SIES College of Arts, Science & Commerce

RISE WITH EDUCATION

NAAC REACCREDITED - 'A' GRADE

ISO 9001 : 2008

S.I.E.S College of Arts, Science and Commerce(Empowered Autonomous)
Sion(W), Mumbai – 400 022.

CERTIFICATE

This is to certify that Miss/Mr **_AJAY GAUTAM IRLE**
Roll No. TCS2526020_ has successfully completed the necessary course of experiments in the subject of **Web Service** during the academic year **2025 – 2026** complying with the requirements of **University of Mumbai**, for the course of **TYBSC Computer Science [Semester-V].**

Prof. In-Charge
RAJESH RAMNARESH YADAV

Examination date:

Examiner's Signature & Date:

Signature of HOD
DR. MANOJ SINGH

College Seal

Name of Subject Incharge: Mr. Rajesh Yadav

SIES College of Arts, Science & Commerce

RISE WITH EDUCATION

NAAC REACCREDITED - 'A' GRADE

ISO 9001 : 2008

S.I.E.S College of Arts, Science and Commerce(Empowered Autonomous)
Sion(W), Mumbai – 400 022.

CERTIFICATE

This is to certify that Miss/Mr **AJAY GAUTAM IRLE** Roll No. **TCS2526020** has successfully completed the necessary course of experiments in the subject of **Web Service** during the academic year **2025 – 2026** complying with the requirements of **University of Mumbai**, for the course of **TYBSC Computer Science [Semester-V].**

8/9/25

Prof. In-Charge
Rajesh Ramnaresh Yadav

Examination date:

Examiner's Signature & Date:

Signature of HOD
DR. MANOJ SINGH

College Seal

Name of Subject Incharge: Mr. Rajesh Yadav

Total Att. = 100%

SIES COLLEGE OF ARTS SCIENCE AND COMMERCE (EMPOWERED AUTONOMOUS)
SION - WEST , MUMBAI-22

Name of the Student: AJAY . G. IRLE **Roll No:** TCS2526020

Class : TYCS **Semester:** V

Subject: WEB SERVICES

Year: 2025-26

INDEX

Sr.No	DATE	TITLE OF PRACTICAL	SIGNATURE OF FACULTY
1) a.	07/07/25	A) Develop a simple SOAP web service using PHP with a function getTemperature(city) and consume it using a php client.	7/7/25
b.	07/07/25	B) Write a SOAP based php ws that display SOAP msg XML request & response.	
c.	07/07/25	c) Develop a simple SOAP web service with a function getTemperature(city) and consume it using a HTML form that take city as input & display onSubmit button.	
d.	07/07/25	d) Write a SOAP Fault Testing Program.	
2) a.	14/07/25	a) SOAP Web Service for Weather Data using PHP & OpenWeatherMap	14/7/25
b.	14/07/25	b) Create a simple HTML form that takes user input (city name) & then fetches weather from PHP Soap service.	
3) a)	17/07/25	Create a php SOAP service that provides a currency conversion & test it with client script	22/7/25
b)	17/07/25	Dynamic Currency Convertor	

Name & Signature of Subject Incharge.
Examiner

Name & Signature of

Name of Subject Incharge: Mr. Rajesh Yadav

Sr. No	Date	Practical Aim	Signature
4)	28/07/25	PHP Soap based program for factorial	28/7/25
5)	28/07/25	PHP Soap based program for gradding based on marks	
6)	28/07/25	a) Create & consume timeserver webservice using ~~web service~~ nodejs.	
	04/08/25	b) PHP client program to consume your Node.js time server REST API	4/8/25
7)	04/08/25	a) Create a Rest API using php that accepts a username via URL & returns a welcome msg in JSON format.	
		b) Create a Rest API using php that accepts a username via URL & returns a welcome msg in JSON format using postman.	
8)	04/08/25	Build a simple WS in php that provides :- live stock trading dashboard.	
9)	11/08/25	PHP-based WS for UGC that: • accepts a clg name as input • Returns the NAAC rating from a MySQL database.	11/8/25
10)	11/08/25	Image upload API	

SIES

Web Services with API
PRACTICAL No. 1-10

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1

AIM :

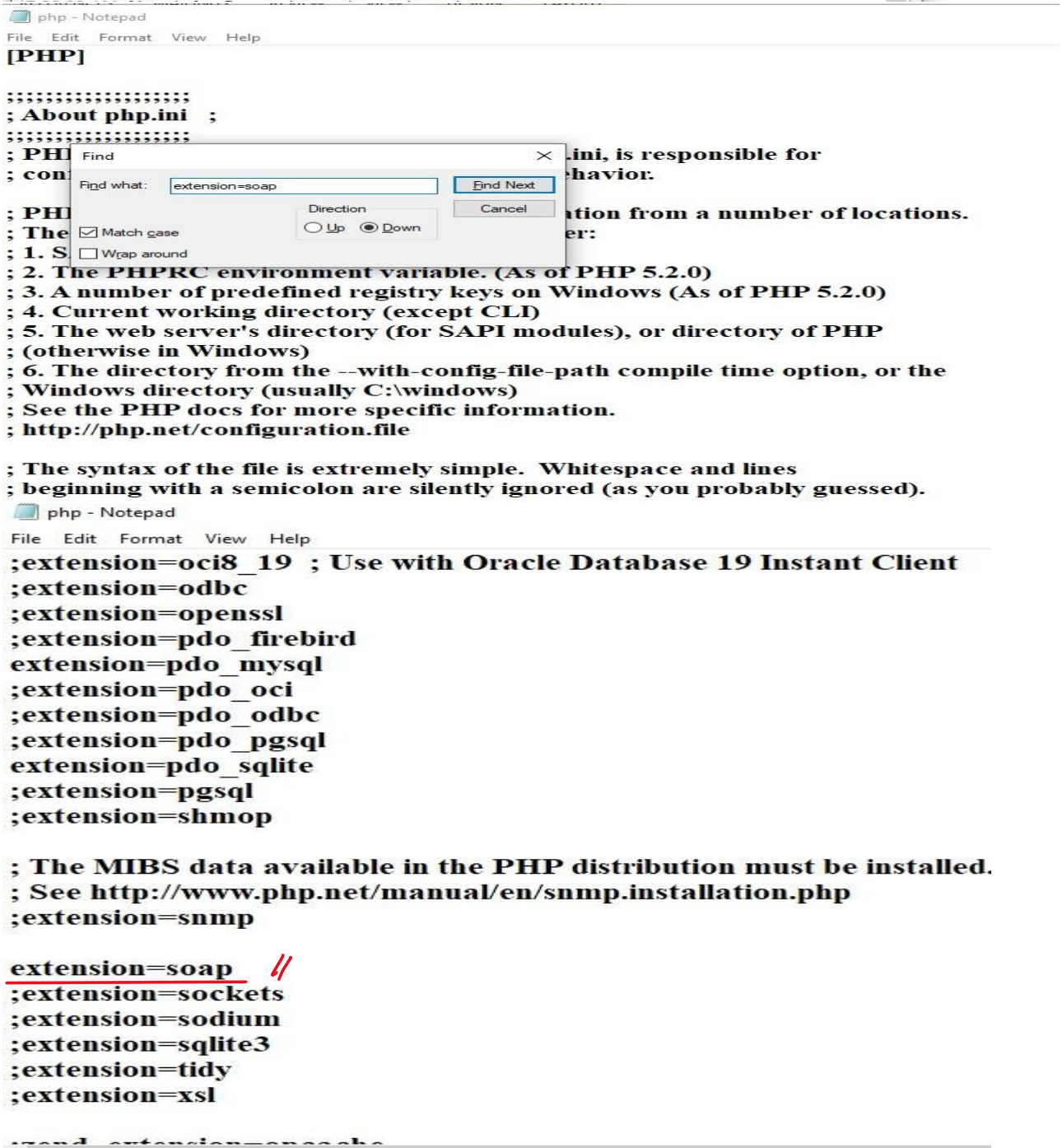
1-A) Develop a simple SOAP web service using PHP with a func on getTemperature(city) and consume it using a PHP client.

DESCRIPTION:

1. php.ini settings

In Xampp ,in config open php.ini find “extension=soap” and remove “;”from front of extension=soap

Name of Subject Incharge: Mr. Rajesh Yadav



Name of Subject Incharge: Mr. Rajesh Yadav

2. server.php

In server.php, we created the getTemperature function and defined an array cityTemperatures where some cities temperatures are stored. We then checked if the city exists in the array using array_key_exists, and returned the temperature if found, otherwise displayed "City not found." We also initialized a SOAP server with a URI, added the function using addFunction, and handled incoming requests using $server>handle().

3. client.php

In client.php, we created a SOAP client using the SoapClient class with the WSDL set to null and passed in an array containing the location and uri of the SOAP server. We then defined a city ("Mumbai") and called the getTemperature method on the client, storing the result in $result. Finally, we printed the response using echo.

CODE:

server.php file

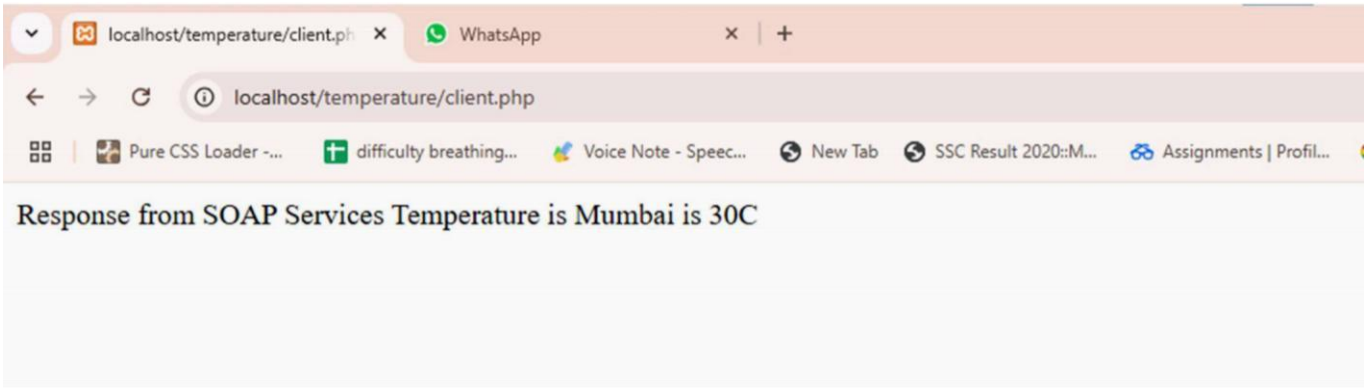
```
<?php function
getTemperature($city){
        $cityTemperatures=[
                "Mumbai"=>"30°C",
                "Delhi"=>"52°C",
                "Chennai"=>"47°C",
                "Kolkata"=>"28°C",
                "Bengaluru"=>"32°C"
];

        if (array_key_exists($city,$cityTemperatures)){
        return "Temperature in $city is" .$cityTemperatures[$city];
        }else{
        return "City not found.";
        }
}
$options=['uri' => 'http://localhost/Temperature1'];
$server=new SoapServer(null,$options);
$server->addFunction('getTemperature');
$server->handle();
?>
```

client.php

```
<?php
```

```
$options=[
'location'  => 'http://localhost/Temperature1/server.php',
'uri' => 'http://localhost/Temperature1'
];
$client=new SoapClient(null,$options);
$city=”Mumbai”;
$result=$client->getTemperature($city);
echo"Response from Soap Service:$result";
?>
```

