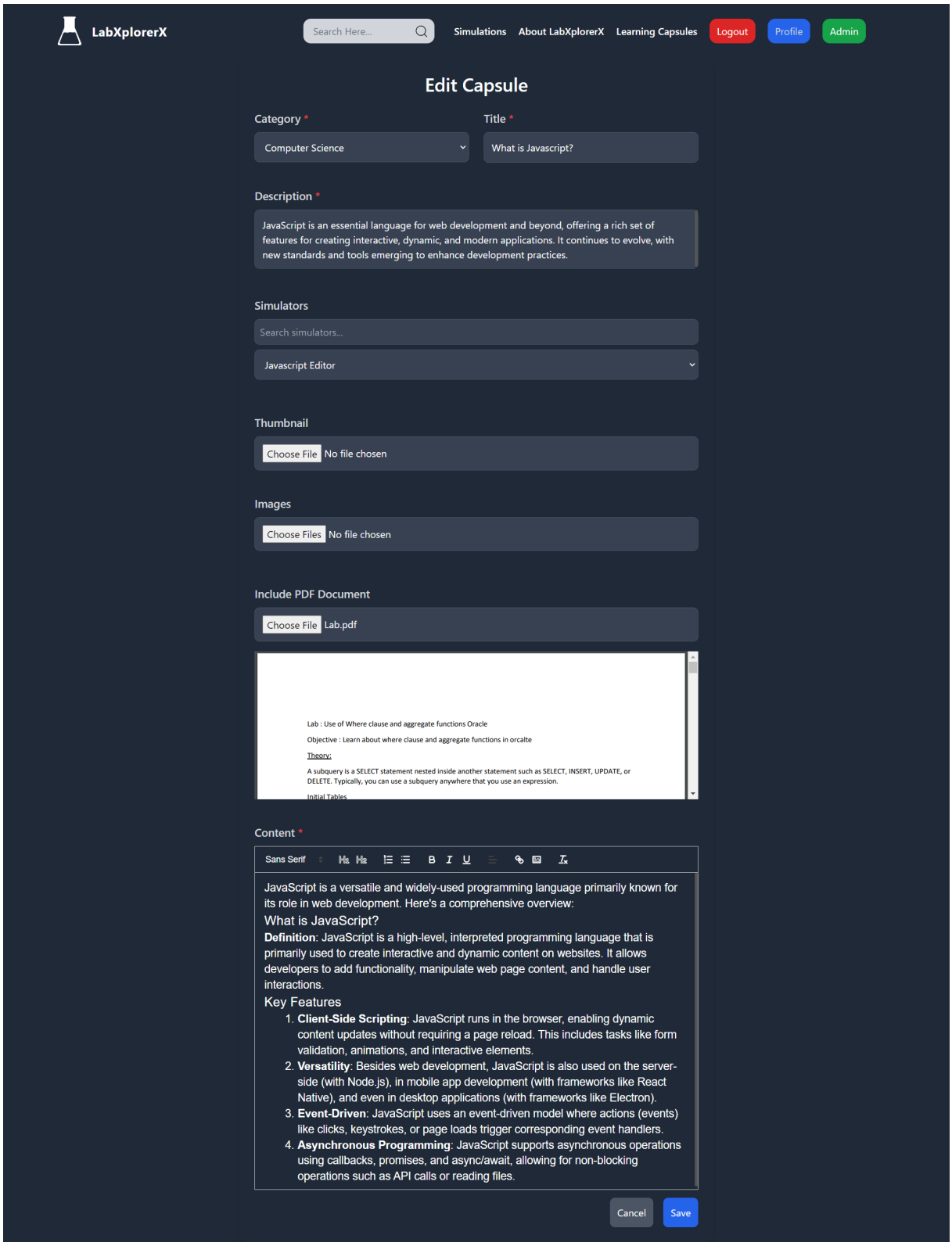


Figure 5.15: Edit Capsule

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

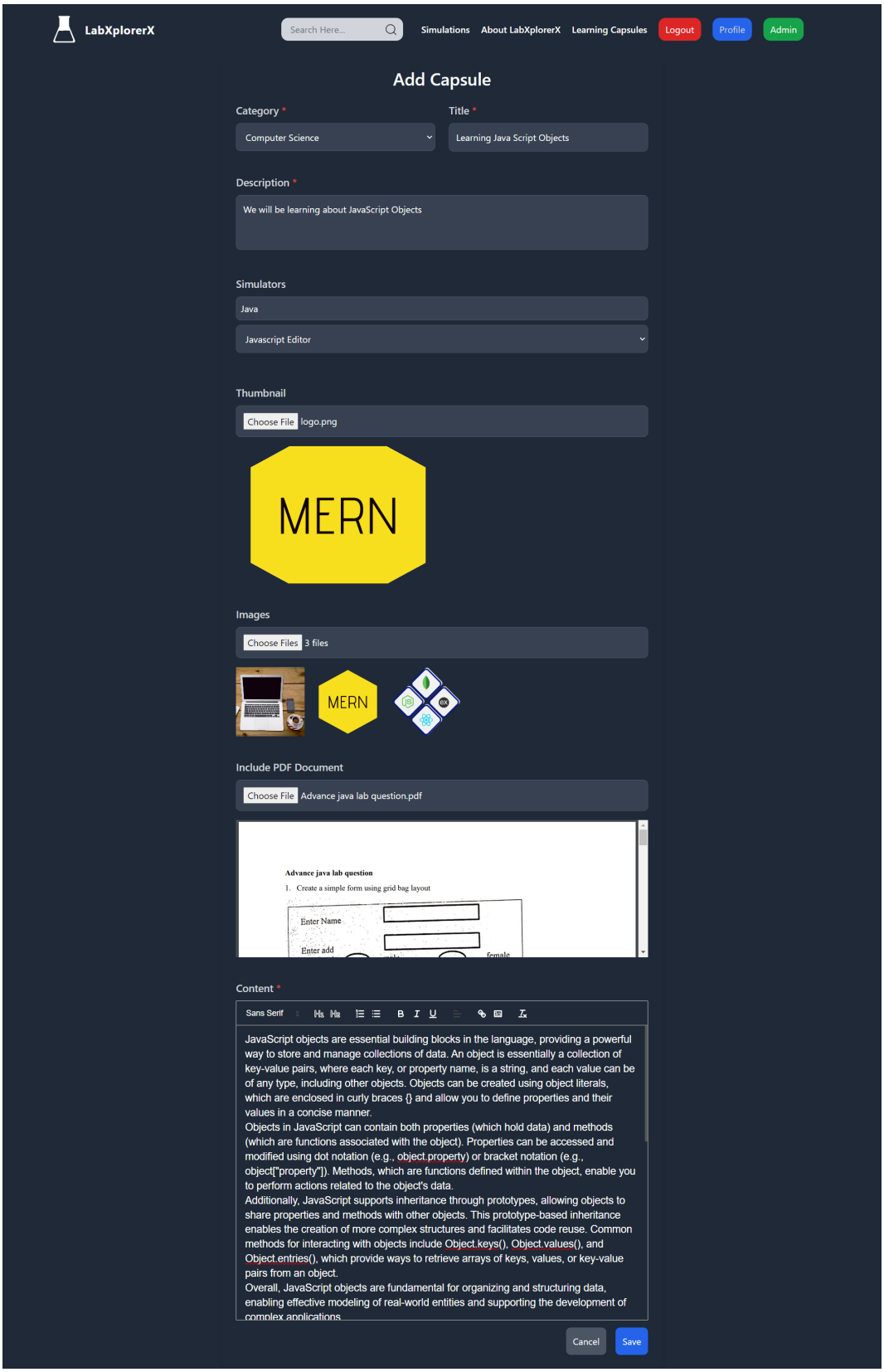


Figure 5.16: Add Capsule

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

