|  |  |  |
| --- | --- | --- |
| 1\_Create own API with FREE hosting and domain  <https://www.youtube.com/watch?v=vxNQ2YGUt5M> | | |
|  |  |  |
|  |  | table with the columns id, name, email, |
|  |  | MySQL table with the columns id, name, email, and profile\_pic using the following SQL statement:  CREATE TABLE employee(  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(255) NOT NULL,  email VARCHAR(255) NOT NULL UNIQUE,  profile\_pic VARCHAR(255)  ); |
|  | Origin | C:\xampp\htdocs\1\_api\php\_api\employee.php |
|  |  |  |
|  |  | <?php  echo json\_encode("Hello Gurudev"); |
|  | Call the page  10.32 | <http://3.86.144.98/api/pr1/employee.php>  <http://localhost/1_api/php_api/employee.php> |
|  |  |  |
|  |  | Coonection .php file create  <?php  class dbObj  {      var $servername = "localhost";      var $username = "root";      var $password = '';      var $dbname = "employee";      var $conn;  //   var $servername = "localhost";  //   var $username = "mohali728\_codex";  //   var $password = "<YOUR\_PASSWORD>";  //   var $dbname = "mohali728\_codex";  //   var $conn;    function getConnstring()    {      $con = mysqli\_connect($this->servername, $this->username, $this->password, $this->dbname) or die("Connection failed: " . mysqli\_connect\_error());      /\* check connection \*/      if (mysqli\_connect\_errno()) {        printf("Connect failed: %s\n", mysqli\_connect\_error());        exit();      } else {        $this->conn = $con;      }      return $this->conn;    }  }  ?> |
|  |  | Test connection: |
|  |  | // Test connection  $db = new dbobj();  $conn = $db->getConnstring();  ?>  <div class="heading\_container heading\_center">  <h2>    Connection  <span>Status</span>  </h2>  <p style="background-color: green; color: white; padding: 10px;">  <?php echo "Congratulatim You are Connected successful in your database!";?>  </p>  </div> |
|  |  | Call the page : |
|  |  | http://localhost/1\_api/php\_api/connection.php |
|  |  |  |
|  |  | Add code in employee.php  C:\xampp\htdocs\1\_api\php\_api\employee.php |
|  |  | <?php  // echo json\_encode("Hello Gurudev");  include("connection.php");  // Include the connection file  // Create a new instance of the dbobj class and get the connection  $db = new dbobj();  $conn = $db->getConnstring();  $sql = "SELECT \* FROM employee";  $result = $conn->query($sql);  $response = array();  if ($result->num\_rows > 0) {      while ($row = $result->fetch\_assoc()) {          array\_push($response, $row);      }  }  // Output JSON encoded response  header('Content-Type:application/json');  echo json\_encode($response); |
|  |  | Response check both url and POSTMAN |
|  |  | http://localhost/1\_api/php\_api/employee.php |
|  |  |  |
|  |  | http://localhost/1\_api/php\_api/employee.php |
|  |  |  |
|  |  | Http status code : |
|  |  |  |
|  |  | Connection setup create : Get and Post method |
|  |  | GET Methods : |
|  |  | <?php  include("connection.php");  // Include the connection file  // Create a new instance of the dbobj class and get the connection  $db = new dbobj();  $conn = $db->getConnstring();  // Store the connection object  $request\_method = $\_SERVER["REQUEST\_METHOD"];  switch ($request\_method) {      case 'GET':          getEmployees($conn);  // Pass the connection to the function          break;      default:          header("HTTP/1.0 405 METHOD NOT IMPLEMENTED");          break;  }  function getEmployees($conn)  {      $sql = "SELECT \* FROM employee";      $result = $conn->query($sql);  // Use the passed connection      $response = array();        if ($result->num\_rows > 0) {          while ($row = $result->fetch\_assoc()) {              array\_push($response, $row);          }      }        header('Content-Type: application/json');      echo json\_encode($response);  }  ?> |
|  |  | Call page again through post man |
|  |  |  |
|  |  | Video :2 |
|  |  |  |
|  |  |  |
|  |  | 2\_ Create REST API for mobile app CRUD operations\_v720P  <https://www.youtube.com/watch?v=vxNQ2YGUt5M> |
|  |  | Employee.php to employee convert using .htaccess  C:\xampp\htdocs\1\_api\php\_api\.htaccess |
|  |  | RewriteEngine On  RewriteRule ^employee/?$ employee.php [NC,L]  RewriteRule ^employee/([0-9]+)/?$ employee.php?id=$1 [NC,L] |
|  |  | Call the page  <http://localhost/1_api/php_api/employee> |
|  |  | Verify in postman |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | Raw Data: |
|  |  | [  {  "name": "John Doe",  "email": "john.doe@example.com",  "profilePic": "profilePicURL1"  },  {  "name": "Jane Smith",  "email": "jane.smith@example.com",  "profilePic": "profilePicURL2"  },  {  "name": "Michael Johnson",  "email": "michael.johnson@example.com",  "profilePic": "profilePicURL3"  },  {  "name": "Emily Davis",  "email": "emily.davis@example.com",  "profilePic": "profilePicURL4"  },  {  "name": "David Brown",  "email": "david.brown@example.com",  "profilePic": "profilePicURL5"  },  {  "name": "Sophia Taylor",  "email": "sophia.taylor@example.com",  "profilePic": "profilePicURL6"  },  {  "name": "James Anderson",  "email": "james.anderson@example.com",  "profilePic": "profilePicURL7"  },  {  "name": "Olivia Martinez",  "email": "olivia.martinez@example.com",  "profilePic": "profilePicURL8"  },  {  "name": "William Wilson",  "email": "william.wilson@example.com",  "profilePic": "profilePicURL9"  },  {  "name": "Isabella Moore",  "email": "isabella.moore@example.com",  "profilePic": "profilePicURL10"  }  ] |
|  |  | POST Method : |
