# Introduction

**1.1 OVERVIEW**

Welcome to our DBMS CRUD (Create, Read, Update, Delete) mini project, where we delve into the dynamic world of database management systems (DBMS) and their role in facilitating efficient data manipulation. In this project, we explore the core functionalities of CRUD operations within the context of a robust and interactive web application. The DBMS serves as the foundation for storing, retrieving, and managing structured data, offering a structured approach to organizing information. Whether it's MySQL, PostgreSQL, or another system, the DBMS provides a reliable framework for data storage and retrieval.

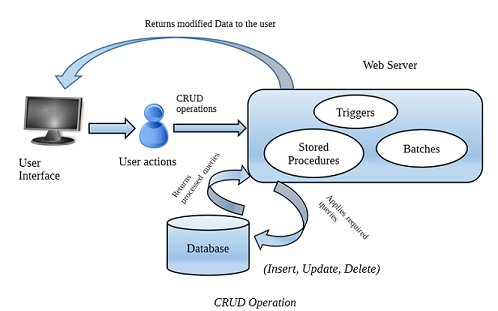
CRUD operations, consisting of Create, Read, Update, and Delete, form the backbone of our project, enabling users to interact with the database seamlessly. We explore each operation in depth, elucidating its significance in data management and manipulation. Through the power of SQL (Structured Query Language), we execute queries to perform CRUD operations as shown in [Fig 1] ensuring data integrity and consistency.

Fig 1: CRUD Operation

Our mini project showcases the seamless integration of PHP, HTML, CSS, and JavaScript to create a dynamic and user-friendly interface for interacting with the DBMS. PHP serves as the server-side scripting language, facilitating communication between the web server and the database.

HTML structures the content, CSS enhances the visual presentation, and JavaScript adds interactivity and responsiveness to the user interface. Together, these technologies provide a seamless user experience, enabling users to perform CRUD operations with ease.

The mini project's architecture follows best practices, with modular components for PHP scripts, CSS stylesheets, JavaScript files, and other assets. This modular approach enhances scalability and maintainability, allowing developers to extend the project's functionality with ease. Key features include robust input validation, error handling, and authentication mechanisms to ensure data security and integrity. Additionally, we prioritize responsive design, ensuring optimal performance across various devices and screen sizes.

**1.2 PROBLEM STATEMENT: BUILDING A CRUD WEB APPLICATION**

The College Event Management System Project defines such a operations to store, retrieve, update, and delete data in a user-friendly manner. To address this need, we propose the development of a dynamic CRUD (Create, Read, Update, Delete) web application for College Event Management System.

**1.3 PROBLEM DISCUSSION**

In this MySQL mini-project for a College Event Management System, we design a database schema comprising user\_details, event\_details, venue\_details, and booking\_details tables. Utilizing PHP and XAMPP server, we develop a CRUD application to create, read, update, and delete records within these tables, facilitating efficient management of college events. Challenges include ensuring data validation to prevent security vulnerabilities like SQL injection, implementing robust authentication mechanisms, and considering scalability for future system enhancements. The system aims to provide a user-friendly interface for organizers to create and manage events, participants to register for events, and administrators to oversee bookings and venue allocations. With MySQL handling data storage and PHP managing server-side logic, XAMPP serves as a local development environment for testing and debugging before deployment. Careful consideration is given to user interface design to enhance usability and navigation. The project emphasizes data integrity, security, and scalability to meet the diverse needs of the college event management process.

**1.4 PROGRAMMING ENVIRONMENTS**

A programming environment for databases encompasses a set of tools, languages, and frameworks used to interact with and manage databases effectively. At its core, developers utilize SQL (Structured Query Language) to communicate with the database management system (DBMS) and perform operations such as querying, updating, and managing data. Integrated Development Environments (IDEs) and text editors provide essential features like syntax highlighting, code completion, and debugging capabilities for writing SQL queries and database scripts.

**2. PROPOSED SYSTEM**

**2.1 PROPOSED SYSTEM FOR USERS**:

* **User Registration and Authentication**: Users should be able to register for an account and authenticate themselves securely to access the system.
* **Data Manipulation**: Users can perform CRUD (Create, Read, Update, Delete) operations on the data within their authorized scope.
* **Data Retrieval**: Users can search, filter, and retrieve information from the database based on their requirements.
* **Profile Management**: Users have the ability to manage their profiles, update personal information, and change account settings.
  + **Data Visualization**: Users may have access to visual representations of data such as Tables, or Descriptions for better understanding and analysis.

**2.2 PROPOSED SYSTEM FOR ADMINISTRATORS:**

* **User Management:** Administrators can manage user accounts, including creating new accounts, updating user profiles, and resetting passwords.
* **Access Control:** Administrators have the authority to define user roles, permissions, and access levels to control who can access and manipulate data within the system.
* **Data Management:** Administrators can perform advanced data management tasks such as Monitoring and Deleting from The Particular Table.
* **Monitoring and Reporting:** Administrators can monitor system performance, track user activities, and generate reports to analyse system usage and identify potential issues.

**3. SYSTEM REQUIREMENTS**

**3 .1 HARDWARE REQUIREMENTS:**

* PROCESSOR: Intel core i3 or above
* RAM: 512 MB Min
* HARD DISK: 40 GB min

**3.2 SOFTWARE REQUIREMENTS:**

* OPERATING SYSTEM: Windows 7 above Versions
* LANGUAGE(Server Side): PHP
* WEB TECHNOLOGY: HTML, CSS, JAVA SCRIPT
* WEB SERVER: XAMPP
* BACK END: MySQL

**3.3 FUNCTIONALITIES:**

Functional requirements for a database management system (DBMS) project outline the specific functionalities and features that the system must provide to meet the needs of its users and stakeholders. Here are some common functional requirements for a DBMS project:

* **User Authentication and Authorization**:
  + - The system should support user authentication mechanisms, such as username/password, two-factor authentication, or integration with external authentication providers.
    - Administrators should be able to define user roles and permissions to control access to database resources based on user roles and privileges.

* **Data Management**:
  + CRUD Operations: Users should be able to perform Create, Read, Update, and Delete operations on database records.
  + Data Validation: The system should enforce data validation rules to ensure the accuracy, completeness, and consistency of data entered into the database.
  + Referential Integrity: Implement referential integrity constraints to maintain data integrity and enforce relationships between related tables.

**4. SYSTEM DESIGN**

* 1. **DATA FLOW DESIGN:**

For the college event management project, designing a data flow diagram (DFD) entails mapping how data traverses through various components like processes, data stores, and external entities. In this scenario, data flows represent the movement of event-related information across the system.

Processes within the DFD signify distinct activities or operations conducted within the system. These could include managing events, handling user details, managing venues, and processing bookings. Each process has designated inputs and outputs, showcasing the flow of information within the system. The data stores within the DFD are repositories where data pertinent to events, users, venues, and bookings are stored and retrieved as required.

* A data flow model is diagrammatic representation of the flow and exchange of information within a system. Data flow models are used to graphically represent the flow of data in an information system by describing the process involved in transferring data.
* User can login into the Event Management System and can fill the user credentials such as user id and password.
* If the user is admin then he/she can manage the booking details ,(in the project the admin can only delete the booking details from the booking details table) according to the user ID. Can keep monitoring on booked details. Shown in [Fig 2].