Figure 5.17: After Addition of Capsule

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

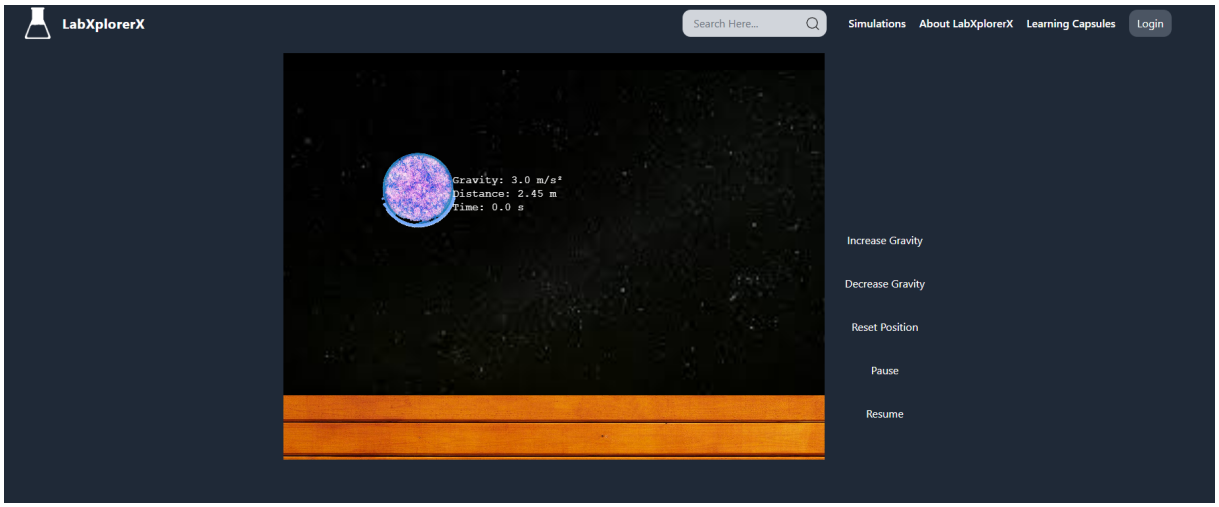


Figure 5.18: Gravity Simulator

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.

