

AWT Practical Week - 13 Task

Web Services Using PHP

- 1) Consume web-service(API) using php & display it on webpage from <https://openweathermap.org/> or any other web-service.

Code:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Consuming API</title>
9   <link rel="stylesheet" href="../css/style.css">
10  <style>
11    @import url('https://fonts.googleapis.com/css2?family=Noto+Sans+Display:wght@400;500;600&display=swap');
12
13    * {
14      margin: 0;
15      padding: 0;
16      font-family: 'Noto Sans Display', sans-serif;
17    }
18  </style>
19 </head>
20
21 <body>
22   <div class="header">My Weather App - 19IT035 Krutik Gadhiya</div>
23   <form class="location">
24     <input required type="text" name="city" class="input">
25     <button type="submit" name="getWeather" class="button">GO!</button>
26   </form>
27   <?php
28   if (isset($_GET['getWeather'])) {

```

```

28   if (isset($_GET['getWeather'])) {
29     $city = $_GET['city'];
30     $weatherUrl = "https://api.openweathermap.org/data/2.5/weather?q=$city&appid=248249feaaf37fd426460b1713t";
31     // $res = file_get_contents($weatherUrl);
32     $ch = curl_init();
33     curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
34     curl_setopt($ch, CURLOPT_URL, $weatherUrl);
35     $res = curl_exec($ch);
36     $res = json_decode($res, true);
37     // echo "<pre>";
38     // print_r($res);
39     // echo "</pre>";
40     $iconUrl = "http://openweathermap.org/img/wn/" . $res['weather'][0]['icon'] . "@2x.png";
41     $seaLevel = @$res['main']['sea_level'] ? $res['main']['sea_level'] : "&#x268A;";
42     $grndLevel = @$res['main']['grnd_level'] ? $res['main']['grnd_level'] : "&#x268A;";
43     $gust = @$res['wind']['gust'] ? $res['wind']['gust'] : "knots : "&#x268A;";
44     ?>
45     <main class="main-container">
46       <div class="main-weather">
47         <div class="icon"><img src=<?php echo $iconUrl ?> alt=""></div>
48         <h1 class="temperature"><?php echo $res['main']['temp']; ?>°C</h1>
49         <h2 class="h2"><?php echo $res['weather'][0]['main']; ?></h2>
50         <h3 class="h3"><?php echo $res['weather'][0]['description']; ?></h3>
51         <div class="other-details">
52           <p class="property">Feels Like: <span class="value"><?php echo $res['main']['feels_like']; ?> °C</span>
53           <p class="property">MAX Temperature: <span class="value"><?php echo $res['main']['temp_max']; ?> °
54           <p class="property">MIN Temperature: <span class="value"><?php echo $res['main']['temp_min']; ?> °
55           <p class="property">Pressure: <span class="value"><?php echo $res['main']['pressure']; ?> hPa</span>