OUTPUT:



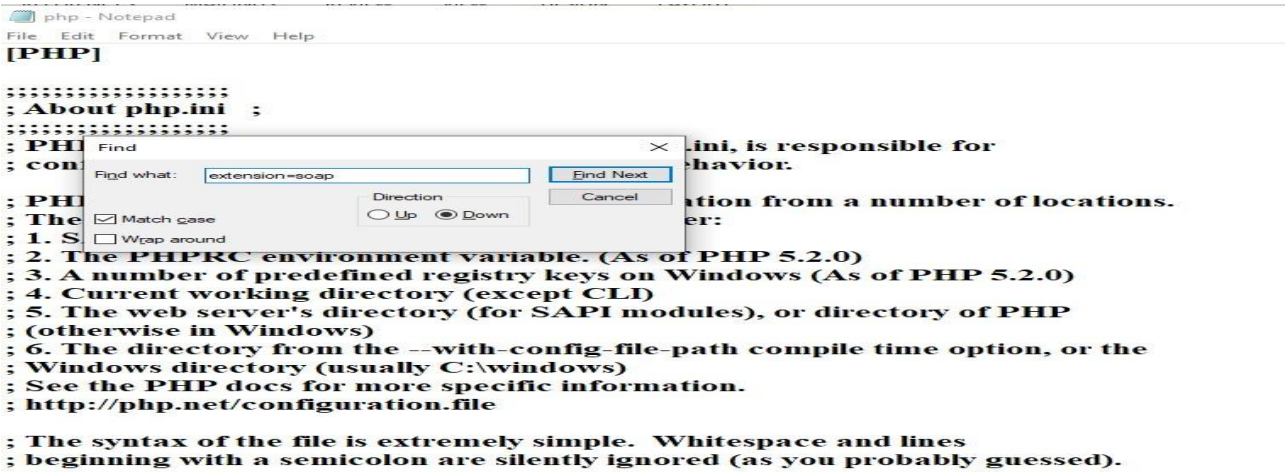
AIM :

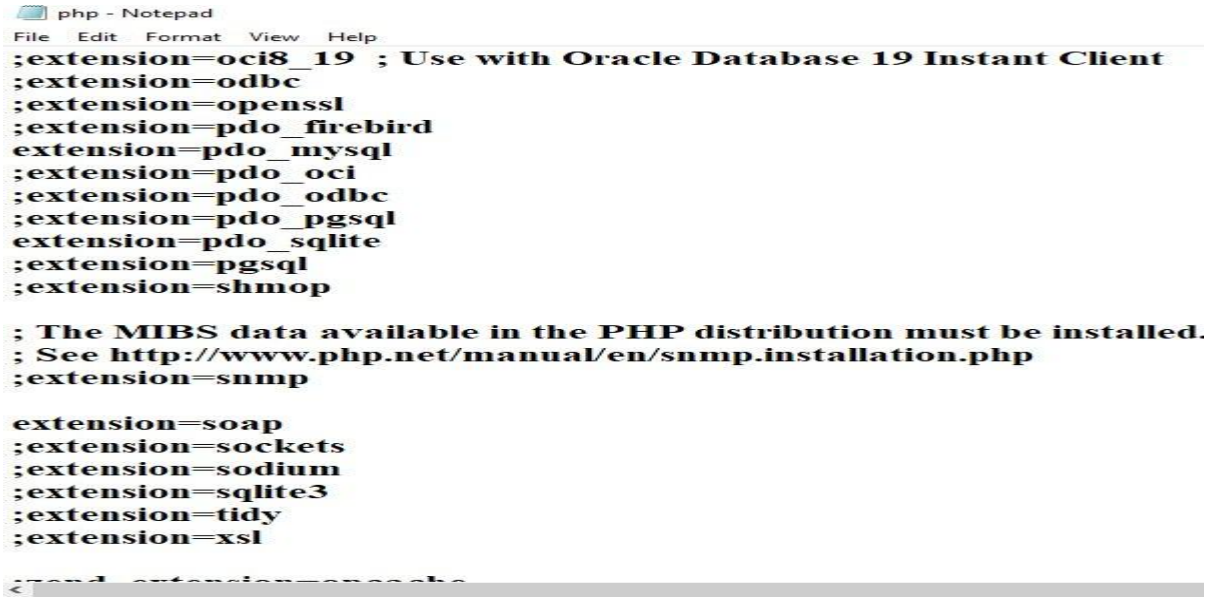
1-B) Write a Soap based PHP web Services that display Soap Message XML Request and Response for the same program.

DESCRIPTION:

1. php.ini settings

In Xampp ,in config open php.ini find “extension=soap” and remove “;”from front of extension=soap





2. server.php

In server.php, we created the getTemperature function and defined an array cityTemperatures where some cities temperatures are stored. We then checked if the city exists in the array using array_key_exists, and returned the temperature if found, otherwise displayed "City not found." We also initialized a SOAP server with a URI, added the function using addFunction, and handled incoming requests using $server>handle().

3. client.php

In client.php, we created a SOAP client with additional options like 'trace' => 1 to capture the request and response details, and 'exception' => true to handle errors using exceptions. We then passed the city name "Mumbai" to the getTemperature method and echoed the response. To help with debugging, we also printed the raw SOAP request and response XML using __getLastRequest() and __getLastResponse(). If any error occurs during the call, it is caught with a SoapFault exception and displayed accordingly.

CODE:

server.php file

```
<?php function
getTemperature($city){
        $cityTemperatures=[
                "Mumbai"=>"45°C",
                "Delhi"=>"52°C",
                "Chennai"=>"47°C",
                "Kolkata"=>"28°C",
                "Bengaluru"=>"32°C"
```

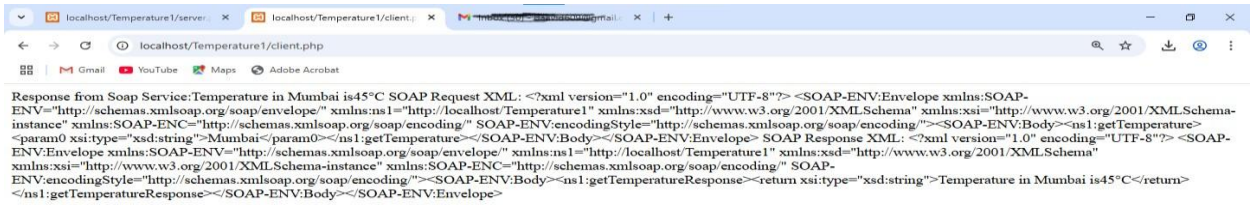
```
];

        if (array_key_exists($city,$cityTemperatures)){        return
"Temperature in $city is" .$cityTemperatures[$city];
        }else{
         return "City not found.";
        }

}

$options=['uri' => 'http://localhost/Temperature1'];
$server=new SoapServer(null,$options);

$server->addFunction('getTemperature');

$server->handle();
```

?> **client.php**

```
<?php

$options=[

 'location'  => 'http://localhost/Temperature1/server.php',

 'uri' => 'http://localhost/Temperature1',

 'trace' =>1,

 'exception'=>true

];

try{

$client=new SoapClient(null,$options);

$city="Mumbai";

$result=$client->getTemperature($city); echo"Response

from Soap Service:$result\n\n";



echo"SOAP Request XML:\n"; echo htmlentities($client-

>__getLastRequest())."\n\n";
```

```
echo "SOAP Response XML:\n"; echo htmlentities($client-
>__getLastResponse())."\n\n";


}catch(SoapFault $e){
$response="SOAP FAULT:" .$e->getMessage();
}
?>
```

OUTPUT:



Name of Subject Incharge: Mr. Rajesh Yadav

SIES

Web Services with API
Practical No.1

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY GAUTAM IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	14/07/2025	**Practical No**	2

A) AIM :

SOAP Web Service for weather data using Php and OpenWeatherMap.

B) DESCRIPTION:

At first, we visit OpenWeatherMap and log in using our account credentials. This platform provides access to a wide range of weather data through its API services. After obtaining an API key, we integrate it into our PHP-based SOAP web service. The WeatherService class uses this key to fetch real-time temperature and weather descriptions for a given city. To ensure the SOAP XML response remains intact and error-free, we disable PHP error output using ini_set('display_errors', 0) and error_reporting(0). This prevents any warnings or notices from interfering with the structured XML output expected by SOAP clients. The client script then calls the getTemp() method remotely, retrieves the weather data, and displays it in a user-friendly format.

