Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering & Technology
Class: Third Year B.Tech CSE (Semester V)
Course: Full Stack Development

GIT Repo URL: https://github.com/prateekp1304/FSD

Roll No: 08
Name: Prateek Patel
PRN: 1032210532

FSD Laboratory 04

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.

PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development. Its architecture encompasses several components and follows a request-response model. Here's an overview of the PHP architecture:

Client-Server Model: PHP operates in a client-server model. A client (typically a web browser) sends a request to a server, and PHP processes that request on the server side before sending a response back to the client.

Web Server: PHP scripts are executed on the server side by a web server. Common web servers that support PHP include Apache, Nginx, and Microsoft IIS.
PHP Interpreter: The core of PHP's architecture is the PHP interpreter, which reads and executes PHP scripts. It interprets the PHP code and produces output that is sent to the web server.

PHP Parser: The PHP parser reads the PHP script and translates it into an intermediate form, often known as the "opcode." This intermediate code is then executed by the PHP runtime.

Zend Engine: The Zend Engine is the open-source scripting engine that executes PHP code. It handles the compilation of PHP scripts into intermediate code and their execution.
Extensions: PHP supports extensions, which are modules that provide additional functionality. Extensions can be either bundled with PHP or added separately. Examples include database extensions (MySQL, PostgreSQL), XML processing, and more.

Database Connectivity: PHP can connect to various databases, allowing developers to interact with database systems like MySQL, PostgreSQL, SQLite, and others. This enables the creation of dynamic, data-driven web applications.

PHP Libraries: PHP comes with a set of standard libraries that provide additional functionality. These libraries cover areas such as file handling, regular expressions, networking, and more.

Session Management: PHP includes built-in session management capabilities. Sessions allow developers to store and retrieve data on a per-user basis, enabling the creation of stateful web applications.

Web Server Module: In the case of web servers like Apache, PHP can be integrated as a module. This allows for better performance and more direct interaction with the web server.

Output Buffering: PHP supports output buffering, which allows developers to capture the output of PHP scripts before it is sent to the client. This can be useful for modifying or manipulating the output before it reaches the browser.

2. Steps for Database connectivity in PHP.

Connecting PHP to a database involves several steps. Here's a general guide on how to establish database connectivity in PHP:

- Choose a Database: Decide on the database system you want to use (e.g., MySQL, PostgreSQL, SQLite).

- Install Database Software: Install and configure the chosen database software on your server or use a hosted database service.

- PHP Database Extension: Ensure that the appropriate PHP extension for your chosen database is installed and enabled. Common extensions include mysqli for MySQL and pgsql for PostgreSQL.

- Database Connection Parameters: Gather the necessary information for connecting to the database:
  Hostname or IP address of the database server
  Database username
  Database password
  Database name
  Open a Connection:

- Use PHP functions to open a connection to the database.

- Execute SQL Queries: Once the connection is established, you can execute SQL queries using PHP to interact with the database.

- Handle Errors: Implement error handling to manage potential issues with the database connection or queries.

- Close the Connection: Always close the database connection when you're done using it to free up resources.

- Security Considerations: Use prepared statements or parameterized queries to prevent SQL injection attacks. Avoid storing sensitive information, such as database credentials, directly in your PHP scripts.

FAQ:

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

Ans:

- Dynamic Content Generation: Server-side scripting allows the generation of dynamic content based on user input, database queries, and other server-side operations. This dynamic content can be customized for each user or session.

- Database Interaction: Server-side scripting enables interaction with databases, allowing the storage and retrieval of data. This is essential for building data-driven web applications.

- Enhanced Security: Server-side scripts are executed on the server, and only the results are sent to the client. This helps in securing sensitive logic and data, reducing the risk of exposing critical information to users.

- Code Protection: Server-side code is not visible to the end-users, providing a layer of security and intellectual property protection for the underlying application logic.

- Platform Independence: Server-side scripting languages, such as PHP, are often platform-independent. This means that the same code can run on different types of servers, offering flexibility in deployment.

- Easy Maintenance: Centralized server-side code is easier to maintain compared to client-side code. Updates and changes can be applied on the server, affecting all clients without requiring individual updates on each client machine.

2. What is XAMPP and phpMyAdmin?

Ans:

XAMPP (Cross-platform, Apache, MySQL, PHP, and Perl):

XAMPP is a free, open-source cross-platform web server solution stack package developed by Apache Friends. It includes Apache HTTP Server, MySQL database, PHP, and Perl. XAMPP provides a convenient way to set up a local web development environment on your computer, allowing you to develop and test web applications before deploying them to a live server.

phpMyAdmin:

phpMyAdmin is a web-based tool written in PHP used for managing MySQL databases. It provides a graphical user interface to perform various database management tasks such as creating, modifying, and deleting databases, tables, and fields. phpMyAdmin simplifies the administration of MySQL databases, making it more accessible to users who may not be comfortable with command-line interfaces.