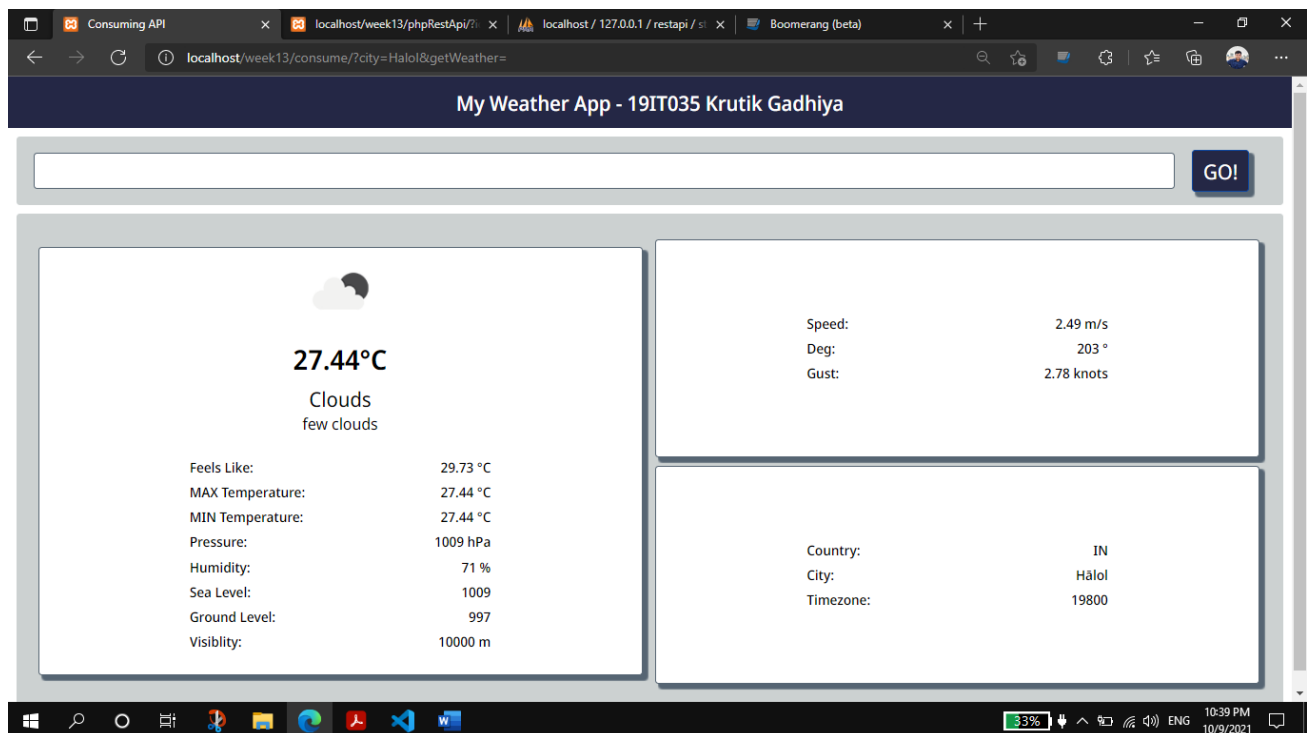
```

```

index.php - week13 - Visual Studio Code
index.php consume U x index.php phpRestApi U testRestApijs U database.php U
consume > index.php > html > body > main-main-container > div.main-weather > div.other-details > p.property > span.value
52 <p class="property">Feels Like: <span class="value"><?php echo $res['main']['feels_like']; ?> °C</span></p>
53 <p class="property">MAX Temperature: <span class="value"><?php echo $res['main']['temp_max']; ?> °C</span></p>
54 <p class="property">MIN Temperature: <span class="value"><?php echo $res['main']['temp_min']; ?> °C</span></p>
55 <p class="property">Pressure: <span class="value"><?php echo $res['main']['pressure']; ?> hPa</span></p>
56 <p class="property">Humidity: <span class="value"><?php echo $res['main']['humidity']; ?> %</span></p>
57 <p class="property">Sea Level: <span class="value"><?php echo $seaLevel; ?></span></p>
58 <p class="property">Ground Level: <span class="value"><?php echo $grndLevel; ?></span></p>
59 <p class="property">Visibility: <span class="value"><?php echo $res['visibility']; ?> m</span></p>
60 </div>
61 </div>
