FSD Laboratory 04

Name: Shabbar Adamjee
Roll No.: PB57
PRN: 1032221508

Aim: Write server-side script in PHP to perform form validation and create database application using PHP and MySQL to perform insert, update, delete and search operations.

Objectives:

1. To understand Server-side Scripting.
2. To learn database connectivity using PHP-MySQL.
3. To perform insert, update, delete and search operations on database.

Theory:

1. PHP Architecture.

LAMP Stack Architecture

- **L**inux: The operating system.
- **A**pache: The web server.
- **M**ySQL: The database.
- **P**HP: The programming language.
  This is the traditional architecture for PHP-based web applications. PHP scripts are executed by the Apache web server, which communicates with the MySQL database.

MVC (Model-View-Controller) Pattern

- **Model**: Handles the data and business logic. Interacts with the database and processes user input.
- **View**: Represents the presentation layer, displaying the data to the user. It contains HTML, CSS, and PHP for output.
- **Controller**: Acts as a bridge between the Model and View, processing user requests and directing data from the Model to the View.
  This pattern separates concerns and is widely used in modern PHP frameworks like Laravel, Symfony, and CodeIgniter.

Layered Architecture

- **Presentation Layer**: Handles everything related to the user interface (UI) and user interaction.
- **Business Logic Layer**: Contains the core business rules and logic of the application.
- **Data Access Layer**: Responsible for accessing and managing data, typically interacting with databases, APIs, etc.
- **Database Layer**: Where the actual data resides, usually in a relational database like MySQL or PostgreSQL.

Service-Oriented Architecture (SOA)

- In more complex PHP applications, a service-oriented architecture is sometimes implemented. This involves creating independent services (often microservices) that perform specific functions and communicate with each other through APIs.
- SOAP and REST are common communication protocols in PHP applications.

Microservices Architecture

- Breaks down the application into small, independently deployable services. Each microservice handles a specific piece of functionality and communicates with other services using HTTP/REST or message queues like RabbitMQ.
- In PHP, you may use frameworks like Lumen (a lighter version of Laravel) to develop microservices.

Event-Driven Architecture

- Applications can be designed to respond to events (user actions, system events, etc.), decoupling event producers from event consumers.
- PHP supports event-driven architectures, particularly when integrated with tools like Swoole or ReactPHP to handle asynchronous tasks and real-time event streams.

2. Steps for Database connectivity in PHP.

   There are three extensions that PHP provides to connect with MySQL include mysqli, mysqlInd, and pdo_mysql.

   We are using mysqli in this assignment.

   PHP **mysqli_connect() function** is used to connect with MySQL database. It returns *resource* if connection is established or null.

