



Experiment No. 7



Title: REST APIs IN PHP

Batch: B2**Roll No.: 16010420117****Experiment No.:7****Aim: REST APIs IN PHP****Resources needed:** Windows OS, Web Browser, Editor, XAMPP Server**Pre Lab/ Prior Concepts:**

Students should have prior knowledge of HTML/CSS/Basic Programming, web services.

Theory:

HTTP is designed considering the REST (Representational State Transfer) architecture which defines how to develop and consume the web service. REST API provides easy to implement without less complexity and stateless web service. SOAP (Simple Object Access Protocol architecture) based communication is another approach web services can adopt.

REST API supports multiple data formats such as Command Separated Value (CSV), JavaScript Object Notation (JSON), Extensible Markup Language (XML) as compare to SOAP [1][2].

Many REST APIs use JSON (or JavaScript Object Notation) to carry responses from REST API endpoints. PHP natively supports converting data to JSON format from PHP variables and vice versa through its json extension [1].

To get a JSON representation of a PHP variable, use `json_encode()`:

```
$data = array(1, 2, "three");
$jsonData = json_encode($data);
echo $jsonData;
```

Result will be : [1, 2, "three"]

Similarly, if you have a string containing JSON data, you can turn it into a PHP variable using `json_decode()`:

```
<?php
$jsonData = "[1, \"KJSCE\", 2001]";
$data = json_decode($jsonData);
print_r($data);
?>
```

Result will be:

```
Array(
[0] => 1
[1] => KJSCE
[2] => 2001
```

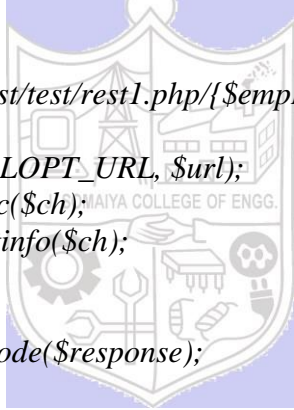
)

There is no direct translation between PHP objects and JSON objects, these objects are an associative array.

Retrieving resources:

To retrieve the resource, GET method of HTTP is used. It uses the **curl** extension to format an HTTP request, set parameters on it, send the request, and get the returned information. To retrieve information about the resource, first construct a URL representing the endpoint for that resource; it initializes a curl resource and provides the constructed URL to it. Finally, the curl object is executed, which sends the HTTP request, waits for the response, and returns it as demonstrated in the below code snippet.

```
<?php
$empId = '1';
$url = "http://localhost/test/rest1.php/{ $empId}";
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
$response = curl_exec($ch);
$resultInfo = curl_getinfo($ch);
curl_close($ch);
// decode the JSON
$empJson = json_decode($response);
Print_r($empJson);
?>
```

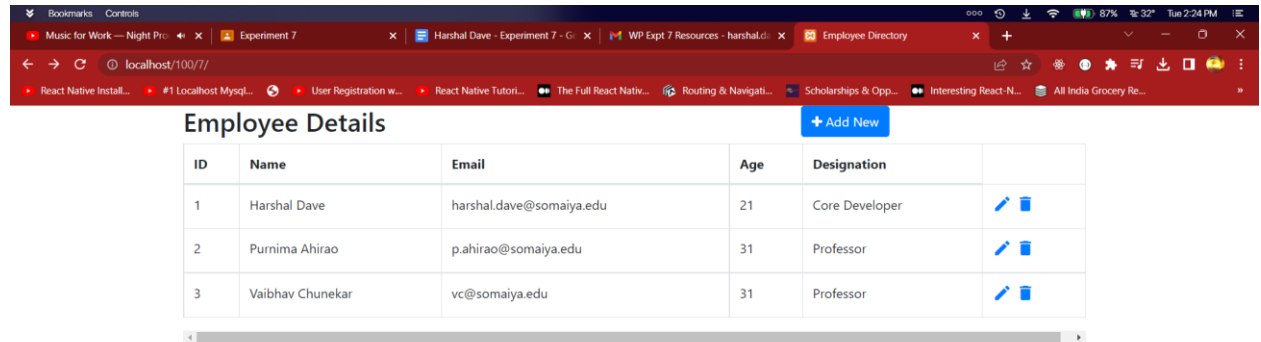


In this case, the response is JSON data, which is decoded and handed off for future processing. The PUT method of HTTP is used to update the resource and DELETE method of HTTP is used to delete the resource.