62 <div class="details">
63 <div class="wind-container">
64 <div class="other-details">
65 <p class="property">Speed: <span class="value"><?php echo $res['wind']['speed']; ?> m/s</span></p>
66 <p class="property">Deg: <span class="value"><?php echo $res['wind']['deg']; ?> °</span></p>
67 <p class="property">Gust: <span class="value"><?php echo $gust; ?></span></p>
68 </div>
69 </div>
70 <div class="city-info">
71 <div class="other-details">
72 <p class="property">Country: <span class="value"><?php echo $res['sys']['country']; ?></span></p>
73 <p class="property">City: <span class="value"><?php echo $res['name']; ?></span></p>
74 <p class="property">Timezone: <span class="value"><?php echo $res['timezone']; ?></span></p>
75 </div>
76 </div>
77 </div>
78 </main>
79 <?php

```

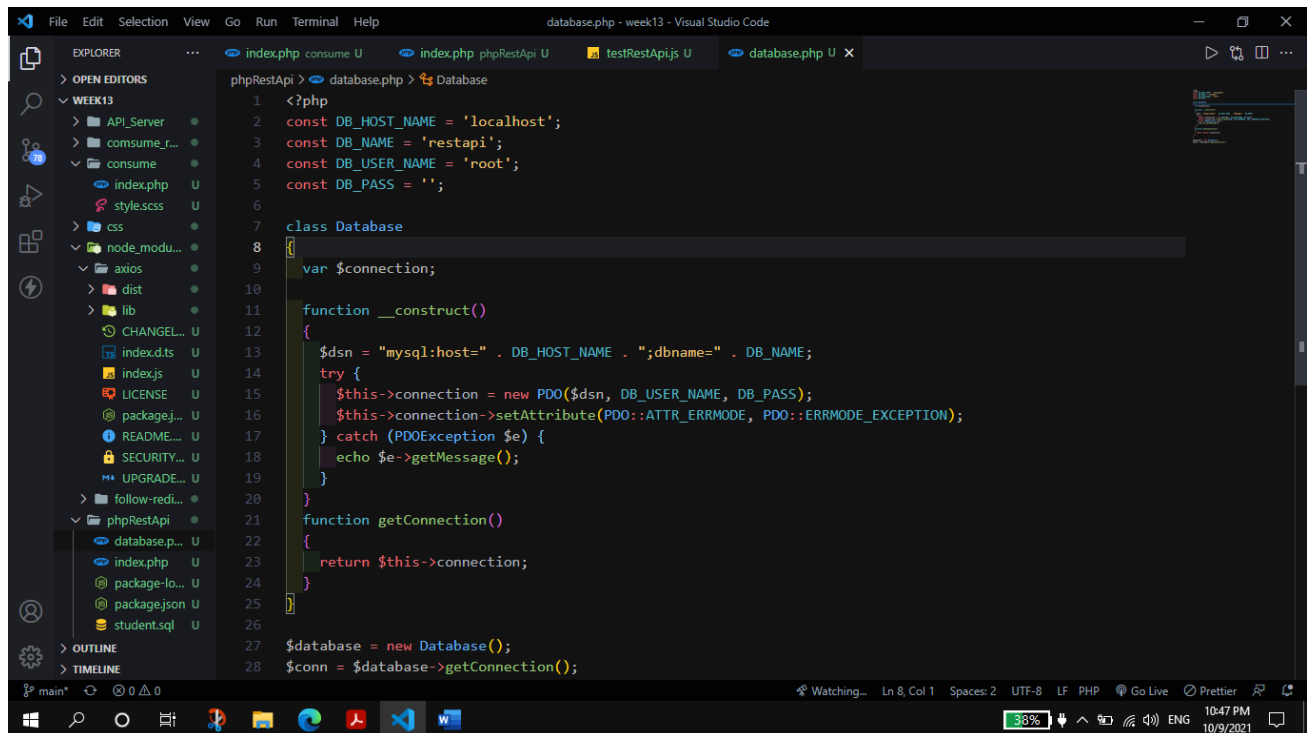
Output:



2) Create REST- API using PHP and Consume it in Python/Java/Nodejs.

- API will receive Student Id using GET method and response with Student Details in JSON from Database.
- API will receive student data(json) to update student details in database using POST Method]

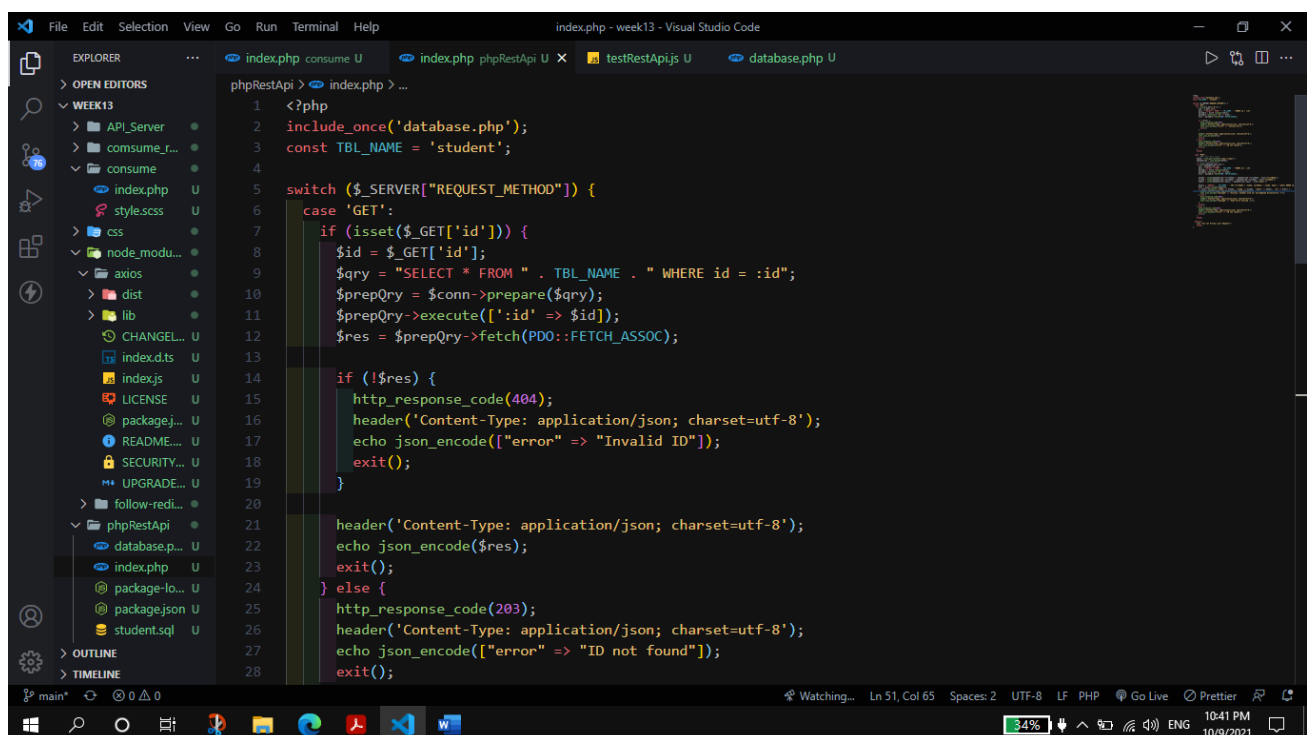
Code: Database.php