   Code:

```
<?php
$servername = "localhost";     takes the server name.
$username = "username";       takes the username for that server.
$password = "password";       provide password for that username.

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

FAQ:

1. What are the advantages of Server-side Scripting?

Improved Security

- Since the code is executed on the server and not exposed to the client, sensitive information (like database queries, authentication mechanisms, and file paths) remains hidden. This protects the application from certain types of client-side attacks, such as tampering with scripts, accessing confidential data, or manipulating form inputs.
- Data validation, authentication, and authorization are handled securely on the server, reducing the risk of breaches.

Database Interaction

- Server-side scripting is essential for interacting with databases. It allows developers to query, insert, update, and delete data from databases dynamically. This is especially important for building content management systems (CMS), e-commerce platforms, and social media sites that require frequent updates to data.
- It also ensures that user inputs are sanitized and securely stored in the database, preventing SQL injection attacks.

Reduces Client Load

- By executing scripts on the server, the client's device (e.g., a browser on a mobile phone) does not have to perform heavy processing. This is particularly beneficial for devices with limited computing power or slower internet connections.
- Server-side scripts handle all the computation-intensive tasks, making websites faster and more responsive on the client side.

Better Control Over File and System Operations

- Server-side scripts can interact with the server's file system, allowing operations like reading, writing, and deleting files. This is important for tasks such as file uploads, logging, and server-side caching.
- Server-side scripts can also run shell commands, perform system diagnostics, or manage sessions and cookies more securely.

Data Integrity and Processing

- Server-side scripting allows for central processing of data, ensuring data integrity and consistency across different sessions and users. Since all data processing is done on the server, there's less chance for discrepancies or manipulation by users.