Book a Event

Event Details

Edit User Information

Admin Management

Deleting Booking

Details

Login Page

Fig 2: Flow of the Design (CEMS)

**4.2 ENTITY RELATIONAL DIAGRAM:**

User\_details

Event\_details

Booking\_details

Venue\_details

Register\_by

has

has

Can\_view

N

1 N

N 1 N

M

N M N

Caption : Entity Relations Diagrams which shows relation between tables

**4.3** **SCHEMA DIAGRAM:**

user\_details:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UserNo | UserID | UserFullName | UserPassword | UserType | UserEmail | UserImage | UserImageName |

venue\_details:

|  |  |  |
| --- | --- | --- |
| VenueID | VenueName | VenueInfo |

event\_details:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EventID | EventName | EventDate | EventTime | EventCategory | EventDescription | EventTicketPrice | VenueID | UserID |

booking\_details:

|  |  |  |  |
| --- | --- | --- | --- |
| BookingID | BookingTimeStamp | UserID | EventID |

Caption : Schema Diagram which Represents tables with their keys

**4.4 TABLE DESCREPTION:**

**User Details:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **constraints** | **Description** |
| UserNo | Int | 10 | Not Null | User No |
| UserID | Varchar | 10 | Primary key | User ID |
| UserFullName | Varchar | 80 | Null | Name of the user |
| UserPassword | Varchar | 12 | Null | User Password |
| UserType | Varchar | 11 | Null | User Type |
| UserEmail | Varchar | 50 | Null | Email of the user |
| UserImage | Longblob | No | Null | User image |
| UserImageName | Varchar | 50 | Null | User image name |

**Venue Details:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** |  | **Size** | **constraints** | **Description** |
| VenueID | Int |  | 11 | Primary key | Venue ID |
| VenueName | Varchar |  | 80 | Null | Venue name |
| VenueInfo | Varchar |  | 50 | Null | Venue information |

**Booking Details:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **constraints** | **Description** |
| BookingID | Int | 10 | Primary Key | Booking ID |
| BookingTimeStamp | Timestamp | No | Null | Booking timestamp |
| UserID | Varchar | 10 | Primary key | User ID |

**Event Details:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** |  | **Size** | **constraints** | **Description** |
| EventID | Int |  | 10 | Primary key | Event ID |
| EventName | varchar |  | 100 | Null | Name of the event |
| EventDate | Date | No | | Null | Event date |
| EventTime | Time | No | | Null | Event time |
| EventCategory | Varchar | 100 | | Null | Type of event |
| EventDescription | Text | No | | Null | Event details |
| EventTicketPrice | Double | 20 | | Null | Price of the event |
| VenueID | Int | 11 | | Primary key | Venue ID |
| UserID | Varchar | 10 | | Primary key | User ID |

**5. DETAILS OF TECHNOLOGIES USED**

**5.1 XAMPP SERVER:**

## **What is XAMPP?**

Let’s take a closer look at what XAMPP is and how it works:

For local development and testing needs, XAMPP provides a full-featured web server solution encompassing Apache, MySQL, PHP and Perl as its letters spell out its name – ideal for local testing in environments without server infrastructures such as Cloud flare or similar solutions. XAMPP also comes equipped with the web-based utility php MyAdmin for administering MySQL databases easily, while providing developers with an easily installable and preconfigured environment to quickly build local web servers for testing purposes. Note that XAMPP should not be used in production settings but is rather tailored for standalone development lab environments.

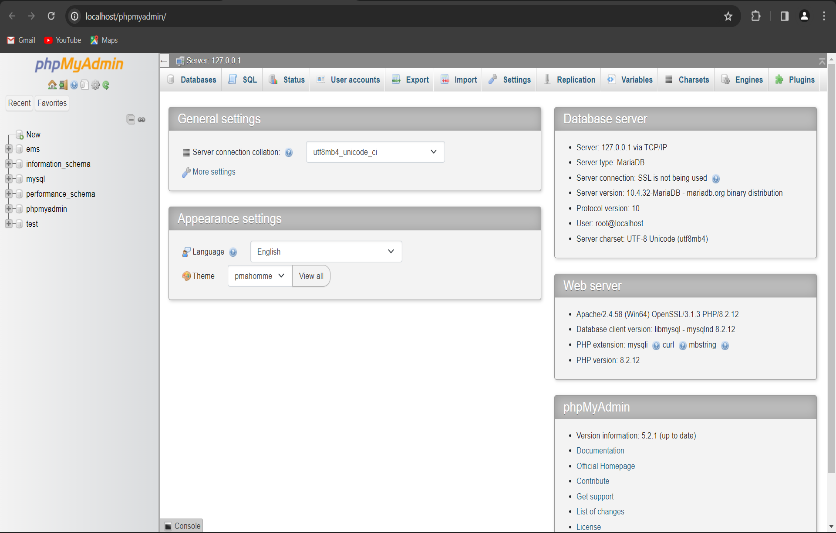


Fig 3: PHP my admin Page

### **XAMPP Definition**

Programmers can create and test websites locally on their PC without the need of external server access or internet connectivity. As it streamlines the setup and management process for local web development software, this approach is widely appreciated among beginners as well as veterans alike.

## **What does XAMPP do and what is XAMPP used for?**

Here’s a quick rundown of the XAMPP suite of tools and what they perform. XAMPP provides an ideal local web server environment on your PC for testing and developing websites and web apps. Developers can build and test websites/apps without worrying about connectivity to an outside server if using this tool locally on PCs without connecting directly. Some of its most frequently utilized functions include.

When it comes to developing and testing websites and apps online, XAMPP provides developers with an efficient tool. No more needing to be online while debugging code or testing functionality or design.

**5.2 MySQL:**

## **MySQL is a relational database management system**

Databases are the essential data repository for all software applications. For example, whenever someone conducts a web search, logs in to an account, or completes a transaction, a database system is storing the information so it can be accessed in the future.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one to one, one to many, unique, required, or optional, and “pointers” between different tables. The database enforces these rules so that with a well-designed database your application never sees data that’s inconsistent, duplicated, orphaned, out of date, or missing. The “SQL” part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

## **MySQL is open source**

[Open source](https://developer.oracle.com/open-source/what-is-open-source/) means it’s possible for anyone to use and modify the oftware. Anybody can download MySQL software from the internet and use it without paying for it. You can also change its source code to suit your needs. MySQL software uses the GNU General Public License (GPL) to define what you may and may not do with the software in different situations.

If you feel uncomfortable with the GNU GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from Oracle. See the MySQL Licensing Information section for more information.

## **MySQL works in client/server or embedded systems**

MySQL Database is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application-programming interfaces (APIs). We also provide MySQL as an embedded multithreaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

## **5.3 MYSQL BENEFITS**

MySQL is fast, reliable, scalable, and easy to use. It was originally developed to handle large databases quickly and has been used in highly demanding production environments for many years. Although MySQL is under constant development, it offers a rich and useful set of functions. MySQL’s connectivity, speed, and security make it highly suited for accessing databases on the internet.

**5.3.1 MYSQL’S KEY BENEFITS INCLUDE:**

**Ease of use:** Developers can install MySQL in minutes, and the database is easy to manage.

**Reliability:** MySQL is one of the most mature and widely used databases. It has been tested in a wide variety of scenarios for more than 25 years, including by many of the world’s largest companies. Organizations depend on MySQL to run business-critical applications because of its reliability.

**Scalability:**MySQL scales to meet the demands of the most accessed applications. MySQL’s native replication architecture enables organizations such as Facebook to scale applications to support billions of users.