```

1  <?php
2  const DB_HOST_NAME = 'localhost';
3  const DB_NAME = 'restapi';
4  const DB_USER_NAME = 'root';
5  const DB_PASS = '';
6
7  class Database
8  {
9      var $connection;
10
11     function __construct()
12     {
13         $dsn = "mysql:host=" . DB_HOST_NAME . ";dbname=" . DB_NAME;
14         try {
15             $this->connection = new PDO($dsn, DB_USER_NAME, DB_PASS);
16             $this->connection->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
17         } catch (PDOException $e) {
18             echo $e->getMessage();
19         }
20     }
21
22     function getConnection()
23     {
24         return $this->connection;
25     }
26
27     $database = new Database();
28     $conn = $database->getConnection();
  
```

index.php (api)



```

1  <?php
2  include_once('database.php');
3  const TBL_NAME = 'student';
4
5  switch ($SERVER["REQUEST_METHOD"]) {
6      case 'GET':
7          if (isset($_GET['id'])) {
8              $id = $_GET['id'];
9              $qry = "SELECT * FROM " . TBL_NAME . " WHERE id = :id";
10             $prepQry = $conn->prepare($qry);
11             $prepQry->execute([':id' => $id]);
12             $res = $prepQry->fetch(PDO::FETCH_ASSOC);
13
14             if (!$res) {
15                 http_response_code(404);
16                 header('Content-Type: application/json; charset=utf-8');
17                 echo json_encode(["error" => "Invalid ID"]);
18                 exit();
19             }
20
21             header('Content-Type: application/json; charset=utf-8');
22             echo json_encode($res);
23             exit();
24         } else {
25             http_response_code(203);
26             header('Content-Type: application/json; charset=utf-8');
27             echo json_encode(["error" => "ID not found"]);
28             exit();
29         }
30     }
  
```