- Complex calculations, business logic, and processing tasks can be handled server-side, reducing errors and maintaining control over data flow.

2. What is XAMPP and phpMyAdmin?

XAMPP

XAMPP is an open-source, cross-platform web server solution stack package that provides a complete environment to develop, test, and deploy web applications. It simplifies the installation and configuration of various components required for running dynamic web applications. The name XAMPP stands for:

- **X**: Cross-platform (Windows, macOS, Linux)
- **A**: Apache (the web server)
- **M**: MySQL or MariaDB (the database management system)
- **P**: PHP (server-side scripting language)
- **P**: Perl (optional scripting language)

Key Components of XAMPP:

Apache: The most widely used web server, it handles HTTP requests and serves web pages to clients.

MySQL / MariaDB: A relational database management system (RDBMS) that stores the application's data. MySQL was traditionally used, but many XAMPP distributions now come with MariaDB, a fork of MySQL.

PHP: A server-side scripting language used for dynamic content generation. It works with Apache to process scripts and generate HTML content.

Perl: A high-level programming language (optional but included for legacy compatibility).

phpMyAdmin: A web-based tool used for managing MySQL or MariaDB databases, explained further below.

phpMyAdmin
phpMyAdmin is an open-source, web-based database management tool that provides an easy-to-use graphical interface for managing MySQL or MariaDB databases. It’s included in XAMPP and similar server stacks like WAMP, LAMP, and MAMP.

Key Features of phpMyAdmin:

- **Database Management**: Create, modify, and delete databases easily via a graphical user interface (GUI).
- **Table Management**: Manage tables within a database, including creating, deleting, modifying fields, and indexing.
- **Query Execution**: Execute SQL queries directly from the interface for more complex database operations.
- **Data Import/Export**: Easily import or export data in various formats like SQL, CSV, and XML.
- **User Management**: Add, edit, or remove users and manage their privileges for secure database access.
- **Backup and Restore**: Perform backups of your database and restore them as needed.

3. What are the two ways to connect to a database in PHP?

In PHP, there are two primary ways to connect to a database:

MySQLi (MySQL Improved)
MySQLi is an improved version of the original MySQL extension, providing both procedural and object-oriented approaches to interact with MySQL databases.

Key Features:

- Supports procedural and object-oriented styles.
- Offers prepared statements to prevent SQL injection.
- Provides enhanced performance and error handling over the original MySQL extension.

PDO (PHP Data Objects)
PDO is a more flexible and powerful database access layer in PHP. It supports multiple databases such as MySQL, PostgreSQL, SQLite, etc., and provides a consistent API for interacting with them.

Key Features:

- Supports multiple databases (MySQL, PostgreSQL, SQLite, etc.), making it more versatile than MySQLi.
- Uses prepared statements to prevent SQL injection.
- Provides error handling with exceptions, making debugging easier.

MySQLi is ideal when you're only working with MySQL and want a simpler, faster option.
PDO is a better choice if you need to work with multiple types of databases and require more advanced features like prepared statements and exception handling.

Output: Screenshots of the output to be attached.

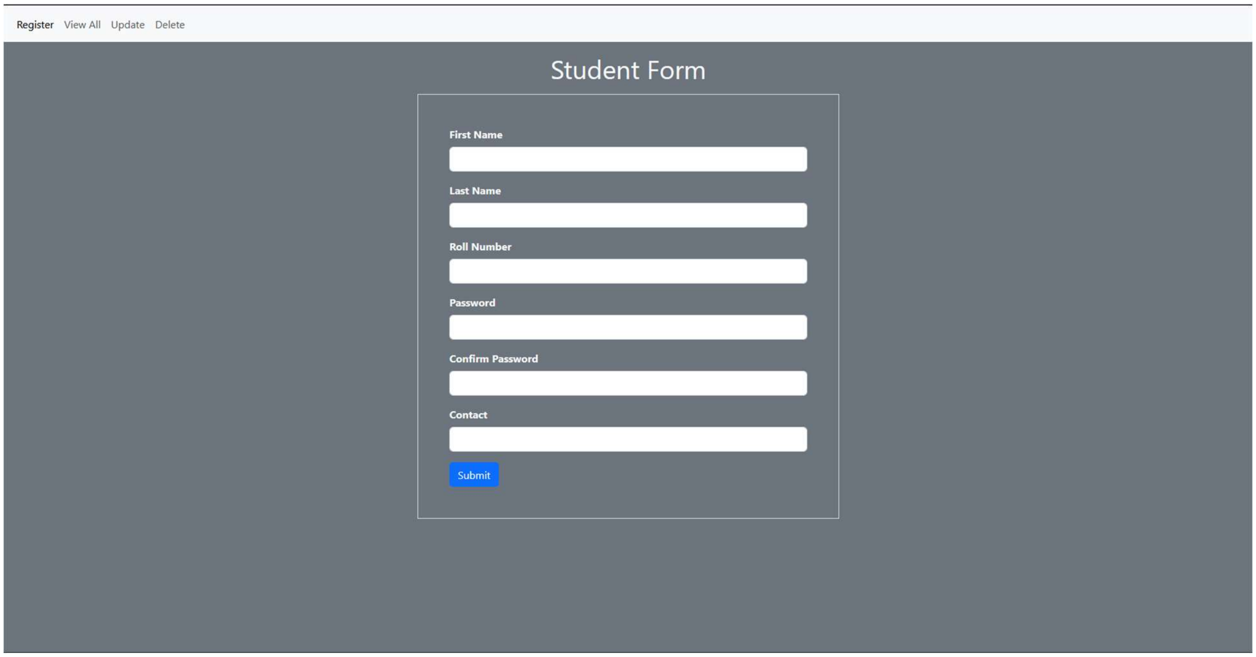
Sample Problem Statements:

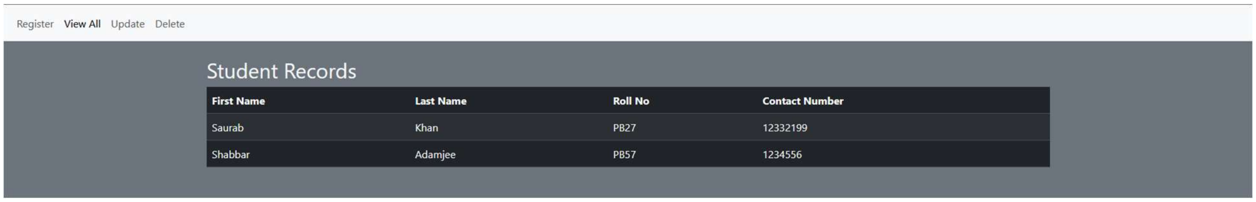
PHP CRUD Operations

1.Student can create a PHP form or use existing/ implemented HTML form for Student's Registration System with the fields mentioned: First name,Last name, Roll No/ID, Password, Confirm Password,Contact number and perform following operations

1.Insert student details -First name,Last name, Roll No/ID, Password, Confirm Password,Contact number
2.Delete the Student records based on Roll no/ID
3.Update the Student details based on Roll no/ID- Example students can update their contact details based on searching the record with Roll no.
4.Display the Updated student details or View the Students record in tabular format.

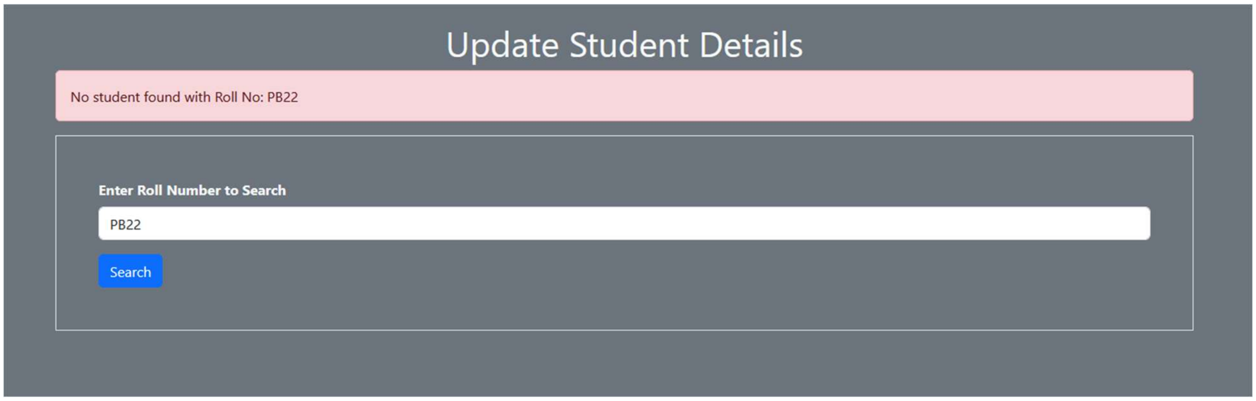
Apply Form Validation on the necessary fields using PHP/Javascript





Student Records

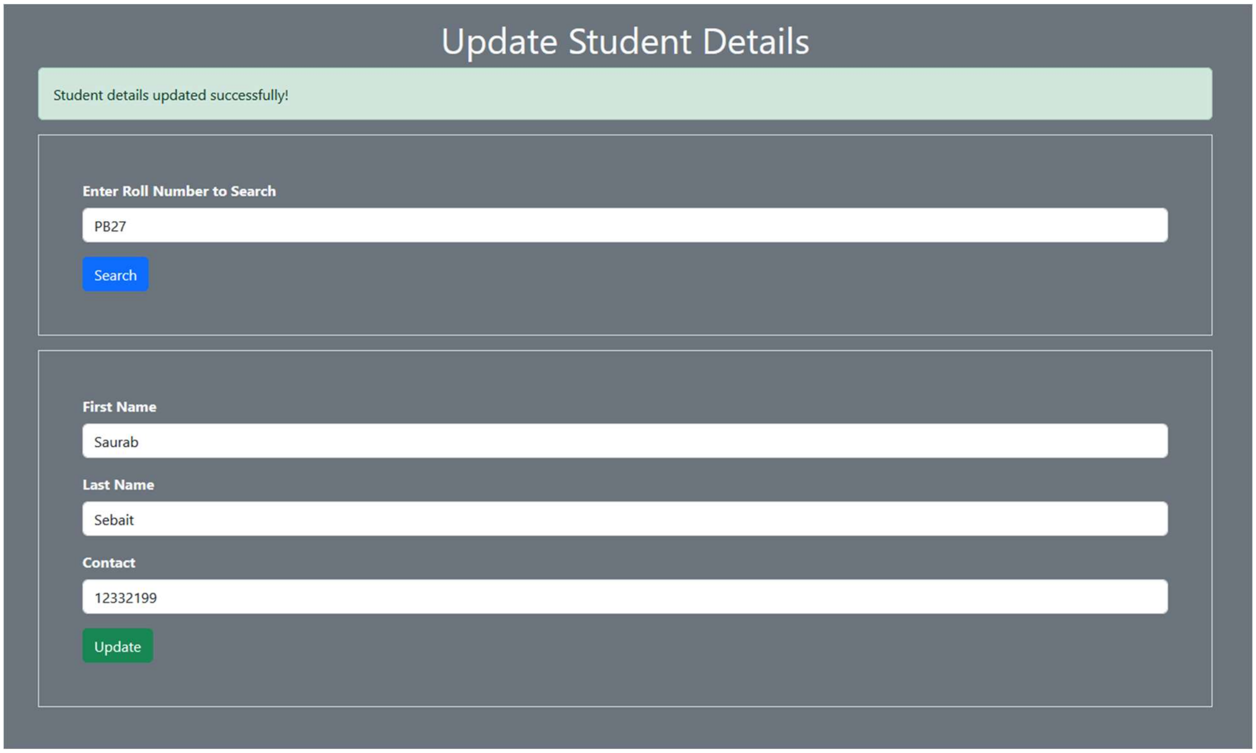
First Name	Last Name	Roll No	Contact Number
Saurab	Khan	PB27	12332199
Shabbar	Adamjee	PB57	1234556



Update Student Details

No student found with Roll No: PB22

Enter Roll Number to Search: PB22



Update Student Details

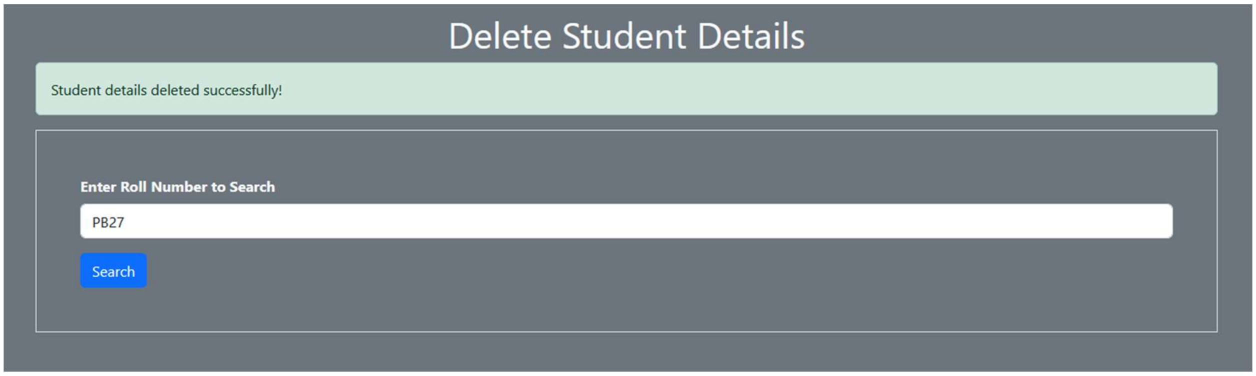
Student details updated successfully!