**Performance:** MySQL HeatWave is faster and less expensive as demonstrated by multiple standard industry benchmarks, including TPC-H, TPC-DS, and CH-benCHmark

**High availability:** MySQL delivers a complete set of native, fully integrated replication technologies for high availability and disaster recovery. For business-critical applications, and to meet service-level agreement commitments, customers can achieve

* Recovery point objective = 0 (zero data loss)
* Recovery time objective = seconds (automatic failover)

**Security:** Data security entails protection and compliance with industry and government regulations, including the European Union General Data Protection Regulation, the Payment Card Industry Data Security Standard, the Health Insurance Portability and Accountability Act, and the Defense Information Systems Agency’s Security Technical Implementation Guides. MySQL Enterprise Edition provides advanced security features, including authentication/authorization, transparent data encryption, auditing, data masking, and a database firewall.

**Flexibility:**The MySQL Document Store gives users maximum flexibility in developing traditional SQL and NoSQL schema-free database applications. Developers can mix and match relational data and JSON documents in the same database and application.

**5.4 PHP:**

## **What is PHP? & What Does PHP Mean?**

PHP stands for “Hypertext Pre-processor,” and earlier, it was known as “Personal Homepage.” The best part developers love is embedding the same into HTML, which is simple for newcomers and delivers programmers with many advanced features. It helps write concise code and develop more exemplary apps with improved RFCs (Request for Comment).

## **What Are the Uses Of PHP?**

There are multiple uses of PHP that make it a strong scripting language option for web development. It operates over the web server and then processes all client requests into HTML files. Additionally, PHP is a general-purpose language; developers can use it to code for different applications. Another good use of PHP is that it is compatible with different operating systems, like macOS, Windows, and Linux. Furthermore, it is possible to use PHP for varying web servers, e.g., OpenBSD, Nginx, and Apache. PHP is compatible with different cloud environments like Amazon AWS and Microsoft Azure.

Overall, PHP has a flexible structure and can general different types of formats, like PNG, JPEG, GIF, and PDFs.

**Essentially, PHP can help with two applications:**

* **Command-line scriptin**g- You can use PHP script for the command-line scripting to conduct administrative work like PDF generation and email transfer.
* **Server-side scripting**– Web developers can use PHP to develop dynamic web apps and websites.

## **Top 5 Practical Applications of PHP**

### **1. Web Development**

PHP is identified as one of the finest solutions for developing websites for any purpose. Its simplicity and ease of use drive top user experience and seamless integration with HTML code, eliminating the barrier in generating dynamic content and helping build blocks for modern web apps.

### **2. Ecommerce Websites**

PHP stands out when experiencing rich features, especially for online shopping platforms. You get help with easy language integration, payment gateway capabilities, databases such as MySQL, and shopping carts, which aid in secure and coherent transactions. If one is comfortable with popular frameworks of Magneto or WooCommerce. This will streamline your online store development with its pre-built modules that help retailers with product catalog management, order processing, and customer management. In addition, you make your practices easier by enabling real-time inventory management, delivering customers personalized shopping experiences, and further helping with user engagement and retention.

### **3**. **Content Management Systems**

The world’s top CMS platforms, WordPress, Joomla, and Drupal, depend on PHP, which delivers the benefits of handling dynamic content, database integration, and user authentication. Further, using these CMS platforms, users have experienced creating, updating, and managing data consistently, including modifying the published content using web browsers and easy data-based integration.

### **4. Social Media Platforms**

Online community forums deal with massive user interactions, content management, and data processing. So, this scripted language assists in real-time data processing, including rich media content processing, and aids in sessions and cookies, making it one of the top options for developing social media applications.

### **5. Enterprise Applications**

PHP is a treasure for building enterprise-level applications. It efficiently contributes to and manages their data, user roles, and permissions, making it one of the best options for customer relationship management (CRM), human resources portals, and inventory management platforms. Further, enterprises can also leverage populist frameworks like Lavavel and Symfony that provide them with significant tools to develop robust apps for whatever unique requirements businesses seek to fulfill.

**Advantages of PHP/PHP Features:**

To understand PHP, it is vital to address its features. There are multiple notable ones available:

* **Cross-Platform**– PHP can efficiently work on different operating systems, like Linux, Windows, and Mac.
* **Simple to Grasp**– Another benefit of this language is that even beginners can quickly adapt to PHP programming. It is one of the more accessible languages to pick up.
* **Open-Source**– Anyone can use PHP for their web development requirements since the original code is open-source. Anyone can adapt it and build upon the code further.
* **Supportive Broad Developer Community**– Access PHP’s vibrant online community, with up-to-date tutorials, documentation, and FAQs available for all developers.
* **Multi-Database Syncing**– You can connect this scripting language to non-rational and rational databases, like MongoDB, Postgress, and MySQL.
* **High-Quality Performance**– PHP operates faster than other scripts like ASP and JSP since it uses personalized memory. Therefore, the loading time and workload are relatively shorter.
* **Embedded**– Developers can embed PHP scripts with HTML scripts and tags.
* **Loosely-Typed Language**– Developers can use a variable without declaring the datatype during scripting and is automatically used during execution depending on the data type containing the related valuation.
* **Syntax Familiarity**– The syntax of PHP is straightforward to understand and is suitable for different web development needs.
* **Error Reporting**– There are pre-defined error reporting constants available under PHP. It can produce real-time error warnings, like E\_WARNING, E\_PARSE, and E\_STRICT.
* **Better Control Quality**– There is no need for excess or long codes or scripts with PHP. It has higher control on web solutions; developers can make changes without extra coding.
* **High-Security**– Among the different scripting language types available, PHP is one of the most secure options with multiple security levels in place. The multi-layered structure safeguards against hacking attempts, malware, etc.
* **Compatible with The Cloud-**cloud computing is at its apex, eliminating managing, patching, and scaling clunky servers. PHP applications are deployed on cloud servers, and they assist in achieving excellent scalability as they support different cloud services.
* **Cost-Effective-** being an open source makes it completely free. It can work with databases like MySQL, PostgreSQL, and Apache, reducing the costs of creating a website with PHP.

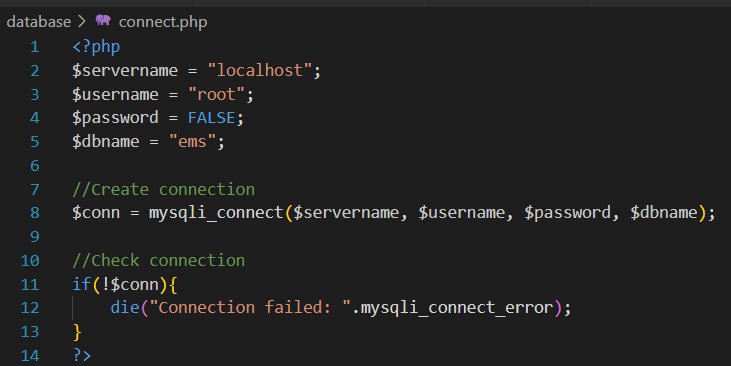


Fig 4: PHP Code for database connection

**5.5 HTML:**

## **What is HTML?**

To understand "HTML" from front to back, let's look at each word that makes up the abbreviation:

**Hypertext**: text (often with embeds such as images, too) that is organized in order to connect related items

**Markup**: a style guide for typesetting anything to be printed in hardcopy or soft copy format

**Language**: a language that a computer system understands and uses to interpret commands.

HTML determines the structure of web pages. This structure alone is not enough to make a web page look good and interactive. So you'll use assisted technologies such as CSS and JavaScript to make your HTML beautiful and add interactivity, respectively.