Activity:

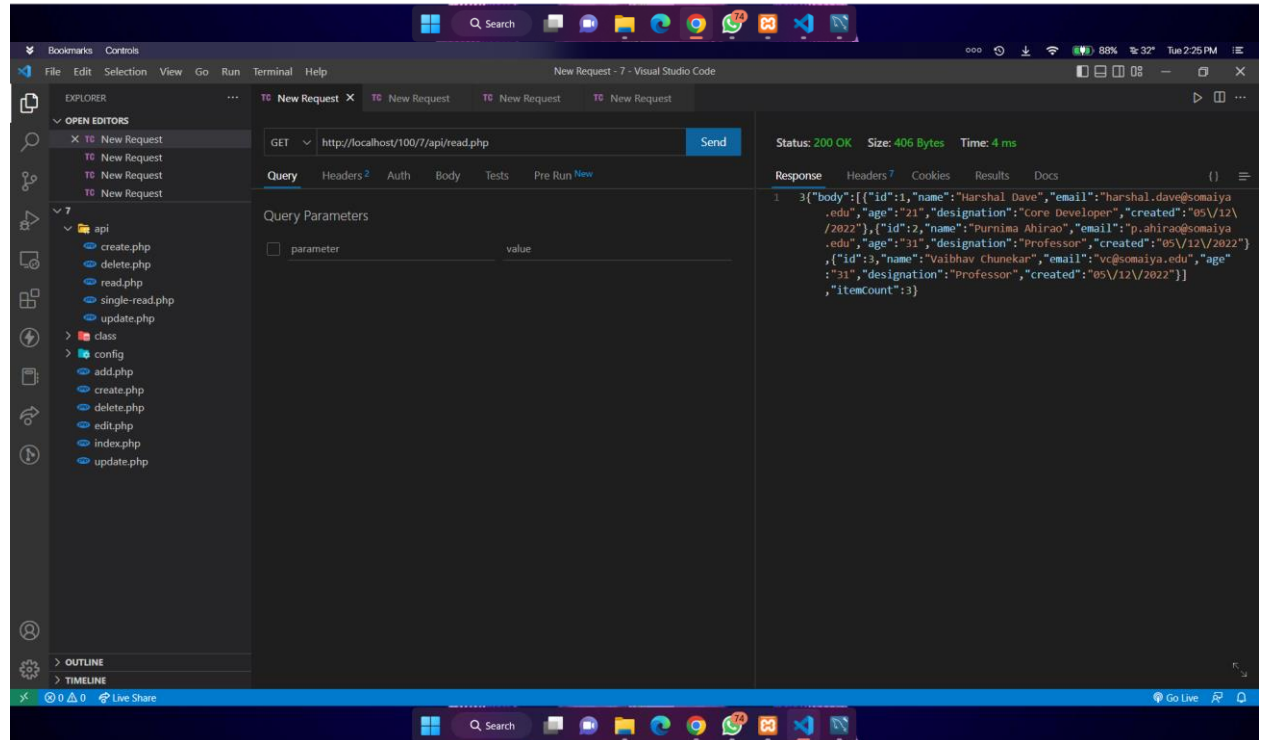
- Implement REST API to retrieve information of a resource
 - Accept the required resource details from the user and using REST API retrieve information of the required resource.
-

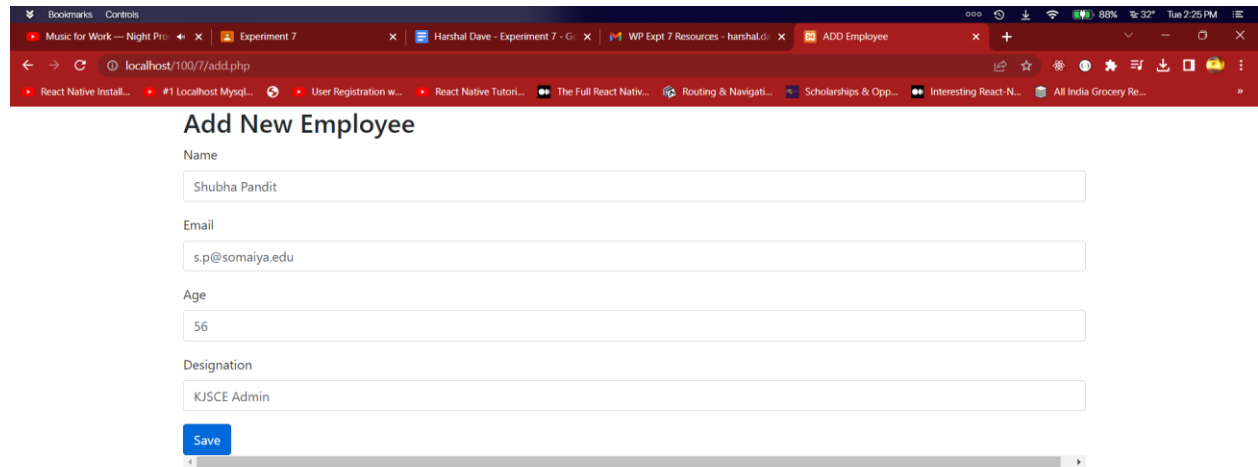
Output(Code with result Snapshot)



Employee Details [+ Add New](#)

ID	Name	Email	Age	Designation
1	Harshal Dave	harshal.dave@somaiya.edu	21	Core Developer
2	Purnima Ahirao	p.ahirao@somaiya.edu	31	Professor
3	Vaibhav Chunekar	vc@somaiya.edu	31	Professor





