|  |
| --- |
| **ID – 2022HT66033**  **Name – Raj Patil**  **Subject – Service Oriented Computing**  **Aim - To design and implement a service-based application for any application of your choice** |

**Theory & Implementation:**

**Postman:**

Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.

While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API.

In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python. Postman is based on a wide range of extremely user-friendly power tools. For more than 8 million users, Postman has become a tool of convenience.

Following are the reasons why Postman is used:

1. **Accessibility-** One can use it anywhere after installing Postman into the device by simply logging in to the account.
2. **Use Collections**-Postman allows users to build collections for their API-calls. Every set can create multiple requests and subfolders. It will help to organize the test suites. **3. Test development-** To test checkpoints, verification of successful HTTP response status shall be added to every API- calls.

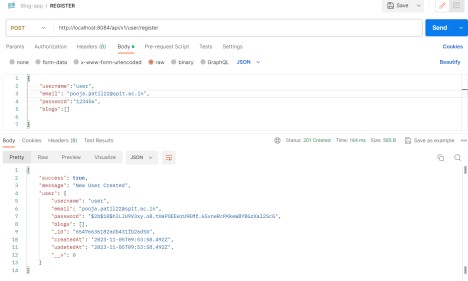
**4. Automation Testing**-Tests can be performed in several repetitions or iterations by using the Collection Runner or Newman, which saves time for repeated tests. **5. Creating Environments-** The design of multiple environments results in less replication of tests as one can use the same collection but for a different setting. 6. Debugging- To effectively debug the tests, the postman console helps to track what data is being retrieved.

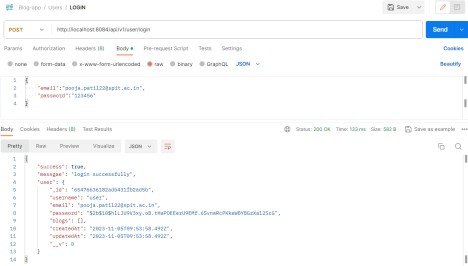
**7. Collaboration**- You can import or export collections and environments to enhance the sharing of files. You may also use a direct connection to share the collections. **8. Continuous integration**-It can support continuous integration. In the context of Postman, a "collection payload" typically refers to the data or information that you include within a Postman collection. A Postman collection is a container for organizing and grouping a set of API requests, tests, and related information. This "payload" within a collection can include various components such as:

* Requests: The primary payload within a Postman collection is the set of API requests.Each request includes details such as the HTTP method (e.g., GET, POST, PUT, DELETE)
* Environment Variables:Environment variables allow you to store and manage dynamic values that can be reused across multiple requests within the collection. ● Tests and Scripts:Collections can include test scripts written in JavaScript.est scripts are used to automate the validation of API responses and perform various actions based on the response data
* Documentation:Add documentation to a Postman collection to provide information about the API, its endpoints, request examples, and any other relevant details ● Folder Structure:Collections can be organized into folders to further structure and categorize the contained requests and related information.
* Global variables:Global variables are typically used for storing values that are relevant to the entire collection.
* Request Examples:Within each request, you can include multiple examples of the request, each with its own payload. This is useful for testing different scenarios or data variations for a specific API endpoint.