In this case, I like to break down the three technologies – HTML, CSS, and JavaScript – this way: they are like a human body.

* HTML is the skeleton,
* CSS is the skin,
* and JavaScript is the circulatory, digestive, and respiratory systems that brings the structure and the skin to life.

You can also look at HTML, CSS, and JavaScript this way: HTML is the structure of a house, CSS is the interior and exterior decor, and JavaScript is the electricity, water system, and many other functional features that make the house livable.

## **HTML Tags**

Since HTML defines the markup for a particular web page, you'll want the text, images, or other embeds to appear in certain ways.

For example, you might want some text to be big, other text to be small, and some to be bold, italic, or in bullet point form.

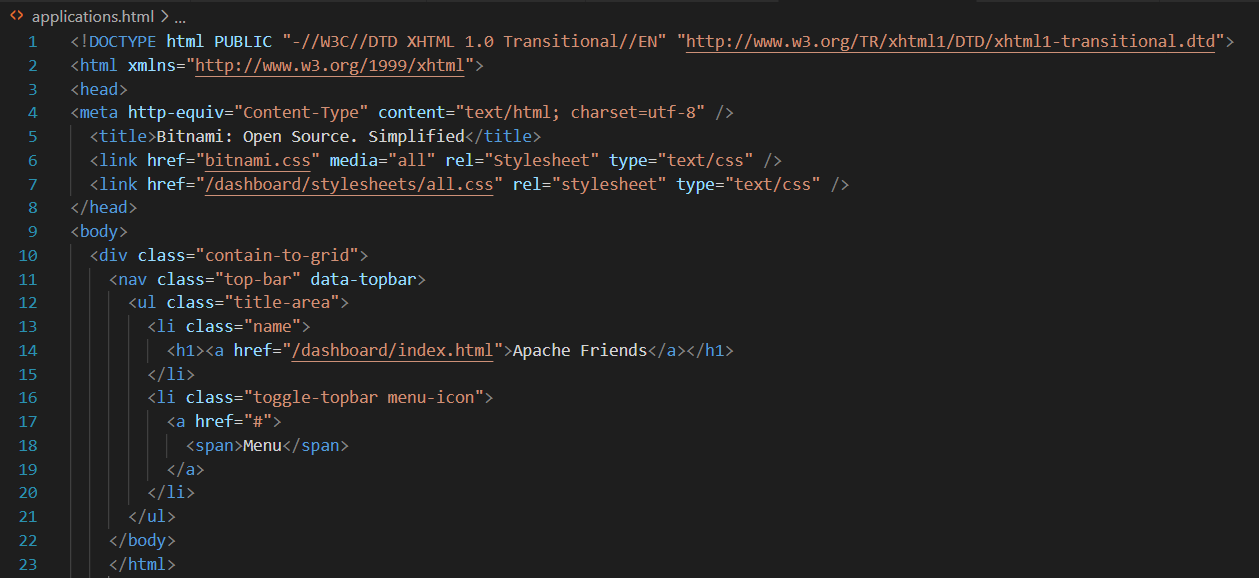
HTML has "tags" that let you get this done For example[Fig 5]. So, there are tags to create headings, paragraphs, bolded words, italicized words, and more.

Fig 5: HTML Code of Home Page

**5.6 CASCADING STYLE SHEET**

**What is CSS?**

**CSS** (Cascading Style Sheets) allows you to create great-looking web pages, but how does it work under the hood? This article explains what CSS is with a simple syntax example and also covers some key terms about the language.

**CSS syntax**

CSS is a rule-based language — you define the rules by specifying groups of styles that should be applied to particular elements or groups of elements on your web page.

For example, you can decide to have the main heading on your page to be shown as large red text. The following code shows a very simple CSS rule that would achieve the styling described above:

* In the below example, the CSS rule opens with a selector. This *selects* the HTML element that we are going to style. In this case, we are styling level one headings (h1).

h1 {

color: red;

font-size: 5em;

}

* We then have a set of curly braces { }.
* This example contains two declarations, one for color and the other for font-size. Each pair specifies a property of the element(s) we are selecting (h1 in this case), then a value that we'd like to give the property.

CSS properties have different allowable values, depending on which property is being specified. In our example, we have the color property, which can take various color values. We also have the font-size property. This property can take various size units as a value.