Bookmarks Controls

Experiment 7 | Harshal Dave - Experiment 7 - G... | WP Expt 7 Resources - harshal.d... | ADD Employee

localhost/10077/add.php

Add New Employee

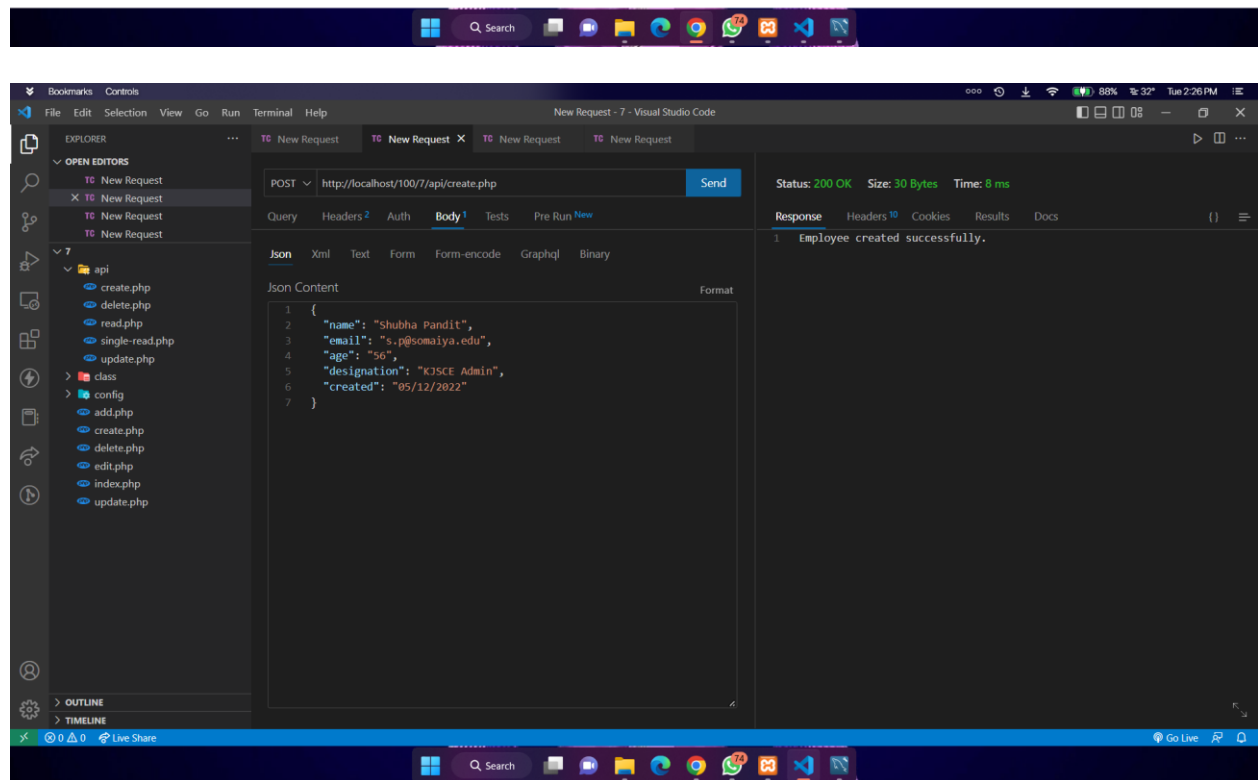
Name

Email

Age

Designation

Save



Employee Details [+ Add New](#)

ID	Name	Email	Age	Designation	
1	Harshal Dave	harshal.dave@somaiya.edu	21	Core Developer	
2	Purnima Ahirao	p.ahirao@somaiya.edu	31	Professor	
3	Vaibhav Chunekar	vc@somaiya.edu	31	Professor	
4	Shubha Pandit	s.p@somaiya.edu	56	KJSCE Admin	