|  |  | <?php  include("connection.php");  // Include the connection file  // Create a new instance of the dbobj class and get the connection  $db = new dbobj();  $conn = $db->getConnstring();  // Store the connection object  $request\_method = $\_SERVER["REQUEST\_METHOD"];  switch ($request\_method) {      case 'GET':          getEmployees($conn);  // Pass the connection to the function          break;      case 'POST':          $data = json\_decode(file\_get\_contents('php://input'), true);          insertEmployee($conn, $data["name"], $data["email"], $data["profilePic"]);  // Pass the connection and data to the function            break;      default:          header("HTTP/1.0 405 METHOD NOT IMPLEMENTED");          break;  }  // Function to get employees  function getEmployees($conn)  {      $sql = "SELECT \* FROM employee";      $result = $conn->query($sql);  // Use the passed connection      $response = array();      if ($result->num\_rows > 0) {          while ($row = $result->fetch\_assoc()) {              array\_push($response, $row);          }      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to insert a new employee  function insertEmployee($conn, $name, $email, $profilePic)  {      // Corrected SQL query with the correct variable $profilePic      $sql = "INSERT INTO employee (name, email, profile\_pic) VALUES ('".$name."', '".$email."', '".$profilePic."')";      $response = array();        if ($conn->query($sql)) {          // Success          header("HTTP/1.0 201 Created");          $response = array(              'status' => 1,              'status\_message' => 'Employee Added Successfully.'          );      } else {          // Failed          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Addition Failed.'          );      }      header('Content-Type: application/json');      echo json\_encode($response);  } |
|  |  | Call the page through postaman  <http://localhost/1_api/php_api/employee.php>  <http://localhost/1_api/php_api/employee> |
|  |  |  |
|  |  | Call again Get method |
|  |  |  |
|  |  | PUT Method |
|  |  | <?php  include("connection.php");  // Include the connection file  // Create a new instance of the dbobj class and get the connection  $db = new dbobj();  $conn = $db->getConnstring();  // Store the connection object  $request\_method = $\_SERVER["REQUEST\_METHOD"];  switch ($request\_method) {      case 'GET':          getEmployees($conn);  // Pass the connection to the function          break;      case 'POST':          $data = json\_decode(file\_get\_contents('php://input'), true);          insertEmployee($conn, $data["name"], $data["email"], $data["profilePic"]);  // Pass the connection and data to the function          break;      case 'PUT':          $id = intval($\_GET["id"]);          $data = json\_decode(file\_get\_contents('php://input'), true);          updateEmployee($conn, $id, $data["name"]);  // Pass the connection here as well          break;      default:          header("HTTP/1.0 405 METHOD NOT IMPLEMENTED");          break;  }  // Function to get employees  function getEmployees($conn)  {      $sql = "SELECT \* FROM employee";      $result = $conn->query($sql);  // Use the passed connection      $response = array();      if ($result->num\_rows > 0) {          while ($row = $result->fetch\_assoc()) {              array\_push($response, $row);          }      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to insert a new employee  function insertEmployee($conn, $name, $email, $profilePic)  {      $sql = "INSERT INTO employee (name, email, profile\_pic) VALUES ('".$name."', '".$email."', '".$profilePic."')";      $response = array();        if ($conn->query($sql)) {          header("HTTP/1.0 201 Created");          $response = array(              'status' => 1,              'status\_message' => 'Employee Added Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Addition Failed.'          );      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to update an existing employee  function updateEmployee($conn, $id, $name)  // Accept connection as a parameter  {      $sql = "UPDATE employee SET name='".$name."' WHERE id='".$id."'";      $response = array();        if ($conn->query($sql)) {          // Success          $response = array(              'status' => 1,              'status\_message' => 'Employee Updated Successfully.'          );      } else {          // Failed          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Updation Failed.'          );      }        header('Content-Type: application/json');      echo json\_encode($response);  } |
|  |  |  |
|  |  | Call the page  <http://localhost/1_api/php_api/employee/1> |
|  |  |  |
|  | Call the page | <http://localhost/1_api/php_api/employee.php> |
|  |  | Delete Method : |
|  |  | <?php  include("connection.php");  // Include the connection file  // Create a new instance of the dbobj class and get the connection  $db = new dbobj();  $conn = $db->getConnstring();  // Store the connection object  $request\_method = $\_SERVER["REQUEST\_METHOD"];  switch ($request\_method) {      case 'GET':          getEmployees($conn);  // Pass the connection to the function          break;      case 'POST':          $data = json\_decode(file\_get\_contents('php://input'), true);          insertEmployee($conn, $data["name"], $data["email"], $data["profilePic"]);  // Pass the connection and data to the function          break;      case 'PUT':          $id = intval($\_GET["id"]);          $data = json\_decode(file\_get\_contents('php://input'), true);          updateEmployee($conn, $id, $data["name"]);  // Pass the connection here as well          break;      case 'DELETE':          $id = intval($\_GET["id"]);          DeleteEmployee($conn, $id);  // Pass the connection to the function          break;      default:          header("HTTP/1.0 405 METHOD NOT IMPLEMENTED");          break;  }  // Function to get employees  function getEmployees($conn)  {      $sql = "SELECT \* FROM employee";      $result = $conn->query($sql);  // Use the passed connection      $response = array();      if ($result->num\_rows > 0) {          while ($row = $result->fetch\_assoc()) {              array\_push($response, $row);          }      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to insert a new employee  function insertEmployee($conn, $name, $email, $profilePic)  {      $sql = "INSERT INTO employee (name, email, profile\_pic) VALUES ('".$name."', '".$email."', '".$profilePic."')";      $response = array();        if ($conn->query($sql)) {          header("HTTP/1.0 201 Created");          $response = array(              'status' => 1,              'status\_message' => 'Employee Added Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Addition Failed.'          );      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to update an existing employee  function updateEmployee($conn, $id, $name)  // Accept connection as a parameter  {      $sql = "UPDATE employee SET name='".$name."' WHERE id='".$id."'";      $response = array();        if ($conn->query($sql)) {          // Success          $response = array(              'status' => 1,              'status\_message' => 'Employee Updated Successfully.'          );      } else {          // Failed          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Updation Failed.'          );      }        header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to delete an existing employee  function DeleteEmployee($conn, $id)  // Accept connection as a parameter  {      $sql = "DELETE FROM employee WHERE id='".$id."'";      $response = array();        if ($conn->query($sql)) {          // Success          $response = array(              'status' => 1,              'status\_message' => 'Employee Deleted Successfully.'          );      } else {          // Failed          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee deletion Failed.'          );      }        header('Content-Type: application/json');      echo json\_encode($response);  } |