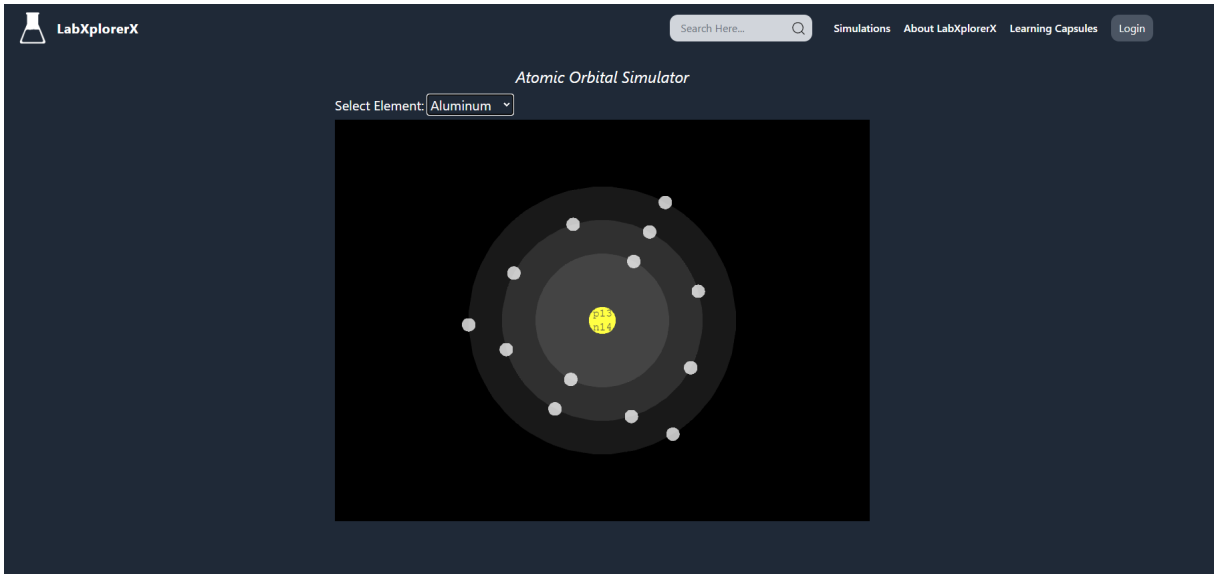


Figure 5.20: Atom Simulator

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

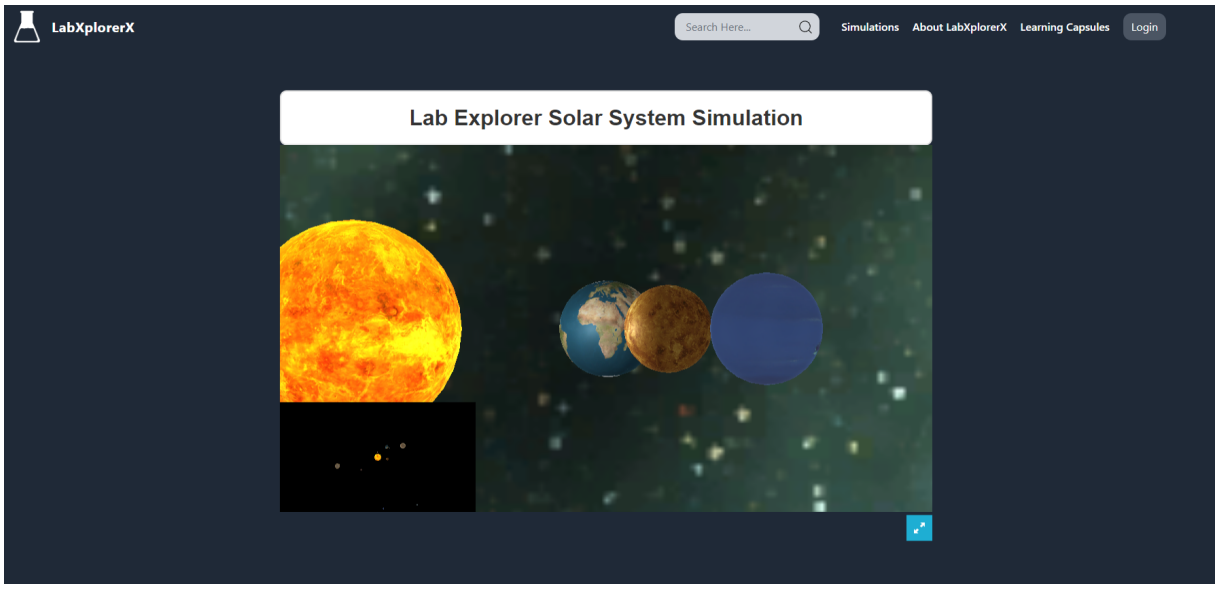


Figure 5.21: Solar System Simulator