Edit Employee

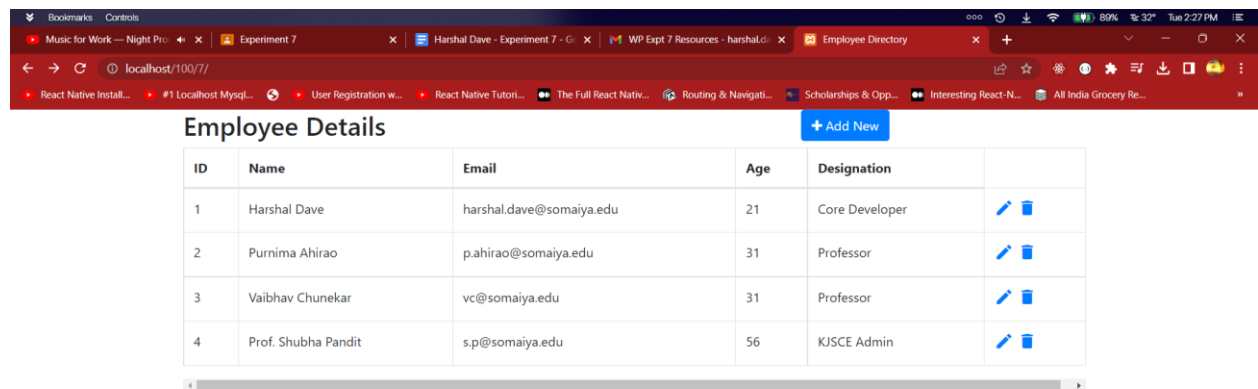
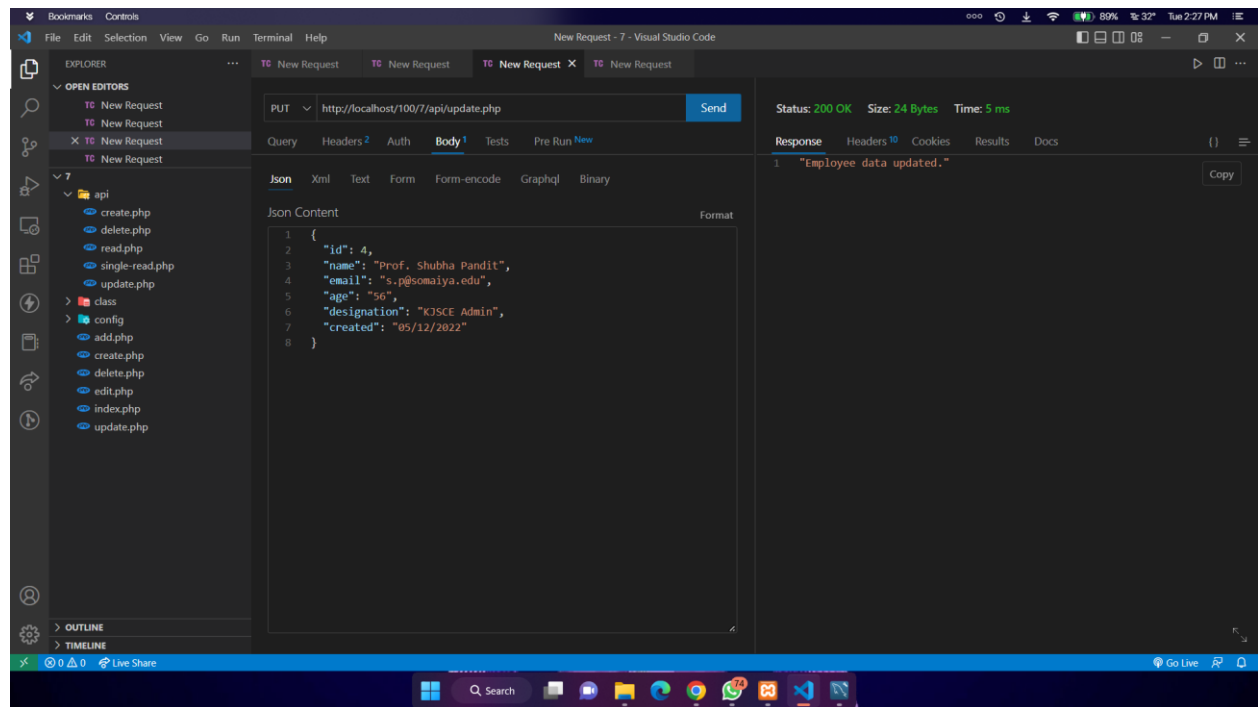
Name

