

VITVARA Technologies

Internship Report

FULL STACK WEB DEVELOPMENT

NAME: LEKHANA SAMUDRA

USN: 1SU20CS030

BRANCH: COMPUTER SCIENCE

COLLEGE: SRINIVAS UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATE OF INTERNSHIP: 17-JULY-2023 TO 16-AUGUST-2023

COMPANT GUIDE: SHRIKANTH

(Chief Executive Officer)

COMPANY: VITVARA TECHNOLOGIES

#3-25/7, Vinayaka Udaya nagar

Behind KPT Junction, Airport Rd,

Kadri Hills, Mangaluru, Karnataka 575004

**Introduction**

Vitvara is Mangalore based Startup Company established in 2011. Vitvara was nurtured by a group of entrepreneurs with a sole mission of establishing a dedicated Research & Development Cell to fertilize the innovations of budding engineers. We are specialized for Software and educational services. The internship at Vitvara Technologies provided hands-on experience in the field of Android app development. The participants were introduced to the latest tools and technologies in the market, and given the opportunity to put their theoretical knowledge into practical one. I had the opportunity to immerse myself in the world of web development. This report highlights the key concepts and technologies I learned over the course of the internship.

**Overview of Internship Duration**

* Duration: **17-July-2023 to 16-August-2023**
* Total Duration: **4 weeks**

**Learning Objectives**

My primary learning objectives during the internship were as follows:

1. Gain a solid understanding of basic IT concepts.
2. Master the fundamentals of HTML, CSS, JavaScript, and jQuery.
3. Develop the ability to create and manipulate databases using MySQL.
4. Learn the basics of server-side programming with PHP.
5. Implement interactive login, signup, and data handling functionalities.

**Internship Breakdown**

**Introduction to Basic IT Concepts (2 Days)**

In today's digitally interconnected world, understanding the fundamental concepts of information technology (IT) is essential. This overview delves into three key areas: internet fundamentals, client-server architecture, and networking basics. These concepts serve as the building blocks for the modern technological landscape.

**Internet Fundamentals**

The internet, often referred to as the "Information Superhighway," is a global network of interconnected computers and devices. It has revolutionized the way information is accessed, shared, and communicated. The internet operates based on several core principles:

**1. Packet Switching:**

Data is divided into small packets that are sent independently over the network. This approach ensures efficient use of resources and reliable data transmission.

**2. Protocols:**

Protocols are rules and standards that govern data exchange and communication between devices on the internet. Key protocols include HTTP (Hypertext Transfer Protocol) for web communication and SMTP (Simple Mail Transfer Protocol) for email transmission.

**3. IP Addresses:**

IP (Internet Protocol) addresses uniquely identify devices on the internet. IPv4 and IPv6 are the two major versions of IP addressing, with IPv6 introduced to address the limitations of IPv4's address space.

**4. Domain Names:**

Domain names provide human-readable names for websites, translating them into IP addresses. The Domain Name System (DNS) serves as the internet's address book.

**Client-Server Architecture**

Client-server architecture is a fundamental model for distributed computing where tasks are divided between clients (user devices) and servers (centralized resources). This architecture enables efficient resource utilization and scalability:

**1. Clients:**

Clients request services or resources from servers. They can range from personal computers and smartphones to IoT devices.

**2. Servers:**

Servers store and manage data, applications, and services. They respond to client requests by processing information and delivering the requested content.

**3. Request-Response Model:**

Communication in client-server architecture follows a request-response pattern. Clients send requests, and servers respond with the appropriate data or action.

**4. Scalability:**

Client-server architecture allows for scalability by adding more servers to handle increased demand. Load balancers distribute client requests across multiple servers.

**Networking Basics**

Networking forms the backbone of IT infrastructure, enabling devices to communicate and share resources across networks. Key networking concepts include:

**1. Local Area Network (LAN) and Wide Area Network (WAN):**

LANs are confined to a limited geographic area, such as a home, office, or campus. WANs cover larger distances and connect multiple LANs.

**2. Routers and Switches:**

Routers connect networks and direct data between them, while switches facilitate data exchange within a network.

**3. Protocols and OSI Model:**

The OSI (Open Systems Interconnection) model defines a conceptual framework for networking. It comprises seven layers, each responsible for specific functions, such as data encapsulation, routing, and application interaction.

**4. Network Security:**

Network security is crucial to protect data from unauthorized access and attacks. Concepts like firewalls, encryption, and intrusion detection systems are employed to ensure network security.

**Conclusion**

A foundational understanding of internet fundamentals, client-server architecture, and networking basics is essential for anyone navigating the digital landscape. As technology continues to evolve, these concepts provide the framework for innovations that shape the way we communicate, access information, and interact with the world. Whether it's the way we browse the web, engage with social media, or build robust applications, these IT concepts underpin the digital experiences that define our modern lives.

**Basic HTML (2 Days):**

HTML (Hypertext Markup Language) serves as the backbone of web development, providing the structure that defines the content and layout of web pages. In this section, we will explore the fundamental aspects of HTML, including its structure, tags, elements, and attributes, while also creating a simple static webpage.

**Understanding the Structure of HTML Documents**

HTML documents are composed of various components that structure the content and define how it should be displayed in web browsers. The core structure includes the following elements:

**1. Document Type Declaration (DOCTYPE):**

The DOCTYPE declaration defines the HTML version being used and ensures that browsers interpret the document correctly.

**2. HTML Element:**

The **<html>** element encapsulates the entire HTML document. It contains the **<head>** and **<body>** sections.

**3. Head Element:**

The **<head>** section contains metadata about the document, such as the title, character encoding, and links to external resources.

**4. Body Element:**

The **<body>** section holds the visible content of the webpage, including text, images, links, and other media.

**Learning About HTML Tags, Elements, and Attributes**

HTML uses a system of tags, elements, and attributes to structure and describe content. Tags are enclosed in angle brackets and indicate the beginning and end of elements. Elements consist of a pair of opening and closing tags, while attributes provide additional information about elements. Let's explore a few essential HTML tags:

**1. Heading Tags (<h1> to <h6>):**

Heading tags are used to define headings of varying levels of importance. **<h1>** is the highest level, representing the main heading, while **<h6>** is the lowest level.

**2. Paragraph Tag (<p>):**

The **<p>** tag is used to define paragraphs of text. It's one of the basic building blocks for structuring content.

**3. Anchor Tag (<a>):**

The **<a>** tag creates hyperlinks, allowing users to navigate to other web pages or resources. It requires an **href** attribute specifying the target URL.

**4. Image Tag (<img>):**

The **<img>** tag embeds images in web pages. The **src** attribute specifies the image source (URL), while the **alt** attribute provides alternative text for accessibility.

**Creating a Simple Static Webpage Using HTML**

**Conclusion**

Understanding the structure of HTML documents, the usage of tags, elements, and attributes, and the creation of a simple static webpage forms the foundation of web development. HTML's role in shaping the content and layout of web pages cannot be overstated. As we delve deeper into web development, we'll build upon this foundation to create more dynamic and interactive experiences for users.

**Advanced HTML (3 Days):**

In the realm of web development, mastering advanced HTML techniques is crucial for creating dynamic and accessible web content. This section explores how to create complex structures using forms, tables, and multimedia elements. Additionally, we'll emphasize the significance of semantic HTML in enhancing accessibility and search engine optimization (SEO).

**Creating Complex Structures with Forms, Tables, and Multimedia Elements**

**Forms (<form>):**

Forms are essential for user interaction and data collection. They enable users to input information, such as text, numbers, and selections. Form elements include text inputs, radio buttons, checkboxes, and buttons.

Html Code

<form action="/submit" method="post">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<label for="gender">Gender:</label>

<input type="radio" id="male" name="gender" value="male">

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="female">

<label for="female">Female</label>

<input type="submit" value="Submit"> </form>

**Tables (<table>):**

Tables are used for organizing tabular data. They consist of rows (**<tr>**) and cells (**<td>** for data cells, **<th>** for header cells).

Html Code

<table>

<tr>

<th>Name</th>

<th>Age</th>

</tr>

<tr>

<td>John</td>

<td>25</td>

</tr>

<tr>

<td>Jane</td>

<td>30</td>

</tr>

</table>

**Multimedia Elements (<video> and <audio>):**

HTML supports embedding multimedia elements such as videos and audio tracks directly into web pages.

Html Code

<video controls> <source src="video.mp4" type="video/mp4"> Your browser does not support the video tag. </video>

<audio controls> <source src="audio.mp3" type="audio/mpeg"> Your browser does not support the audio tag. </audio>

**Emphasizing Semantic HTML for Accessibility and SEO**

**Semantic Elements:**

Semantic HTML uses elements that convey meaning to both browsers and developers. Examples include **<header>**, **<nav>**, **<main>**, **<article>**, **<section>**, and **<footer>**. These elements enhance the structure of a webpage, aiding screen readers and search engines in understanding content.

Html Code

<header>

<h1>Welcome to Our Website</h1>

<nav>

<ul>

<li><a href="/">Home</a></li>

<li><a href="/about">About</a></li>

<li><a href="/contact">Contact</a></li>

</ul>

</nav>

</header>

<main>

<article> <h2>Exploring Web Development</h2>

<p>Learn about advanced HTML techniques and their significance.</p>

</article>

</main>

<footer> <p>&copy; 2023 Example Company. All rights reserved.</p>

</footer>

**Accessibility and SEO:**

Semantic HTML contributes to improved accessibility by providing a logical structure for assistive technologies. It also enhances SEO, as search engines can better understand the content and context of a webpage.

**Conclusion**

Mastering advanced HTML techniques opens up a world of possibilities in web development. By creating complex structures using forms, tables, and multimedia elements, and by embracing semantic HTML for accessibility and SEO, developers can craft engaging and inclusive web experiences. As technology continues to evolve, these skills remain foundational for building web content that resonates with users and search engines alike.

**CSS (3 Days)**

**CSS Mastery: Crafting Visual Excellence and Responsive Designs**

Cascading Style Sheets (CSS) stand at the forefront of web design, providing the means to transform plain HTML into visually appealing and responsive web pages. Over the next three days, we will explore the essence of CSS, from its introduction to advanced layout techniques, while also delving into the realm of responsive design principles.

**Introduction to CSS for Styling HTML Elements**

CSS is the design language of the web, allowing developers to define the appearance and layout of HTML elements. It operates through a set of rules, each comprising selectors, properties, and values. Let's delve into the core components:

**Selectors:**

Selectors pinpoint HTML elements to which CSS rules apply. They can be based on elements, classes, IDs, attributes, and more.

css code

p { color: blue; }

/\* Selecting by class \*/ .button { background-color: #f1c40f; }

/\* Selecting by ID \*/ #header { font-size: 24px; }

**Properties and Values:**

Properties dictate what aspect of an element to style, while values define how the styling is applied.

css code

h1 { font-family: 'Arial', sans-serif; font-weight: bold; }

.button { background-color: #3498db; color: white; }

.container { width: 80%; margin: 0 auto; }

**Learning About Basic Layout Techniques**

CSS empowers developers to structure and position elements effectively. A few essential layout techniques include:

**Box Model:**

The box model governs how elements are sized and spaced, comprising content, padding, border, and margin.

css code

.box { width: 200px; padding: 20px; border: 1px solid #ddd; margin: 10px; }

**Flexbox:**

Flexbox is a powerful layout tool that simplifies the arrangement of elements in rows or columns.

css code

.container { display: flex; justify-content: space-between; } .item { flex: 1; }

**Grid Layout:**

Grid layout allows for more complex arrangements, creating both rows and columns.

css code

.container { display: grid; grid-template-columns: repeat(3, 1fr); grid-gap: 20px; }

**Implementing Responsive Design Principles**

Responsive design ensures web pages adapt seamlessly to various screen sizes and devices. Achieve this with media queries and fluid layouts:

**Media Queries:**

Media queries apply specific styles based on the device's characteristics, such as screen width.

css code

/\* Apply styles when screen width is less than 600px \*/ @media (max-width: 600px) { .menu { display: none; } }

**Fluid Layouts:**

Fluid layouts use relative units (percentages) instead of fixed units (pixels), allowing content to expand and contract as the screen size changes.

css code

.container { width: 90%; margin: 0 auto; }

/\* Images scale with container \*/ .img-responsive { max-width: 100%; height: auto; }

**Conclusion**

CSS is the artist's palette in web development, enabling the transformation of raw HTML into visually captivating and responsive websites. With a solid grasp of selectors, properties, and values, along with proficiency in basic layout techniques and responsive design principles, developers can craft exceptional user experiences that seamlessly adapt to the ever-evolving digital landscape. As we continue to explore the realm of web development, CSS will remain an indispensable tool in the pursuit of design excellence and user satisfaction.