|  |  |  |
|  |  |  |
|  |  | http://localhost/1\_api/php\_api/employee/8 |
|  |  |  |
|  |  | <http://localhost/1_api/php_api/employee.php> |
|  |  | **Single Employee Call :** |
|  |  | <?php  include("connection.php");  // Include the connection file  // include("auth.php");  //Auth Start  // $auth = new authObj();  // $isAuthorized = $auth->authenticate();  // if(!$isAuthorized)  // {  //  header("HTTP/1.0 401");  //  exit;  // }  //Auth End  // Create a new instance of the dbobj class and get the connection  $db = new dbobj();  $conn = $db->getConnstring();  // Store the connection object    $request\_method = $\_SERVER["REQUEST\_METHOD"];  switch ($request\_method) {      case 'GET':          if (isset($\_GET["id"])) {              // Fetch a single employee by ID              $id = intval($\_GET["id"]);              getEmployeeById($conn, $id);          } else {              // Fetch all employees              getEmployees($conn);          }          break;      case 'POST':          $data = json\_decode(file\_get\_contents('php://input'), true);          insertEmployee($conn, $data["name"], $data["email"], $data["profilePic"]);          break;      case 'PUT':          $id = intval($\_GET["id"]);          $data = json\_decode(file\_get\_contents('php://input'), true);          updateEmployee($conn, $id, $data["name"]);          break;      case 'DELETE':          $id = intval($\_GET["id"]);          DeleteEmployee($conn, $id);          break;      default:          header("HTTP/1.0 405 METHOD NOT IMPLEMENTED");          break;  }  // Function to get all employees  function getEmployees($conn)  {      $sql = "SELECT \* FROM employee";      $result = $conn->query($sql);      $response = array();      if ($result->num\_rows > 0) {          while ($row = $result->fetch\_assoc()) {              array\_push($response, $row);          }      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to get a single employee by ID  function getEmployeeById($conn, $id)  {      $sql = "SELECT \* FROM employee WHERE id='".$id."'";      $result = $conn->query($sql);      $response = array();      if ($result->num\_rows > 0) {          $response = $result->fetch\_assoc();      } else {          // Handle employee not found          header("HTTP/1.0 404 Not Found");          $response = array(              'status' => 0,              'status\_message' => 'Employee Not Found.'          );      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to insert a new employee  function insertEmployee($conn, $name, $email, $profilePic)  {      $sql = "INSERT INTO employee (name, email, profile\_pic) VALUES ('".$name."', '".$email."', '".$profilePic."')";      $response = array();        if ($conn->query($sql)) {          header("HTTP/1.0 201 Created");          $response = array(              'status' => 1,              'status\_message' => 'Employee Added Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Addition Failed.'          );      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to update an existing employee  function updateEmployee($conn, $id, $name)  {      $sql = "UPDATE employee SET name='".$name."' WHERE id='".$id."'";      $response = array();        if ($conn->query($sql)) {          $response = array(              'status' => 1,              'status\_message' => 'Employee Updated Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Updation Failed.'          );      }        header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to delete an existing employee  function DeleteEmployee($conn, $id)  {      $sql = "DELETE FROM employee WHERE id='".$id."'";      $response = array();        if ($conn->query($sql)) {          $response = array(              'status' => 1,              'status\_message' => 'Employee Deleted Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee deletion Failed.'          );      }        header('Content-Type: application/json');      echo json\_encode($response);  } |
|  |  |  |
|  |  |  |
|  |  | |
|  |  | Call the page |
|  |  | <https://www.facebook.com/krishnendu.dalui.5/> |
|  |  | Authentication : |
|  |  | C:\xampp\htdocs\1\_api\php\_api\auth.php |
|  |  | <?php  class authObj  {      var $token = "Y29kZXg6cmVzdF9hcGlfdGVzdAo=";      function authenticate()      {          $headers = getallheaders();          if (!array\_key\_exists('Authorization', $headers)) {              header("HTTP/1.0 400");              echo json\_encode(["error" => "Authorization header is missing"]);              exit;          }            if (substr($headers['Authorization'], 0, 7) !== 'Bearer ') {              header("HTTP/1.0 400");              echo json\_encode(["error" => "Bearer keyword is missing"]);              exit;          }            $bearerToken = trim(substr($headers['Authorization'], 7));          return $bearerToken === $this->token;      }  }  ?> |
|  |  |  |