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

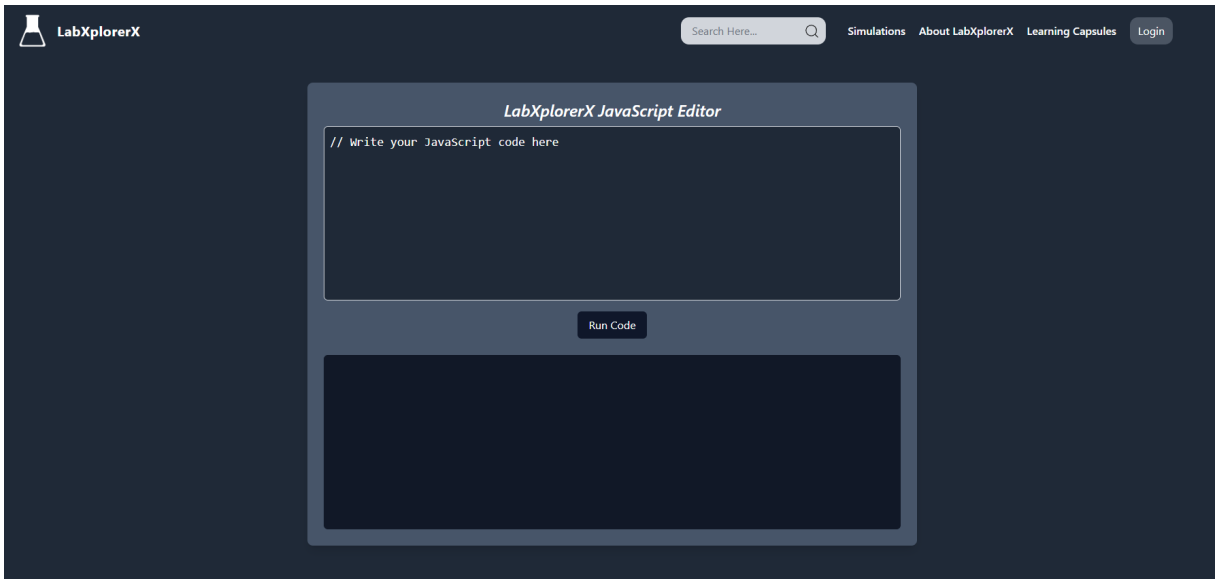


Figure 5.22: JavaScript Editor

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.