Enter Roll Number to Search: PB27

First Name: Saurab

Last Name: Sebait

Contact: 12332199



Delete Student Details

Student details deleted successfully!

Enter Roll Number to Search

PB27

2. Student can create a PHP form or use existing/ implemented HTML form for Library Management System with the fields mentioned: Book name, ISBN No, Book title, Author name , Publisher name and perform following operations

1.Insert Book details -Book name, ISBN No, Book title, Author name, Publisher name
2.Delete the Book records based on ISBN No
3.Update the Book details based on ISBN No- Example students can update wrong entered book details based on searching the record with ISBN No.
4.Display the Updated Book details or View the Book Details records in tabular format.

Apply Form Validation on the necessary fields using PHP/Javascript

3. Student can create a PHP form or use existing/ implemented HTML form for Employee Management System with the fields mentioned: Employee name, Employee ID, Department_name, Phone number , Joining Date and perform following operations

1.Insert Employee details -Employee name, Employee ID, Department_name, Phone number , Joining Date
2.Delete the Employee records based on Employee ID
3.Update the Employee details based on Employee ID- Example students can update Employee details based on searching the record with Employee ID.
4.Display the Updated Employee details or View the Employee Details records in tabular format.

Apply Form Validation on the necessary fields using PHP/Javascript

4. Student can create a PHP form or use existing/ implemented HTML form for Flight Booking Management System with the fields mentioned: Passenger name, From, to, date,Departure date,Arrival date, Phone number , Email ID and perform following operations

1.Insert Passenger details -Passenger name, From, to, date,Departure date,Arrival date, Phone number , Email ID
2.Delete the Passenger records based on Phone Number
3.Update the Passenger details based on Phone Number - Example students can update Flight Booking details based on searching the record with Phone Number.
4.Display the Updated Flight Booking details or View the Flight Booking Details records in tabular format.

Apply Form Validation on the necessary fields using PHP/Javascript.

Technologies Student Should Use:
XAMPP
PHP for Server-side Scripting
MySQL as a backend Database