C) CODE:

Server.php

```
<?php
//Disable error output to avoid breaking SOAP XML response
ini_set('display errors',0);
error_reporting(0);

class WeatherService{
        private $apikey='eed34cf9392a35de629ca8c147cb963d';

        public function getTemp($city){
                $city=trim($city);
```

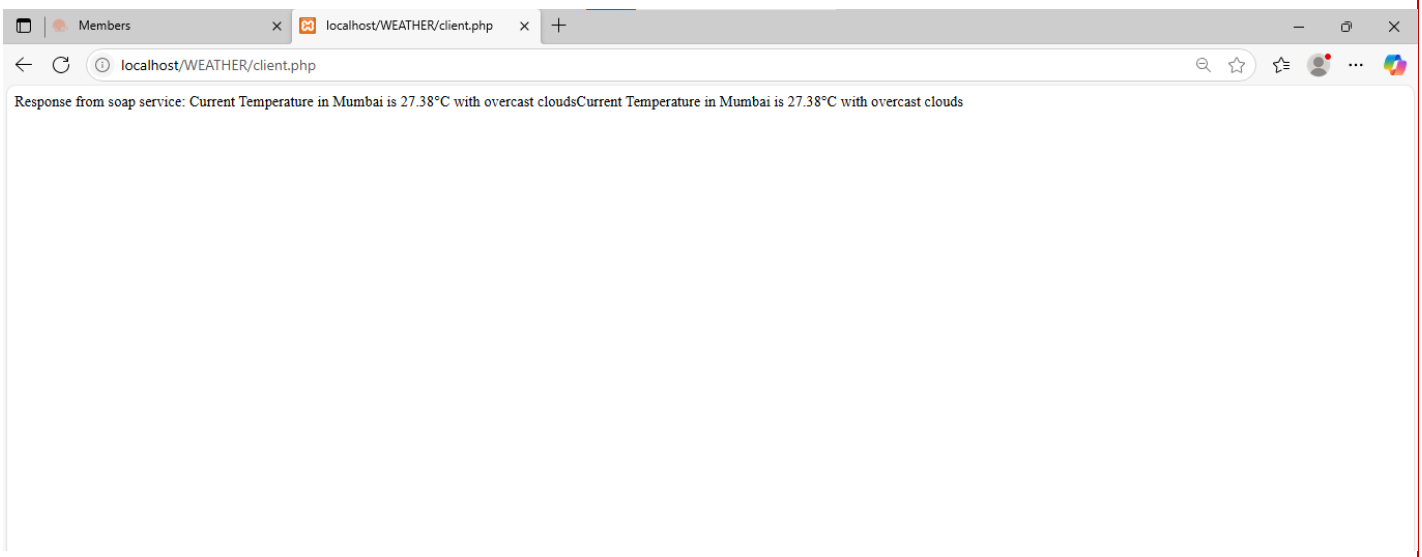
Name of Subject Incharge: Mr. Rajesh Yadav

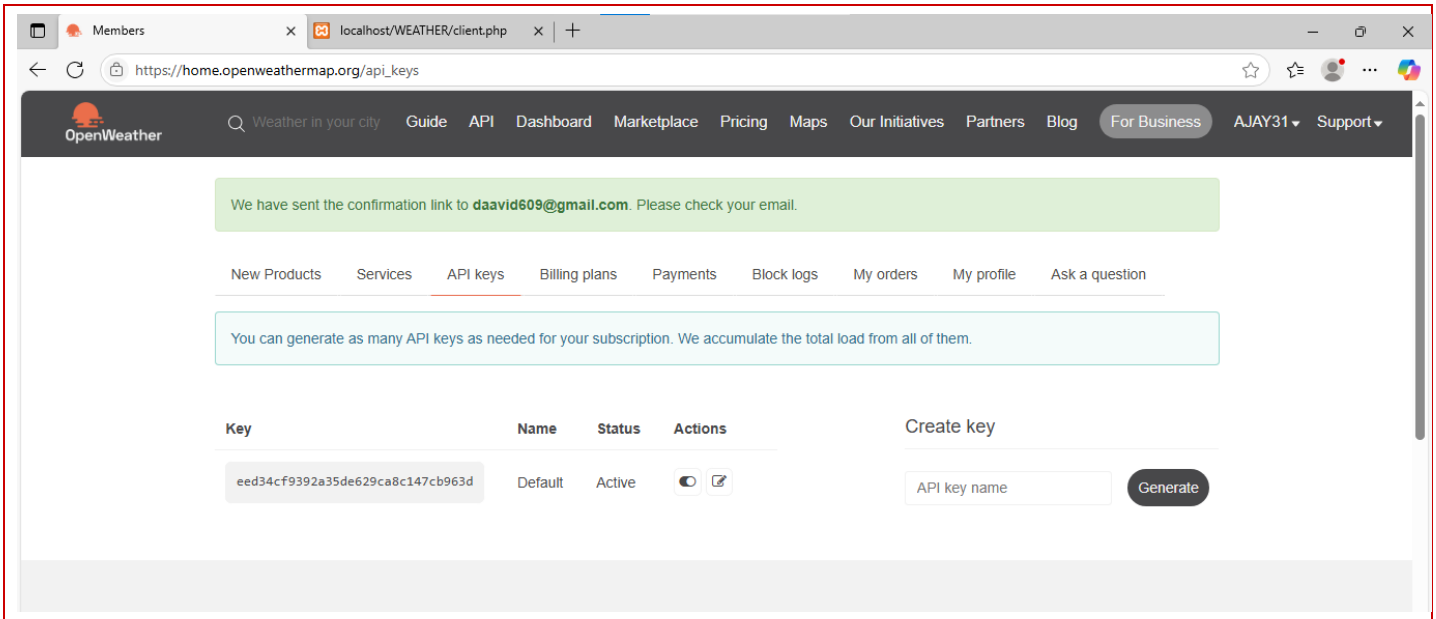
```
                if(!$city){
                        return "Error:City name required.";
                }


        $url="https://api.openweathermap.org/data/2.5/weather?q=".urlencode($city)
."&appid={$this->apikey}&units=metrics";

                $response=$this->curlGet($url);
                if(!$response){
                        return"Failed to fetch wether data from API.";
                }
                $data=json_decode($response,true);
                if(!isset($data['main']['temp'])){
                        return "City not found or API error.";
                }
                $temp=$data['main']['temp'];
                $desc=$data['weather'][0]['description']??'No Description';
                return "Current Temperature in {$city} is {$temp}°C with {$desc}";
        }

        private function curlGet($url){
        $ch=curl_init($url);
        curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
        curl_setopt($ch,CURLOPT_FAILONERROR,true);
        curl_setopt($ch,CURLOPT_CONNECTTIMEOUT,10);
        curl_setopt($ch,CURLOPT_TIMEOUT,15);
        $data=curl_exec($ch);
        curl_close($ch);
        return $data?:false;
        }
}

$options=['uri'=>'https://localhost/WEATHER'];
$server=new SoapServer(null,$options);
$server->setClass('WeatherService');
$server->handle();
?>
```

Client.php

```
<?php
try{
        $options = [
                "location" => "http://localhost/WEATHER/server.php",
                "uri" => "http://localhost/WEATHER",
                "trace" =>1,
        ];

        $client = new SoapClient(null, $options);
        $city = "Mumbai";
        $result = $client->getTemp($city);
        echo "Response from soap service: $result";
        echo nl2br(htmlspecialchars($result));
        }catch(SoapFault $e){
                echo "SOAP ERROR ".htmlspecialchars($e->getMessage());
        }
?>
```

D) OUTPUT:





A) AIM :

b) Create a simple HTML formthat let users input a city name and then fetches the weather from your php SOAP service via the php client.

B) DESCRIPTION:

AT first we visit OpenWeatherMap to register and obtain an API key. This key enables authenticated access to their live weather data. Using PHP, we build a SOAP-based server (Server.php) that defines a WeatherService class. It connects to OpenWeatherMap via a CURL request, retrieves temperature and weather description for a specified city, and returns it in a user-readable format. To prevent SOAP XML responses from being corrupted by unexpected PHP error output, we disable error reporting using ini_set('display_errors', 0) and error_reporting(0). On the frontend, Index.html allows users to input a city name, which is passed to Form_handler.php. This handler acts as a SOAP client, calls the server method getTemp(), and displays the result cleanly on a response page. The application thus provides a seamless experience for users to fetch and view current weather conditions from any city.

C) CODE:

Server.php

```
<?php
// Disable error output to avoid breaking SOAP XML response
ini_set('display_errors', 0);  // Fixed: 'display errors' → 'display_errors'
```

```
error_reporting(0);

class WeatherService {
    private $apikey = 'eed34cf9392a35de629ca8c147cb963d';

    public function getTemp($city) {
        $city = trim($city);
        if (!$city) {
            return "Error: City name required.";
        }

        $url = "https://api.openweathermap.org/data/2.5/weather?q=" . urlencode($city)
. "&appid={$this->apikey}&units=metric";

        $response = $this->curlGet($url);
        if (!$response) {
            return "Failed to fetch weather data from API.";  // Fixed: "wether" →
"weather"
        }

        $data = json_decode($response, true);
        if (!isset($data['main']['temp'])) {
            return "City not found or API error.";
        }

        $temp = $data['main']['temp'];
        $desc = $data['weather'][0]['description'] ?? 'No Description';
        return "Current Temperature in {$city} is {$temp}°C with {$desc}";
    }

    private function curlGet($url) {
        $ch = curl_init($url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_FAILONERROR, true);
        curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 10);
        curl_setopt($ch, CURLOPT_TIMEOUT, 15);
        $data = curl_exec($ch);
        curl_close($ch);
        return $data ?: false;
    }
```

