MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

Department of Information and Communication Technology

Lab Report

Lab report No: 05

Lab report on: Implementation of third-party API (email validation, weather, etc.) by JavaScript and Postman Software.

Course title: Internet and Web Programming Lab

Course Code: ICT-4108

Submitted By -	**Submitted To-**
Md. Shahidul Islam ID : IT-21024 4th Year 1st Semester Session : 2020-2021 Dept. of ICT, MBSTU	Md. Anowar Kabir Lecturer Dept. of ICT,MBSTU

Date of Performance: Date of Submission: 12.07.2025

Lab Report No: 05

Lab report Title: Implementation of third-party API (email validation, weather, etc.) by JavaScript and Postman Software.

Objective:

To understand and implement third-party APIs (such as email validation and weather information retrieval) using JavaScript and test them using Postman.

Tools and Technologies Used:

- Programming Language: JavaScript (Node.js or client-side)
- API Testing Tool: Postman
- APIs Used:
- Abstract API – Email Validation
- OpenWeatherMap API
- Other Tools: Internet connection, Text editor (VS Code), Browser (for client-side JS)

Theory:

Third-party APIs allow developers to access functionalities or data provided by external services. APIs like email validation and weather APIs help improve user experience and reliability by verifying user input and displaying real-time data.

Procedure:

1. Email Validation API Implementation:

- API Used: Abstract Email Validation API
- Endpoint: https://emailvalidation.abstractapi.com/v1/?api_key=YOUR_API_KEY&email=test@example.com

Sample JavaScript Code:

```
const email = "test@example.com";

const apiKey = "MY_API_KEY";

const url =
https://emailvalidation.abstractapi.com/v1/?api_key=${apiKey}&email=${email};

fetch(url) .then(response => response.json()) .then(data => { console.log("Is Valid
Email:", data.is_valid_format.value);

console.log("Deliverability:", data.deliverability); }) .catch(error =>
console.error("Error:", error));
```

Test in Postman:

- ❖ Method: GET
- ❖ URL: Replace email and API key in the above URL
- ❖ Observe JSON response for is_valid_format, deliverability, etc.

2. Weather API Implementation:

- ✓ API Used: OpenWeatherMap
- ✓ Endpoint: https://api.openweathermap.org/data/2.5/weather?q=Dhaka&appid=YOUR_API_KEY&units=metric

Sample JavaScript Code:

```
const city = "Dhaka";

const apiKey = "YOUR_API_KEY";

 const url =
https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units
=metric;

fetch(url) .then(response => response.json()) .then(data =>
{ console.log(Temperature in ${city}: ${data.main.temp}°C);
```

```
 console.log(Weather: ${data.weather[0].description}); }) .catch(error =>
console.error("Error:", error));
```

Test in Postman:

- ❖ Method: GET
- ❖ URL: Use the above URL with your API key
- ❖ Observe response JSON for temperature and weather description

Result:

- ❖ Successfully fetched and displayed:
- ❖ Email validity and deliverability status
- ❖ Current weather information for a given city
- ❖ Verified responses in Postman and JavaScript console

Discussion:

- ♦ Email validation ensures user-entered emails are correctly formatted and possibly reachable.
- ♦ Weather API provides real-time environmental data that can be useful for travel, planning, and display in apps.
- ♦ APIs require keys; some are free with limited usage.
- ♦ API rate limits and CORS issues should be considered when using APIs in browser-based JavaScript.

Conclusion:

We successfully implemented and tested third-party APIs using JavaScript and Postman. This experiment demonstrated the practical use of external APIs for building feature-rich, data-driven web applications.