12.1)Write a PHP Script to interact with MySQL and perform the following i. insert data, ii. retrieve data, iii. update data, iv. authenticate user with form data.

**i. insert data:**

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "database\_name";

// Create connection

$conn = new mysqli($servername,

$username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: "

. $conn->connect\_error);

}

$sqlquery = "INSERT INTO table VALUES

('John', 'Doe', 'john@example.com')"

if ($conn->query($sql) === TRUE) {

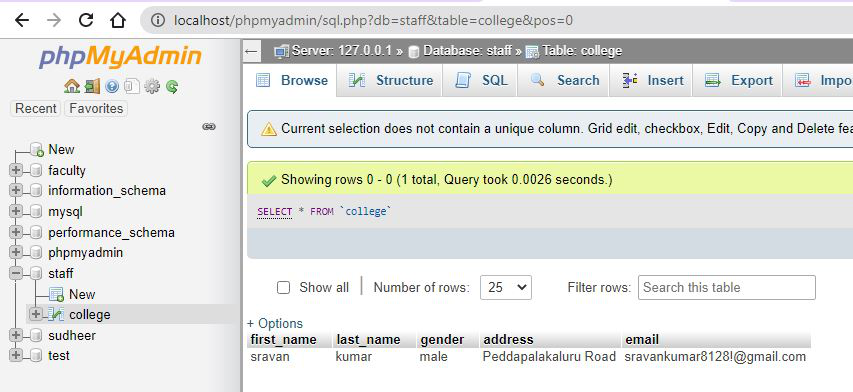
echo "record inserted successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

**Output:**



**ii. retrieve data**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$databasename = "geeksforgeeks";

// CREATE CONNECTION

$conn = new mysqli($servername,

$username, $password, $databasename);

// GET CONNECTION ERRORS

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// SQL QUERY

$query = "SELECT \* FROM `Student Details`;";

// FETCHING DATA FROM DATABASE

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

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

{

// OUTPUT DATA OF EACH ROW

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

{

echo "Roll No: " .

$row["Roll\_No"]. " - Name: " .

$row["Name"]. " | City: " .

$row["City"]. " | Age: " .

$row["Age"]. "<br>";

}

}

else {

echo "0 results";

}

$conn->close();

?>

**Output:**

Roll No: 1 - Name: Ram | City: Delhi | Age: 18

Roll No: 2 - Name: Shyam | City: Mumbai | Age: 19

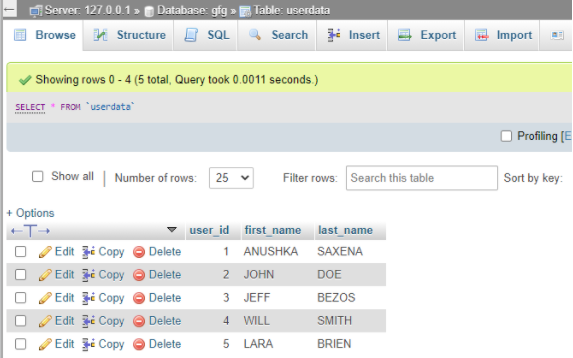
Roll No: 3 - Name: Rohit | City: Chennai | Age: 18

Roll No: 4 - Name: Suresh | City: Kolkata | Age: 20

**iii. update data**

<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
$dbname = "myDB";  
  
// Create connection  
$conn = mysqli\_connect($servername, $username, $password, $dbname);  
// Check connection  
if (!$conn) {  
  die("Connection failed: " . mysqli\_connect\_error());  
}  
  
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";  
  
if (mysqli\_query($conn, $sql)) {  
  echo "Record updated successfully";  
} else {  
  echo "Error updating record: " . mysqli\_error($conn);  
}  
  
mysqli\_close($conn);  
?>

**Output:**



**iv. authenticate user with form data.**

**<?php**

    include('connection.php');

    $username = $\_POST['user'];

    $password = $\_POST['pass'];

       //to prevent from mysqli injection

        $username = stripcslashes($username);

        $password = stripcslashes($password);

        $username = mysqli\_real\_escape\_string($con, $username);

        $password = mysqli\_real\_escape\_string($con, $password);

        $sql = "select \*from login where username = '$username' and password = '$password'";

        $result = mysqli\_query($con, $sql);

        $row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);

       $count = mysqli\_num\_rows($result);

        if($count == 1){

            echo "**<h1><center>** Login successful **</center></h1>**";