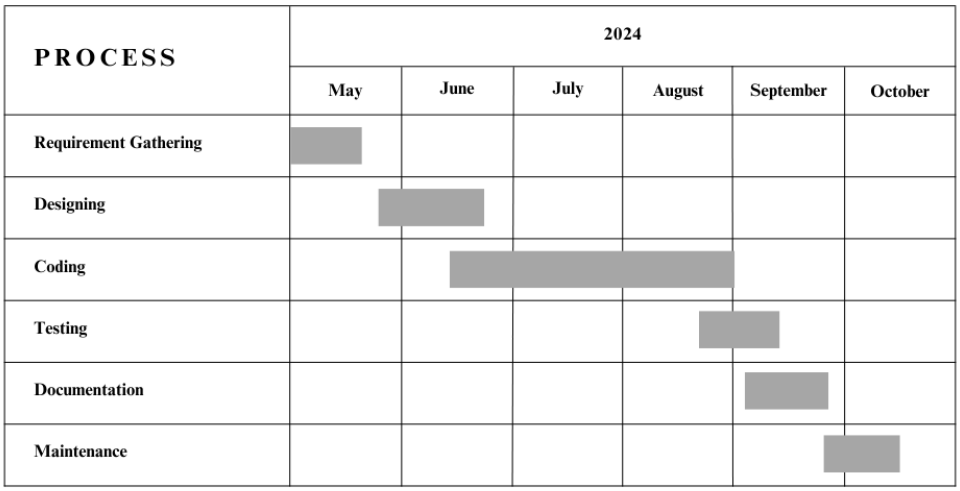


Figure A.23: Gantt Chart of Schedule

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

	id [PK] integer	username character varying (50)	email character varying (100)	password text	created_at timestamp without time zone	updated_at timestamp without time zone	email_verified boolean	email_verification_token character varying (255)
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@labxplorerx.com	$2b$10$O3...	2024-08-07 18:06:11.700329	2024-08-07 18:06:11.700329	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

	id [PK] integer	title character vary	description text	content text	thumbnail character varying (;	images text[]	pdf character var	category character varying (5	author_id integer	created_at timestamp with time	updated_at timestamp with ti
1	6	Atom test	An atom is t...	<p>Struc...	uploads\17231...	{uploads1723131479...	uploads\...	Chemistry	15	2024-08-08 21:2...	2024-08-08 21...
2	7	What is J...	JavaScript i...	<p>JavaScript is ...	uploads\17231...	{uploads1723131928...		Cs	15	2024-08-08 21:3...	2024-08-08 21...
3	8	Understan...	Ohm's Law i...	<p>Ohm'...	uploads\17231...	{uploads1723132242...		Physics	15	2024-08-08 21:3...	2024-08-08 21...
4	9	Understan...	Gravity is th...	<h3>Understandi...	uploads\17231...	{uploads1723132504...		Physics	15	2024-08-08 21:4...	2024-08-08 21...
5	10	Introducti...	The solar sy...	<p>The solar syst...	uploads\17231...	{uploads1723136808...		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...	{uploads1726246274...	uploads\...	Cs	15	2024-09-13 22:3...	2024-09-13 22...

Figure A.25: Table of Capsule

	capsule_id [PK] integer	simulation_id [PK] integer
1	25	7

Figure A.26: Table of Capsule Simulations

	comment_id [PK] integer	capsule_id integer	user_id integer	comment_text text	created_at timestamp without time zone	updated_at timestamp without time zone
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