Name of Subject Incharge: Mr. Rajesh Yadav

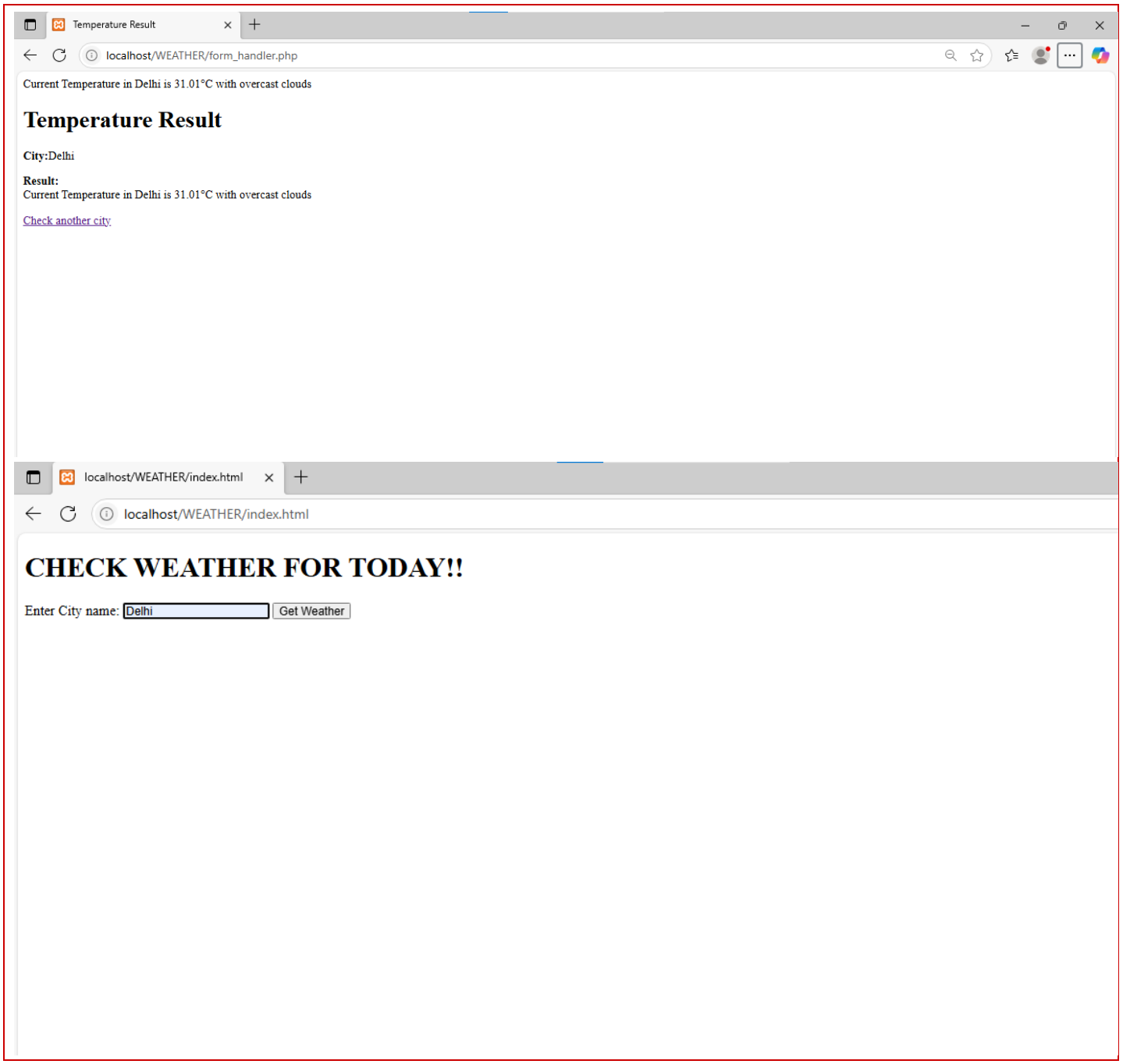
```
}

$options = ['uri' => 'https://localhost/WEATHER'];
$server = new SoapServer(null, $options);
$server->setClass('WeatherService');
$server->handle();
?>
Form_handler.php
<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $city = trim($_POST['city']);

    if (empty($city)) {
        echo "Please enter city name";
        exit;
    }

    $options = [
        "location" => "http://localhost/WEATHER/server.php",
        "uri" => "http://localhost/WEATHER",
        "trace" => 1,
    ];

    try {
        $client = new SoapClient(null, $options);
        $result = $client->getTemp($city);
    } catch (Exception $e) {
        $result = "Error: " . $e->getMessage();
    }
    echo $result;
} else {
    echo "Invalid request method";
}
?>
<!DOCTYPE html>
<html>
<head><title>Temperature Result</title></head>
<body>
<h1>Temperature Result</h1>
<p><strong>City:</strong><?= htmlspecialchars($city ?? '') ?></p>
```

```
<p><strong>Result:</strong><br><?= nl2br(htmlspecialchars($result)) ?></p>
<a href="index.html">Check another city</a>
</body>
</html>
```

Index.html

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>CHECK WEATHER FOR TODAY!!</h1>
<form method="POST" action="">
   <label for="city">Enter City name:</label>
   <input type="text" id="city" name="city">
   <button type="submit">Get Weather</button>
</form>
</body>
</html>
```

D) OUTPUT:



Name of Subject Incharge: Mr. Rajesh Yadav

SIES

Web Services with API
Practical No.1

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY G IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	16/07/2025	**Practical No**	3

A) AIM : Create a php SOAP service that provides currency conversion (convertCurrency(from , to ,amount)) and test it with a client script.

B) DESCRIPTION:

The CurrencyConverter application is a PHP-based system that facilitates currency conversion using a SOAP web service. It consists of two primary components: the server-side CurrencyConverter class and a client script. The class internally stores currency exchange rates with respect to USD and provides a public method named convertCurrency that takes three parameters—source currency (from), target currency (to), and the monetary amount to be converted. It ensures consistency by converting currency codes to uppercase and precision by treating the amount as a float. The conversion logic first transforms the source amount into its USD equivalent and then computes the value in the target currency, returning the final result rounded to two decimal places. On the client side, the script sets up a SOAP client to communicate with the locally hosted service, supplying the required parameters and displaying the converted result or handling errors gracefully if any occu

C) CODE:

Server.php

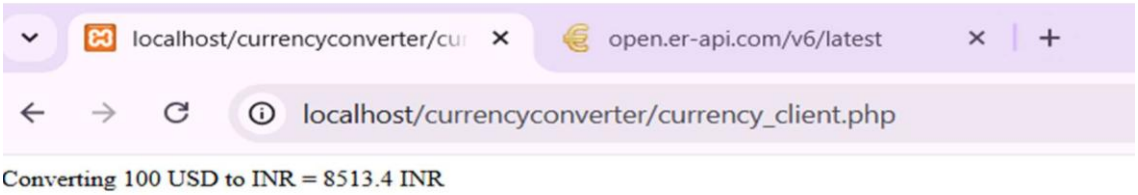
```
<?php
class   CurrencyConverter{      private
$rates=[
                "USD"=>1.0,
                "EUR"=>0.91,
                "INR"=>85.134,
```

Name of Subject Incharge: Mr. Rajesh Yadav

```
                "GBP"=>0.78,
                "JPY"=>139.0,
        ];
            public func on convertCurrency($from,$to,$amount){
                $from=strtoupper($from);
            $to=strtoupper($to);
                                    $amount=floatval($amount);
            if(!isset($this->rates[$from])||!isset($this->rates[$to])){
        return "Currency Code Not Supported.";
                        }
            $usdAmount=$amount/$this->rates[$from];
        $convertedAmount=$usdAmount*$this->rates[$to];                  return
round($convertedAmount,2);
            }
}

$op ons = ['uri'=>'h p://localhost/currencyconverter'];
$server = new SoapServer(null,$op ons);
$server->setClass('currencyconverter');
$server->handle();

Client.php
<?php
$op ons=[
        'loca on'=>'h p://localhost/currencyconverter/currency_server.php',          'uri'=>'h
p://localhost/currencyconverter'
];
```

```
try{
                                        $client=new SoapClient(null,$op ons);
                $from="USD";
                $to=”INR”;
                $amount=100;
                $result=$client->convertCurrency($from,$to,$amount);
                echo"Converting $amount $from to $to = $result$to”;
 }catch(Exception $e){
 echo”ERROR:”.$e->getMessage();
 }
 ?>
```

D) OUTPUT:



A) AIM :

Dynamic Currency Converter Soap Service with php.

B) DESCRIPTION:

Open Xamp go to config in which open php.ini

Press Ctrl+f and find Allow_url_open=On //change to on if off

Server.php

A class is declared with name CurrencyConverter in which a an API is used that takes the real me data and uses that to convert the currency

A public function is used to access the private apiKey and private rates A public function named convertCurrency with three parameter from, to and amount

from is used to as from which currency the amount should be converted and to is used as to which type of currency the amount should be converted amount is used to take the amount

$from = strtoupper($from);

$to = strtoupper($to);

 Converts both currency codes to uppercase to ensure consistency

$amount = floatval($amount);

Ensures the amount is a float for accurate calcula on if(!isset($this->rates[$from])||!isset($this->rates[$to])){

return "Currency Code Not Supported.";

}

Checks if both the source and target currencies exist in the $this->rates array. If either is missing, it returns an error message.

$usdAmount = $amount / $this->rates[$from]; $convertedAmount =
$usdAmount * $this->rates[$to];
First converts the source amount to a USD equivalent.
Then converts that USD amount into the target currency.
After that returns the final result rounded to 2 decimal places, which is standard for currency.

Client.php

This PHP script sets up a SOAP client to communicate with a currency conversion web service hosted locally. It specifies the source and target currencies (USD to INR) and the amount to convert (100). The convertCurrency method is called via the SOAP client, and the result is displayed. If there's an error during the process, the exception is caught and its message is shown.

C) CODE:

Server.php

```
<?php
class CurrencyConverter {
   private $apiUrl = "h ps://open.er-api.com/v6/latest/USD";     private
$rates = [];

   public func on __construct() {
      $json = @file_get_contents($this->apiUrl);
if ($json !== false) {
         $data = json_decode($json, true);
if ($data && isset($data['rates'])) {
            $this->rates = $data['rates'];
         }
      }
   }

   public func on convertCurrency($from, $to, $amount) {
      $from = strtoupper($from);
      $to = strtoupper($to);
```

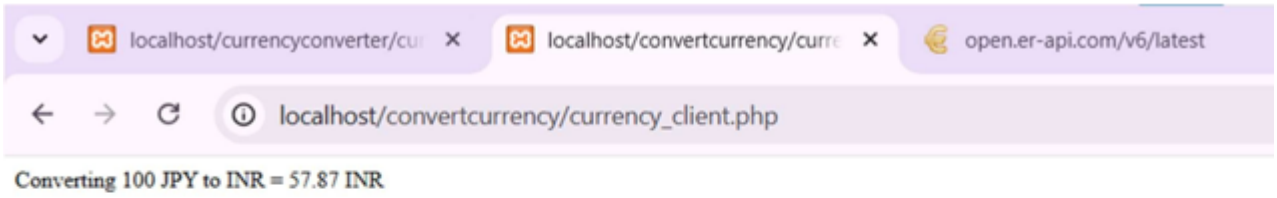
```
        $amount = floatval($amount);

        if (empty($this->rates)) {
            return "Error fetching exchange rates.";
        }

        if (!isset($this->rates[$from]) || !isset($this->rates[$to])) {
return "Currency Code Not Supported.";
        }

        $usdAmount = $amount / $this->rates[$from];
        $convertedAmount = $usdAmount * $this->rates[$to];
return round($convertedAmount, 2);
    }
}
$op ons = ['uri' => 'h p://localhost/convertcurrency'];
$server = new SoapServer(null, $op ons);
$server->setClass('CurrencyConverter');
$server->handle();
?>
```

Client.php