**JavaScript Fundamentals (5 Days)**

**JavaScript Fundamentals: Empowering Interactivity and Dynamic Web Development**

In the dynamic realm of web development, JavaScript serves as the engine that breathes life into web pages. Over the next five days, we will embark on a journey through JavaScript's fundamental concepts. From its introduction as a programming language for web development to understanding variables, data types, functions, conditional statements, loops, and basic DOM manipulation, we will explore the core facets that make JavaScript a cornerstone of modern web experiences.

**Introduction to JavaScript as a Programming Language**

JavaScript, often referred to as the "language of the web," is a versatile programming language that brings interactivity and dynamism to web pages. It's executed directly in browsers, enabling real-time interactions without the need for constant server requests.

**Script Tag and External Scripts:**

JavaScript code can be included within HTML using the **<script>** tag or as an external script file linked with the **src** attribute.

Html Code

<script> alert("Hello, world!"); </script>

<!-- External Script -->

<script src="script.js"></script>

**Learning About Variables, Data Types, and Functions**

**Variables:**

Variables store data that can be used and manipulated throughout the script. They are declared using **var**, **let**, or **const**.

Javascript Code

var age = 25; let name = "John"; const pi = 3.14159;

**Data Types:**

JavaScript supports various data types, including numbers, strings, booleans, objects, arrays, and more.

Javascript Code

var num = 42; var message = "Hello, world!"; var isTrue = true;

var person = { firstName: "Jane", lastName: "Doe" }; var fruits = ["apple", "banana", "orange"];

**Functions:**

Functions are blocks of code that can be reused and called with specific inputs (arguments).

Javascript Code

function greet(name) { return "Hello, " + name + "!"; } var result = greet("Alice");

// Returns "Hello, Alice!"

**Conditional Statements and Loops**

**Conditional Statements (if, else if, else):**

Conditional statements execute different blocks of code based on specified conditions.

Javascript Code

var age = 18;

if (age < 18)

{ console.log("You are a minor."); }

else if (age >= 18 && age < 65)

{ console.log("You are an adult."); }

else { console.log("You are a senior citizen."); }

**Loops (for, while):**

Loops are used to execute a block of code repeatedly.

Javascript Code

// For loop for (var i = 1; i <= 5; i++)

{ console.log("Iteration " + i); }

// While loop var count = 0; while (count < 3)

{ console.log("Count: " + count); count++; }

**Implementing Basic DOM Manipulation and Interactivity**

**Document Object Model (DOM):**

The DOM is a programming interface that allows scripts to interact with HTML and XML documents. It enables the modification of content and structure dynamically.

Html Code

<div id="myDiv">Hello, DOM!</div>

Javascript Code

var element = document.getElementById("myDiv"); element.innerHTML = "Hello, Updated DOM!";

**Event Handling:**

JavaScript facilitates event handling, enabling interactivity by responding to user actions.

Html Code

<button id="myButton">Click me</button>

Javascript Code

var button = document.getElementById("myButton"); button.addEventListener("click", function() { alert("Button clicked!"); });

**Conclusion**

JavaScript is the dynamic force behind modern web development, driving interactivity and responsive user experiences. Armed with an understanding of variables, data types, functions, conditional statements, loops, and basic DOM manipulation, developers can create interactive web pages that adapt and respond to user actions. As we continue our exploration of the web development landscape, JavaScript's capabilities will empower us to bring our visions to life, turning static content into engaging digital interactions.

**MySQL and SQL Queries (3 Days)**

**MySQL and SQL Queries: Unveiling the Power of Data Management**

In the dynamic landscape of web development, efficient data management is essential for creating robust and scalable applications. Over the course of the next three days, we will delve into the realm of relational databases, exploring their significance, mastering basic SQL syntax for data manipulation and querying, and creating and managing databases using MySQL.

**Understanding Relational Databases and Their Importance**

**Relational Databases:**

Relational databases organize data into structured tables with predefined relationships, facilitating efficient data retrieval and manipulation.

**Tables and Relationships:**

Tables consist of rows (records) and columns (fields). Relationships define how tables are connected based on shared data.

**Importance in Web Development:**

Relational databases enable seamless storage and retrieval of data, crucial for applications with user accounts, e-commerce transactions, content management, and more.

**Learning Basic SQL Syntax for Data Manipulation and Querying**

**SELECT Statement:**

The **SELECT** statement retrieves data from one or more tables.

SQL Code

SELECT first\_name, last\_name FROM employees;

**INSERT Statement:**

The **INSERT** statement adds new records to a table.

SQL Code

INSERT INTO customers (first\_name, last\_name) VALUES ('John', 'Doe');

**UPDATE Statement:**

The **UPDATE** statement modifies existing records in a table.

SQL Code

UPDATE products SET price = 49.99 WHERE id = 123;

**DELETE Statement:**

The **DELETE** statement removes records from a table.

SQL Code

DELETE FROM orders WHERE order\_date < '2023-01-01';

**Creating and Managing Databases Using MySQL**

**Creating Databases:**

Use the **CREATE DATABASE** statement to create a new database.

SQL Code

CREATE DATABASE ecommerce;

**Creating Tables:**

Tables are defined using the **CREATE TABLE** statement, specifying columns and data types.

SQL Code

CREATE TABLE products ( id INT PRIMARY KEY, name VARCHAR(255), price DECIMAL(10, 2) );

**Managing Data:**

Use various SQL statements to manage data, including **SELECT**, **INSERT**, **UPDATE**, and **DELETE**.

**Querying Data:**

SQL's power lies in querying data. Complex queries retrieve specific information from multiple tables.

SQL Code

SELECT customers.first\_name, orders.order\_date FROM customers JOIN orders ON customers.id = orders.customer\_id WHERE customers.country = 'USA';

**Conclusion**

The mastery of MySQL and SQL queries is indispensable for web developers navigating the data-driven world of web applications. By understanding the significance of relational databases, grasping the essentials of SQL syntax for data manipulation and querying, and adeptly creating and managing databases using MySQL, developers equip themselves with the tools needed to craft applications that efficiently handle and present data. As we continue our exploration of web development, the knowledge gained during these three days will remain integral to building applications that not only function effectively but also deliver a seamless and engaging user experience.

**PHP for Server-Side Programming (1 Day)**

**PHP for Server-Side Programming: Unleashing Dynamic Web Power**

In the realm of web development, PHP emerges as a powerful server-side scripting language, transforming static web pages into dynamic and interactive experiences. Over the next day, we will embark on an exploration of PHP, from its introduction as a server-side scripting language to integrating it with HTML for dynamic web pages. Additionally, we will develop a foundational understanding of PHP's pivotal role in server-side processing.

**Introduction to PHP as a Server-Side Scripting Language**

**PHP's Role:**

PHP (Hypertext Preprocessor) is a scripting language primarily used for server-side web development. It runs on the server before the page is sent to the client, allowing dynamic content generation.

**Advantages:**

PHP empowers developers to create dynamic and interactive web pages that adapt to user input and data changes.

**Integrating PHP with HTML for Dynamic Web Pages**

**Embedding PHP in HTML:**

PHP code can be seamlessly embedded within HTML, denoted by **<?php** and **?>** tags.

Php Code

<!DOCTYPE html>

<html> <head>

<title>Dynamic Page</title>

</head> <body> <h1>Welcome, <?php echo "User"; ?>!</h1> <p>Today's date is <?php echo date("Y-m-d"); ?>.</p> </body> </html>

**Dynamic Content:**

PHP allows the insertion of dynamic content into HTML, such as user-specific greetings and real-time date information.

**Developing a Basic Understanding of PHP's Role in Server-Side Processing**

**Server-Side Processing:**

PHP is executed on the server, enabling tasks like handling form submissions, interacting with databases, and generating dynamic content before sending it to the client.

**Processing User Input:**

PHP can process user input from forms and perform actions based on that input.

Php Code

<?php if ($\_SERVER["REQUEST\_METHOD"] == "POST")

{ $username = $\_POST["username"]; $password = $\_POST["password"];

// Perform login authentication // ... } ?>

**Communicating with Databases:**

PHP seamlessly interacts with databases to retrieve, update, and manage data.

Php Code

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "myDB";

$conn = new mysqli($servername, $username, $password, $dbname);

$sql = "SELECT id, name, email FROM users";

$result = $conn->query($sql);

if ($result->num\_rows > 0)

{ while($row = $result->fetch\_assoc())

{ echo "ID: " . $row["id"] . " Name: " . $row["name"] . " Email: " . $row["email"]; } }

else { echo "No results found."; }

$conn->close(); ?>

**Conclusion**

PHP's prowess in server-side scripting propels web development into a realm of dynamic and interactive experiences. As we embrace PHP's role in generating dynamic content, seamlessly integrating it with HTML, and harnessing its potential for server-side processing, we unlock the ability to create web applications that respond to user input, access databases, and adapt content on the fly. This journey through PHP marks a pivotal step in our exploration of web development, opening doors to a world of personalized and efficient digital experiences.

**jQuery (1 Day)**

**jQuery: Elevating Interactivity Through Simplified DOM Manipulation**

In the ever-evolving landscape of web development, jQuery stands as a prominent JavaScript library that streamlines DOM manipulation and event handling. Over the course of the next day, we will embark on a journey through jQuery's offerings. From its introduction as a library designed to simplify DOM manipulation and event handling to implementing basic animations and interactivity, we will explore how jQuery enhances web development by providing a concise and efficient toolset.

**Introduction to jQuery Library for Simplifying DOM Manipulation and Event Handling**

**jQuery's Role:**

jQuery is a lightweight JavaScript library that simplifies DOM manipulation, making it easier to traverse, manipulate, and interact with HTML elements.

**Benefits:**

jQuery reduces the complexity of traditional JavaScript code, enabling developers to achieve more with fewer lines of code.

**Embedding jQuery:**

jQuery can be included in web pages via a script tag or linked from a content delivery network (CDN).

Html Code

<script src="jquery.min.js"></script>

**Implementing Basic Animations and Interactivity Using jQuery**

**DOM Manipulation:**

jQuery simplifies selecting and manipulating DOM elements. The **$** (dollar sign) function is used to select elements by their CSS selectors.

Javascript Code

// Hide an element $("#myElement").hide(); // Change text content $(".title").text("New Title");

**Event Handling:**

jQuery streamlines attaching event handlers to elements, allowing actions to be triggered in response to user interactions.

Javascript Code

// Handle button click $("#myButton").click(function() { alert("Button clicked!"); });

**Basic Animations:**

jQuery offers smooth animations with methods like **.fadeIn()**, **.fadeOut()**, **.slideUp()**, and **.slideDown()**.

Javascript Code

// Fade in an element $("#myElement").fadeIn();

**Interactivity:**

jQuery enables adding interactive elements, enhancing user experience.