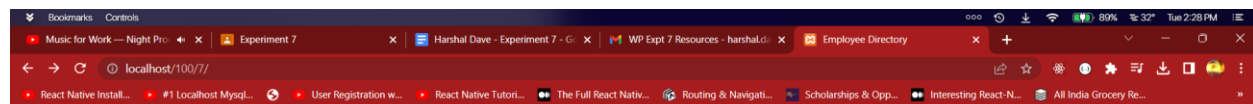
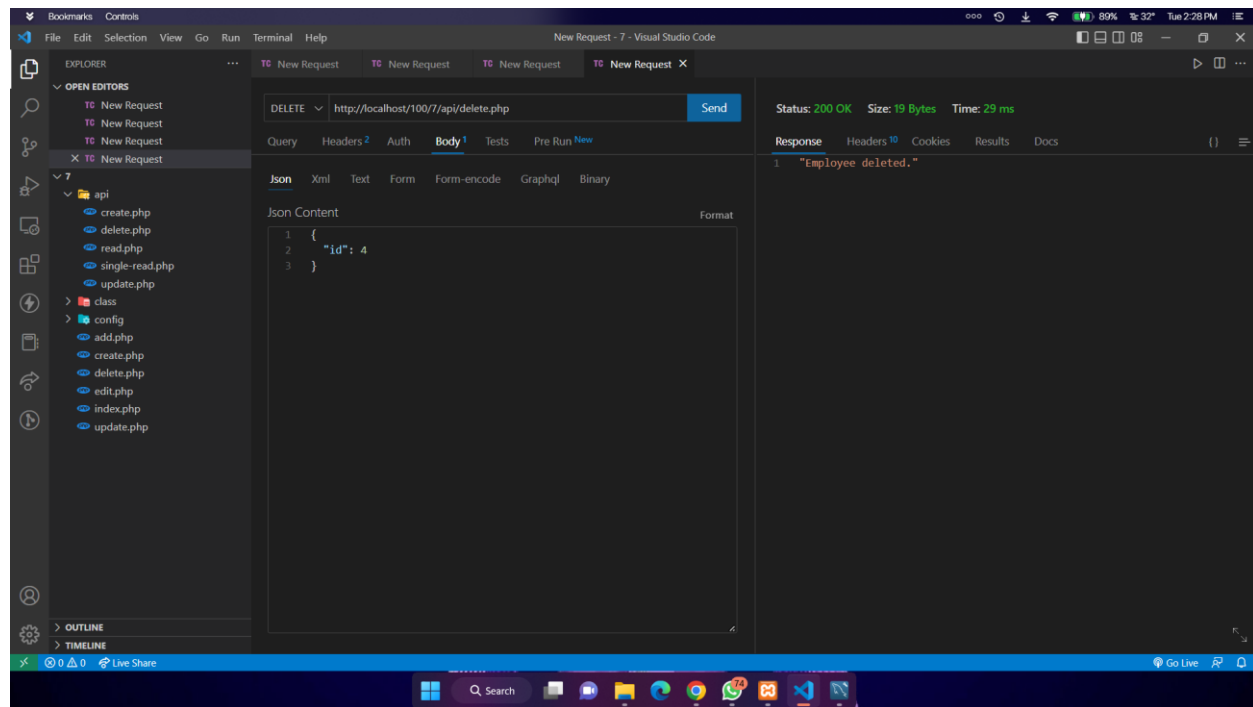
Email

Age







Designation

[Save](#)





Employee Details

ID	Name	Email	Age	Designation	
1	Harshal Dave	harshal.dave@somaiya.edu	21	Core Developer	 
2	Purnima Ahirao	p.ahirao@somaiya.edu	31	Professor	 
3	Vaibhav Chuneekar	vc@somaiya.edu	31	Professor	 

api/create.php

<?php

```
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: POST");
header("Access-Control-Max-Age: 3600");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
```



```

Authorization, X-Requested-With");
include_once '../config/database.php';
include_once '../class/employees.php';
$db = new Database();
$db = $db->getConnection();
$item = new Employee($db);
$data = json_decode(file_get_contents("php://input"));
$item->name = $data->name;
$item->email = $data->email;
$item->age = $data->age;
$item->designation = $data->designation;
// $item->created = date('Y-m-d H:i:s');

if($item->createEmployee()){
    echo 'Employee created successfully.';
} else{
    echo 'Employee could not be created.';
}
?>

```

api/delete.php

```

<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: POST");
header("Access-Control-Max-Age: 3600");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
Authorization, X-Requested-With");

include_once '../config/database.php';
include_once '../class/employees.php';

$db = new Database();
$db = $db->getConnection();

$item = new Employee($db);

```

```

$data = json_decode(file_get_contents("php://input"));

$item->id = $data->id;

if($item->deleteEmployee()){
    echo json_encode("Employee deleted.");
} else{
    echo json_encode("Data could not be deleted");
}
?>

```

api/read.php

```

<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");

include_once '../config/database.php';
include_once '../class/employees.php';
$db = new Database();
$db = $db->getConnection();
$item = new Employee($db);
$stmt = $item->getEmployees();
$itemCount = $stmt->rowCount();

echo json_encode($itemCount);
if($itemCount > 0){

    $employeeArr = array();
    $employeeArr["body"] = array();
    $employeeArr["itemCount"] = $itemCount;
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)){
        extract($row);
        $e = array(
            "id" => $id,
            "name" => $name,

```

```

        "email" => $email,
        "age" => $age,
        "designation" => $designation,
        "created" => $created
    );
    array_push($EmployeeArr["body"], $e);
}
echo json_encode($EmployeeArr);
}
else{
    http_response_code(404);
    echo json_encode(
        array("message" => "No record found.")
    );
}
?>

```

api/single-read.php

```

<?php
    header("Access-Control-Allow-Origin: *");
    header("Content-Type: application/json; charset=UTF-8");
    header("Access-Control-Allow-Methods: POST");
    header("Access-Control-Max-Age: 3600");
    header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
Authorization, X-Requested-With");
    include_once '../config/database.php';
    include_once '../class/employees.php';
    $database = new Database();
    $db = $database->getConnection();
    $item = new Employee($db);
    $item->id = isset($_GET['id']) ? $_GET['id'] : die();

    $item->getSingleEmployee();
    if($item->name != null){
        // create array
        $emp_arr = array(

```

```

        "id" => $item->id,
        "name" => $item->name,
        "email" => $item->email,
        "age" => $item->age,
        "designation" => $item->designation,
        "created" => $item->created
    );

    http_response_code(200);
    echo json_encode($emp_arr);
}

else{
    http_response_code(404);
    echo json_encode("Employee not found.");
}
?>