```
<?php
$op ons=[
                                                'loca on'=>'h
p://localhost/convertcurrency/currency_server.php',
                                    'uri'=>'h p://localhost/convertcurrency'
];
try{
                                    $client=new SoapClient(null,$op ons);
                        $from="JPY";
                $to="INR";
                        $amount=100;
        $result=$client->convertCurrency($from,$to,$amount);    echo
"Conver ng $amount $from to $to = $result $to";
```

```
}catch(Excep on $e){
                    echo "Error: ".$e->getMessage();
} ?>
```

D) OUTPUT:



SIES

Web Services with API
Practical No.4

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	28/07/2025	**Practical No**	4

A) AIM :

AIM: Write a PHP SOAP server and client where the server exposes a method to return factorial of a number.

DESCRIPTION: The server-side script defines a FactorialService class with a method named factorial() that computes the factorial of a given non-negative integer. It first validates the input: if the number is negative, it returns an error message; if the number is zero, it correctly returns 1. For other positive integers, it uses a for loop to compute the factorial iteratively. The SOAP server is initialized with a local URI and registers the FactorialService class, making its factorial method accessible to any SOAP client that connects to it.

The client-side script sets up a SoapClient that connects to the server via the location and uri parameters. It calls the factorial() method remotely with $n = 5 and prints the result. Basic error handling is included using a try-catch block to catch and display any SoapFault errors that occur during the call.

CODE:

Server.php

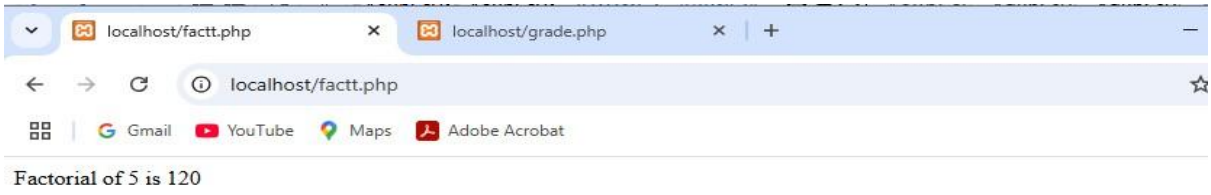
```
<?php
class FactorialService {
   public function factorial($n) {
      $n = intval($n);
if ($n < 0) {
return "Error: Number should  be positive";
      }
if ($n === 0) {
return 1;
      }
      $result = 1;
for ($i = 1; $i <= $n; $i++) {
```

```
            $result *= $i;
        }         return
$result;
    }
}
$options = ['uri' => 'http://localhost/'];
$server = new SoapServer(null, $options);
$server->setClass('FactorialService');
$server->handle();
?>
```

Client.php

```
<?php
$options = [
'location' => 'http://localhost/server2.php',
'uri' => 'http://localhost/',
'trace' => 1
];
$n=5;
try {
$client = new SoapClient(null, $options);
$result = $client->factorial($n);
echo "Factorial of", $n." is ". $result;
} catch (SoapFault $e) {
echo "SOAP Error: ", $e->getMessage();
}
?>
```

OUTPUT:



SIES

Web Services with API
Practical No.5

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	28/07/2025	**Practical No**	5

A) AIM :

Develop a SOAP webservice to return grade based on marks.Use NON-WSDL mode

DESCRIPTION :

The Server.php script defines a class called Gradeservice, which has a method getGrade() that takes a numerical input for marks, validates it, and returns a corresponding grade based on defined thresholds. Grades range from "A+" for scores above 90 to "F" for scores below 50, and the method also checks for invalid input outside the 0–100 range. The server is initiated using PHP's SoapServer class, exposing the getGrade method over a local URI. On the other hand, Client.php connects to the SOAP server using SoapClient, sends a request with a sample marks value (90 in this case), and retrieves the grade returned by the server. It includes basic error handling for SOAP faults**.**

CODE:

Server.php

```
<?php
class Gradeservice{
public function getGrade($marks){
$marks=floatval($marks); if($marks<0||$marks>100){
return "Invalid Marks";
}
if ($marks>=90) return "A+";
if ($marks>=80) return "A";
if ($marks>=70) return "B";
if ($marks>=60) return "C"; if
($marks>=50) return "D";
return "F";
}
}
```

Name of Subject Incharge: Mr. Rajesh Yadav

```
$options = ['uri' => 'http://localhost/'];
$server = new SoapServer(null, $options);
$server->setClass('Gradeservice');
$server->handle();
?>
```

Client.php

```
<?php
$options = [
'location' => 'http://localhost/gserver.php';
'uri' => 'http://localhost/',
'trace' => 1
];
try {
$client = new SoapClient(null, $options);
$marks=90;
$grade = $client->getGrade($marks);
echo "Grade for". $marks. "marks is". $grade;
} catch (SoapFault $e) {
echo "SOAP Error: ".$e->getMessage();
}
?>
```

OUTPUT:

SIES

Web Services with API
Practical No.6

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	28/07/2025	**Practical No**	6

A) AIM :

Create and consume timeserver webservice using node js

DESCRIPTION: This simple Node.js time server is built using Express and CORS to provide real-time date and time information to client applications. When a GET request is made to the /time endpoint, the server responds with the current local time, formatted as a readable string. It uses new Date().toLocaleString() to capture the moment the request is processed, ensuring the client receives an accurate timestamp. Designed to be lightweight and accessible, this server can serve as a backend utility for clock widgets, logging systems, or time-based notifications. With just a few lines of code, it demonstrates the core functionality of a RESTful API responding with dynamic data.

STEPS BEFORE WRITING CODE:

1)Check proper installed or not

```
C:\Users\admin>node -v
v18.16.1
```

2)Make directory for your project:

```
C:\Users\admin>mkdir timeserver-api
```

3)Initialize & install package :

Name of Subject Incharge: Mr. Rajesh Yadav

```
C:\Users\admin\timeserver-api>npm init -y
Wrote to C:\Users\admin\timeserver-api\package.json:

{
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^5.1.0"
  },
  "scripts": {
    "start": "node server.js"
  },
  "name": "timeserver-api",
  "version": "1.0.0",
  "main": "server.js",
  "devDependencies": {},
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}



C:\Users\admin\timeserver-api>
C:\Users\admin\timeserver-api>npm install express cors

up to date, audited 70 packages in 830ms
```

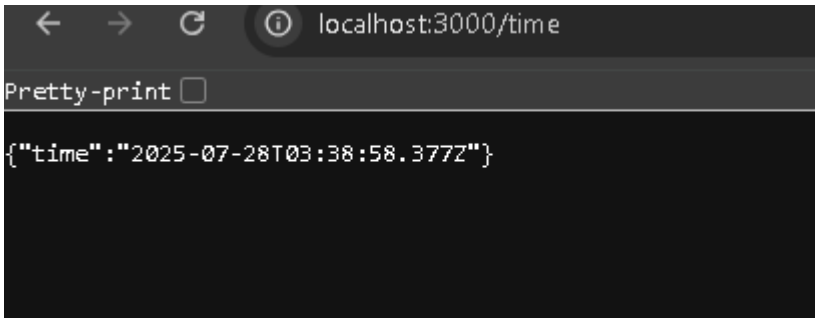
CODE:

```
const e=require('express');
const c=require('cors');
const app-express();
const PORT=3000;
app.use(cors());
app.get('/time'.(req.res)=>{
const currentTime = new Date().toLocaleString();
res.json({tim SarrentTime});
app.listen(PORT.)=>{
console.log('Time server running successfully at
http://localhost: ${PORT}');
})
```

OUTPUT:

Name of Subject Incharge: Mr. Rajesh Yadav

```
C:\Users\admin\timeserver-api>node server.js
Time server is running at http://localhost:3000/time
```



Name of Subject Incharge: Mr. Rajesh Yadav

SIES

Web Services with API
Practical No.6

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY GAUTAM IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	04/08/2025	**Practical No**	6

B) AIM :

PHP client program to consume your Node.js Time Server REST API.

DESCRIPTION:

The PHP script sends a request to a local Node.js server endpoint (/time) and, if the response is successful, decodes the returned JSON and displays the current time on a webpage. This approach allows dynamic content from a Node.js service to be embedded within a PHP-based frontend, offering a practical example of cross-platform interaction in web development.

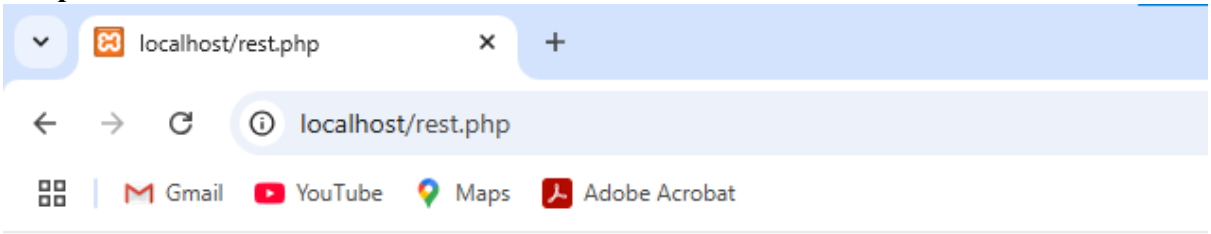
CODE

```
<?php
$apiUrl="http://localhost:3000/time";
$response=file_get_contents($apiUrl);
if($response!==false){
$data=json_decode($response,true);
echo "<h3>Current Time from Node.js API:</h3>";
echo "Time:".htmlspecialchars($data['time']);
}
else{
echo "Failed to connect to the API.";
}
?>
```

Node.js server

```
Node.js command prompt - node  server.js
Your environment has been set up for using Node.js 22.16.0 (x64) and npm.