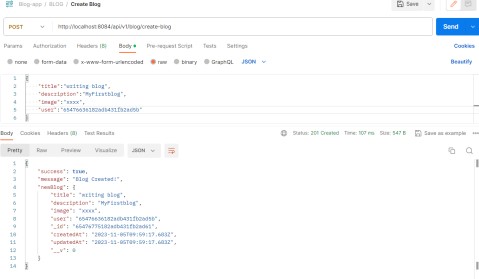
**Implementation**

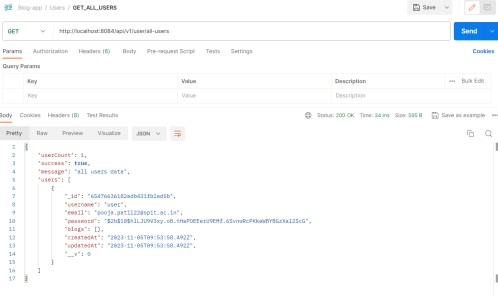
**Post Method Registration:**



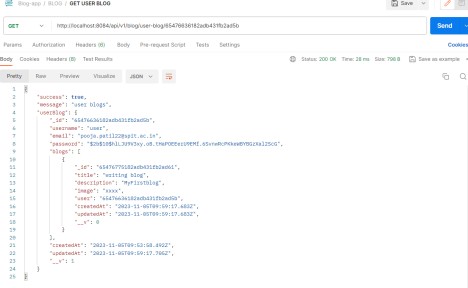
**Login:-**

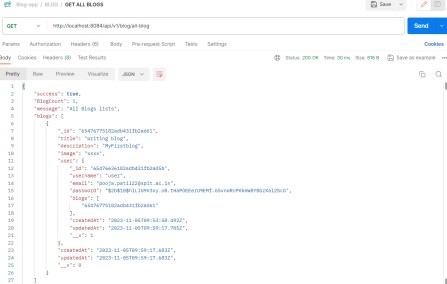
**Create Blog**



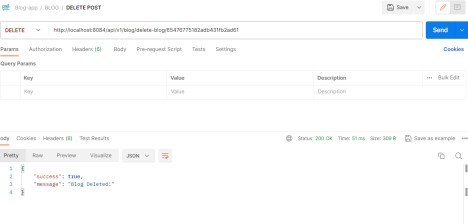
**Get Method: Get All Users**

**Get User Blog**

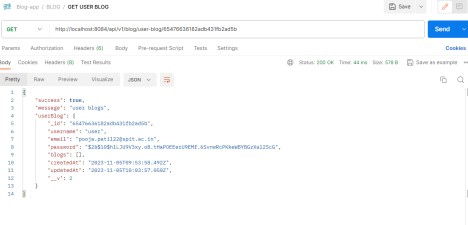