```

api/update.php

```

<?php
    header("Access-Control-Allow-Origin: *");
    header("Content-Type: application/json; charset=UTF-8");
    header("Access-Control-Allow-Methods: POST");
    header("Access-Control-Max-Age: 3600");
    header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
Authorization, X-Requested-With");

    include_once '../config/database.php';
    include_once '../class/employees.php';

    $database = new Database();
    $db = $database->getConnection();

    $item = new Employee($db);

    $data = json_decode(file_get_contents("php://input"));

```

```

$item->id = $data->id;

// employee values
$item->name = $data->name;
$item->email = $data->email;
$item->age = $data->age;
$item->designation = $data->designation;
$item->created = date('Y-m-d H:i:s');

if($item->updateEmployee()){
    echo json_encode("Employee data updated.");
} else{
    echo json_encode("Data could not be updated");
}
?>

```

class/employees.php

```

<?php
class Employee{
    // Connection
    private $conn;

    // Table
    private $db_table = "Employee";

    // Columns
    public $id;
    public $name;
    public $email;
    public $age;
    public $designation;
    public $created;

    // Db connection
    public function __construct($db){

```

```

    $this->conn = $db;
}

// GET ALL
public function getEmployees(){
    $sqlQuery = "SELECT id, name, email, age, designation, created FROM " . $this->db_table
    . """;

    $stmt = $this->conn->prepare($sqlQuery);
    $stmt->execute();
    return $stmt;
}

// CREATE
public function createEmployee(){
    $sqlQuery = "INSERT INTO
        ". $this->db_table . "
        SET
            name = :name,
            email = :email,
            age = :age,
            designation = :designation,
            created = :created";

    $stmt = $this->conn->prepare($sqlQuery);

    // sanitize
    $this->name=htmlspecialchars(strip_tags($this->name));
    $this->email=htmlspecialchars(strip_tags($this->email));
    $this->age=htmlspecialchars(strip_tags($this->age));
    $this->designation=htmlspecialchars(strip_tags($this->designation));
    $this->created=htmlspecialchars(strip_tags($this->created));

    // bind data
    $stmt->bindParam(":name", $this->name);
    $stmt->bindParam(":email", $this->email);
    $stmt->bindParam(":age", $this->age);

```

```

$stmt->bindParam(":designation", $this->designation);
$stmt->bindParam(":created", $this->created);

if($stmt->execute()){
    return true;
}
return false;
}

// READ single
public function getSingleEmployee(){
    $sqlQuery = "SELECT
        id,
        name,
        email,
        age,
        designation,
        created
    FROM
        ". $this->db_table ."
    WHERE
        id = ?
    LIMIT 0,1";

    $stmt = $this->conn->prepare($sqlQuery);
    $stmt->bindParam(1, $this->id);
    $stmt->execute();
    $dataRow = $stmt->fetch(PDO::FETCH_ASSOC);

    $this->name = $dataRow['name'];
    $this->email = $dataRow['email'];
    $this->age = $dataRow['age'];
    $this->designation = $dataRow['designation'];
    $this->created = $dataRow['created'];
}

// UPDATE

```

```

public function updateEmployee(){
    $sqlQuery = "UPDATE
        ". $this->db_table ."
        SET
            name = :name,
            email = :email,
            age = :age,
            designation = :designation,
            created = :created
        WHERE
            id = :id";

    $stmt = $this->conn->prepare($sqlQuery);

    $this->name=htmlspecialchars(strip_tags($this->name));
    $this->email=htmlspecialchars(strip_tags($this->email));
    $this->age=htmlspecialchars(strip_tags($this->age));
    $this->designation=htmlspecialchars(strip_tags($this->designation));
    $this->created=htmlspecialchars(strip_tags($this->created));
    $this->id=htmlspecialchars(strip_tags($this->id));

    // bind data
    $stmt->bindParam(":name", $this->name);
    $stmt->bindParam(":email", $this->email);
    $stmt->bindParam(":age", $this->age);
    $stmt->bindParam(":designation", $this->designation);
    $stmt->bindParam(":created", $this->created);
    $stmt->bindParam(":id", $this->id);

    if($stmt->execute()){
        return true;
    }
    return false;
}

// DELETE

```



```

function deleteEmployee(){
    $sqlQuery = "DELETE FROM " . $this->db_table . " WHERE id = ?";
    $stmt = $this->conn->prepare($sqlQuery);

    $this->id=htmlspecialchars(strip_tags($this->id));

    $stmt->bindParam(1, $this->id);

    if($stmt->execute()){
        return true;
    }
    return false;
}
?>

```

config/database.php

```

<?php
class Database {
    private $host = "localhost";
    private $database_name = "100";
    private $username = "root";
    private $password = "123456789";
    public $conn;
    public function getConnection(){
        $this->conn = null;
        try{
            $this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->
database_name, $this->username, $this->password);
            $this->conn->exec("set names utf8");
        }catch(PDOException $exception){
            echo "Database could not be connected: " . $exception->getMessage();
        }
        return $this->conn;
    }
}