C:\Users\admin>npm init -y
Wrote to C:\Users\admin\package.json:

{
  "name": "admin",
  "version": "1.0.0",
  "main": ".mongorc.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}



C:\Users\admin>node server.js
node:internal/modules/cjs/loader:1404
  throw err;
  ^

Error: Cannot find module 'C:\Users\admin\server.js'
    at Function._resolveFilename (node:internal/modules/cjs/loader:1401:15)
    at defaultResolveImpl (node:internal/modules/cjs/loader:1057:19)
    at resolveForCJSWithHooks (node:internal/modules/cjs/loader:1062:22)
    at Function._load (node:internal/modules/cjs/loader:1211:37)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
    at wrapModuleLoad (node:internal/modules/cjs/loader:235:24)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:171:5)
    at node:internal/main/run_main_module:36:49 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}

Node.js v22.16.0

C:\Users\admin>cd timeserver-api

C:\Users\admin\timeserver-api>node server.js
Time Server running at http://localhost:3000
```

Output:-



Current Time from Node.js API:

Time:8/4/2025, 7:43:32 AM

Name of Subject Incharge: Mr. Rajesh Yadav

SIES

Web Services with API
Practical No.7

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY GAUTAM IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	04/08/2025	**Practical No**	7

A) **AIM :** Create a REST API using php that accepts a username via URL and returns a welcome message in json format.

DESCRIPTION:

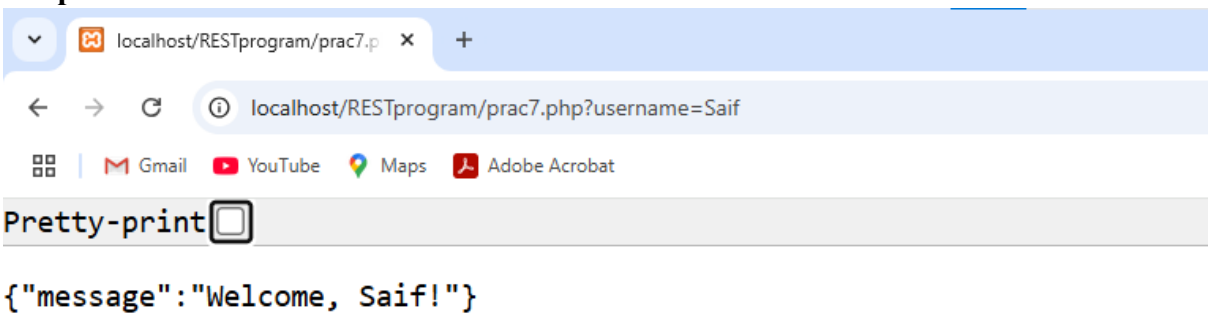
This PHP script is a simple API endpoint that returns a personalized welcome message in JSON format. It only accepts GET requests and expects a username parameter in the query string. If provided, it responds with a greeting; if missing or if the request method isn't GET, it returns an appropriate error message. The use of htmlspecialchars ensures basic security by preventing potential injection in the output.

Code:-

```
<?php
header('Content-Type: application/json');
if($_SERVER['REQUEST_METHOD']==='GET'){
if (isset($_GET['username']) && !empty($_GET['username'])){
$username=htmlspecialchars($_GET['username']);
$response=[
'message'=>"Welcome, $username!"
];
}
else{
$response=[
'error'=>'Username parameter is missing.'
];
}
}
else{
http_response_code(405);
$response=[
```

```
'error'=>'Only GET requests are allowed.'
];
}
echo json_encode($response);
?>
```

Output:-



B)Aim: Create a REST API using php that accepts a username via URL and returns a welcome message in json format using POSTMAN.

DESCRIPTION:

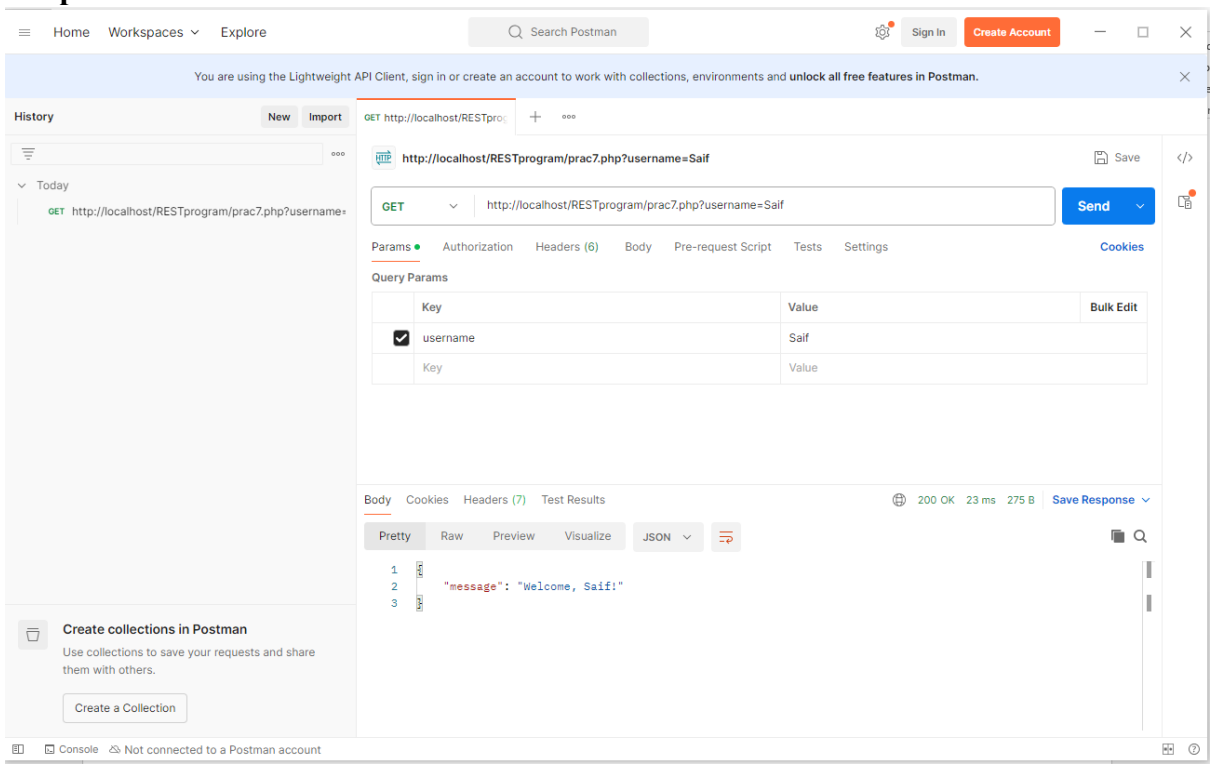
REST API that responds with a JSON message, intended for use with tools like Postman. When accessed via a GET request with a username parameter in the URL (e.g., ?username=Alice), it returns a personalized welcome message like {"message":"Welcome, Alice!"}. If the username is missing or the request method is not GET, it responds with a clear error message. The use of htmlspecialchars ensures the username input is safely handled before being echoed back.

CODE:

```
<?php
header('Content-Type: application/json');
if($_SERVER['REQUEST_METHOD']==='GET'){
if (isset($_GET['username']) && !empty($_GET['username'])){
$username=htmlspecialchars($_GET['username']);
$response=[
'message'=>"Welcome, $username!"
];
```

```
}
else{
$response=[
'error'=>'Username parameter is missing.'
];
}
}
else{
http_response_code(405);
$response=[
'error'=>'Only GET requests are allowed.'
];
}
echo json_encode($response);
?>
```

Output:-



Name of Subject Incharge: Mr. Rajesh Yadav

SIES

Web Services with API
Practical No.8

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY GAUTAM IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	04/08/2025	**Practical No**	8

A) AIM :

Build a simple web service in php that provides:

- NSE index
- BSE index
- Gold rate

Fetched from a MYSQL database and a php client(web page) that consumes and displays them like a live stock trading dashboard

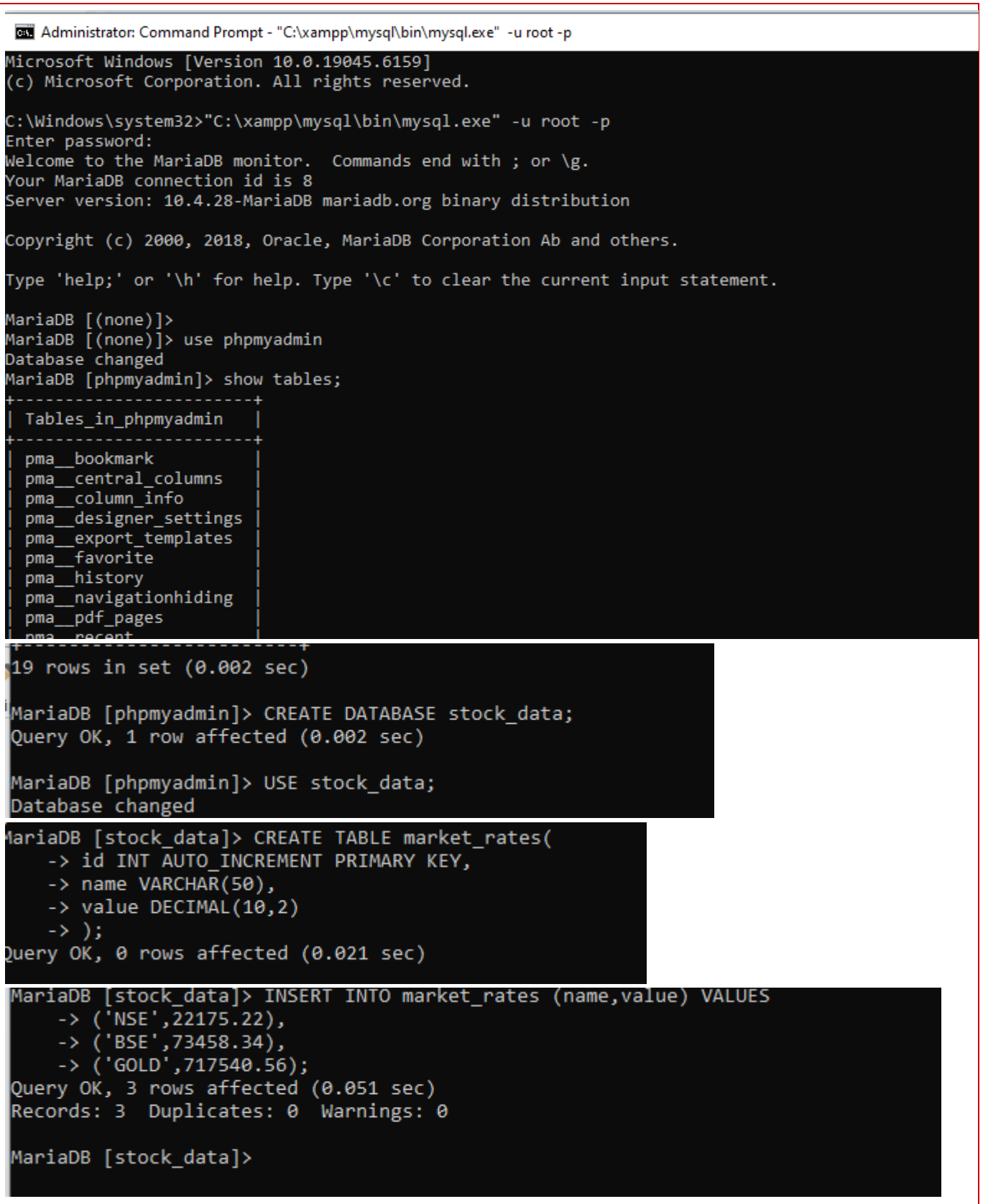
DESCRIPTION:

This is a simple PHP-based stock market dashboard. The api.php script connects to a MySQL database and returns current market rates in JSON format, while index.php fetches this data and displays live values for NSE, BSE, and Gold on a web page. Perfect for lightweight real-time updates.

STEPS BEFORE WRITING CODE IN NOTEPAD:

1) OPEN COMMAND PANEL BY RUNNING AS ADIMINTRATIVE .
2) Enter location of MySQL bin and -u root -p command .
3) For password,double click enter(press enter 2 times)
4) Now we can create or use any database and table as per need.

Name of Subject Incharge: Mr. Rajesh Yadav