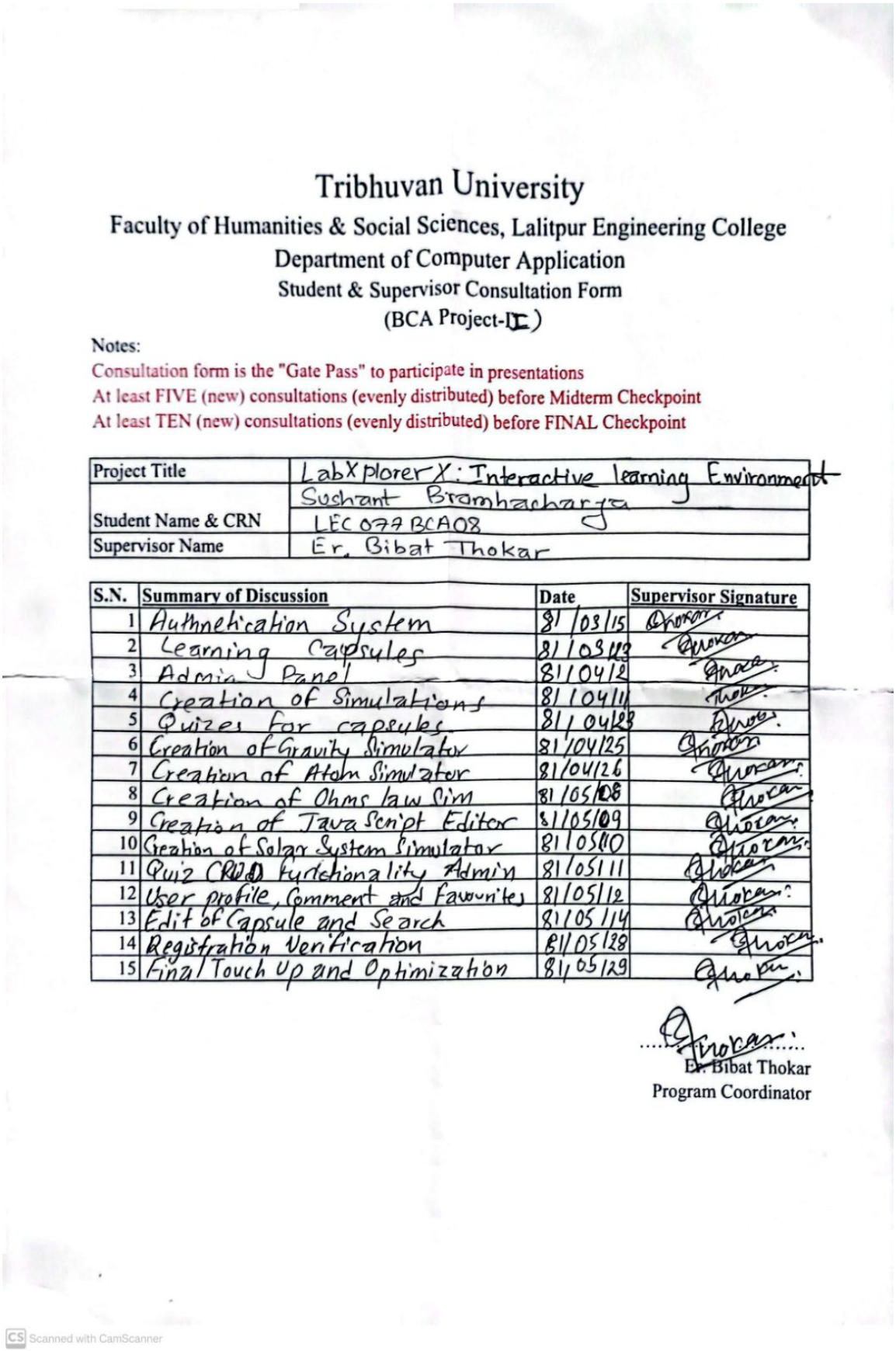
	id [PK] integer	title character varying (255)	capsule_id integer
1	3	Basic Quiz on Ohms Law	8
2	4	Effects of Resistance on Current in Ohms Law	8
3	5	Gravity and Time to Fall	9
4	6	Basic Concepts of Gravity	9
5	7	What is the atomic number of Hydrogen and how many protons does it hav...	6

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	LabXplorerX: Interactive learning Environment
Student Name & CRN	Sushant Bramhacharya LEC077BCA08
Supervisor Name	Er. Bibat Thokar

S.N.	Summary of Discussion	Date	Supervisor Signature
1	Authnehication System	81/03/15	
2	Learning Capsules	81/03/23	
3	Admin Panel	81/04/2	
4	Creation of Simulations	81/04/11	
5	Quizes for capsules	81/04/28	
6	Creation of Gravity Simulator	81/04/25	
7	Creation of Atom Simulator	81/04/26	
8	Creation of Ohms law Sim	81/05/08	
9	Creation of Java Script Editor	81/05/09	
10	Creation of Solar System Simulator	81/05/10	
11	Quiz CRUD functionality Admin	81/05/11	
12	User profile, Comment and Favourites	81/05/12	
13	Edit of Capsule and Search	81/05/14	
14	Registration Verification	81/05/28	
15	Final Touch up and Optimization	81/05/29	

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.