Javascript Code

// Toggle element visibility $("#toggleButton").click(function() { $("#toggleElement").toggle(); });

**Conclusion**

jQuery's prowess in simplifying DOM manipulation and event handling significantly streamlines the web development process. As we explore its capabilities, we realize how jQuery transforms complex JavaScript tasks into concise, elegant code. By integrating jQuery's functionality to implement basic animations and interactivity, developers empower web applications with enhanced user experiences and dynamic content presentation. As we continue our journey through the web development landscape, jQuery serves as a powerful tool that accelerates our ability to create engaging, interactive, and visually appealing digital interactions.

**Implementing Login, Signup, and Data Handling (3 Days)**

**Implementing Login, Signup, and Data Handling: Building Secure User Systems**

In the realm of web development, creating secure and functional login and signup systems is paramount to providing users with a seamless experience. Over the next three days, we will dive into the intricacies of integrating HTML, CSS, JavaScript, PHP, and MySQL to craft robust user authentication systems. We will learn to handle user input, validate data, and securely store and retrieve user information, ensuring the integrity and security of user interactions.

**Integrating Knowledge of HTML, CSS, JavaScript, PHP, and MySQL**

**User Authentication Flow:**

User authentication involves registration, login, and session management.

Registration (Signup):

1. Collect user details using HTML forms.
2. Validate and sanitize user input using JavaScript and PHP.
3. Insert user data into the MySQL database.

Authentication (Login):

1. Validate user credentials using PHP and MySQL.
2. Start a session for the authenticated user.
3. Redirect to the user's dashboard.

**Handling User Input and Validating Data**

Front-End Validation (JavaScript):

JavaScript can validate user input in real time, offering instant feedback.

Javascript Code

function validateForm()

{ var username = document.getElementById("username").value;

if (username.length < 4)

{ alert("Username must be at least 4 characters long."); return false; } // Other validations... }

Back-End Validation (PHP):

PHP should further validate and sanitize user input before storing it.

Php Code

$username = $\_POST["username"];

if (strlen($username) < 4)

{ echo "Username must be at least 4 characters long."; return; } // Other validations...

**Securely Storing and Retrieving User Information in a Database**

**Storing Passwords Securely:**

Hash passwords using algorithms like bcrypt before storing them in the database.

Php Code

$hashedPassword = password\_hash($password, PASSWORD\_BCRYPT);

**Authenticating Users:**

Compare stored hashed passwords with the entered password during authentication.

Php Code

if (password\_verify($enteredPassword, $storedHashedPassword)) { // Successful authentication } else { // Authentication failed }

**Session Management:**

Use PHP's built-in session management to maintain user sessions securely.

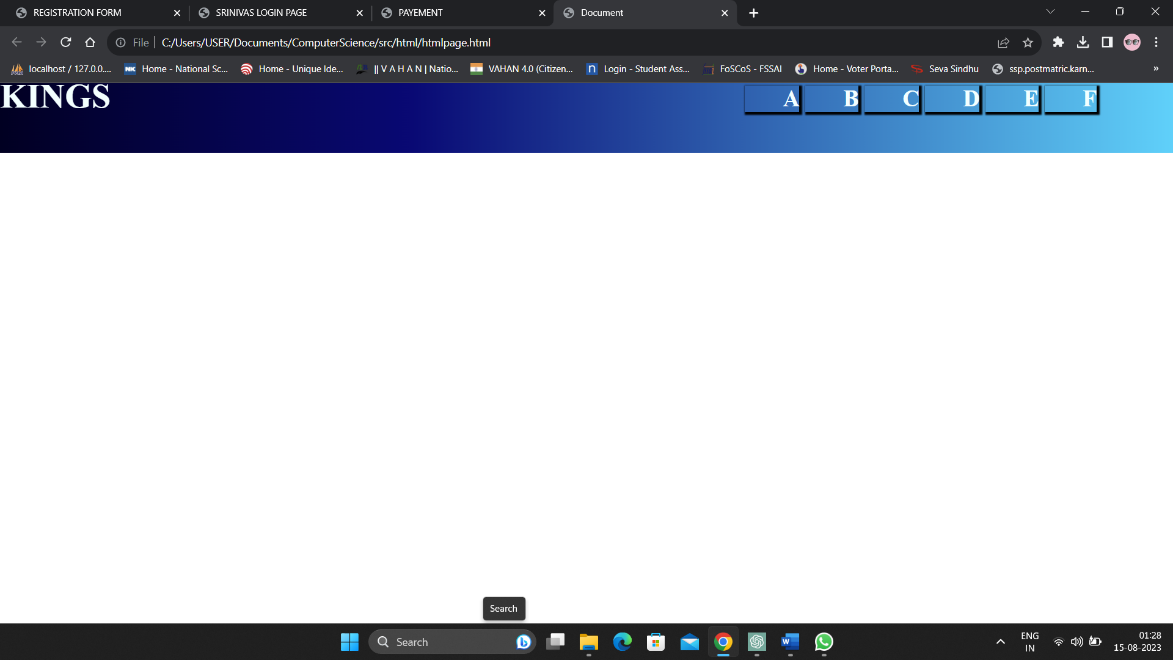
Php Code

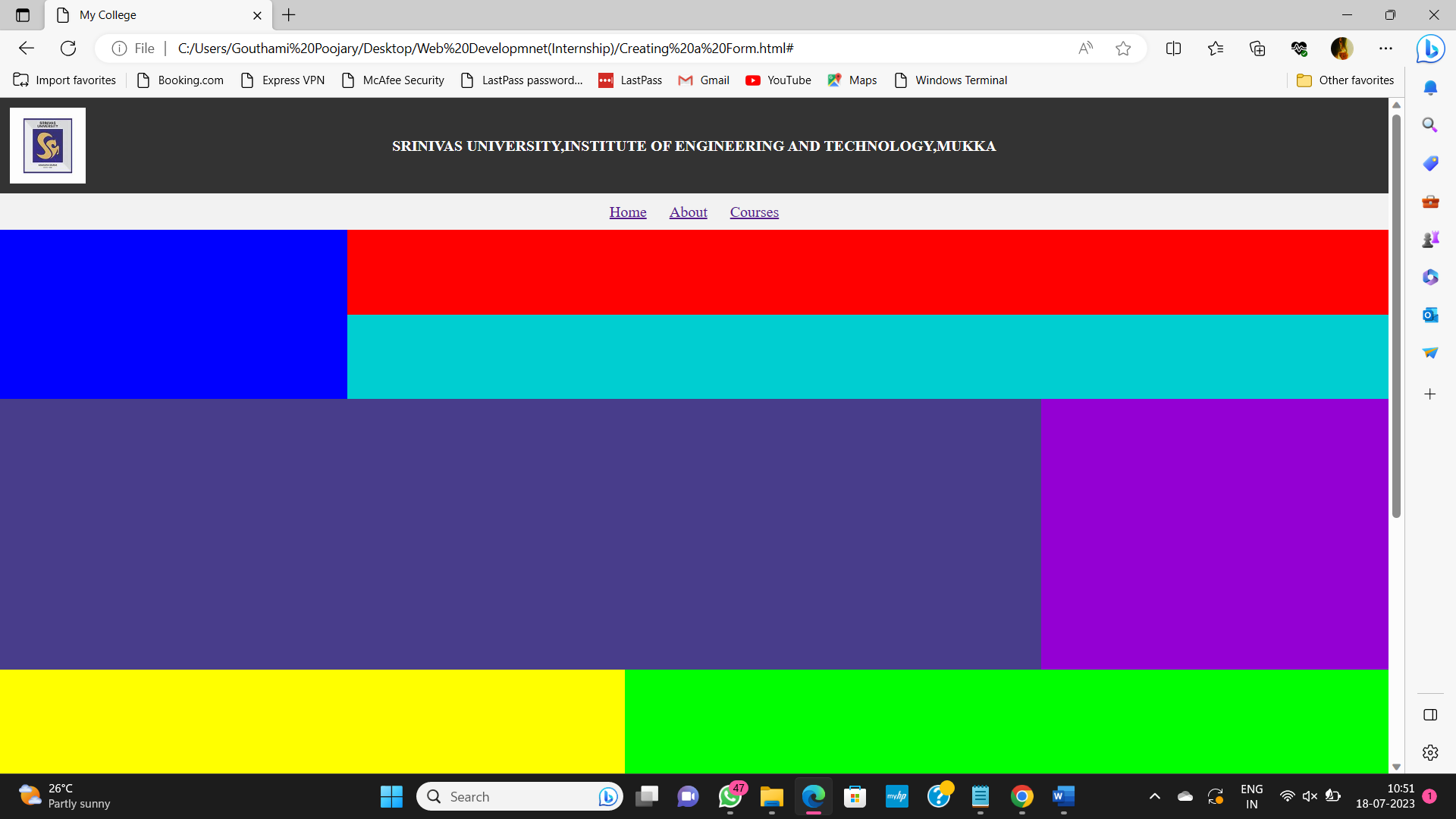
session\_start(); $\_SESSION["user\_id"] = $userId;