**Get All Blogs**

**Delete Method: Delete Blog**

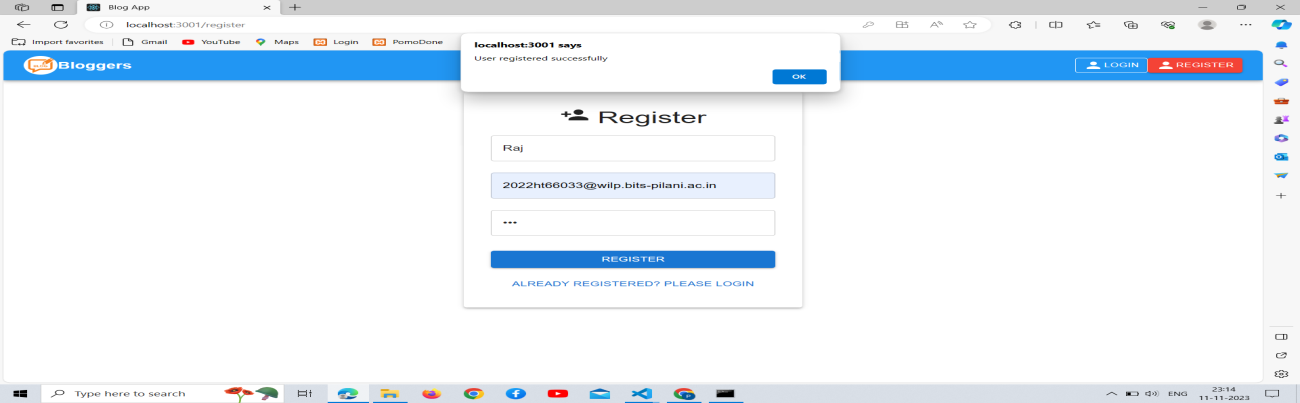


**Update visible in Postman Output**

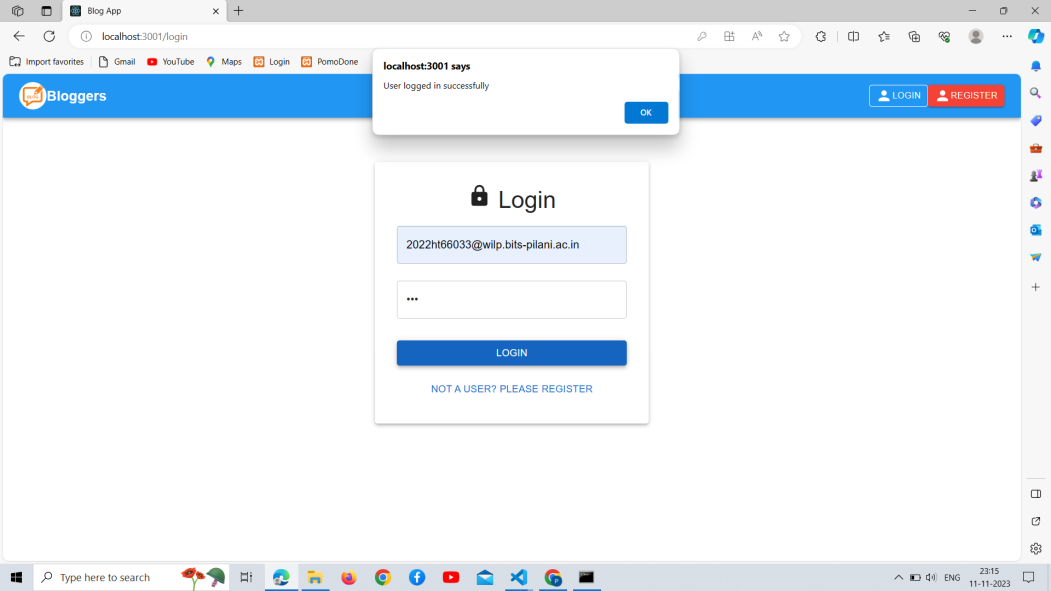


**Website UI:**

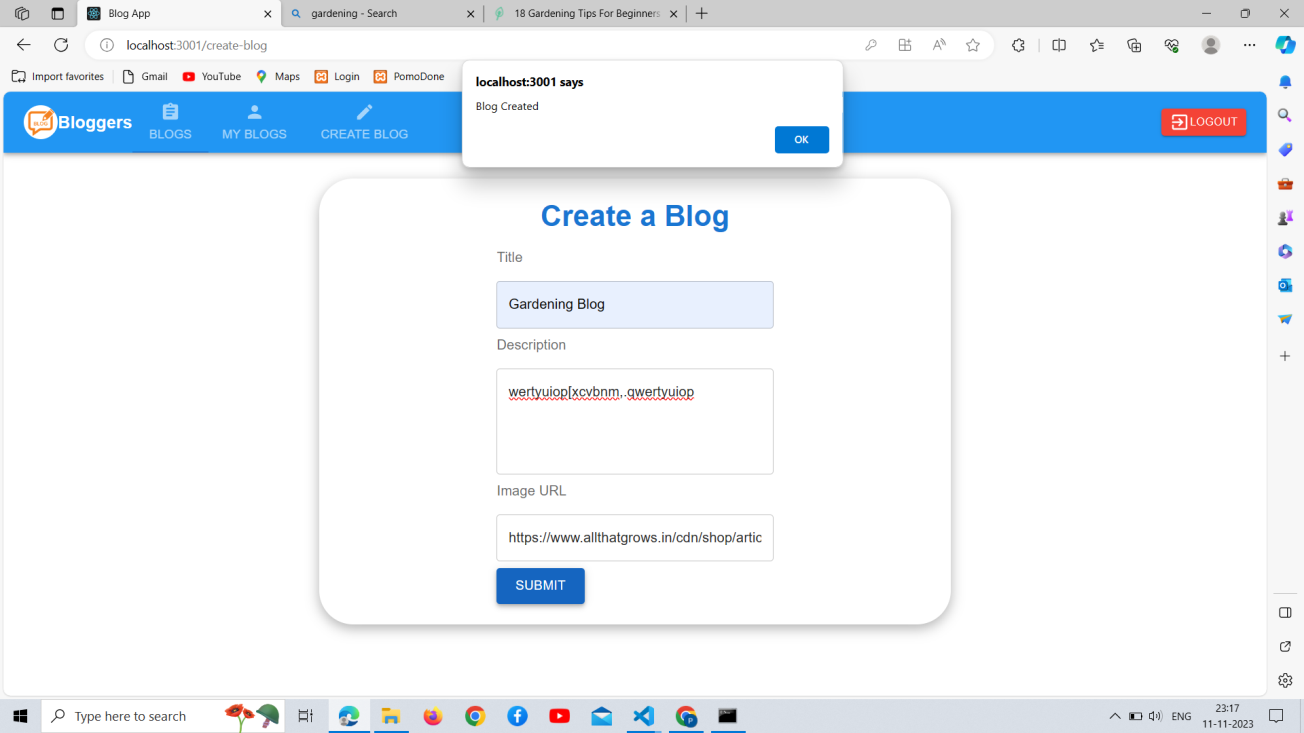
**Login Page:**

****

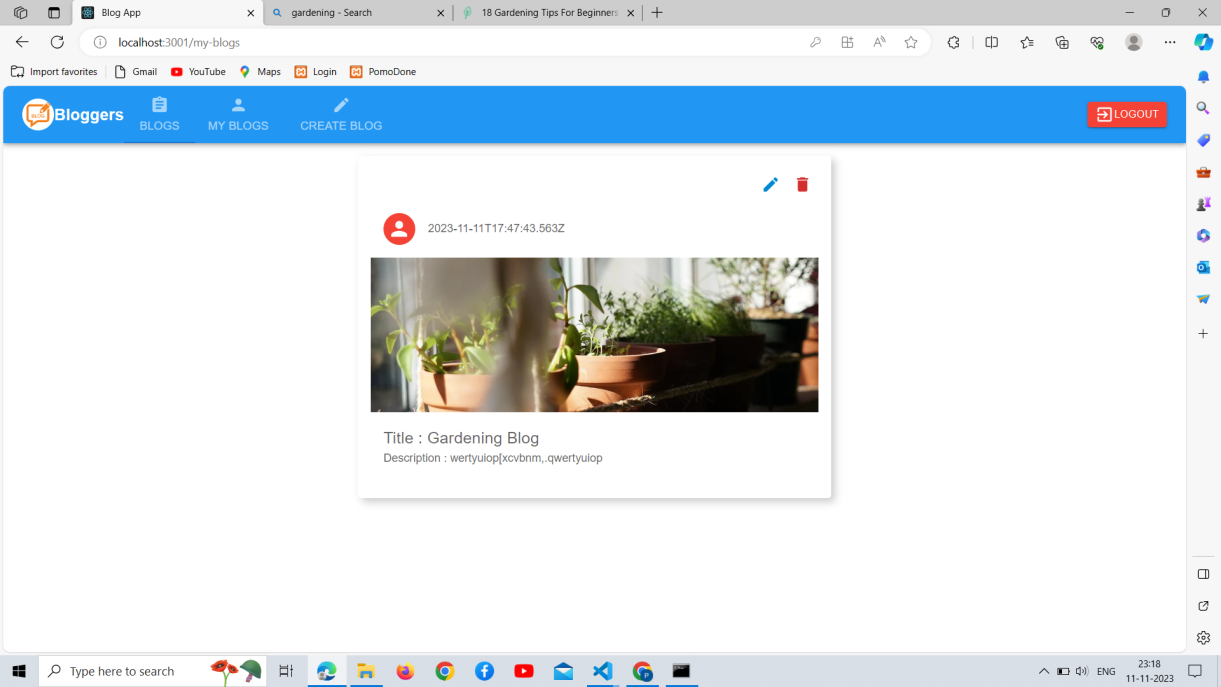