|  |  | <?php  include("connection.php");  // Include the connection file  include("auth.php");  // Auth Start  $auth = new authObj();  $isAuthorized = $auth->authenticate();  if(!$isAuthorized)  {      header("HTTP/1.0 401");      exit;  }  // Auth End  // Create a new instance of the dbobj class and get the connection  $db = new dbobj();  $conn = $db->getConnstring();  // Store the connection object    $request\_method = $\_SERVER["REQUEST\_METHOD"];  switch ($request\_method) {      case 'GET':          if (isset($\_GET["id"])) {              // Fetch a single employee by ID              $id = intval($\_GET["id"]);              getEmployeeById($conn, $id);          } else {              // Fetch all employees              getEmployees($conn);          }          break;      case 'POST':          $data = json\_decode(file\_get\_contents('php://input'), true);          insertEmployee($conn, $data["name"], $data["email"], $data["profilePic"]);          break;      case 'PUT':          $id = intval($\_GET["id"]);          $data = json\_decode(file\_get\_contents('php://input'), true);          updateEmployee($conn, $id, $data["name"]);          break;      case 'DELETE':          $id = intval($\_GET["id"]);          DeleteEmployee($conn, $id);          break;      default:          header("HTTP/1.0 405 METHOD NOT IMPLEMENTED");          break;  }  // Function to get all employees  function getEmployees($conn)  {      $sql = "SELECT \* FROM employee";      $result = $conn->query($sql);      $response = array();      if ($result->num\_rows > 0) {          while ($row = $result->fetch\_assoc()) {              array\_push($response, $row);          }      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to get a single employee by ID  function getEmployeeById($conn, $id)  {      $sql = "SELECT \* FROM employee WHERE id='".$id."'";      $result = $conn->query($sql);      $response = array();      if ($result->num\_rows > 0) {          $response = $result->fetch\_assoc();      } else {          // Handle employee not found          header("HTTP/1.0 404 Not Found");          $response = array(              'status' => 0,              'status\_message' => 'Employee Not Found.'          );      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to insert a new employee  function insertEmployee($conn, $name, $email, $profilePic)  {      $sql = "INSERT INTO employee (name, email, profile\_pic) VALUES ('".$name."', '".$email."', '".$profilePic."')";      $response = array();        if ($conn->query($sql)) {          header("HTTP/1.0 201 Created");          $response = array(              'status' => 1,              'status\_message' => 'Employee Added Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Addition Failed.'          );      }      header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to update an existing employee  function updateEmployee($conn, $id, $name)  {      $sql = "UPDATE employee SET name='".$name."' WHERE id='".$id."'";      $response = array();        if ($conn->query($sql)) {          $response = array(              'status' => 1,              'status\_message' => 'Employee Updated Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee Updation Failed.'          );      }        header('Content-Type: application/json');      echo json\_encode($response);  }  // Function to delete an existing employee  function DeleteEmployee($conn, $id)  {      $sql = "DELETE FROM employee WHERE id='".$id."'";      $response = array();        if ($conn->query($sql)) {          $response = array(              'status' => 1,              'status\_message' => 'Employee Deleted Successfully.'          );      } else {          header("HTTP/1.0 400 Bad Request");          $response = array(              'status' => 0,              'status\_message' => 'Employee deletion Failed.'          );      }        header('Content-Type: application/json');      echo json\_encode($response);  } |
|  |  |  |
|  |  |  |
|  |  | |
|  |  | http://localhost/1\_api/php\_api/employee/2 |
|  | php\_api\_consumer part | |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\consume.php |
|  |  | <?php  // consume.php  function fetchEmployees($url) {      $ch = curl\_init($url);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      $response = curl\_exec($ch);      curl\_close($ch);      return json\_decode($response, true);  }  // Replace with the actual URL of your API project  $apiUrl = 'http://localhost/1\_api/php\_api/employee'; // Adjust the path accordingly  // Fetch all employees  $employees = fetchEmployees($apiUrl);  // Include the view file to display employees  include('view.php');  ?> |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\view.php |
|  |  | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Employee List</title>      <link rel="stylesheet" href="styles.css"> <!-- Link to CSS file -->  </head>  <body>      <div class="container">          <h1>Employee List</h1>            <a href="add\_employee.php" class="btn">Add New Employee</a>          <?php if (!empty($employees)): ?>              <ul class="employee-list">                  <?php foreach ($employees as $employee): ?>                      <li class="employee-item">                          <strong>ID:</strong> <?= htmlspecialchars($employee["id"]) ?> <br>                          <strong>Name:</strong> <?= htmlspecialchars($employee["name"]) ?> <br>                          <strong>Email:</strong> <?= htmlspecialchars($employee["email"]) ?><br>                          <strong>Profile Pic:</strong>                          <?php if (!empty($employee["profile\_pic"])): ?>                              <img src="<?= htmlspecialchars($employee["profile\_pic"]) ?>" alt="<?= htmlspecialchars($employee["name"]) ?>" class="profile-pic">                          <?php else: ?>                              <span>No image available</span>                          <?php endif; ?>                          <form action="process\_delete\_employee.php" method="POST" style="display:inline;">                              <input type="hidden" name="id" value="<?= htmlspecialchars($employee["id"]) ?>">                              <input type="submit" value="Delete" onclick="return confirm('Are you sure you want to delete this employee?');">                          </form>                      </li>                  <?php endforeach; ?>              </ul>          <?php else: ?>              <p>No employees found.</p>          <?php endif; ?>      </div>  </body>  </html> |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\styles.css |