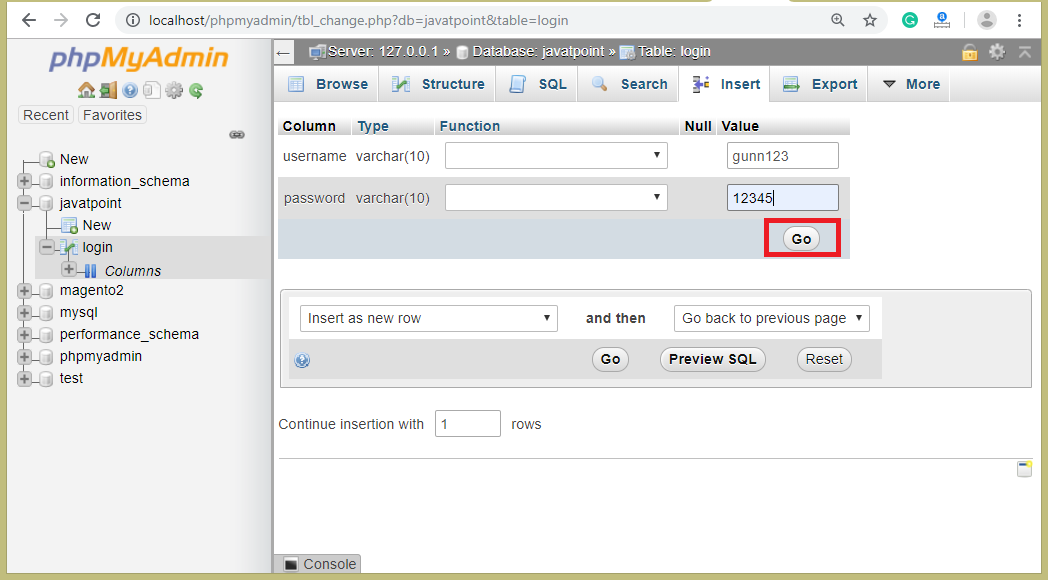
        }

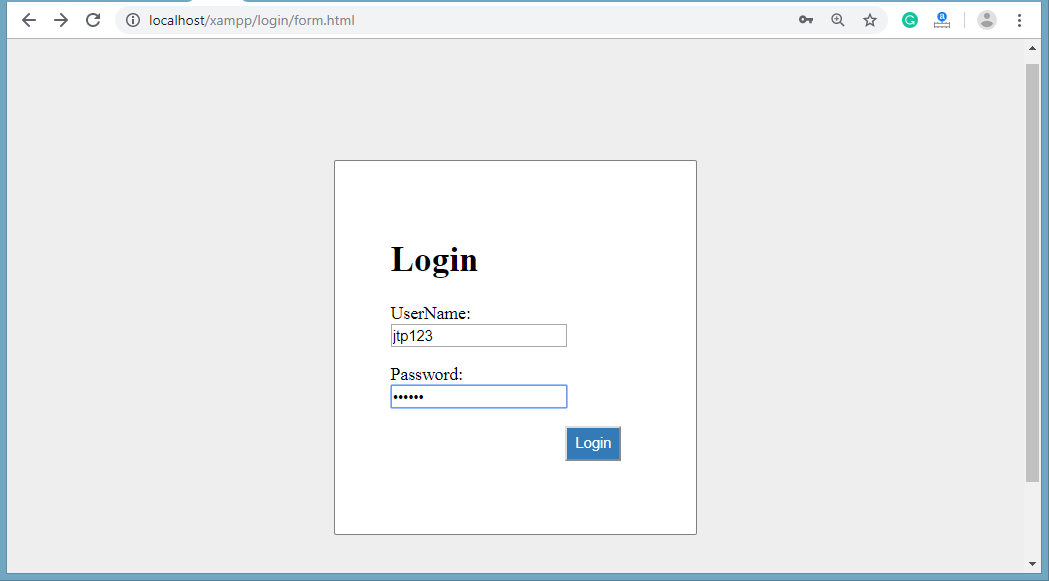
       else{

            echo "**<h1>** Login failed. Invalid username or password.**</h1>**";

        }

**?>**

**Output:** 



12.2) Write a program to interact with MongoDB using i. PHP, ii. Angular JS, iii. React JS, iv. authenticate user with form data.

**i. PHP**

<?php

**require** 'vendor/autoload.php';

// Creating Connection

$con = **new** MongoDB\Client("mongodb://localhost:27017");

// Creating Database

$db = $con->javatpoint;

// Creating Document

$collection = $db->employee;

// Insering Record

$collection->insertOne( [ 'name' =>'Peter', 'email' =>'peter@abc.com' ] );

// Fetching Record

$record = $collection->find( [ 'name' =>'Peter'] );

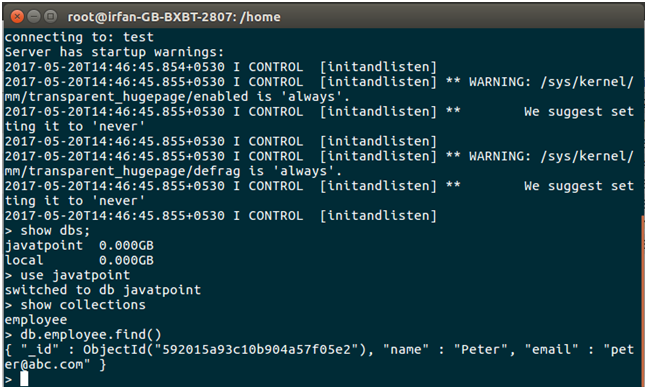
**foreach** ($record **as** $employe) {

echo $employe['name'], ': ', $employe['email']."<br>";

}

?>

**Output:**



**ii. Angular JS**

<script>

angular.module('formExample', [])

.controller('ExampleController', ['$scope', '$http', function($scope, $http) {

$scope.master = {};

$scope.update = function(user) {

if ($scope.formx.$valid) {

$scope.master = angular.copy(user);

$http.post('http://localhost:3300/post', $scope.master

).success(function(data, status, headers, config) {

alert("Success!")