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

Ans:

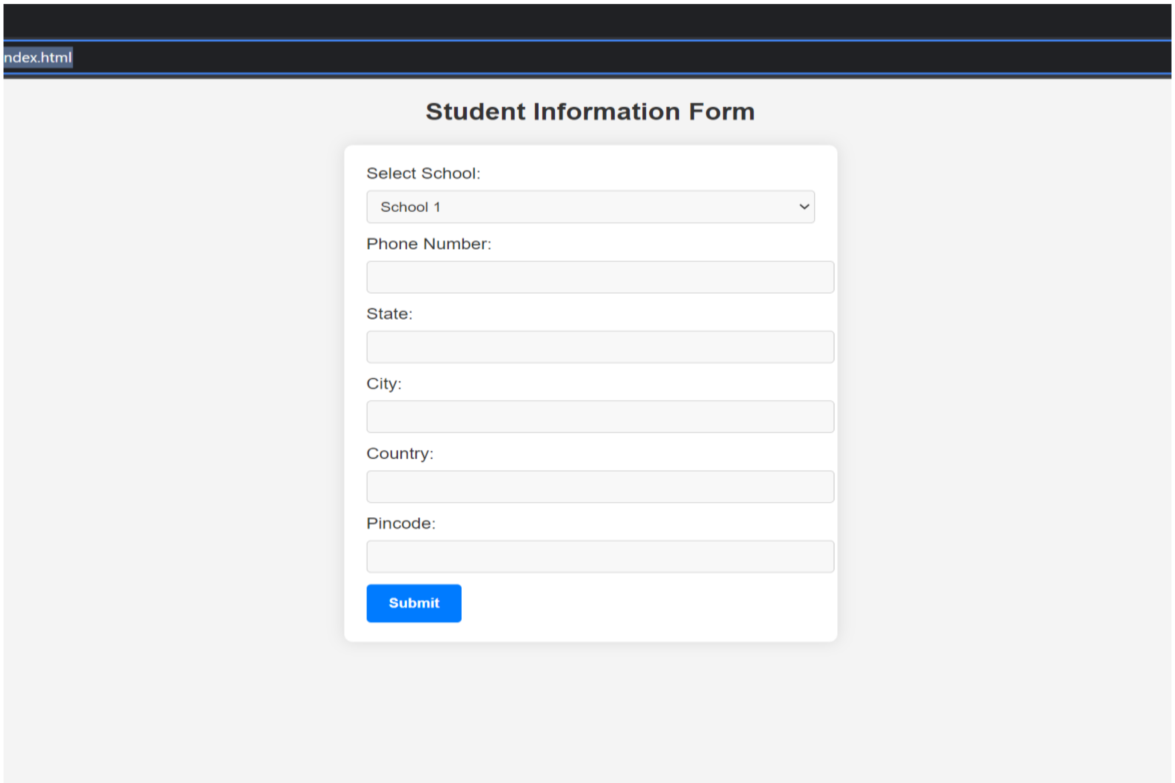
MySQLi (MySQL Improved): MySQLi is an extension in PHP that provides an interface to communicate with MySQL databases. It offers both procedural and object-oriented approaches to interact with databases. MySQLi supports features like prepared statements, transactions, and enhanced security.

PDO (PHP Data Objects):

PDO is a database access layer providing a uniform method of access to multiple databases. It supports various database management systems, making it more flexible. PDO also provides features like prepared statements and error handling.

Output: Screenshots of the output to be attached.

Output Screenshots:



Student Information Form

Select School: School 1

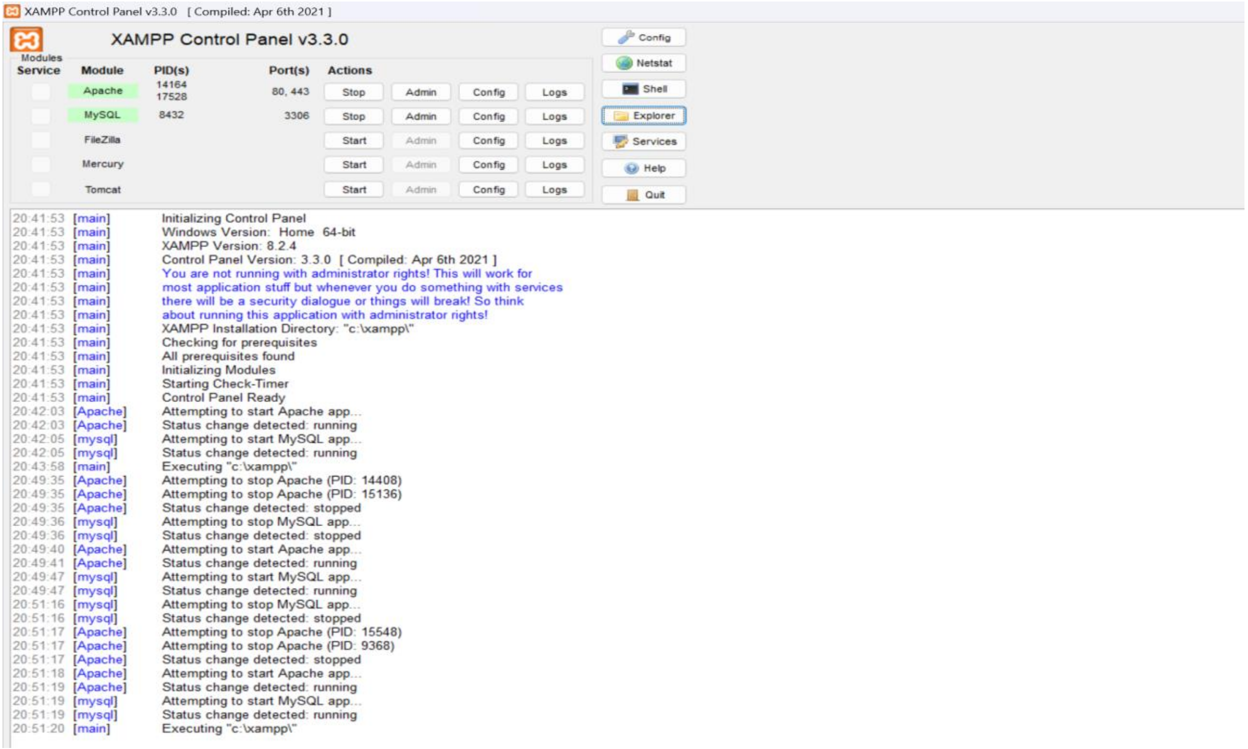
Phone Number:

State:

City:

Country:

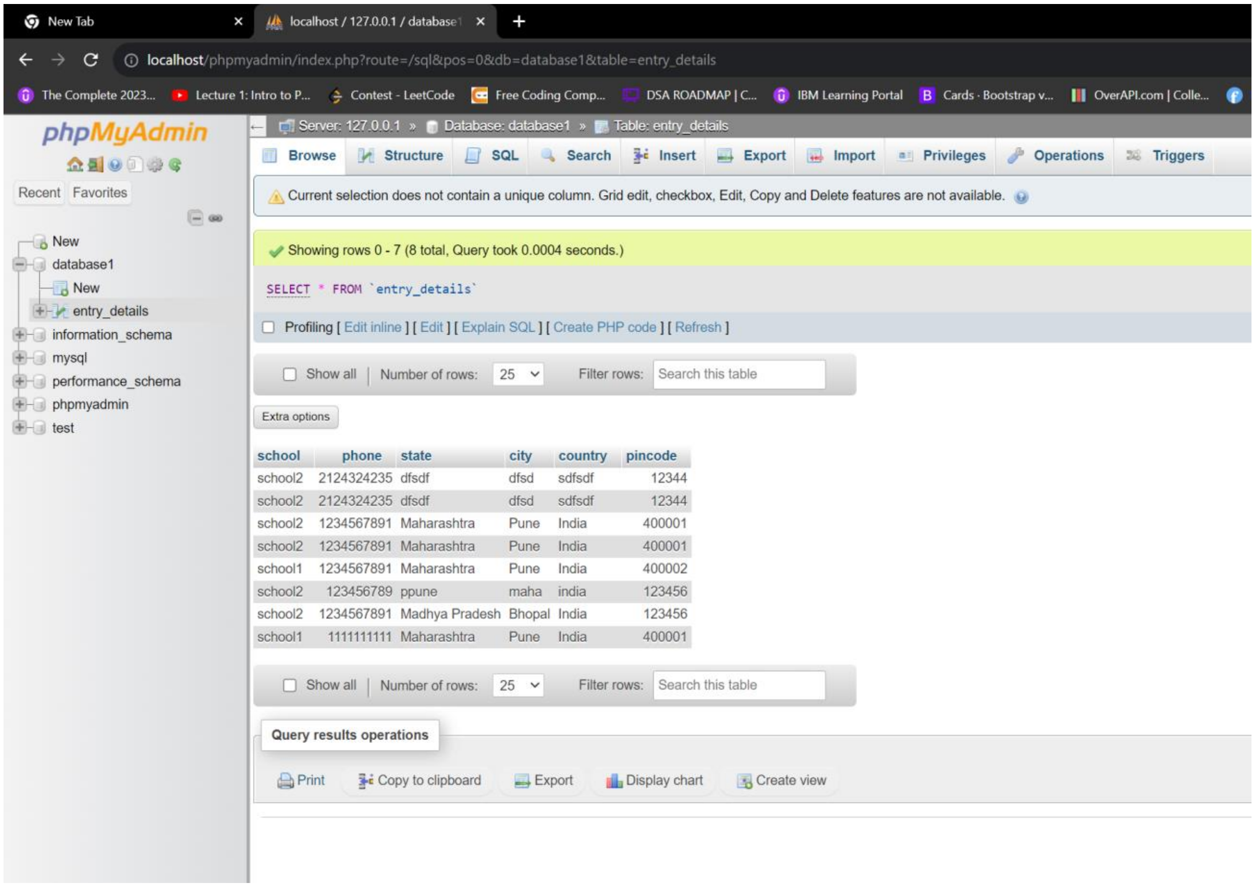
Pincode:


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	14164 17528	80, 443	Stop	Admin	Config	Logs
	MySQL	8432	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
20:41:53  [main]        Initializing Control Panel
20:41:53  [main]        Windows Version:  Home  64-bit
20:41:53  [main]        XAMPP Version: 8.2.4
20:41:53  [main]        Control Panel Version: 3.3.0  [ Compiled: Apr 6th 2021 ]
20:41:53  [main]        You are not running with administrator rights! This will work for
20:41:53  [main]        most application stuff but whenever you do something with services
20:41:53  [main]        there will be a security dialogue or things will break! So think
20:41:53  [main]        about running this application with administrator rights!
20:41:53  [main]        XAMPP Installation Directory: "c:\xampp\"
20:41:53  [main]        Checking for prerequisites
20:41:53  [main]        All prerequisites found
20:41:53  [main]        Initializing Modules
20:41:53  [main]        Starting Check-Timer
20:41:53  [main]        Control Panel Ready
20:42:03  [Apache]      Attempting to start Apache app...
20:42:03  [Apache]      Status change detected: running
20:42:05  [mysql]       Attempting to start MySQL app...
20:42:05  [mysql]       Status change detected: running
20:43:58  [main]        Executing "c:\xampp\"
20:49:35  [Apache]      Attempting to stop Apache (PID: 14408)
20:49:35  [Apache]      Attempting to stop Apache (PID: 15136)
20:49:35  [Apache]      Status change detected: stopped
20:49:36  [mysql]       Attempting to stop MySQL app...
20:49:36  [mysql]       Status change detected: stopped
20:49:40  [Apache]      Attempting to start Apache app...
20:49:41  [Apache]      Status change detected: running
20:49:47  [mysql]       Attempting to start MySQL app...
20:49:47  [mysql]       Status change detected: running
20:51:16  [mysql]       Attempting to stop MySQL app...
20:51:16  [mysql]       Status change detected: stopped
20:51:17  [Apache]      Attempting to stop Apache (PID: 15548)
20:51:17  [Apache]      Attempting to stop Apache (PID: 9368)
20:51:17  [Apache]      Status change detected: stopped
20:51:18  [Apache]      Attempting to start Apache app...
20:51:19  [Apache]      Status change detected: running
20:51:19  [mysql]       Attempting to start MySQL app...
20:51:19  [mysql]       Status change detected: running
20:51:20  [main]        Executing "c:\xampp\"
```

•



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=database1&table=entry_details

Server: 127.0.0.1 » Database: database1 » Table: entry_details

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 7 (8 total, Query took 0.0004 seconds.)

```
SELECT * FROM `entry_details`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 | Filter rows: Search this table

school	phone	state	city	country	pincode
school2	2124324235	dfsdf	dfsd	sdfsdf	12344
school2	2124324235	dfsdf	dfsd	sdfsdf	12344
school2	1234567891	Maharashtra	Pune	India	400001
school2	1234567891	Maharashtra	Pune	India	400001
school1	1234567891	Maharashtra	Pune	India	400002
school2	123456789	ppune	maha	india	123456
school2	1234567891	Madhya Pradesh	Bhopal	India	123456
school1	1111111111	Maharashtra	Pune	India	400001

□ Show all | Number of rows: 25 | Filter rows: Search this table

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

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