**6. SNAPSHOTS:**

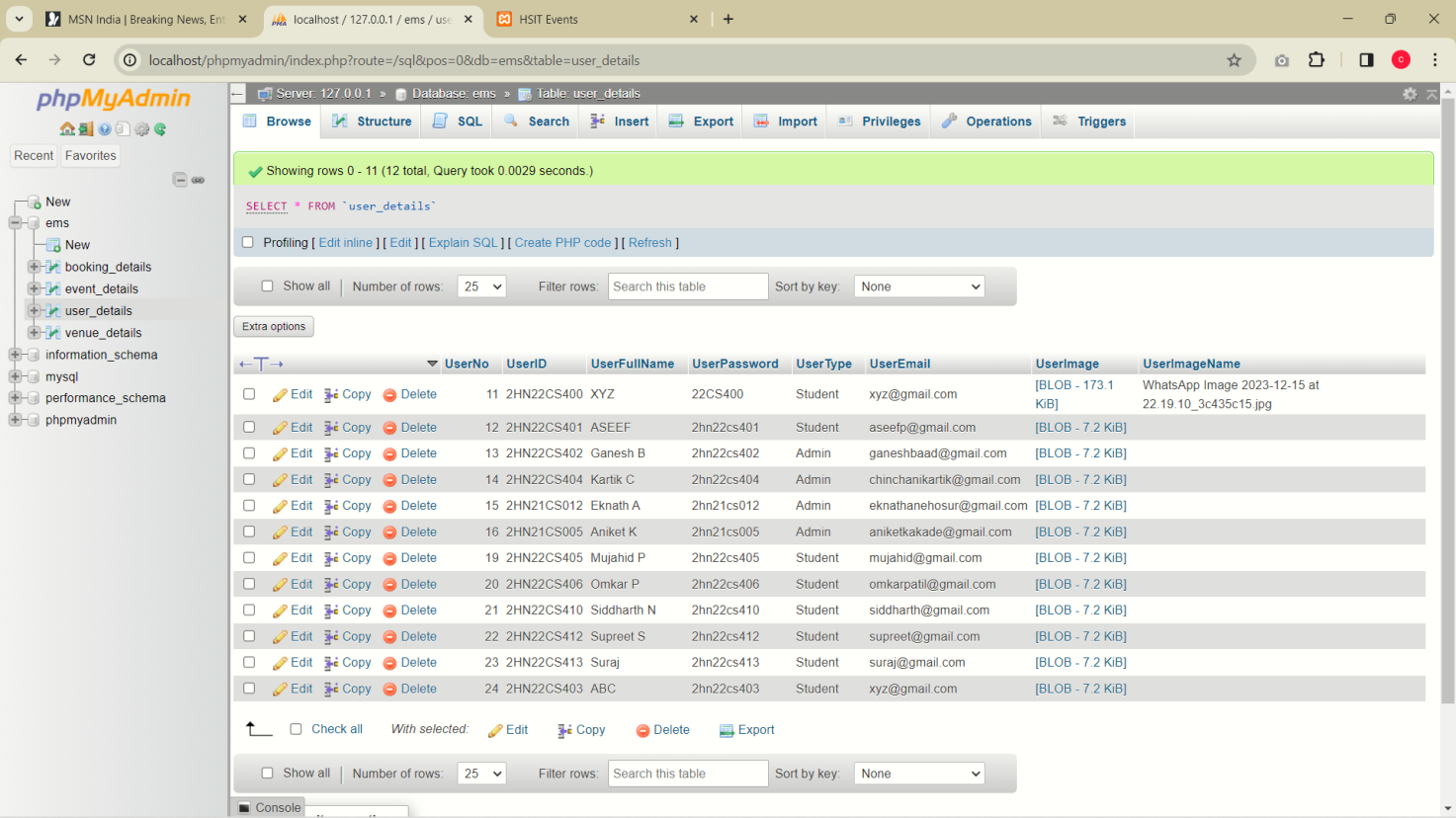
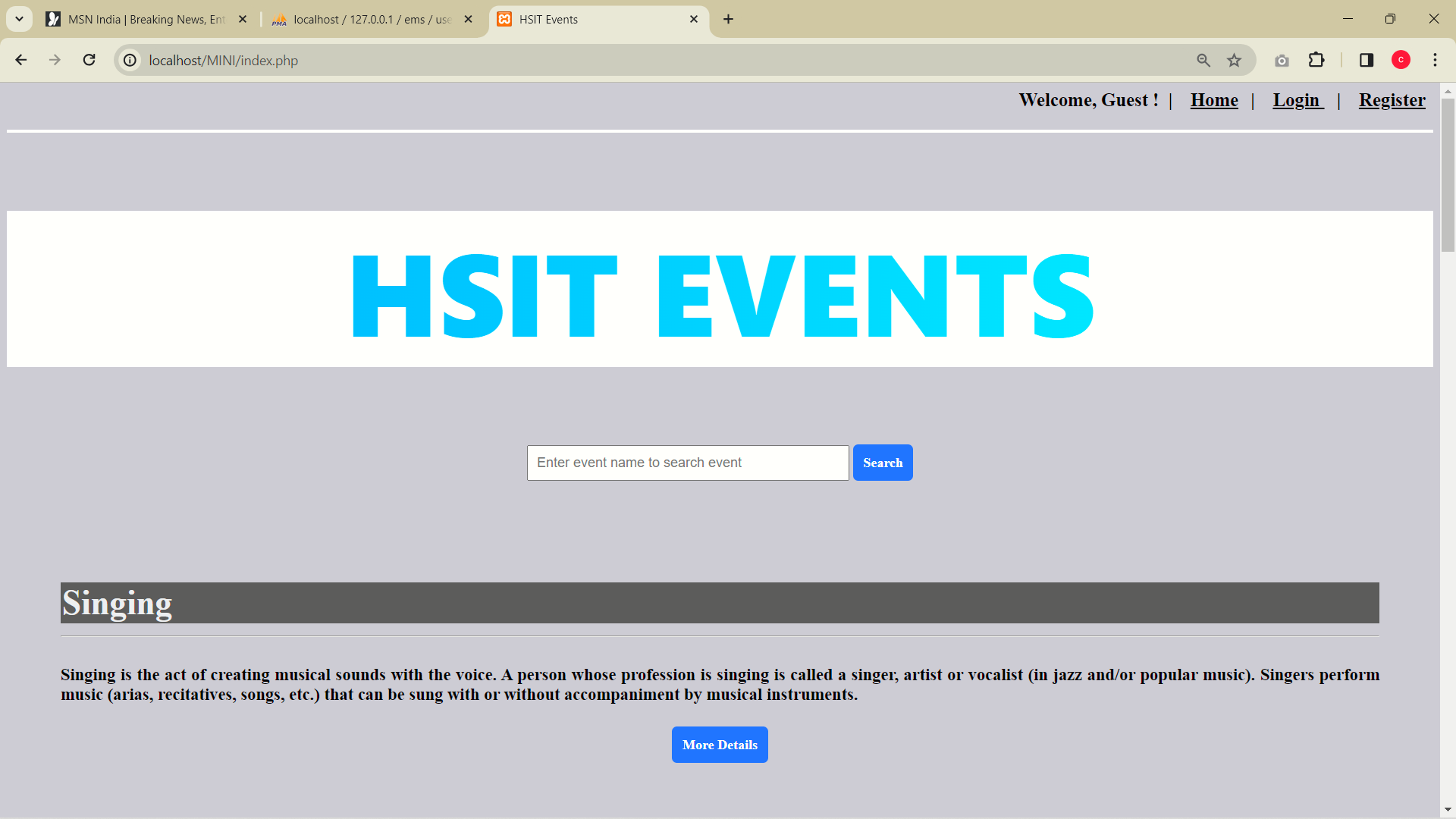
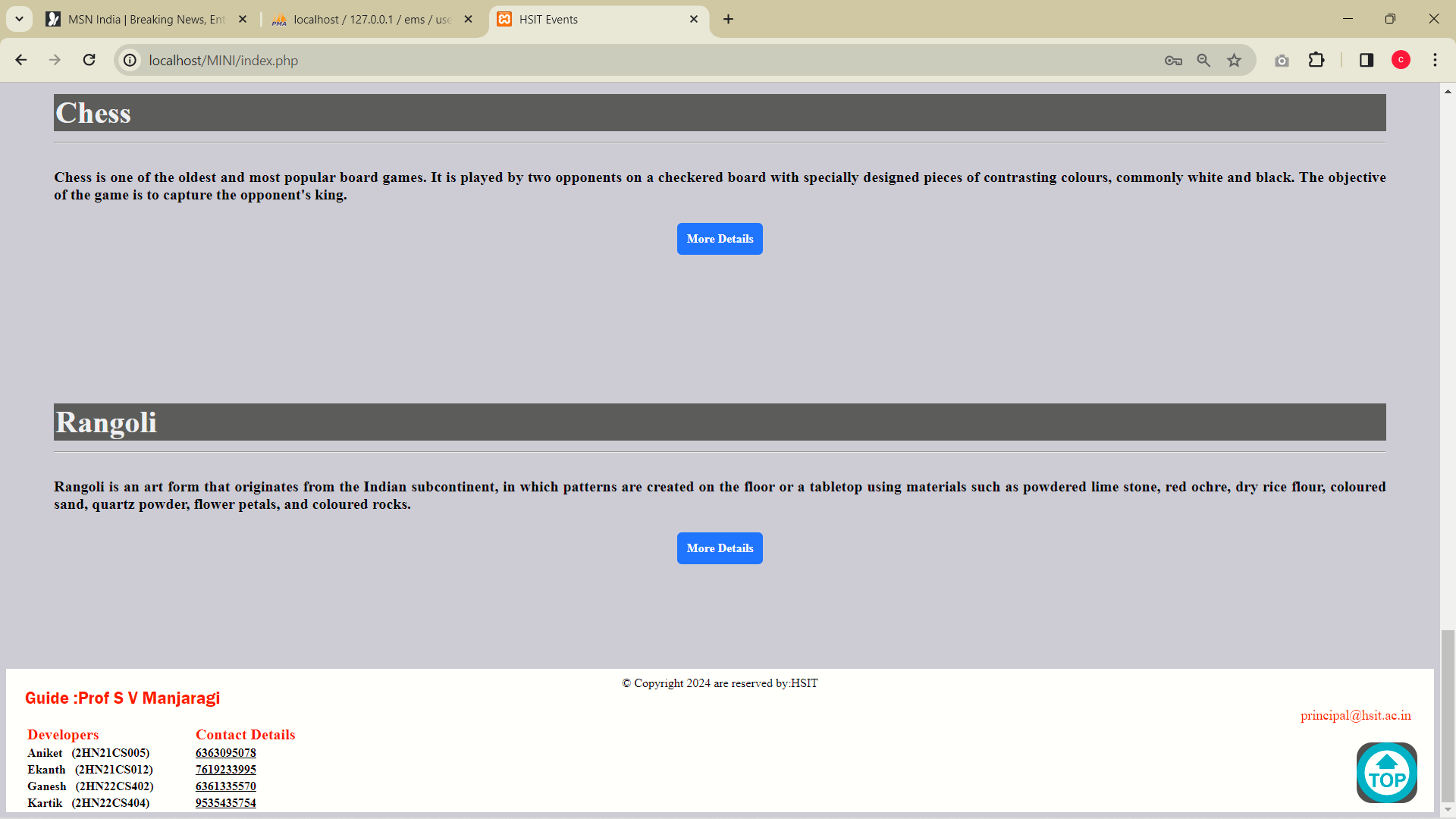
****