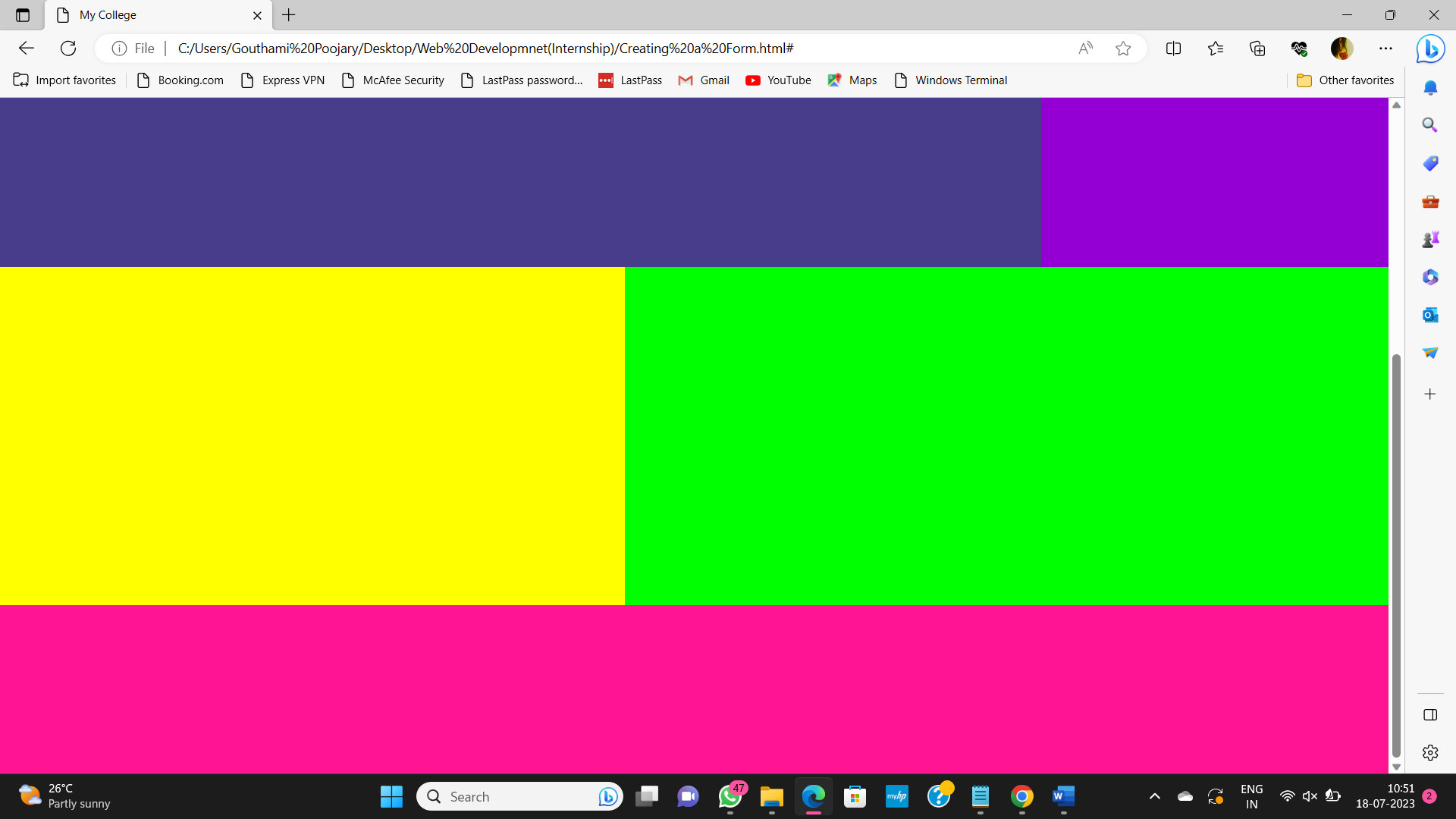
**Conclusion**

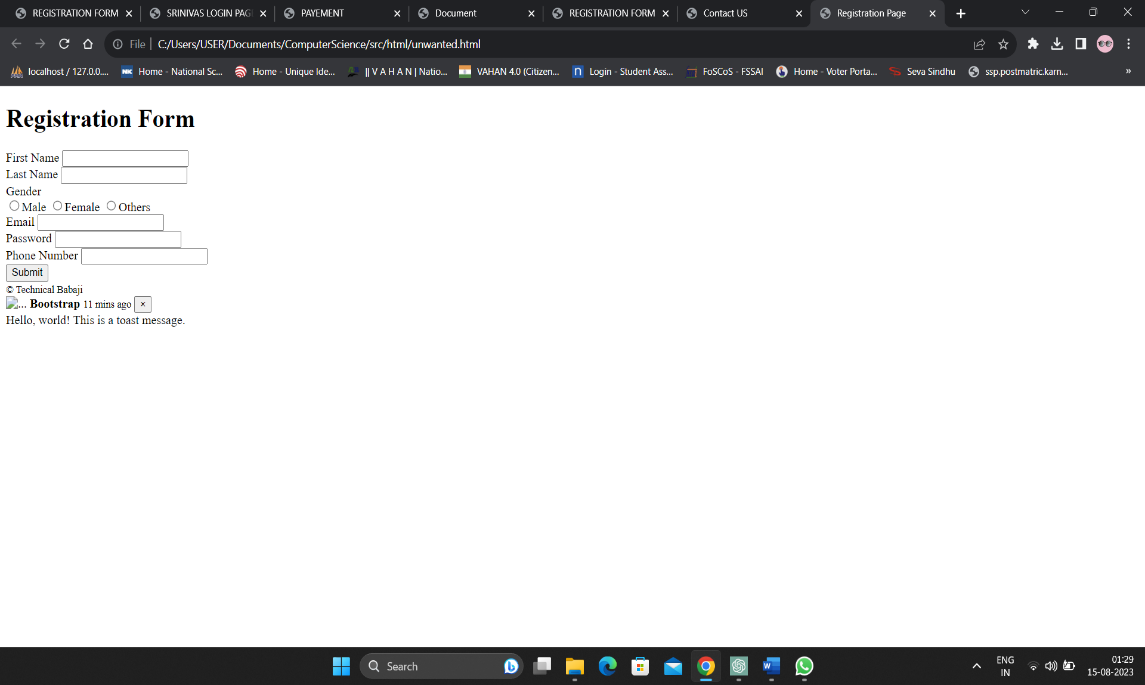
Creating functional login and signup systems is a cornerstone of web development, and mastering this skill is vital for building secure user interactions. By skillfully integrating HTML, CSS, JavaScript, PHP, and MySQL, developers can craft user authentication systems that offer seamless registration, secure login, and session management. Through meticulous data handling, user input validation, secure password storage, and session management, developers ensure the confidentiality and integrity of user data and interactions. As we continue to explore the complexities of web development, the knowledge gained during these three days will play a pivotal role in building applications that prioritize user security and a seamless user experience.

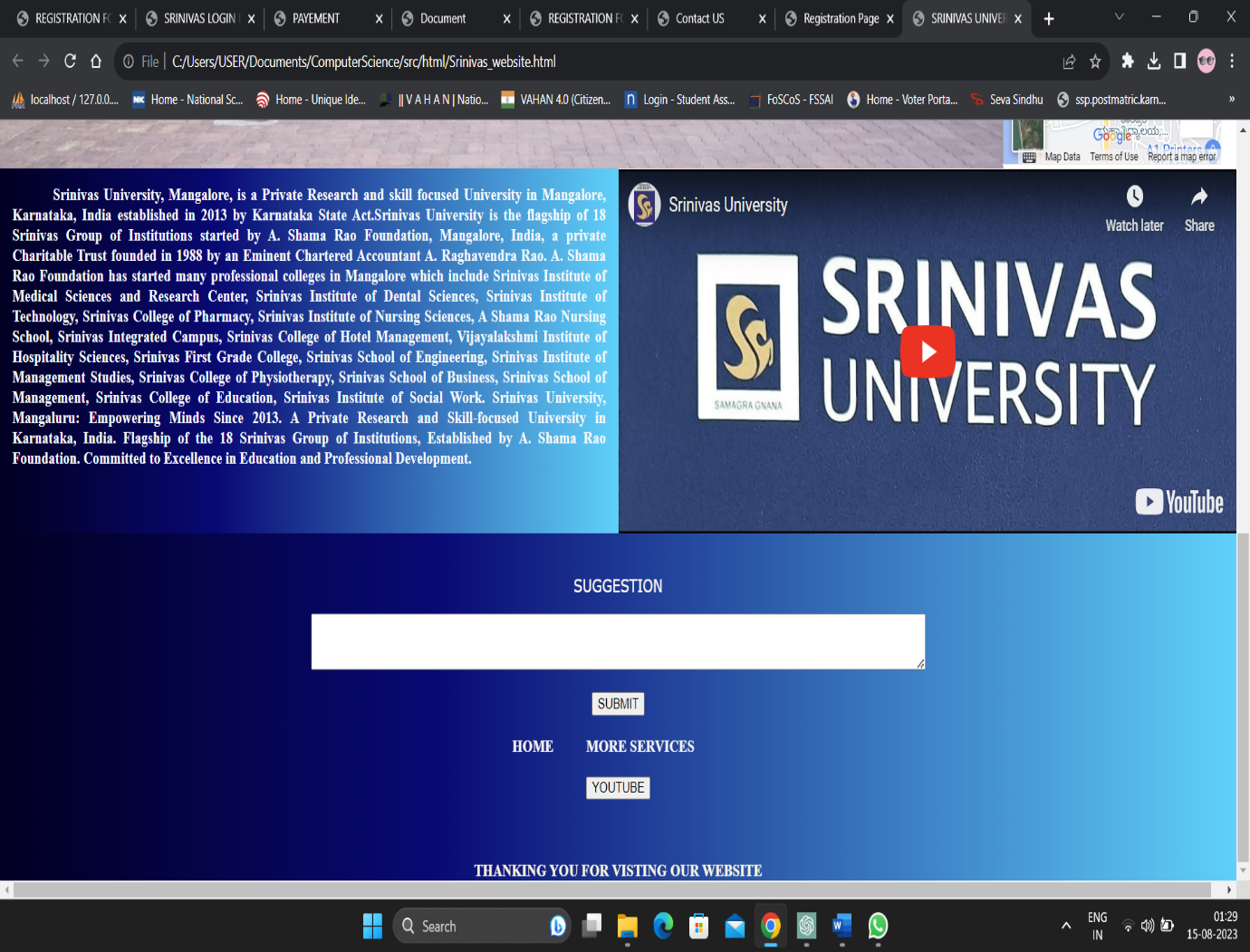
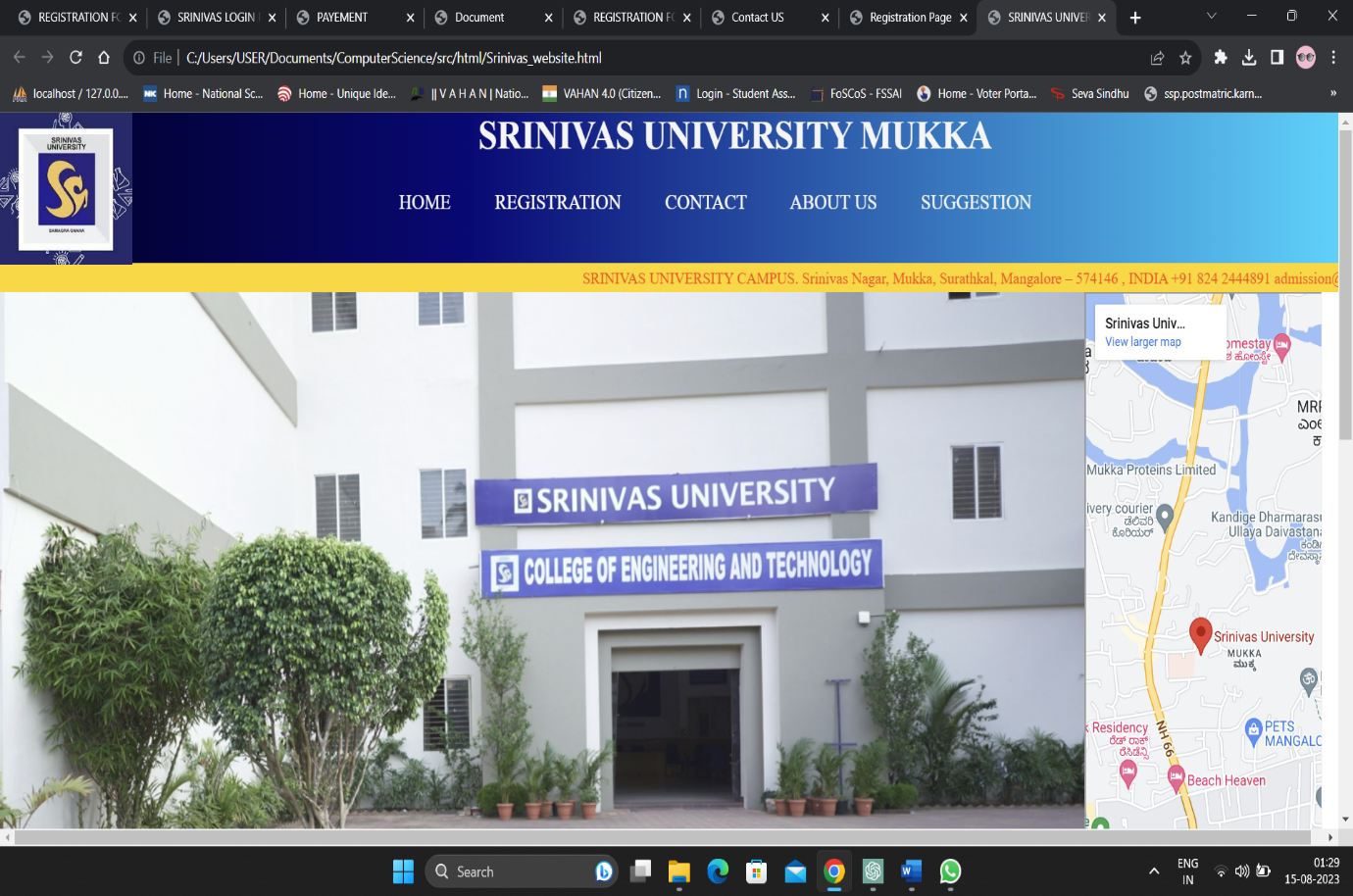
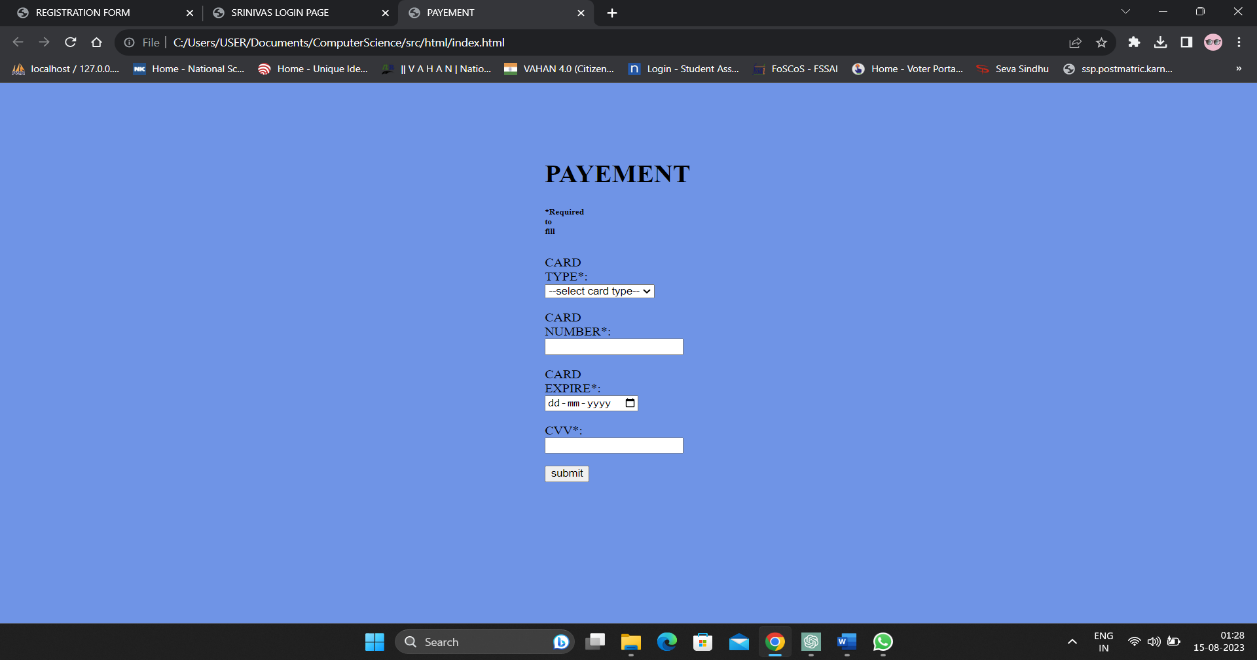
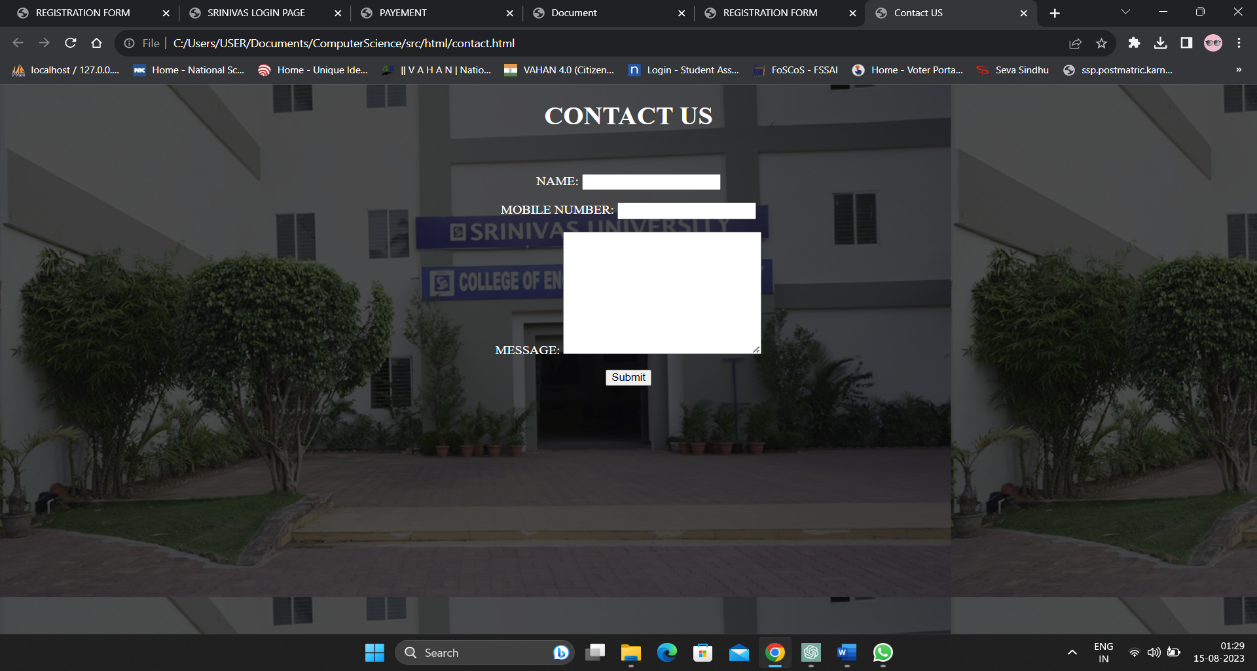
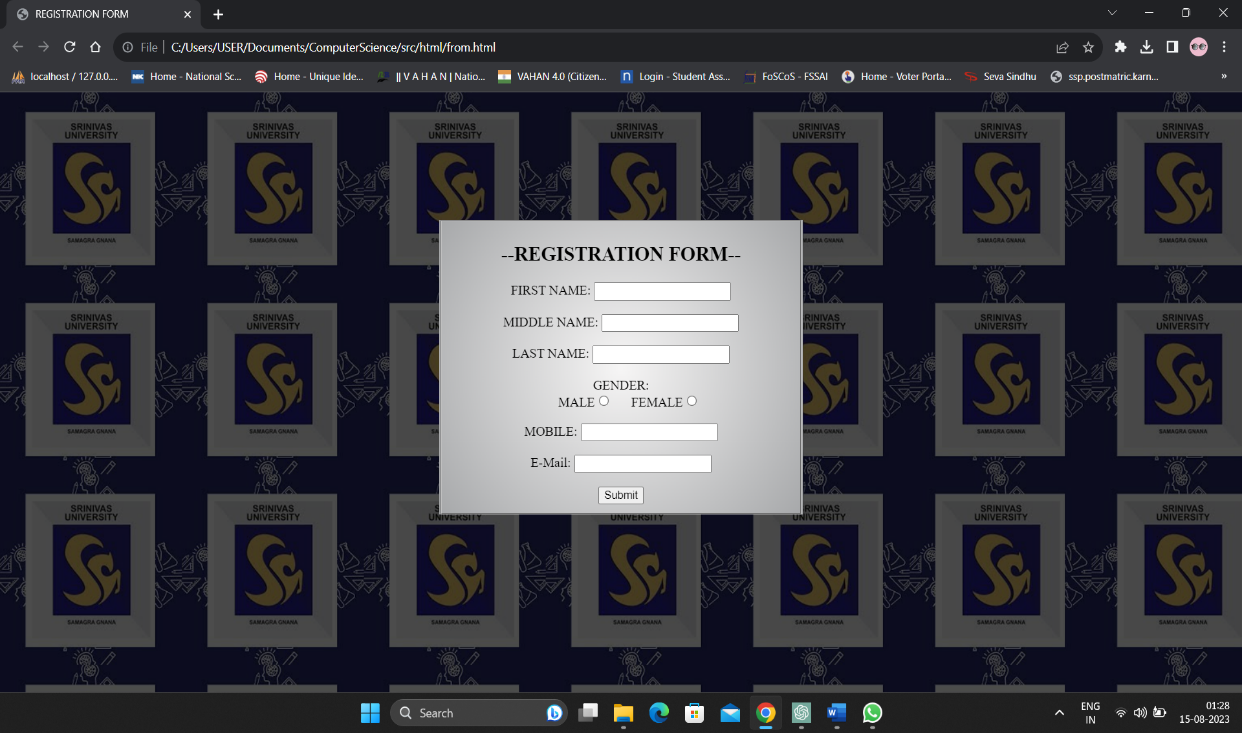
**Screenshot:**