|  |  | body {      font-family: Arial, sans-serif;      background-color: #f4f4f4;      margin: 0;      padding: 20px;  }  .container {      max-width: 800px;      margin: auto;      background: #fff;      padding: 20px;      border-radius: 8px;      box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  }  h1 {      text-align: center;      color: #333;  }  .employee-list {      list-style-type: none;      padding: 0;  }  .employee-item {      background: #f9f9f9;      margin: 10px 0;      padding: 15px;      border-radius: 5px;      box-shadow: 0 1px 3px rgba(0, 0, 0, 0.2);  }  .employee-item strong {      color: #555;  }  .profile-pic {      max-width:250px; /\* Set max width for profile images \*/      max-height:250px; /\* Set max height for profile images \*/      height: auto;     /\* Maintain aspect ratio \*/      border-radius: 10%; /\* Make it circular \*/      display: block;   /\* Make sure the image is on a new line \*/      margin-top: 10px; /\* Add some spacing above the image \*/  }  .btn {      display: inline-block;      padding: 10px 20px; /\* Top/bottom, left/right padding \*/      margin: 10px 0; /\* Spacing around the button \*/      font-size: 16px; /\* Font size \*/      color: white; /\* Text color \*/      background-color: #007bff; /\* Background color \*/      border: none; /\* No border \*/      border-radius: 5px; /\* Rounded corners \*/      text-decoration: none; /\* Remove underline \*/      text-align: center; /\* Center text \*/      transition: background-color 0.3s ease; /\* Smooth background transition \*/  }  .btn:hover {      background-color: #0056b3; /\* Darker background on hover \*/  }  .btn:active {      background-color: #004085; /\* Even darker when active \*/  } |
|  |  |  |
|  |  | |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\process\_add\_employee.php |
|  |  | <?php  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      $name = $\_POST["name"];      $email = $\_POST["email"];      $profilePic = $\_POST["profilePic"];      $apiUrl = 'http://localhost/1\_api/php\_api/employee'; // Adjust the path accordingly      // Prepare data to be sent      $data = json\_encode([          "name" => $name,          "email" => $email,          "profilePic" => $profilePic      ]);      // Initialize cURL      $ch = curl\_init($apiUrl);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      curl\_setopt($ch, CURLOPT\_POST, true);      curl\_setopt($ch, CURLOPT\_POSTFIELDS, $data);      curl\_setopt($ch, CURLOPT\_HTTPHEADER, [          'Content-Type: application/json',          'Content-Length: ' . strlen($data)      ]);      // Execute the request      $response = curl\_exec($ch);      curl\_close($ch);      // Check response      $responseData = json\_decode($response, true);      if (isset($responseData['status']) && $responseData['status'] === 1) {          echo "<p>Employee added successfully!</p>";      } else {          echo "<p>Error adding employee.</p>";      }      // Link back to employee list      echo '<p><a href="consume.php">View Employee List</a></p>';  }  ?> |
|  |  |  |
|  |  |  |
|  |  | <?php  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      $id = intval($\_POST["id"]);      $apiUrl = 'http://localhost/1\_api/php\_api/employee?id=' . $id; // Adjust the path accordingly      // Initialize cURL      $ch = curl\_init($apiUrl);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, "DELETE");      // Execute the request      $response = curl\_exec($ch);      curl\_close($ch);      // Check response      $responseData = json\_decode($response, true);      if (isset($responseData['status']) && $responseData['status'] === 1) {          echo "<p>Employee deleted successfully!</p>";      } else {          echo "<p>Error deleting employee.</p>";      }      // Link back to employee list      echo '<p><a href="consume.php">View Employee List</a></p>';  }  ?> |
|  |  |  |
|  |  |  |
|  |  | **subtitle.to/**https://www.zee5.com/tv-shows/details/sa-re-ga-ma-pa-2024/0-6-4z5539095/sa-re-ga-ma-pa-2024-september-01-2024/0-1-6z5612493 |
|  |  |  |
|  |  | E Mon Byakul - Jokhon Tokhon  E Mon Byakul - Jokhon Tokhon  Deke Jay - Barey Barey  E Mon Byakul - Jokhon Tokhon  Deke Jay - Barey Barey  Tobu Stobdhota Eshey Dhora  Day Amar - Binar Tar E  E Mon Byakul - Jokhon Tokhon  Deke Jay - Barey Barey  Tobu Stobdhota Eshey Dhora  Day Amar - Binar Tar E  Janina - Janina  Janina - Ki Karon  Janina - Janina  Janina - Ki Karon  E Mon - Keno Day Na Ko Sara  Mane Na Baron - Mon  Keno Day Na Ko Sara  Mane Na Baron - Mon  Janina - Janina  Janina - Ki Karon  Sudhu Vabonara Meley Dana  Track By \_Imtiaz\_007  Ke Jane - Aj Keno Mon  Ke Jane - Aj Keno Mon  Dur Akash - Chhutey Chay  Ke Jane - Aj Keno Mon  Dur Akash - Chhutey Chay  Tobu Dupaye Kiser Ba(n)dha  Jeno Thomke Daray  Ke Jane - Aj Keno Mon  Dur Akash - Chhutey Chay  Tobu Dupaye Kiser Ba(n)dha  Jeno Thomke Daray  Janina - Janina  Janina - Ki Karon  Janina - Janina  Janina - Ki Karon  E Mon - Kano Day Na Ko Sara  Mane Na Baron - Mon  Keno Day Na Ko Sara  Mane Na Baron - Mon  Janina - Janina  Janina - Ki Karon  Janina - Janina  Janina - Ki Karon  Janina - Janina  Janina - Ki Karon  Janina - Janina  Janina - Ki Karon  Janina - Janina  Janina - Ki Karon |
|  |  | |
|  |  | NB : Authentication Part : |
|  |  |  |
|  |  | LIVE |
|  |  | <?php  class dbObj  {      // Default environment (change to 'live' for production)      private $environment = 'local'; // Change this to 'live' when deploying      private $servername;      private $username;      private $password;      private $dbname = "employee"; // Your database name      public $conn;      function \_\_construct()      {          if ($this->environment === 'local') {              // Local server credentials              $this->servername = "localhost"; // Localhost              $this->username = "root"; // Local MySQL username              $this->password = ''; // Local MySQL password          } else {              // Live server credentials              $this->servername = "18.206.58.178"; // Your EC2 public IP or RDS endpoint              $this->username = "root"; // Live MySQL username              $this->password = 'D@2015$yuti'; // Live MySQL password          }      }      function getConnstring()      {          // Establish connection          $con = mysqli\_connect($this->servername, $this->username, $this->password, $this->dbname);            // Check connection          if (!$con) {              die("Connection failed: " . mysqli\_connect\_error());          }          $this->conn = $con;          return $this->conn;      }  }  // // Test connection  // $db = new dbObj();  // $conn = $db->getConnstring();  // if ($conn) {  //     echo "Connected successfully!";  // }  // ?> |
|  |  |  |