```
Administrator: Command Prompt - "C:\xampp\mysql\bin\mysql.exe"  -u root -p
Microsoft Windows [Version 10.0.19045.6159]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>"C:\xampp\mysql\bin\mysql.exe" -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
MariaDB [(none)]> use phpmyadmin
Database changed
MariaDB [phpmyadmin]> show tables;
+------------------------+
| Tables_in_phpmyadmin   |
+------------------------+
| pma__bookmark          |
| pma__central_columns   |
| pma__column_info       |
| pma__designer_settings |
| pma__export_templates  |
| pma__favorite          |
| pma__history           |
| pma__navigationhiding  |
| pma__pdf_pages         |
| pma__recent            |
+------------------------+
19 rows in set (0.002 sec)

MariaDB [phpmyadmin]> CREATE DATABASE stock_data;
Query OK, 1 row affected (0.002 sec)

MariaDB [phpmyadmin]> USE stock_data;
Database changed
MariaDB [stock_data]> CREATE TABLE market_rates(
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(50),
    -> value DECIMAL(10,2)
    -> );
Query OK, 0 rows affected (0.021 sec)

MariaDB [stock_data]> INSERT INTO market_rates (name,value) VALUES
    -> ('NSE',22175.22),
    -> ('BSE',73458.34),
    -> ('GOLD',717540.56);
Query OK, 3 rows affected (0.051 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [stock_data]>
```

CODE:

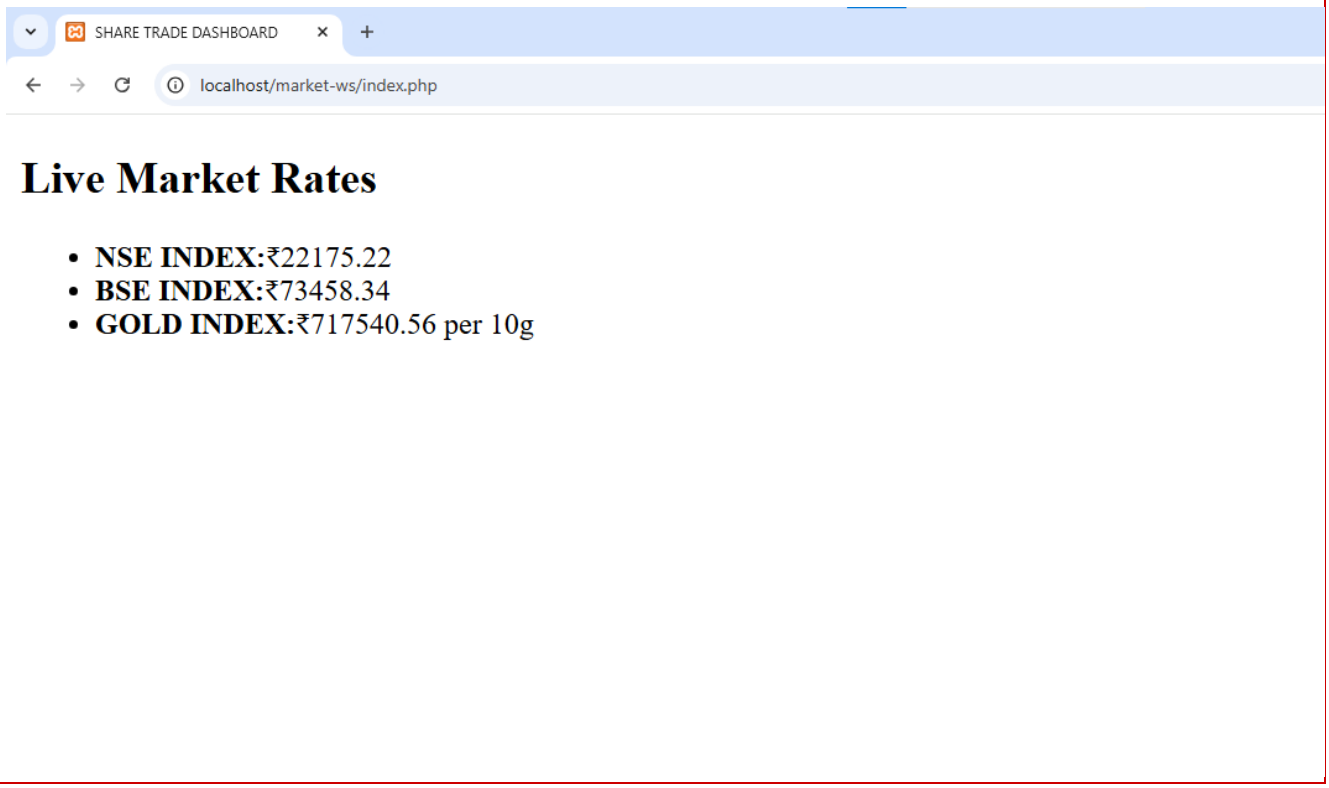
Index.php

```
<?php
$response=file_get_contents("http://localhost/market-ws/api.php");
$data=json_decode($response,true);
?>
<!DOCTYPE html>
<html>
<head><title>SHARE TRADE DASHBOARD</title></head>
<body>
<h2>Live Market Rates</h2>
<ul>
<li><strong>NSE INDEX:</strong>₹<?php echo $data['NSE'];?></li>
<li><strong>BSE INDEX:</strong>₹<?php echo $data['BSE'];?></li>
<li><strong>GOLD INDEX:</strong>₹<?php echo $data['GOLD'];?> per 10g</li>
</ul>
</body>
</html>
```

Api.php

```
<?php
header("Content-Type:application/json");
header("Access-Control-Allow-Origin:*");
$host="localhost";
$user="root";
$pass="";
$db="stock_data";
$conn=new mysqli($host,$user,$pass,$db);
if($conn->connect_error){
        echo json_encode(["ERROR"=>"DATABASE CONNECTION FAILED"]);
        exit;
}
$sql="SELECT name,value FROM market_rates";
$result=$conn->query($sql);
$rates=[];
while($row=$result->fetch_assoc()){
        $rates[$row['name']]=$row['value'];
}
echo json_encode($rates);
$conn->close();
?>
```

OUTPUT:



Name of Subject Incharge: Mr. Rajesh Yadav

SIES

Web Services with API
Practical No.9

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY G IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	11/08/2025	**Practical No**	9

A) AIM :

PHP _based WEB SERVICE for UGC that:
Accepts a college name as input.
Returns the NAAC rating for the college

B) DESCRIPTION:

```
C:\Windows\system32>

C:\Windows\system32>"C:\xampp\mysql\bin\mysql.exe" -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use phpmyadmin
Database changed
MariaDB [phpmyadmin]> create database mydbs;
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [phpmyadmin]> CREATE TABLE clg_rating (    clg_name VARCHAR(100),    naac_rating VARCHAR(20));
Query OK, 0 rows affected (0.009 sec)
```