****

****



VITVARA Technologies

Internship Project Report

**RFID-Based School Management System**

Aim: The aim of this project is to develop a web-based School Management System that utilizes RFID technology to automate attendance tracking, access control, library management, and fee payment tracking, improving overall operational efficiency and security within the school environment.

Description: The "RFID-Based School Management System" is a comprehensive web application developed to streamline various administrative tasks within a school environment. The project utilizes Radio-Frequency Identification (RFID) technology to enhance the efficiency, accuracy, and security of processes such as attendance tracking, access control, library management, and fee payment tracking.

Group Member:

Mohammad Azmal , Lekhana Samudra,Mohammad Ali ,Mohammad Shahil, Rahil, Hamzathul Karrar, Mohammad Suhaib

|  |
| --- |
| **Table of Contents** |
| 1. **Introduction** |
| * + Background and Motivation   + Objectives |
| 1. **System Design** |
| * + System Architecture |
| * + Use Case Diagram |
| * + Database Design |
| 1. **Technologies Used** |
| * + Front-End Development |
| * + Back-End Development |
| * + Database Management |
| * + RFID Integration |
| 1. **Features and Functionality** |
| * + Student Attendance Tracking |
| * + Staff Attendance Tracking |
| * + Access Control |
| * + Library Management |
| * + Fee Payment Tracking |
| 1. **Implementation** |
| * + User Authentication and Authorization |
| * + RFID Reader Integration |
| * + Data Storage and Retrieval |
| * + UI/UX Design |
| * + Error Handling and Validation |
| 1. **Testing** |
| * + Unit Testing |
| * + Integration Testing |
| * + User Acceptance Testing  1. **Screenshot** |
| * + Webpage interface screenshots |
| 1. **Challenges Faced** |
| * + RFID Hardware Compatibility |
| * + Data Security and Privacy Concerns |
| * + Real-time Data Synchronization |
| 1. **Future Enhancements** |
| * + Parental Access Portal |
| * + Performance Optimization |
| * + Mobile Application Integration |
| 1. **Conclusion** |
| * + Project Summary |
| * + Achievements and Learning |

1. **Introduction**

**Executive Summary**

This report documents the development of an RFID-based School Management System as part of a web development internship project. The system aims to enhance the efficiency and security of school operations through the implementation of RFID technology. The project encompasses the design, development, and testing phases of the web application, emphasizing the integration of RFID technology for various school management tasks.

**Background and Motivation**

Traditional school management systems often involve manual record-keeping and administrative tasks, which can be time-consuming and error-prone. RFID technology offers an efficient solution by automating processes such as attendance tracking, access control, and library management. This project aims to leverage RFID technology to create a comprehensive School Management System that streamlines various operations.

**Objectives**

The main objectives of this project are as follows:

* Develop a web-based School Management System integrating RFID technology.
* Automate student and staff attendance tracking using RFID cards.
* Provide access control to secure areas within the school premises.
* Manage the school library and track book borrowing using RFID tags.
* Implement a fee payment tracking system.

**2. System Design**

**System Architecture**

The architecture follows a three-tier model: Presentation Layer (UI), Application Logic Layer (Back-End), and Data Layer (Database). The RFID readers are integrated into the Back-End layer to capture RFID card data.

**Use Case Diagram**

A use case diagram illustrates the interactions between users and the system. It includes actors like students, teachers, and administrators, and depicts use cases such as attendance tracking, access control, library management, and fee payment.

**Database Design**

The database schema includes tables for students, staff, attendance records, library books, fee payments, and user credentials. Relationships are established to maintain data integrity and enable efficient queries.

**3. Technologies Used**

* **Front-End Development:** HTML, CSS, JavaScript
* **Back-End Development:** Php
* **Database Management:** MySQL
* **RFID Integration:** RFID Readers, RFID Tags, RFID Libraries

**4. Features and Functionality**

**Student and Staff Attendance Tracking**

Students and staff members are issued RFID cards that they scan upon entering the school premises. The RFID readers record their attendance, which is then stored in the database.