|  |  | …or create a new repository on the command line echo "# PHI\_API\_PROVIDER" >> README.md  git init  git add README.md  git commit -m "first commit"  git branch -M master  git remote add origin git@github.com:Code-with-nandu/PHI\_API\_PROVIDER.git  git push -u origin master …or push an existing repository from the command line git remote add origin git@github.com:Code-with-nandu/PHI\_API\_PROVIDER.git  git branch -M master  git push -u origin master |
|  |  | Correct Steps to Clone a Git Repository |
|  |  | 1. **Navigate to the Target Directory:**   bash  Copy code  cd /var/www/html/   1. **Use the git clone Command:**   bash  Copy code  git clone https://github.com/Code-with-nandu/PHI\_API\_PROVIDER.git  This will clone the repository into the /var/www/html/api\_p1 directory.   1. Update   Git pull |
|  |  | Call the page :  <https://krishnendudalui.in.net/PHI_API_PROVIDER/connection.php> |
|  |  |  |
|  |  | Correct Steps to Clone a Git Repository cd /var/www/html/PHI\_API\_PROVIDER  git pull |
|  |  | **If disabling directory listing with .htaccess** |
|  |  | *  Open your Apache configuration file (often found in /etc/apache2/sites-available/000-default.conf or similar):   bash  Copy code  sudo nano /etc/apache2/sites-available/000-default.conf   * Look for a block similar to this:   apache  Copy code  <Directory /var/www/html>  AllowOverride All  </Directory>   **Restart Apache**: After making changes, restart Apache:  bash  Copy code  sudo systemctl restart apache2 |
|  |  | **To save and exit the file in nano, follow these steps:** |
|  |  | To save and exit the file in nano, follow these steps:   1. **Save the changes**:    * Press Ctrl + O (this will prompt you to confirm the filename).    * Press Enter to confirm. 2. **Exit nano**:    * Press Ctrl + X.   If you're using a different text editor, let me know which one, and I can provide specific instructions for that editor! |
|  |  | **To prevent the listing of files in your directory at:** |
|  |  | To prevent the listing of files in your directory at https://krishnendudalui.in.net/PHI\_API\_PROVIDER/, you can disable directory listing in your web server configuration. Here’s how you can do it for common web servers: Apache  1. **Edit the .htaccess file** in your PHI\_API\_PROVIDER directory (create it if it doesn’t exist):   bash  Copy code  nano /var/www/html/PHI\_API\_PROVIDER/.htaccess   1. **Add the following line** to disable directory listing:   apache  Copy code  Options -Indexes   1. **Save and exit** the file. 2. **Restart Apache** to apply the changes:   bash  Copy code  sudo systemctl restart apache2 |
|  |  | To ignore a specific file with a name like ~$p api live\_provider.docx in your Git repository, you need to add a pattern to your .gitignore file. Here's how to do it: |
|  |  | 1. **Open your .gitignore file**: If you don’t have one in your repository, you can create it:   bash  Copy code  nano .gitignore   1. **Add the filename to .gitignore**: You can add the specific filename or a pattern. For your case, add:   plaintext  Copy code  ~$p api live\_provider.docx |
|  |  | PHI API Consumer Project Run in Live : |
|  |  | Recreate the view page : |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\view.php |
|  |  |  |
|  |  | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Employee List</title>      <link rel="stylesheet" href="styles.css"> <!-- Link to CSS file -->  </head>  <body>      <div class="container">          <h1>Employee List (REST API PHP Project)</h1>      <a href="https://docs.google.com/document/d/1ApVI0Lue2hgxE6yhT7yt3vaWxici0VCh/edit" class="btn" target="\_blank" >Step By Step Guide </a>      <a href="https://github.com/Code-with-nandu/PHI\_API\_PROVIDER" class="btn" target="\_blank" > API Provider in GITHUB  </a>      <a href="https://github.com/Code-with-nandu/PHI\_API\_PROVIDER"class="btn"target="\_blank">  API consumer Link GITHUB  </a>      <a href="https://krishnendudalui.in.net/PHI\_API\_PROVIDER/employee" class="btn"target="\_blank">JSON View </a>      <br><br>            <a href="add\_employee.php" class="btn">Add New Employee</a>          <?php if (!empty($employees)): ?>              <ul class="employee-list">                  <?php foreach ($employees as $employee): ?>                      <li class="employee-item">                          <strong>ID:</strong> <?= htmlspecialchars($employee["id"]) ?> <br>                          <strong>Name:</strong> <?= htmlspecialchars($employee["name"]) ?> <br>                          <strong>Email:</strong> <?= htmlspecialchars($employee["email"]) ?><br>                          <strong>Profile Pic:</strong>                          <?php if (!empty($employee["profile\_pic"])): ?>                              <img src="<?= htmlspecialchars($employee["profile\_pic"]) ?>" alt="<?= htmlspecialchars($employee["name"]) ?>" class="profile-pic">                          <?php else: ?>                              <span>No image available</span>                          <?php endif; ?>                          <form action="process\_delete\_employee.php" method="POST" style="display:inline;">                              <input type="hidden" name="id" value="<?= htmlspecialchars($employee["id"]) ?>">                              <input type="submit" value="Delete" onclick="return confirm('Are you sure you want to delete this employee?');">                          </form>                      </li>                  <?php endforeach; ?>              </ul>          <?php else: ?>              <p>No employees found.</p>          <?php endif; ?>      </div>  </body>  </html> |
|  |  | Prices showing in descending order :  C:\xampp\htdocs\1\_api\php\_api\_consumer\consume.php |
|  |  | <?php  // consume.php  function fetchEmployees($url) {      $ch = curl\_init($url);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      $response = curl\_exec($ch);      curl\_close($ch);      return json\_decode($response, true);  }  // Replace with the actual URL of your API project  $apiUrl = 'http://localhost/1\_api/php\_api/employee'; // Adjust the path accordingly  // Fetch all employees  $employees = fetchEmployees($apiUrl);  // Sort employees in descending order by ID (or any other field you prefer)  if (!empty($employees)) {      usort($employees, function($a, $b) {          return $b['id'] <=> $a['id']; // Change 'id' to another field if needed      });  }  // Include the view file to display employees  include('view.php');  ?> |