```
29 break;
30
31
32 case 'POST':
33     // TODO: Post data route
34     $body = file_get_contents('php://input');
35     $bodyParsed = json_decode($body);
36     // print_r($bodyParsed);
37     if (isset($bodyParsed->id)) {
38         $id = $bodyParsed->id;
39         $qry = "SELECT * FROM " . TBL_NAME . " WHERE id = :id";
40         $prepQry = $conn->prepare($qry);
41         $prepQry->execute([':id' => $id]);
42         $res = $prepQry->fetch(PDO::FETCH_ASSOC);
43
44         $fname = isset($bodyParsed->firstName) ? $bodyParsed->firstName : $res['firstName'];
45         $lname = isset($bodyParsed->lastName) ? $bodyParsed->lastName : $res['lastName'];
46         $email = isset($bodyParsed->email) ? $bodyParsed->email : $res['email'];
47
48         $query = 'UPDATE ' . TBL_NAME . ' SET firstName = :fname, lastName = :lname, email = :email WHERE id = :id';
49         $res = $conn->prepare($query);
50         if ($res->execute([':fname' => $fname, ':lname' => $lname, ':email' => $email, ':id' => $id])) {
51             header('Content-Type: application/json; charset=utf-8');
52             echo json_encode(["message" => "Success: Student with id: $id Updated Successfully :"]);
53         } else {
54             http_response_code(500);
55             header('Content-Type: application/json; charset=utf-8');
56             echo json_encode(["message" => "Some Error Occured :"]);
57         }
58     }
59     default:
60         echo "Can not Process your Request!";
61         break;
62     }
```

```
48 $query = 'UPDATE ' . TBL_NAME . ' SET firstName = :fname, lastName = :lname, email = :email WHERE id = :id';
49 $res = $conn->prepare($query);
50 if ($res->execute([':fname' => $fname, ':lname' => $lname, ':email' => $email, ':id' => $id])) {
51     header('Content-Type: application/json; charset=utf-8');
52     echo json_encode(["message" => "Success: Student with id: $id Updated Successfully :"]);
53 } else {
54     http_response_code(500);
55     header('Content-Type: application/json; charset=utf-8');
56     echo json_encode(["message" => "Some Error Occured :"]);
57 }
58 exit();
59 } else {
60     http_response_code(203);
61     header('Content-Type: application/json; charset=utf-8');
62     echo json_encode(["error" => "ID not found"]);
63     exit();
64 }
65 break;
66
67 default:
68     echo "Can not Process your Request!";
69     break;
70 }
71 }
```

testRestApi.js

```

1  const axios = require('axios')
2
3  const getStudent = async (id = 1) => {
4      try {
5          // const res = await fetch(`http://localhost/week13/phpRestApi/?id=${id}`, {
6          const res = await axios.get(`http://localhost/week13/phpRestApi/?id=${id}`)
7          console.log(`ID: ${id}`, res.data)
8      } catch (err) {
9          console.log(err)
10     }
11 }
12
13 const updateStudent = async (id = 1, fname = undefined, lname = undefined, email = undefined) => {
14     try {
15         const res = await axios.post('http://localhost/week13/phpRestApi/', {
16             id,
17             firstName: fname,
18             lastName: lname,
19             email
20         })
21         console.log(`ID: ${id}`, res.data)
22     } catch (err) {
23         console.log(err)
24     }
25 }
26
27 const main = async () => {
28     await getStudent()
29     await updateStudent(id = 1, fname = 'KRUTIK')
30     await getStudent()
31     await getStudent(3)
32     await updateStudent(id = 3, fname = 'PragnenKumar')
33     await getStudent(3)
34 }
35 main()

```

Output:

```

ID: 1 {
  id: '1',
  firstName: 'KRUTIK',
  lastName: 'Gad',
  email: '19IT035@charusat.edu.in',
  createdAt: '2021-10-08 22:54:28'
}
ID: 1 { message: 'Success: Student with id: 1 Updated Successfully :)' }
ID: 1 {
  id: '1',
  firstName: 'KRUTIK',
  lastName: 'Gad',
  email: '19IT035@charusat.edu.in',
  createdAt: '2021-10-08 22:54:28'
}
ID: 3 {
  id: '3',
  firstName: 'Pragnen',
  lastName: 'Amdavadi',
  email: '19IT002@charusat.edu.in',
  createdAt: '2021-10-08 22:54:28'
}
ID: 3 { message: 'Success: Student with id: 3 Updated Successfully :)' }
ID: 3 {
  id: '3',
  firstName: 'PragnenKumar',
  lastName: 'Amdavadi',
  email: '19IT002@charusat.edu.in',
  createdAt: '2021-10-08 22:54:28'
}
C:\xampp\htdocs\week13>

```