**Access Control**

Certain areas within the school, such as laboratories or administrative offices, can be restricted using RFID-based access control. Authorized individuals can access these areas by scanning their RFID cards.

**Library Management**

Library books are tagged with RFID tags. Students can borrow and return books by scanning the RFID tags at self-service kiosks. The system updates the database accordingly.

**Fee Payment Tracking**

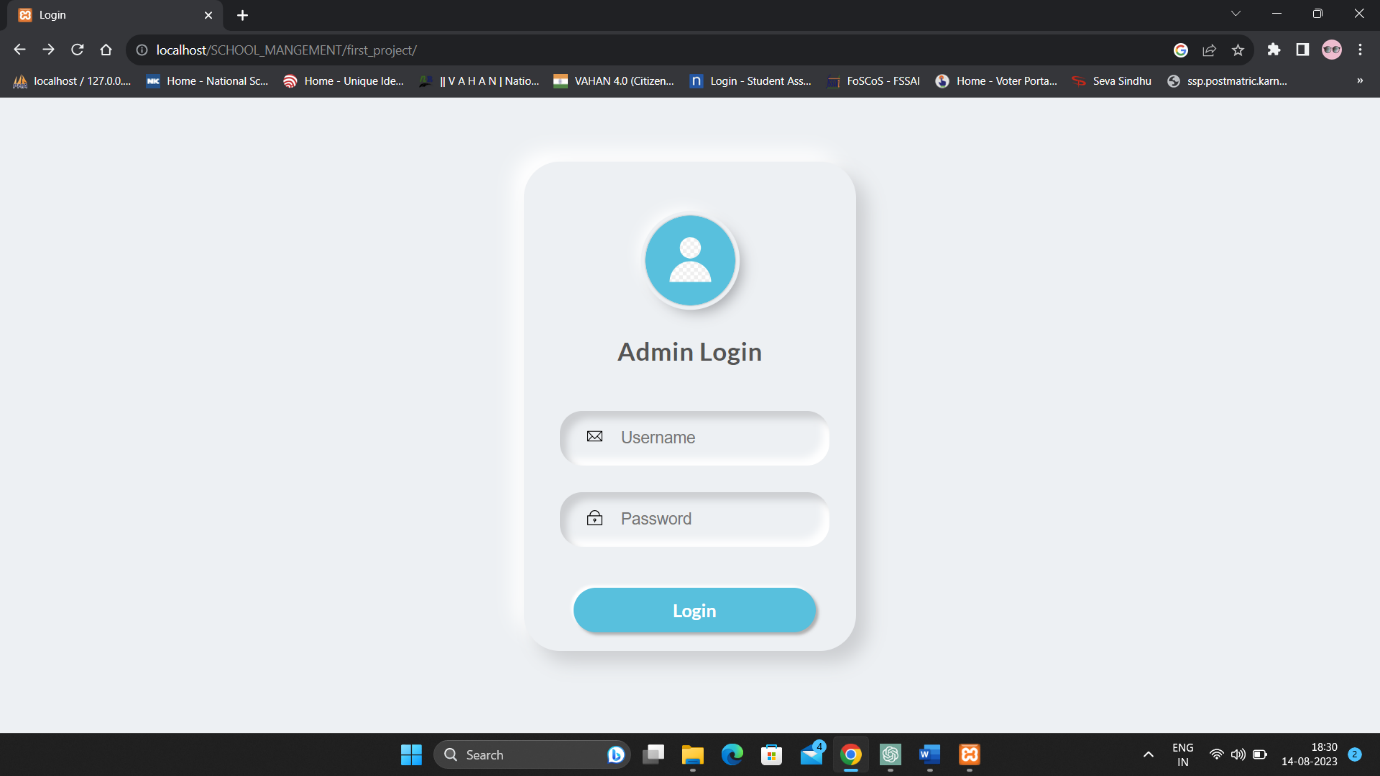
Parents can make fee payments through the web application. The system records the payment details and generates receipts. Students can also pay fines for overdue library books using the same mechanism.