}).error(function(data, status, headers, config) {

alert("Error");

});

}

};

$scope.reset = function() {

$scope.user = angular.copy($scope.master);

};

$scope.reset();

}]);

</script>

**Output:**



**iii) react js**

**Step 1:** Create a react application using the following command

**npx create-react-app foldername**

**Step 2:** Once it is done change your directory to the newly created application using the following command

**cd foldername**

**Step to run the application:** Enter the following command to run the application.

**npm start**

**Backend Setup With NodeJS:**Setup NodeJs for Backend to integrate with frontend.

**Step1:** Make a folder in the root directory using the following command

**mkdir backend**

**Step 2:**Once it is done change your directory to the newly created folder called backend using the following command

**cd backend**

**Step 3:**Run the following command to create configure file

**npm init -y**

**Step 4:**Now Install the mongoose MongoDB using the following command.

**npm i express mongoose mongodb cors**

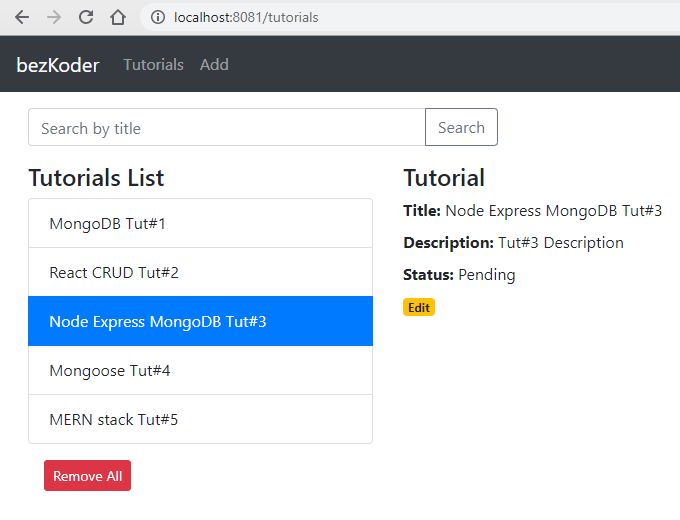
**Step 5:**Create a file that is index.js

**touch index.js**

**Index.js :**

|  |
| --- |
| import React, { Component } from 'react' |
|  | import { Link } from 'react-router-dom' |
|  | import styled from 'styled-components' |
|  |  |
|  | const Collapse = styled.div.attrs({ |
|  | className: 'collpase navbar-collapse', |
|  | })`` |
|  |  |
|  | const List = styled.div.attrs({ |
|  | className: 'navbar-nav mr-auto', |
|  | })`` |
|  |  |
|  | const Item = styled.div.attrs({ |
|  | className: 'collpase navbar-collapse', |
|  | })`` |
|  |  |
|  | class Links extends Component { |
|  | render() { |
|  | return ( |
|  | <React.Fragment> |
|  | <Link to="/" className="navbar-brand"> |
|  | My first MERN Application |
|  | </Link> |
|  | <Collapse> |
|  | <List> |
|  | <Item> |
|  | <Link to="/movies/list" className="nav-link"> |
|  | List Movies |
|  | </Link> |
|  | </Item> |
|  | <Item> |
|  | <Link to="/movies/create" className="nav-link"> |
|  | Create Movie |
|  | </Link> |
|  | </Item> |
|  | </List> |
|  | </Collapse> |
|  | </React.Fragment> |
|  | ) |
|  | } |
|  | } |
|  |  |
|  | export default Links |

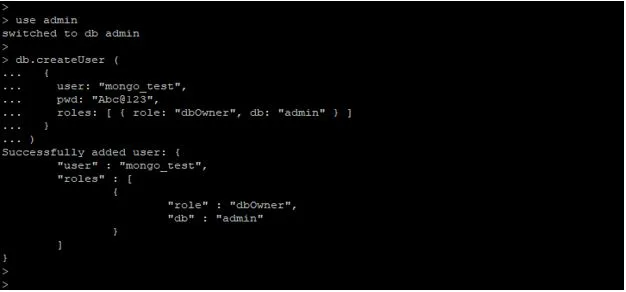
**Output:**



**iv. authenticate user with form data**.

use admin;  
db.createUser(  
{  
user: "mongo\_test",  
pwd: "Abc@123",  
roles: [ { role: "dbOwner", db: "admin" } ]    ## Define role as dbOwner on admin DB.  
}  
)

**output:**

****

vi /etc/mongod.conf

**output:**

****

/etc/init.d/mongod stop  
/etc/init.d/mongod start

**output:**