```

?>

add.php

```

<!DOCTYPE html>
<html lang="en">

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>ADD Employee</title>
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto|Varela+Round|Open+Sans">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <div class="container">
    <div class="table-responsive">
      <div class="table-wrapper">
        <div class="table-title">
          <div class="row">
            <div class="col-sm-8">
              <h2>Add New Employee</h2>
            </div>
          </div>
        </div>
      </div>
    </div>

    <form action="create.php" method="POST" id="myform">

```

```

    <div class="form-group">
        <label>Name</label>
        <input type="text" name="name" class="form-control">
    </div>
    <div class="form-group">
        <label>Email</label>
        <input type="text" name="email" class="form-control">
    </div>
    <div class="form-group">
        <label>Age</label>
        <input type="text" name="age" class="form-control">
    </div>

    <div class="form-group">
        <label>Designation</label>
        <input type="text" name="designation" class="form-control">
    </div>

    <button type="submit" name="submit" class="btn btn-primary">Save</button>

</form>

</div>
</div>
</div>
</body>

</html>

```

create.php

```

<?php
session_start();
if (isset($_POST['name'])) {
    $url = "http://localhost/100/7/api/create.php";
    $ch = curl_init();

```

```

$arr['name'] = $_POST['name'];
$arr['email'] = $_POST['email'];
$arr['age'] = $_POST['age'];
$arr['designation'] = $_POST['designation'];
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $arr);
$result = curl_exec($ch);
curl_close($ch);
$result = json_decode($result, true);

// echo "<pre>";

// print_r($result);
// die();

if (isset($result['status']) && isset($result['code']) && $result['code'] == 10) {
    $_SESSION['success_msg'] = $result['data'];
    header('location:index.php');
    die();
} else {
    echo $result['data'];
}
} else {
    header('location:index.php');
}
}

```

delete.php

```

<?php
session_start();
if (isset($_GET['id']) && isset($_GET['id']) > 0) {
    $curl = curl_init();

    curl_setopt_array($curl, [
        CURLOPT_URL => "http://localhost/100/7/api/delete.php",

```

```

CURLOPT_RETURNTRANSFER => true,
CURLOPT_ENCODING => "",
CURLOPT_MAXREDIRS => 10,
CURLOPT_TIMEOUT => 30,
CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => "DELETE",
CURLOPT_POSTFIELDS => "{\n  \"id\": \" . $_GET['id'] . \"\n}",
CURLOPT_HTTPHEADER => [
    "Accept: */*",
    "Content-Type: application/json",
],
]);

$response = curl_exec($curl);
$error = curl_error($curl);

curl_close($curl);

if ($error) {
    echo "cURL Error #:" . $error;
} else {
    echo $response;
}
} else {
    header('location:index.php');
}

```

edit.php

```

<?php
if (isset($_GET['id']) && isset($_GET['id']) > 0) {
    $url = "https://bootstrapfriendly.com/demo/live-demo/CRUD-operations-with-PHP-cURL-REST-API_1649791942/live/edit.php?token=ABDSC144DSD";
    $ch = curl_init();
    $arr['id'] = $_GET['id'];
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $arr);
    $result = curl_exec($ch);
    curl_close($ch);

    $result = json_decode($result, true);

    // echo '<pre>';
    // print_r($result);
    // echo '</pre>';
} else {
    echo "id not found";
}

?>

<!DOCTYPE html>
<html lang="en">

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Edit Employee</title>
    <link rel="stylesheet"

```

```

href="https://fonts.googleapis.com/css?family=Roboto|Varela+Round|Open+Sans">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <div class="container-lg">
    <div class="table-responsive">
      <div class="table-wrapper">
        <div class="table-title">
          <div class="row">
            <div class="col-sm-12">
              <h2>Edid <b>Employee</b></h2>
            </div>
          </div>
        </div>
      </div>
      <?php
if (isset($result['status']) && isset($result['code']) && isset($result['code']) == 5) {

?>

      <form action="update.php" method="POST" id="myform">

        <table class="table table-bordered">

          <?php foreach ($result['data'] as $list) {

            ?>
            <div class="form-group">
              <input type="hidden" name="id" value="<?php echo $list['id'] ?>"
class="form-control">
            </div>

```

```

        <div class="form-group">
            <label>Name</label>
            <input type="text" name="name" class="form-control">
        </div>
        <div class="form-group">
            <label>Email</label>
            <input type="text" name="email" class="form-control">
        </div>
        <div class="form-group">
            <label>Age</label>
            <input type="text" name="age" class="form-control">
        </div>

        <div class="form-group">
            <label>Designation</label>
            <input type="text" name="designation" class="form-control">
        </div>

        <button type="submit" name="submit" class="btn btn-
primary">Save</button>

    <?php

    } ?>

</form>

<?php

} else {
    //echo $result['data'];
}

?>

</div>