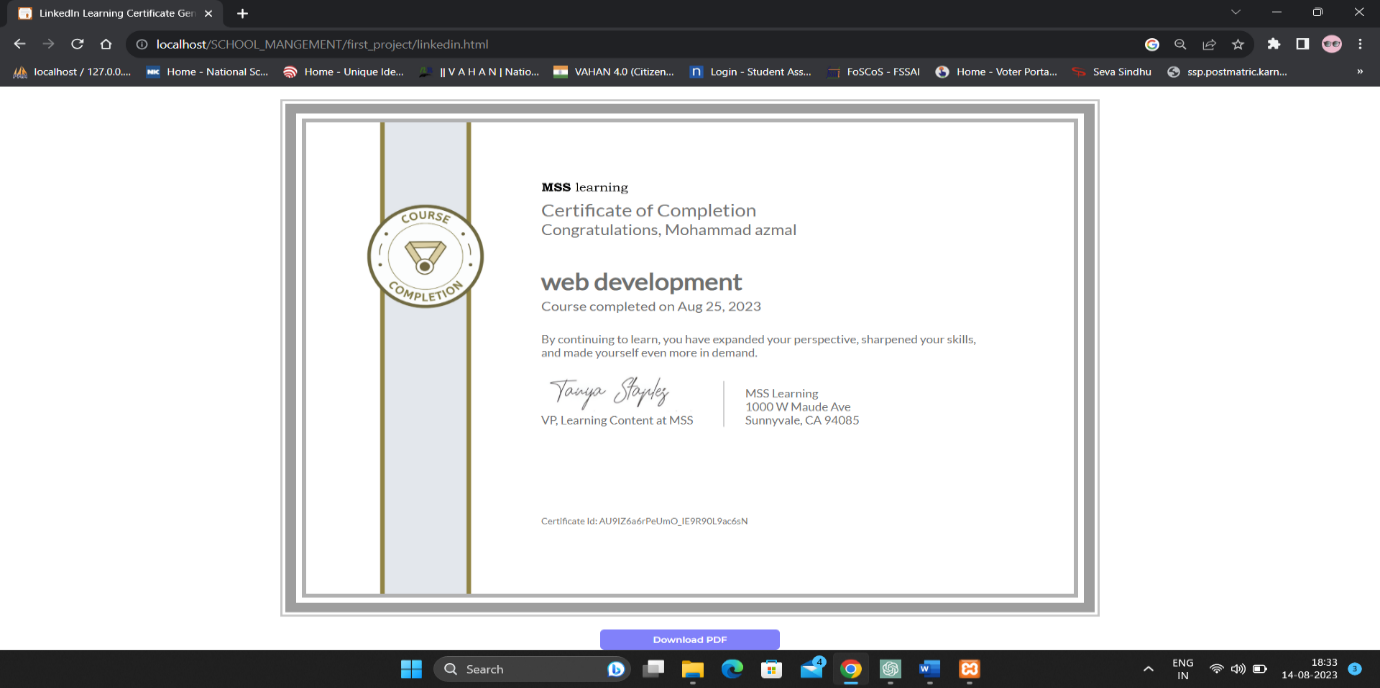
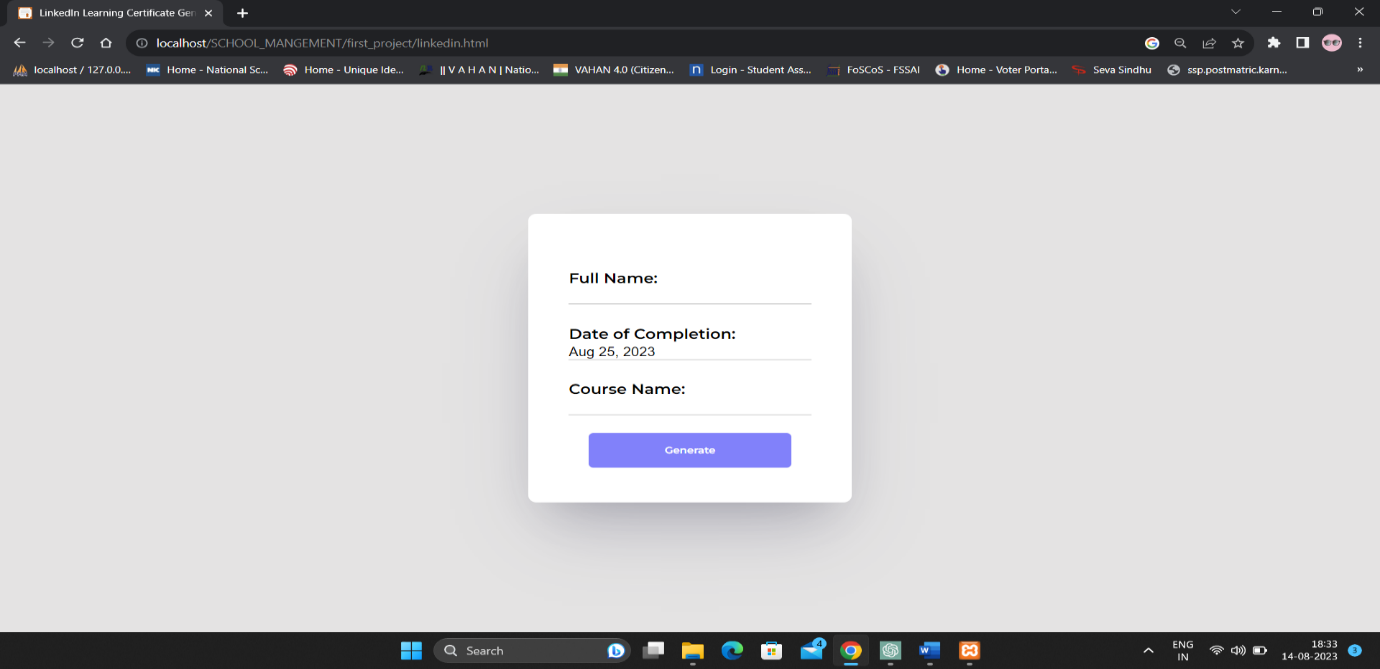
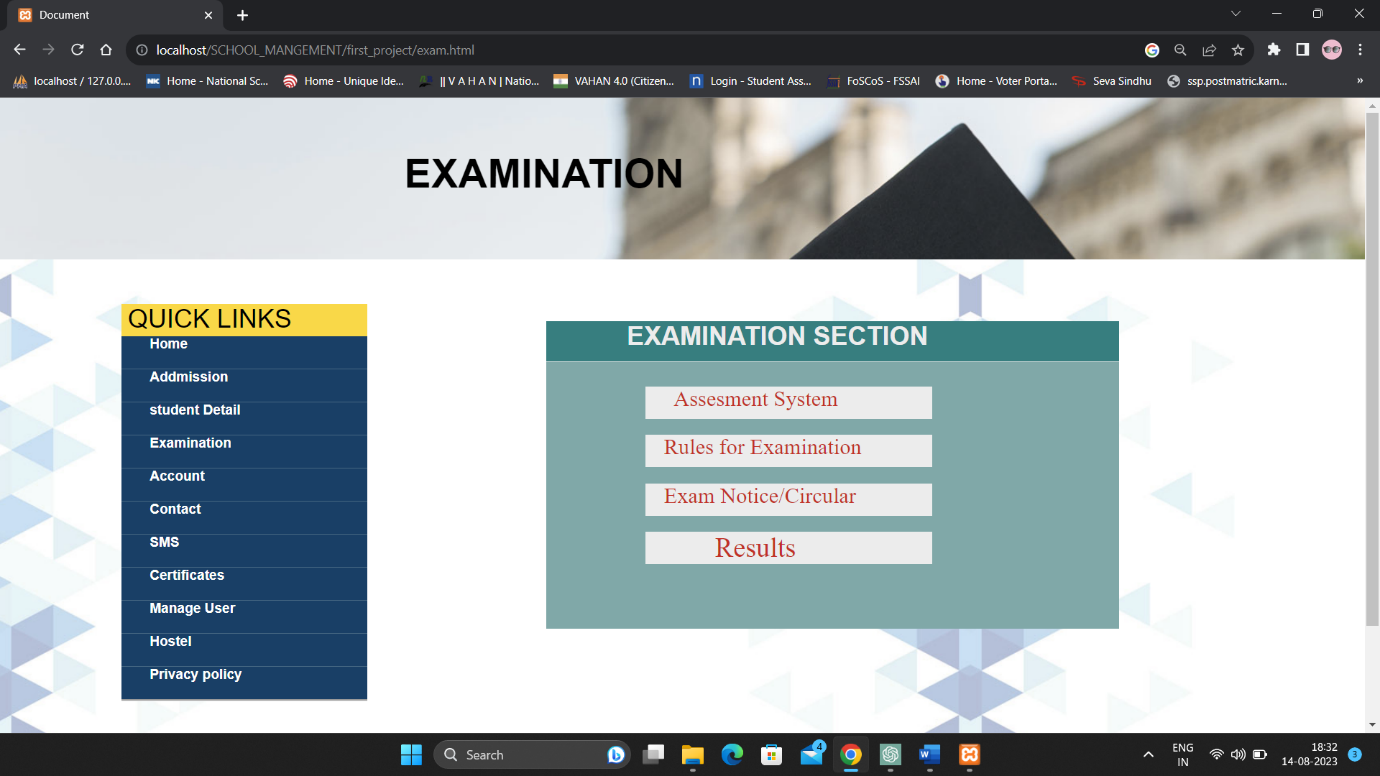
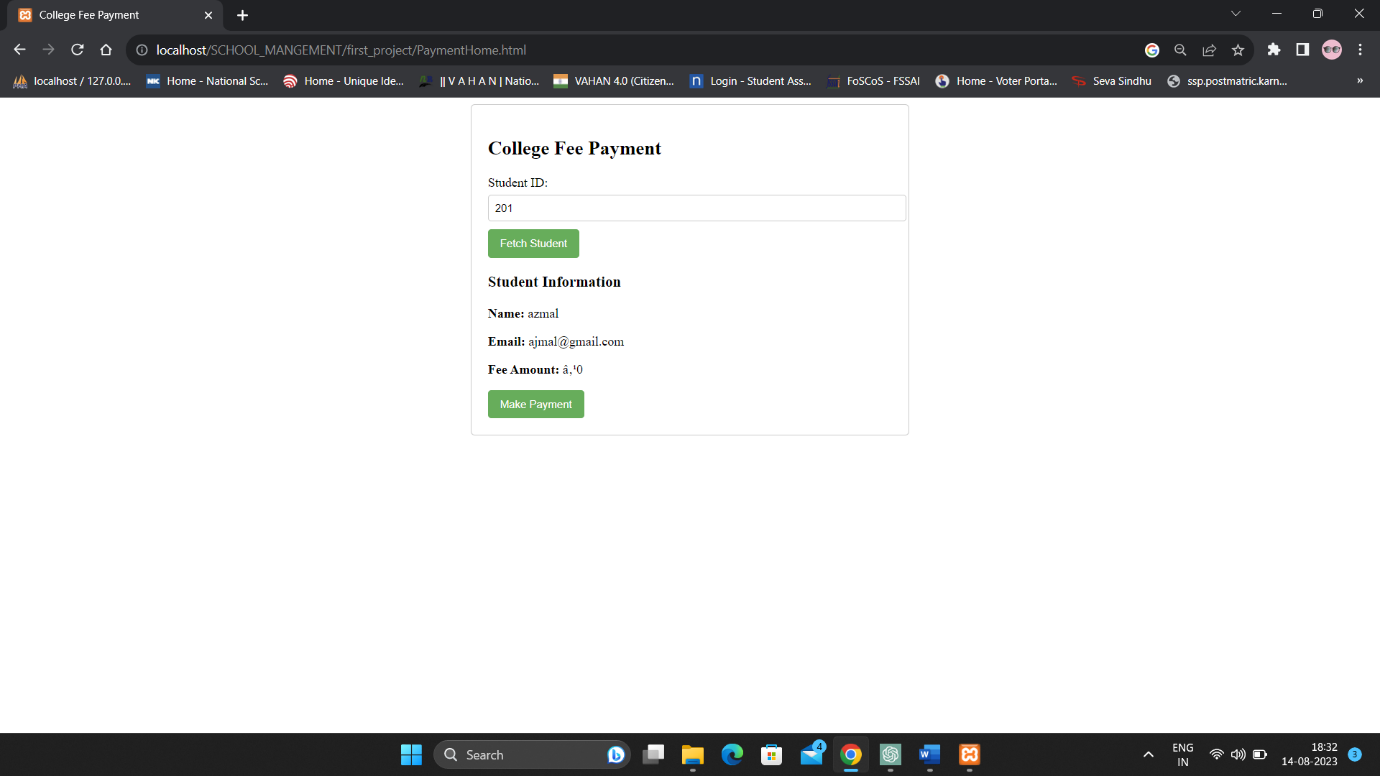
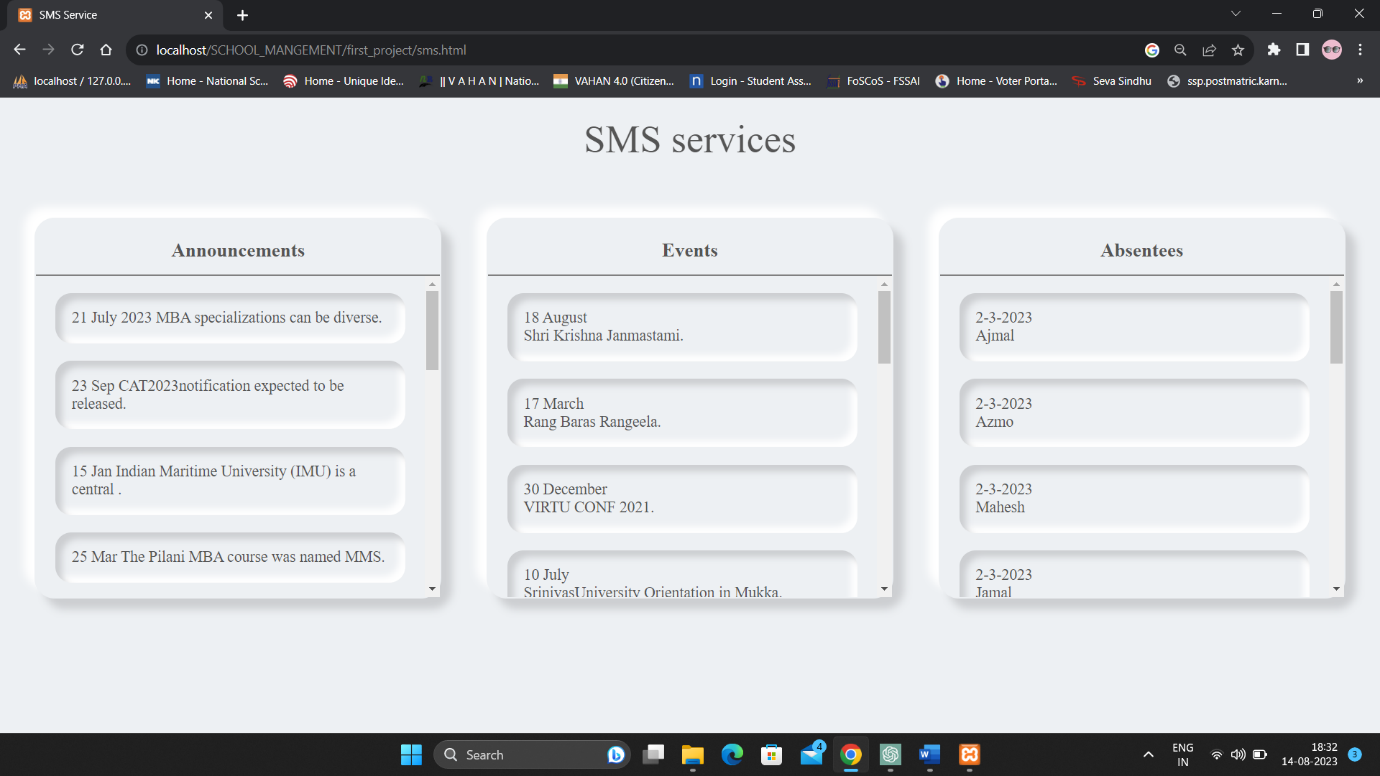
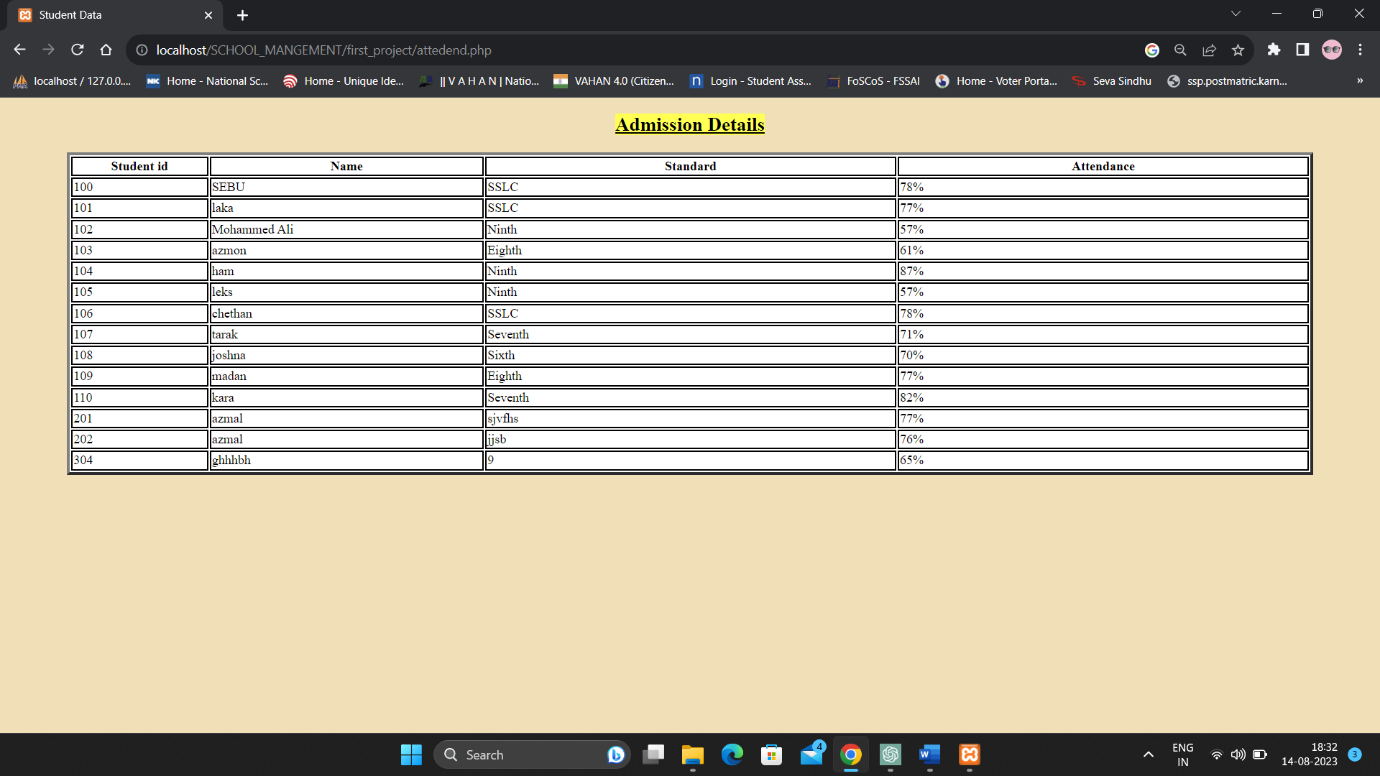
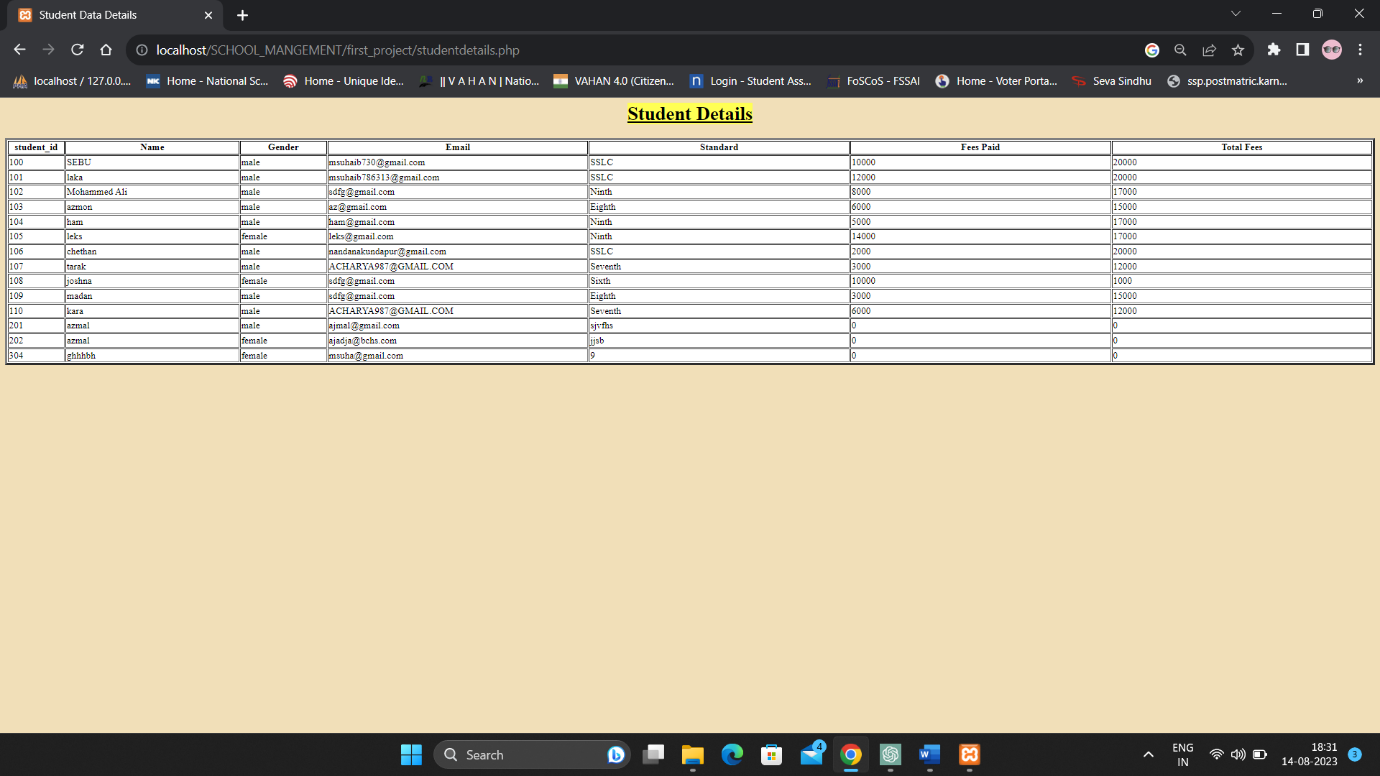
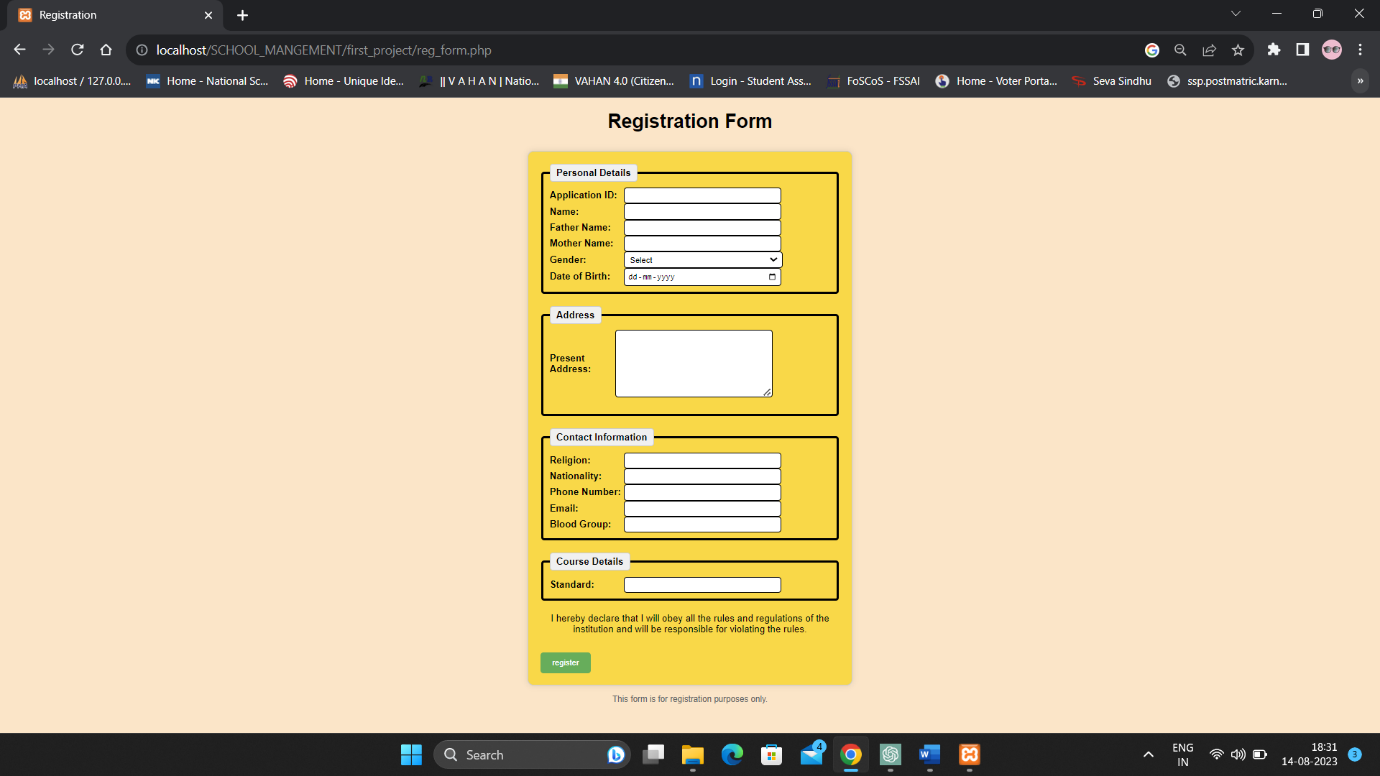
**5. Implementation**