**Login Succeesful:**

****

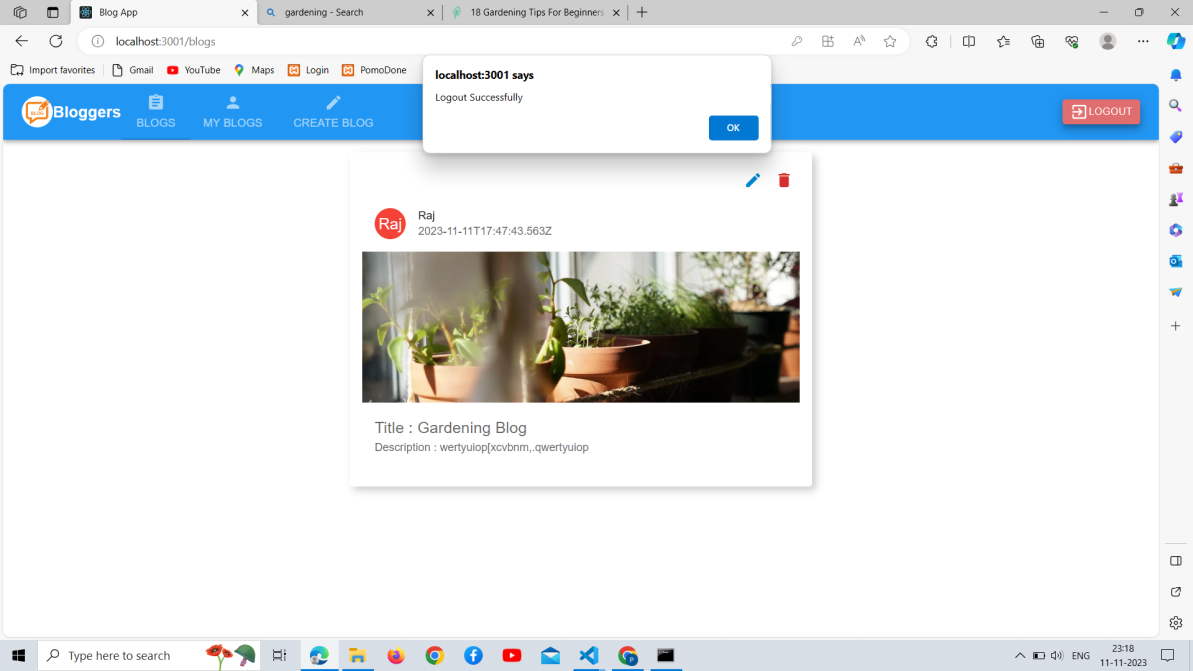
**Blog Created:**

****

**Blog Page:**

****

**Logout Page (Success Message):**

****

**Conclusion:** We have successfully implemented and tested three HTTP

methods (GET, POST and DELETE) in our Node.js application and used Postman to verify their functionality