Fig 6: PHP Home Page and CEMS Home Page

Caption: The phpMyadmin consisting college event management system along with tables.

And CEMS with Home Page.

****

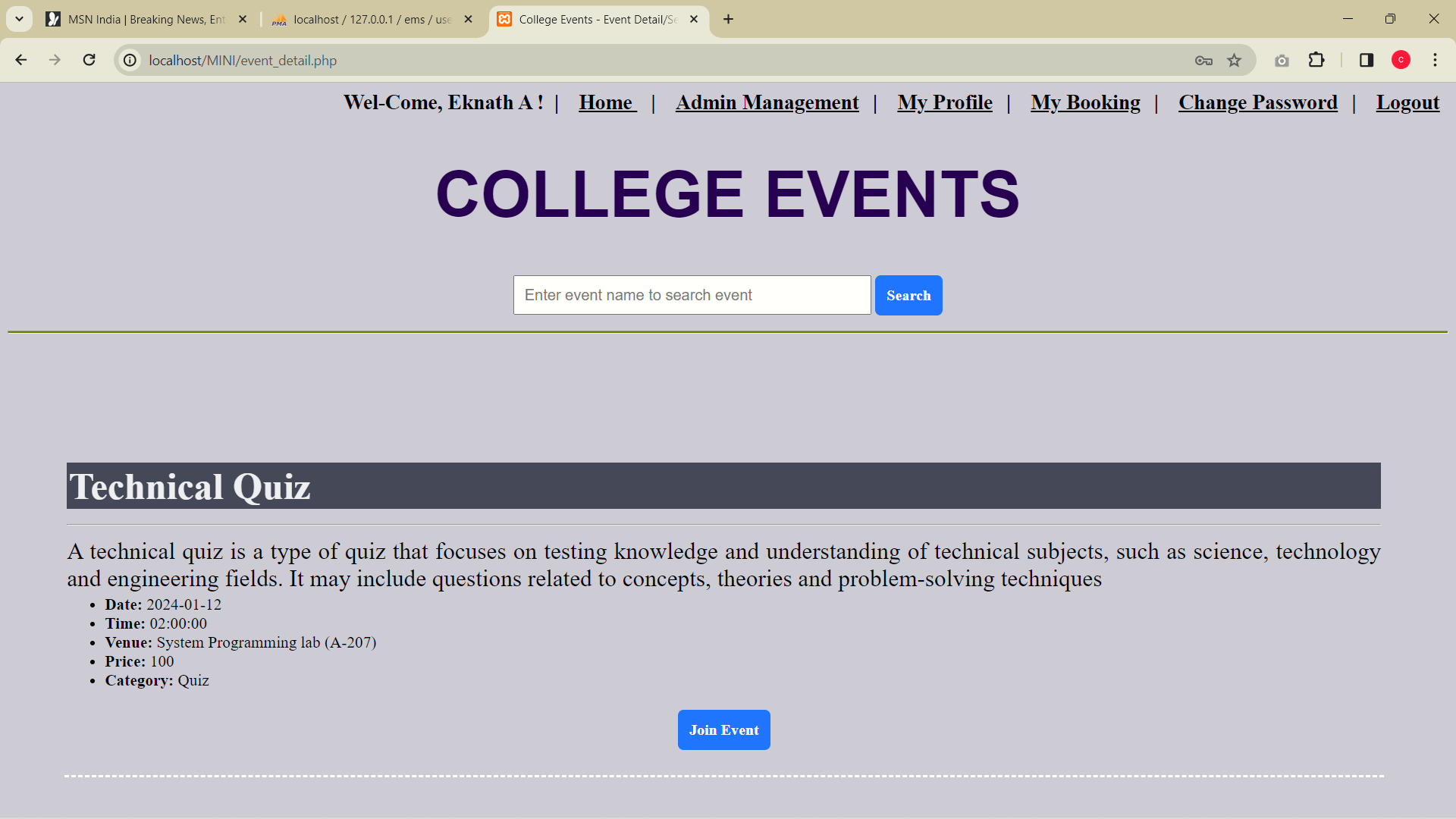