|  |  |  |
|  |  |  |
|  |  | …or create a new repository on the command line echo "# php-api-consumer" >> README.md  git init  git add README.md  git commit -m "first commit"  git branch -M master  git remote add origin git@github.com:Code-with-nandu/php-api-consumer.git  git push -u origin master …or push an existing repository from the command line git remote add origin git@github.com:Code-with-nandu/php-api-consumer.git  git branch -M master  git push -u origin master |
|  |  |  |
|  |  | Correct Steps to Clone a Git Repository |
|  |  | 1. **Navigate to the Target Directory:**   bash  Copy code  cd /var/www/html/   1. **Use the git clone Command:**   bash  Copy code  git clone https://github.com/Code-with-nandu/php-api-consumer.git |
|  |  |  |
|  |  | Call the page : |
|  |  | <https://krishnendudalui.in.net/php-api-consumer/consume.php> |
|  |  |  |
|  |  | // Determine the environment (local or live) |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\consume.php |
|  |  | <?php  // consume.php  function fetchEmployees($url) {      $ch = curl\_init($url);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      $response = curl\_exec($ch);      curl\_close($ch);      return json\_decode($response, true);  }  // Determine the environment (local or live)  $isLocal = true; // Set to false when deploying to live  // Set the API URL based on the environment  $apiUrl = $isLocal ? 'http://localhost/1\_api/php\_api/employee' : 'https://krishnendudalui.in.net/PHI\_API\_PROVIDER/employee';  // Fetch all employees  $employees = fetchEmployees($apiUrl);  // Sort employees in descending order by ID (or any other field you prefer)  if (!empty($employees)) {      usort($employees, function($a, $b) {          return $b['id'] <=> $a['id']; // Change 'id' to another field if needed      });  }  // Include the view file to display employees  include('view.php');  ?> |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\process\_add\_employee.php// Determine the environment (local or live) |
|  |  | <?php  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      $name = $\_POST["name"];      $email = $\_POST["email"];      $profilePic = $\_POST["profilePic"];  // Determine the environment (local or live)  $isLocal = false; // Set to false when deploying to live  // Set the API URL based on the environment  $apiUrl = $isLocal ? 'http://localhost/1\_api/php\_api/employee' : 'https://krishnendudalui.in.net/PHI\_API\_PROVIDER/employee';        // Prepare data to be sent      $data = json\_encode([          "name" => $name,          "email" => $email,          "profilePic" => $profilePic      ]);      // Initialize cURL      $ch = curl\_init($apiUrl);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      curl\_setopt($ch, CURLOPT\_POST, true);      curl\_setopt($ch, CURLOPT\_POSTFIELDS, $data);      curl\_setopt($ch, CURLOPT\_HTTPHEADER, [          'Content-Type: application/json',          'Content-Length: ' . strlen($data)      ]);      // Execute the request      $response = curl\_exec($ch);      curl\_close($ch);      // Check response      $responseData = json\_decode($response, true);      if (isset($responseData['status']) && $responseData['status'] === 1) {          echo "<p>Employee added successfully!</p>";      } else {          echo "<p>Error adding employee.</p>";      }      // Link back to employee list      echo '<p><a href="consume.php">View Employee List</a></p>';  }  ?> |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\process\_delete\_employee.php/ Determine the environment (local or live) |
|  |  | <?php  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      $id = intval($\_POST["id"]);   // Determine the environment (local or live)   $isLocal = false; // Set to true for local testing   // Set the API URL based on the environment   $apiUrl = $isLocal ? 'http://localhost/1\_api/php\_api/employee?id=' . $id : 'https://krishnendudalui.in.net/PHI\_API\_PROVIDER/employee?id=' . $id;      // Initialize cURL      $ch = curl\_init($apiUrl);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, "DELETE");      // Execute the request      $response = curl\_exec($ch);      curl\_close($ch);      // Check response      $responseData = json\_decode($response, true);      if (isset($responseData['status']) && $responseData['status'] === 1) {          echo "<p>Employee deleted successfully!</p>";      } else {          echo "<p>Error deleting employee.</p>";      }      // Link back to employee list      echo '<p><a href="consume.php">View Employee List</a></p>';  }  ?> |
|  |  | Correct Steps to Clone a Git Repository cd /var/www/html/php-api-consumer  git pull |
|  |  | **Call the page**  [**https://krishnendudalui.in.net/php-api-consumer/consume.php**](https://krishnendudalui.in.net/php-api-consumer/consume.php) |
|  |  | **C:\xampp\htdocs\1\_api\php\_api\_consumer\.htaccess**  **.htaccess file added** |
|  |  | **RewriteEngine On**  **RewriteRule ^consume/?$ consume.php [NC,L]**  **RewriteRule ^consume/([0-9]+)/?$ consume.php?id=$1 [NC,L]**  **Options -Indexes** |
|  |  | **Call the page after upadate (without php mode )**  [**https://krishnendudalui.in.net/php-api-consumer/consume**](https://krishnendudalui.in.net/php-api-consumer/consume) |
|  |  |  |
|  |  |  |
|  |  | Direct access forbidden :  <https://krishnendudalui.in.net/php-api-consumer/> |
|  |  |  |
|  |  |  |
|  |  | To ignore a specific file with a name like ~$p api live\_provider.docx in your Git repository, you need to add a pattern to your .gitignore file. Here's how to do it: |
|  |  | 1. **Open your .gitignore file**: If you don’t have one in your repository, you can create it:   bash  Copy code  nano .gitignore   1. **Add the filename to .gitignore**: You can add the specific filename or a pattern. For your case, add:   plaintext  Copy code  ~$p api live\_provider.docx |
|  |  | Bewtify the page : |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\add\_employee.php |
|  |  |  |