```
MariaDB [phpmyadmin]> insert into clg_rating(clg_name,naac_rating) VALUES
    -> ('Delhi University','A+'),
    -> ('Mumbai University','A');
Query OK, 2 rows affected (0.042 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [phpmyadmin]>
```

It enables users to retrieve the NAAC rating of a college by entering its name through a simple web interface. The system consists of two main components: ugc_api.php, **which**

Name of Subject Incharge: Mr. Rajesh Yadav

acts as the backend API, and index.php, which serves as the frontend interface. When a user submits a college name via the form in index.php, the input is sent to ugc_api.php, which connects to a MySQL database (mydbs) and queries the clg_rating table for the corresponding NAAC rating. The result is returned in JSON format and displayed on the webpage.

C) CODE:

Ugc_api.php

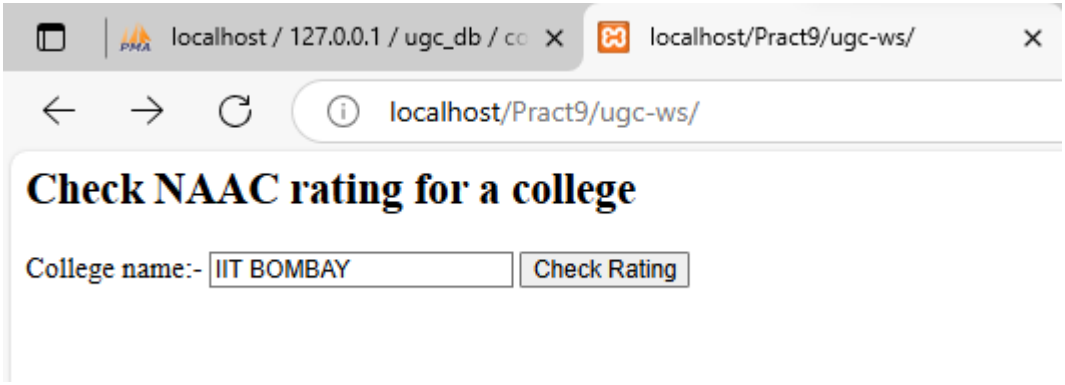
```
<?php
header("Content-Type:application/json");
header("Access-Control-Allow-Origin:*");
$host='localhost';
$user='root';
$pass='';
$db='mydbs';
$conn=new mysqli($host,$user,$pass,$db);
if($conn->connect_error){
echo json_encode(["Error"=>"No college name provided."]);
exit;
}
$stmt=$conn->prepare("SELECT naac_rating FROM clg_rating WHERE clg_name=?");
$stmt->bind_param("s",$college);
$stmt->execute();
$stmt->bind_result($rating);
if($stmt->fetch()){
        echo json_encode(["college"=>$college,"rating"=>$rating]);
}else{
echo json_encode(["college"=>$college,"rating"=>"NOT  FOUND"]);
}
$stmt->close();
$conn->close();
?>
```

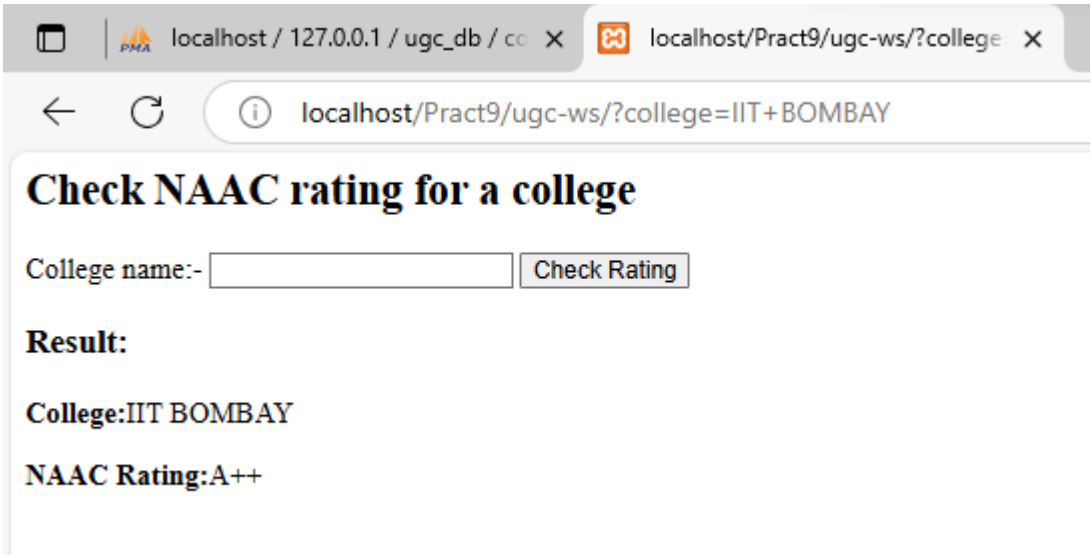
Index.php

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<h2>Check NAAC Rating</h2>
<form method='GET'>
  <label>College name:</label>
  <input type='text' name='college' required>
  <input type='submit' value='Check Rating'>
```

```
</form>

<?php
if (isset($_GET['college'])) {
   $clg = urlencode($_GET['college']);
   $api_url = "http://localhost/hari/ugc_api.php?college=" . $clg;
   $response = file_get_contents($api_url);
   $data = json_decode($response, true);

   echo "<h3>RESULT:</h3>";
   echo "<p><strong>College:</strong> " . htmlspecialchars($data['college']) . "</p>";
   echo "<p><strong>NAAC RATING:</strong> " . htmlspecialchars($data['rating']) . "</p>";
}
?>
</body>
</html>
```

D) OUTPUT:





Name of Subject Incharge: Mr. Rajesh Yadav

SIES

Web Services with API
Practical No.10

DEPARTMENT OF COMPUTER SCIENCE

Name of the Student	AJAY G IRLE	**Roll Number**	TCS2526020
Class	T.Y.B.Sc(Computer Science)	**Batch**	1
Date	11/08/2025	**Practical No**	10

A) AIM :

Image upload WebServices

B) DESCRIPTION:

A simple image upload system using HTML and PHP. The index.html file provides a user-friendly interface with a form that allows users to select and upload an image file. Upon submission, the form sends the image to upload.php, which handles the server-side processing. The PHP script checks the file extension to ensure it is one of the allowed image formats—JPG, JPEG, PNG, or GIF. If valid, the image is moved to the upload/ directory on the server, and a link to the uploaded image is displayed. If the upload fails or the file type is not permitted, appropriate error messages are shown. This setup is ideal for learning basic file handling and validation in PHP.

C) CODE:

Index.html

```
<html>
<body>
<h2>Upload Image</h2>
<form action="upload.php" method="post" enctype="multipart/form-data">
<input type="file" name="image" accept="image/*" required>
<input type="submit" value="Upload">
</form></body>
</html>
```

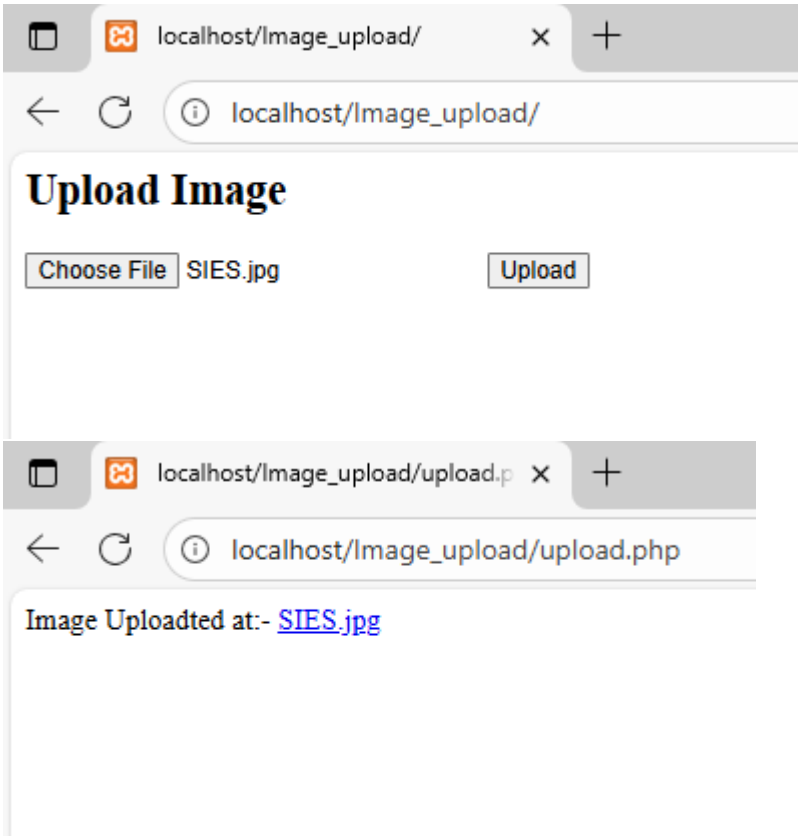
upload.php

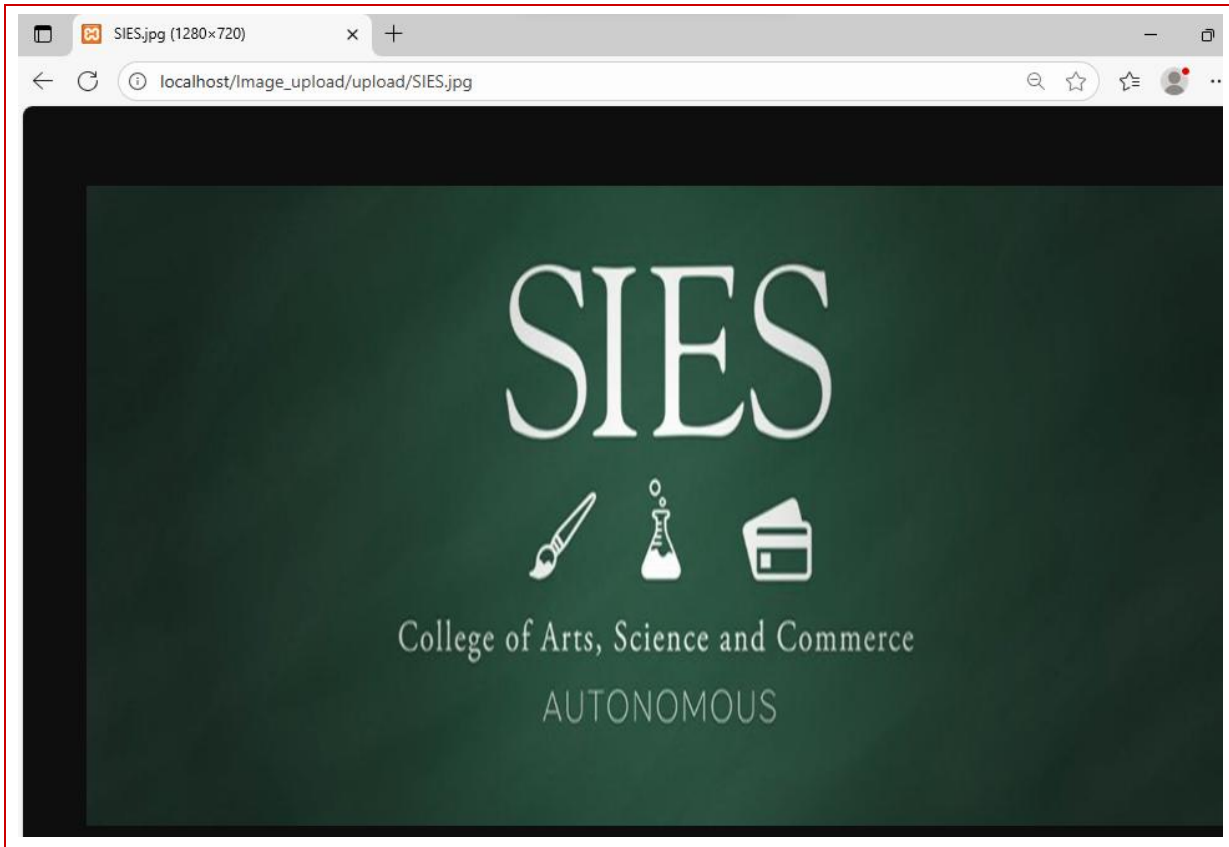
```
<?php
$target_dir="upload/";
$target_file=$target_dir.basename($_FILES["image"]["name"]);
$imageFileType=strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
$allowed=array("jpg","jpeg","png","gif");
```

Name of Subject Incharge: Mr. Rajesh Yadav

```
if(in_array($imageFileType,$allowed)){
        if (move_uploaded_file($_FILES["image"]["tmp_name"],$target_file)){
                echo "Image Uploadted at:- <a href='$target_file'>".basename($_FILES["image"]["name"])."</a>";
        }
        else{
                echo "Error uploading file";
        }

        }
        else{
        echo "Only JPG , PNG , JPEG , GIF files allowed";}
?>
```

D) OUTPUT:





Name of Subject Incharge: Mr. Rajesh Yadav