Fig 7: Footer in Home Page

Fig 8: Event Details

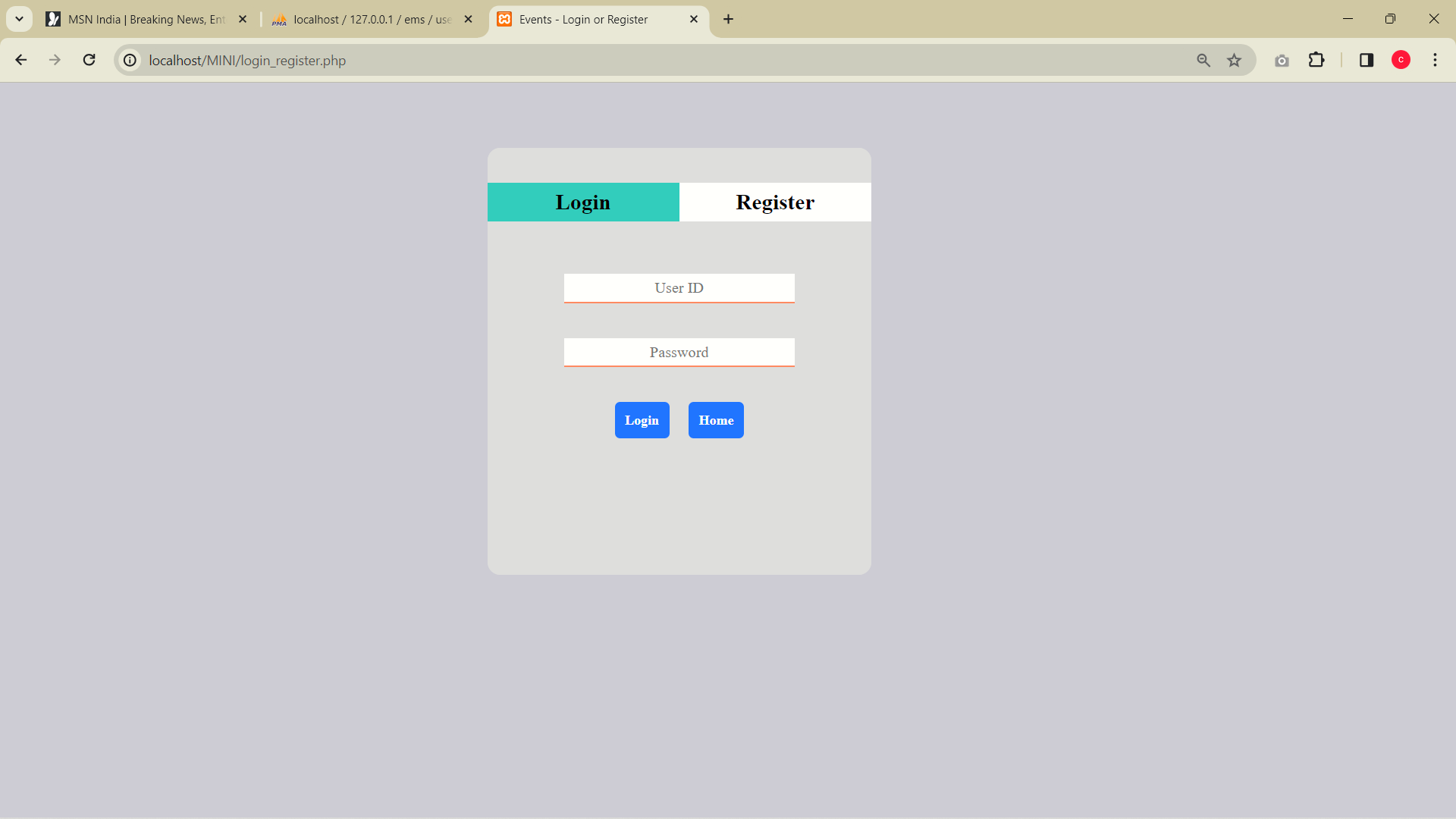
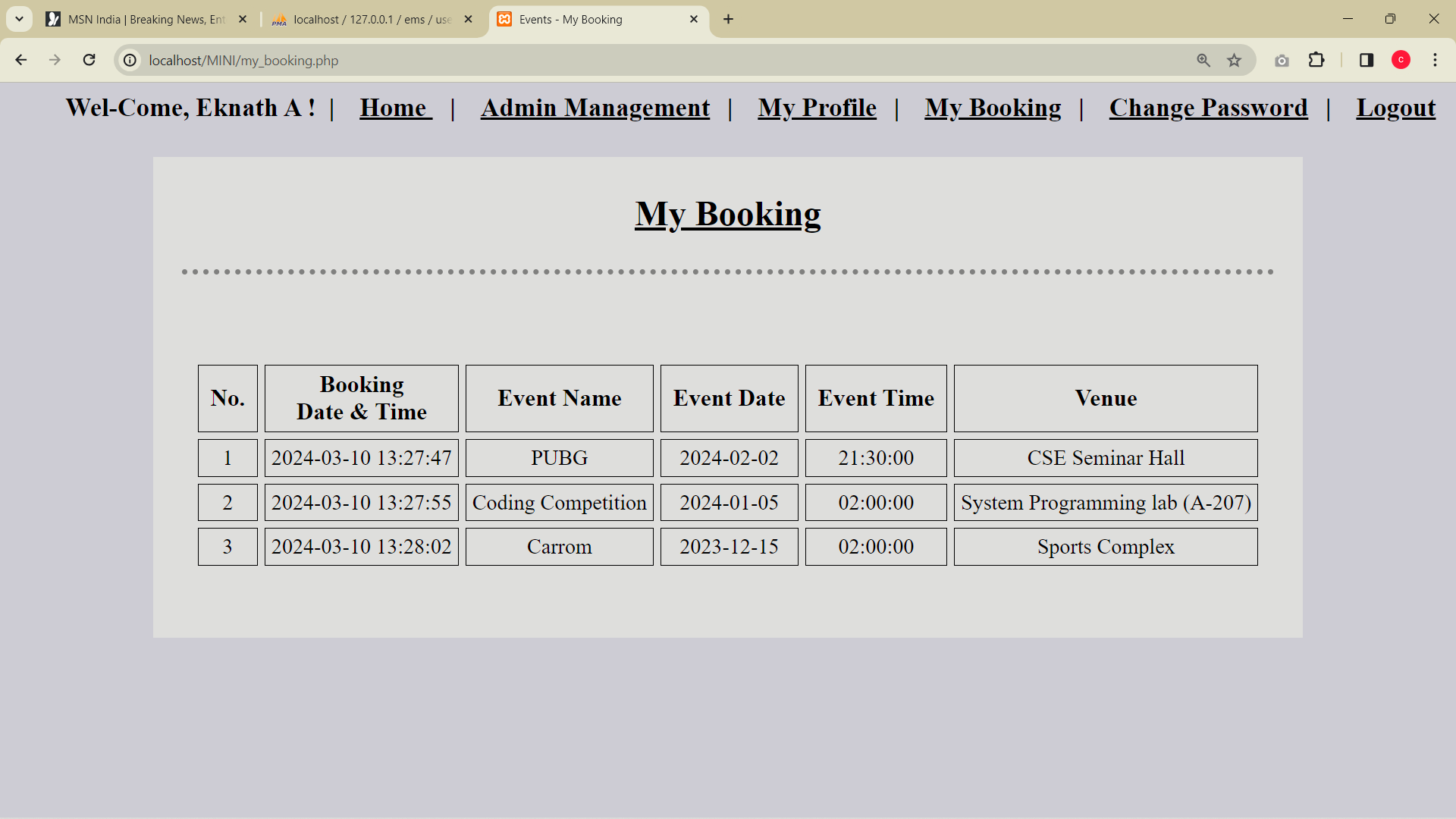


Fig 9: User Login Page and User View (My Booking details)