|  |  | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Add New Employee</title>      <link rel="stylesheet" href="styles.css"> <!-- Link to CSS file -->      <style>          body {              background-image: url('https://gurudev.artofliving.org/wp-content/uploads/2022/10/270-Gurudev-Sri-Sri-Ravi-Shankar.jpg'); /\* Background image \*/              background-size: cover; /\* Cover the entire viewport \*/              background-position: center; /\* Center the image \*/              color: #f0f0f0; /\* Light text color \*/              font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif; /\* Modern font \*/              margin: 0;              padding: 0;          }          .container {              background-color: rgba(0, 0, 0, 0.8); /\* Dark semi-transparent background \*/              padding: 30px;              border-radius: 15px; /\* Rounded corners \*/              max-width: 500px; /\* Limit width \*/              margin: 100px auto; /\* Center the container with top margin \*/              box-shadow: 0 8px 16px rgba(0, 0, 0, 0.5); /\* Add shadow \*/              text-align: center; /\* Center text \*/          }          h1 {              margin-bottom: 20px; /\* Space below heading \*/              font-size: 2em; /\* Larger heading \*/          }          label {              display: block; /\* Make labels block elements \*/              margin: 15px 0 5px; /\* Add some margin \*/              font-weight: bold; /\* Bold labels \*/          }          input[type="text"],          input[type="email"] {              width: calc(100% - 20px); /\* Full width with padding \*/              padding: 10px; /\* Padding for inputs \*/              margin-bottom: 15px; /\* Space below inputs \*/              border: 1px solid #ccc; /\* Light border \*/              border-radius: 5px; /\* Rounded corners \*/              font-size: 1em; /\* Font size \*/          }          input[type="text"]:focus,          input[type="email"]:focus {              border-color: #4CAF50; /\* Green border on focus \*/              outline: none; /\* Remove outline \*/          }          input[type="submit"] {              background-color: #4CAF50; /\* Green background \*/              color: white; /\* White text \*/              border: none; /\* Remove default border \*/              padding: 12px; /\* Padding for button \*/              border-radius: 5px; /\* Rounded corners \*/              cursor: pointer; /\* Pointer cursor on hover \*/              width: 100%; /\* Full width button \*/              font-size: 1em; /\* Font size \*/              transition: background-color 0.3s; /\* Smooth transition \*/          }          input[type="submit"]:hover {              background-color: #45a049; /\* Darker green on hover \*/          }          @media (max-width: 600px) {              .container {                  width: 90%; /\* Responsive width \*/              }          }      </style>  </head>  <body>      <div class="container">          <h1>Add New Employee</h1>          <form action="process\_add\_employee.php" method="POST" enctype="multipart/form-data">              <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="profilePic">Profile Picture (URL):</label>              <input type="text" id="profilePic" name="profilePic" placeholder="Optional">              <input type="submit" value="Add Employee">          </form>      </div>  </body>  </html> |
|  |  |  |
|  |  | C:\xampp\htdocs\1\_api\php\_api\_consumer\process\_add\_employee.php |
|  |  |  |
|  |  | <?php  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      $name = $\_POST["name"];      $email = $\_POST["email"];      $profilePic = $\_POST["profilePic"];      // Determine the environment (local or live)      $isLocal = false; // Set to false when deploying to live      // Set the API URL based on the environment      $apiUrl = $isLocal ? 'http://localhost/1\_api/php\_api/employee' : 'https://krishnendudalui.in.net/PHI\_API\_PROVIDER/employee';      // Prepare data to be sent      $data = json\_encode([          "name" => $name,          "email" => $email,          "profilePic" => $profilePic      ]);      // Initialize cURL      $ch = curl\_init($apiUrl);      curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);      curl\_setopt($ch, CURLOPT\_POST, true);      curl\_setopt($ch, CURLOPT\_POSTFIELDS, $data);      curl\_setopt($ch, CURLOPT\_HTTPHEADER, [          'Content-Type: application/json',          'Content-Length: ' . strlen($data)      ]);      // Execute the request      $response = curl\_exec($ch);      curl\_close($ch);      // Check response      $responseData = json\_decode($response, true);        // Prepare the message      $message = '';      if (isset($responseData['status']) && $responseData['status'] === 1) {          $message = '<p class="success-message">Employee added successfully!</p>';      } else {          $message = '<p class="error-message">Error adding employee. Please try again.</p>';      }      // Link back to employee list      $link = '<p><a href="consume.php" class="view-list-link">View Employee List</a></p>';  }  ?>  <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Add New Employee</title>      <link rel="stylesheet" href="styles.css"> <!-- Link to CSS file -->      <style>          body {              display: flex;              flex-direction: column;              align-items: center;              justify-content: center;              height: 100vh; /\* Full viewport height \*/              background-color: #f4f4f4; /\* Light background color \*/              font-family: Arial, sans-serif; /\* Default font \*/          }          .success-message, .error-message {              padding: 10px;              border-radius: 5px; /\* Rounded corners \*/              margin: 10px 0;          }          .success-message {              color: green;              background-color: rgba(144, 238, 144, 0.2); /\* Light green background \*/              border: 1px solid green;          }          .error-message {              color: red;              background-color: rgba(255, 99, 71, 0.2); /\* Light red background \*/              border: 1px solid red;          }          .view-list-link {              text-decoration: underline;              color: blue;              font-weight: bold;              margin-top: 10px;          }      </style>  </head>  <body>      <div class="message-container">          <?php          // Output the message and link          if (isset($message)) {              echo $message;              echo $link;          }          ?>      </div>  </body>  </html> |
|  |  |  |
|  |  |  |
|  |  |  |