```



```

    </div>
  </div>
</body>

</html>

```

index.php

```

<?php

$curl = curl_init();

curl_setopt_array($curl, [
    CURLOPT_URL => "http://localhost/100/7/api/read.php",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => [
        "Accept: */*",
    ],
]);

$response = curl_exec($curl);
$error = curl_error($curl);

curl_close($curl);

if ($error) {
    echo "cURL Error #:" . $error;
} else {
    $str1 = substr($response, 1);
    $result = json_decode($str1, true);
    // echo $response;
}

```

```

?>

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Employee Directory</title>
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto|Varela+Round|Open+Sans">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <div class="container">
    <div class="table-responsive">
      <div class="table-wrapper">
        <div class="table-title">
          <div class="row">
            <div class="col-sm-8">
              <h2>Employee Details</h2>
            </div>
            <div class="col-sm-4">
              <a href="add.php" class="btn btn-primary add-new"><i class="fa fa-plus"></i>
Add New</a>
            </div>
          </div>
        </div>
      </div>
    </div>
    <form action="" method="POST" id="myform">

```

```

<table class="table table-bordered">
  <thead>
    <tr>
      <th>ID</th>
      <th>Name</th>
      <th>Email</th>
      <th>Age</th>
      <th>Designation</th>
    </tr>
  </thead>
  <tbody>

    <?php foreach ($result['body'] as $list) {
    ?>
      <tr>
        <td><?php echo $list['id'] ?></td>
        <td><?php echo $list['name'] ?></td>
        <td><?php echo $list['email'] ?></td>
        <td><?php echo $list['age'] ?></td>
        <td><?php echo $list['designation'] ?></td>
        <td>
          <a href="edit.php?id=<?php echo $list['id'] ?>" class="edit" title="Edit"
data-toggle="tooltip"><i class="material-icons"></i></a>
          <a href="delete.php?id=<?php echo $list['id'] ?>" class="delete"
title="Delete" data-toggle="tooltip"><i class="material-icons"></i></a>
        </td>
      </tr>

    <?php
    } ?>

  </tbody>
</table>
</form>
</div>
</div>

```

```

</div>
</body>

</html>

```

update.php

```

<?php
session_start();
if (isset($_POST['name'])) {
    $curl = curl_init();

    curl_setopt_array($curl, [
        CURLOPT_URL => "http://localhost/100/7/api/update.php",
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => "",
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 30,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => "PUT",
        CURLOPT_POSTFIELDS => "{\n  \"id\": \"".$_POST['id']."\",\n  \"name\": \"".$_POST['name']."\n  \"email\": \"".$_POST['email']."\n  \"age\": \"".$_POST['age']."\n  \"designation\": \"".$_POST['email']."\n}",
        CURLOPT_HTTPHEADER => [
            "Accept: */*",
            "Content-Type: application/json",
            "User-Agent: Thunder Client (https://www.thunderclient.com)"
        ],
    ]);

    $response = curl_exec($curl);
    $err = curl_error($curl);

    curl_close($curl);

    if ($err) {
        echo "cURL Error #:" . $err;
    }
}

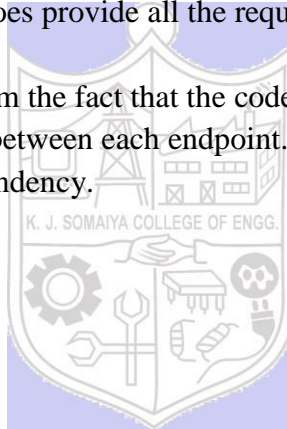
```

```
} else {  
    echo $response;  
}  
} else {  
    header('location:index.php');  
}
```

Questions:-

1. What is the purpose of REST API?
 - Lightweight. One of the main benefits of REST APIs is that they rely on the HTTP standard, which means it's format-agnostic and you can use XML, JSON, HTML, etc. ...
 - Independent. Another benefit of REST APIs is the fact that the client and server are independent. This means we can use the same APIs to integrate on mobile devices as well as web devices.
 - Scalable and flexible.

2. What are the other alternatives to REST API?
 - SOAP: It is a communication protocol that uses XML messages to authenticate, authorize and execute remote code. It tries to provide an interface for remote method execution.
 - GraphQL: While REST focuses on resources and provides a set of endpoints around them to allow for multiple operations, GraphQL focuses on the data. It is more a query language than an actual API, although it does provide all the required tools to access the queried data as well.
 - RPC: The client is abstracted from the fact that the code used is not local, how to contact it and what happens with the data between each endpoint. Instead you just perform the call like you would with a local dependency.



Outcomes: **CO4:** Demonstrate the use advanced features such as REST API, email handling, Localization and internationalization in PHP.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

REST API Architecture in PHP was successfully understood and implemented.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge with date

References:

Books:

1. Thomson PHP and MySQL Web Development Addison-Wesley Professional, 5th Edition 2016.
2. Peter MacIntyre, Kevin Tatro Programming PHP O'Reilly Media, Inc, 4th Edition 2020
3. Frank M. Kromann Beginning PHP and MySQL: From Novice to Professional, Apress 1st Edition, 2018