The implementation phase involves:

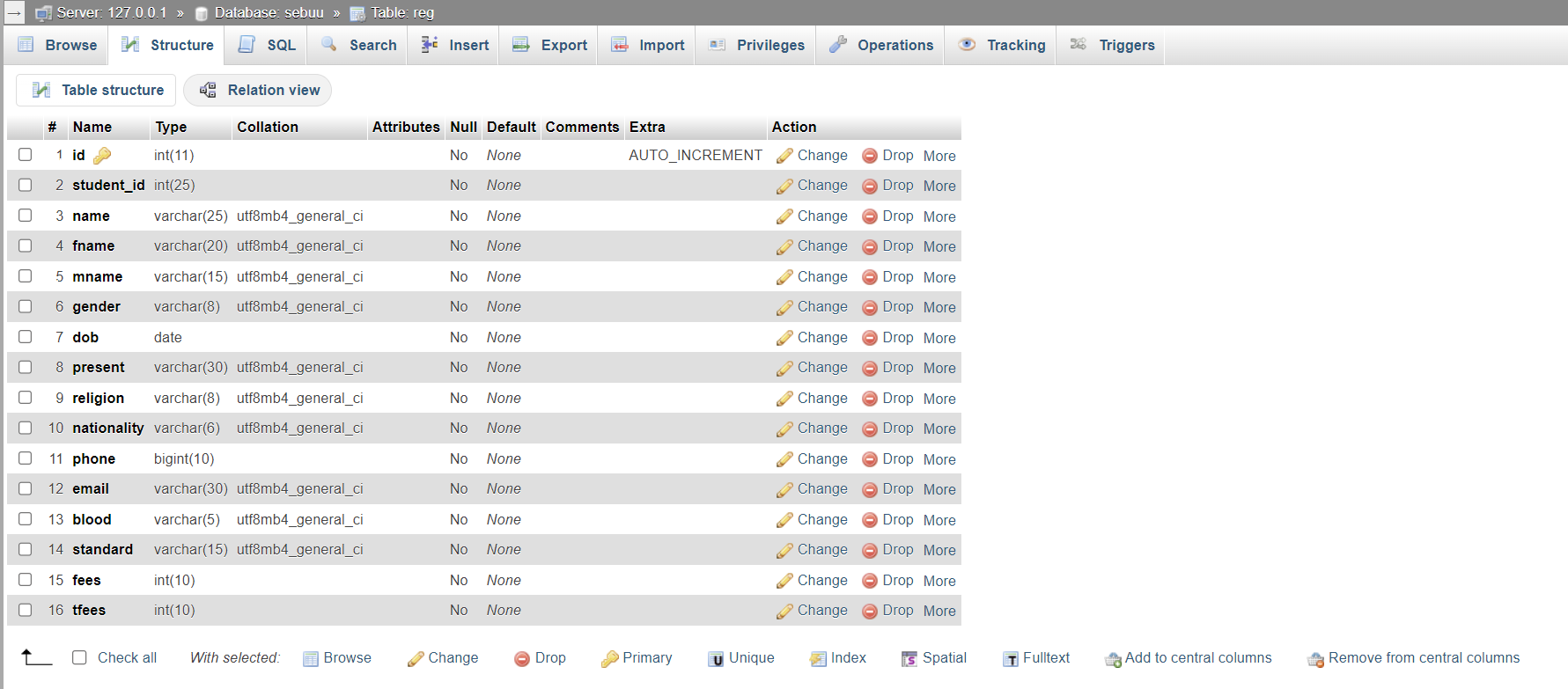
* Setting up user authentication and authorization mechanisms.
* Integrating RFID reader libraries into the Back-End.
* Designing and implementing database interactions for storing and retrieving data.
* Designing the user interface for various features.
* Implementing error handling and input validation to ensure data integrity.

**6. Screenshots**

****

****

**Database Screenshot:**

****

**7. Testing**

Testing is conducted at multiple levels:

* **Unit Testing:** Testing individual components and functions.
* **Integration Testing:** Testing interactions between different modules.
* **User Acceptance Testing:** Involving end-users to ensure the system meets their needs.

**8. Challenges Faced**

* Ensuring compatibility between different RFID reader models and the software.
* Addressing security concerns regarding the storage and transmission of RFID card data.
* Synchronizing real-time data updates across multiple devices.

**9. Future Enhancements**

* Developing a portal for parents to access their child's attendance and academic performance.
* Optimizing system performance for high traffic usage.
* Creating a mobile application for easier access to the system.

**10. Conclusion**

The RFID-based School Management System project successfully achieved its goals of automating school management processes using RFID technology. The system enhances efficiency, accuracy, and security in attendance tracking, access control, library management, and fee payment tracking. This project provided valuable experience in web development, RFID technology integration, and project management, contributing to the intern's skill set and professional growth. As this above project is not completed fully RFID based now.

**Above all the codes or scripts are uploaded in below github link:**

**https://github.com/LekhanaSamudara/Vitvara\_Internship.git**

Top of Form