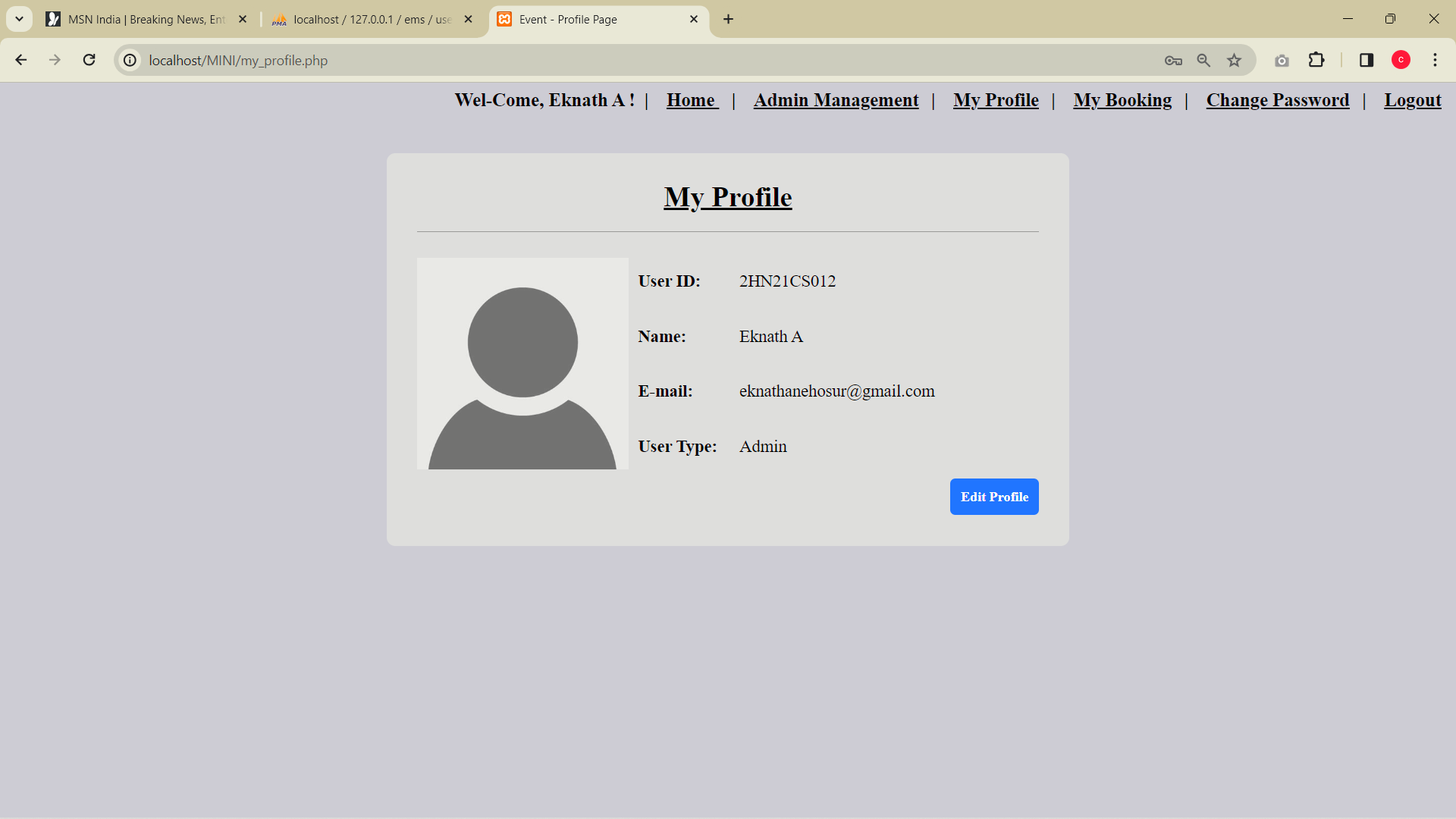
****

Fig 10: User profile and Profile Management

Caption: User can change his Credentials

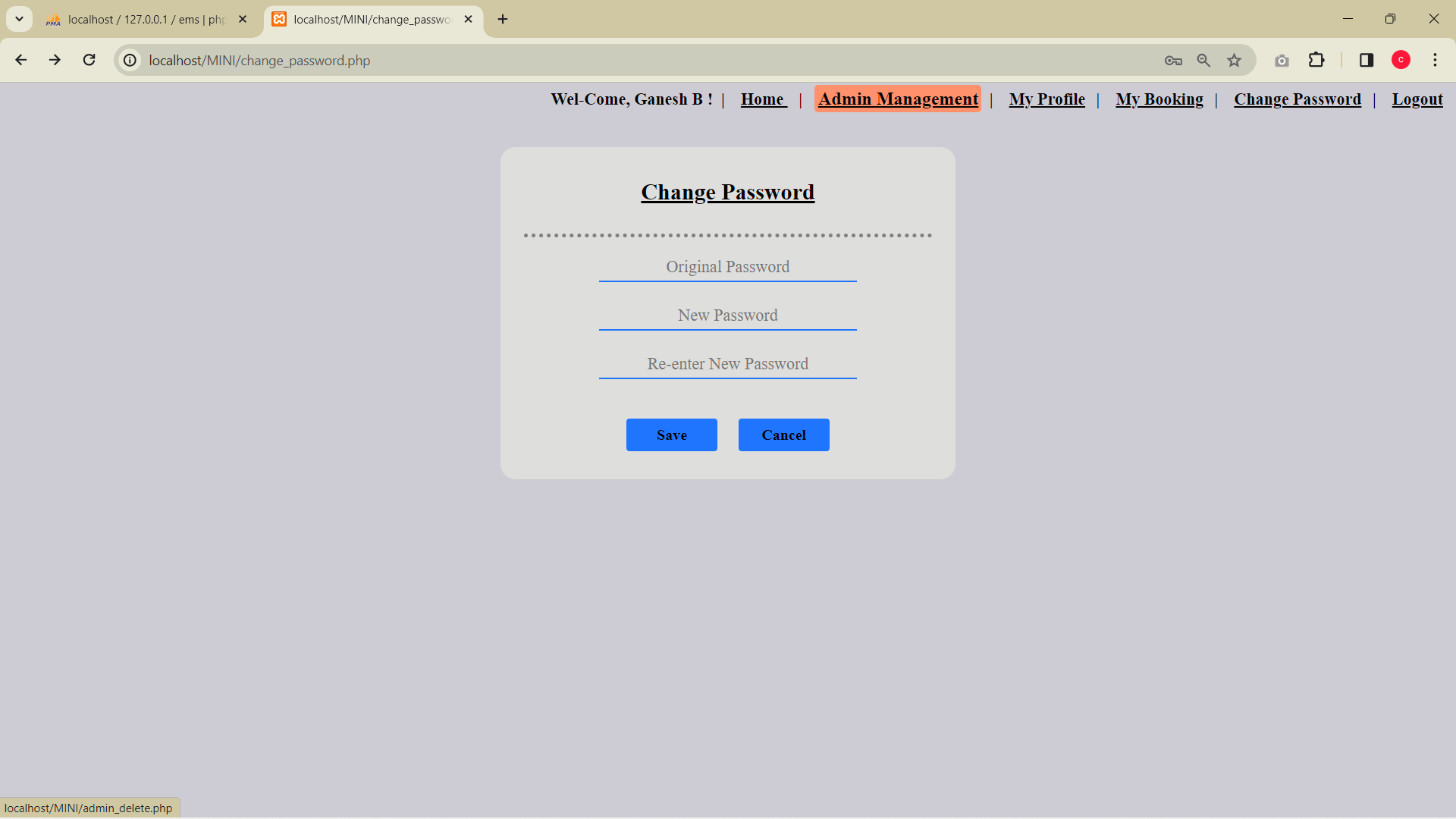
****

Fig 11: Admin Management

Caption: Admin can manage and monitor the Booking Details

**CONCLUSION AND FUTURE STEPS**

Our MySQL mini-project for college event management system offers a streamlined solution for organizing and overseeing campus activities. Through meticulous database design and implementation, we have developed a user-friendly platform that facilitates event creation, scheduling, participant management, and resource allocation. There is no such performance booster for the website it is simple web application to manage the events that are held in the college or in the universities. This mini-project is user friendly and easy to manage and understand.

In future some of the steps may be included in the College Event Management System that are Performance, data integrity ,Changes in the appearance, including more functionality for the admin to administrate and monitor the system. Can include frame works for high performance. Data Backup and recovery.

**REFERENCES**

* <https://developer.mozilla.org/en-US/docs/Learn/CSS>
* <https://www.mysql.com/products/workbench>
* <https://www.oracle.com/in/mysql/what-is-mysql>
* <https://www.investopedia.com/terms/h/html.asp>
* <https://www.w3schools.com/mysql>
* <https://www.youtube.com/watch?v=I2lB7fZE37g>