

EXPERIMENT NO. 3

Name of Student	Anuprita Mhapankar
Class Roll No	28
D.O.P.	20/2/2024
D.O.S.	27/2/2024
Sign and Grade	

AIM: To develop a basic Flask application with multiple routes and demonstrate the handling of GET and POST requests.

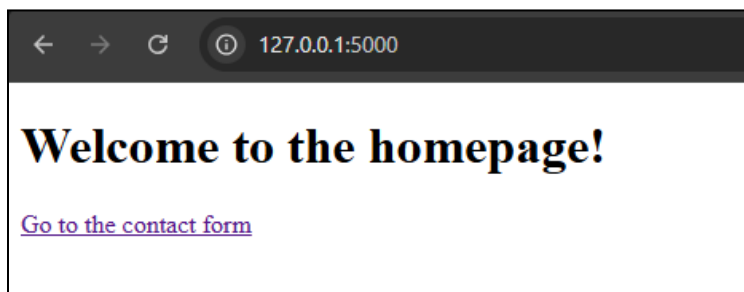
OVERVIEW OF TASKS PERFORMED:

The experiment involves designing a Flask web application with multiple routes. A **homepage (/)** displays a welcome message and supports dynamic personalization using a query parameter. A **contact page (/contact)** contains a form for users to enter their name and email. Upon submission via the **POST** method, the data is processed and displayed on a **thank-you page (/thank_you)**. The application demonstrates both **GET** and **POST** request handling, enabling personalized messages and form data submission.

GITHUB LINK: https://github.com/Anuprita2022-26/WebX_Exp3

OUTPUT

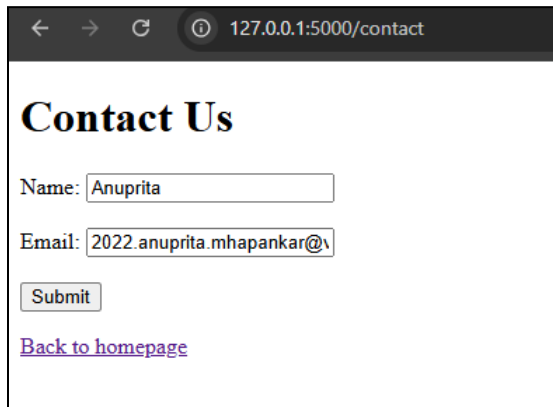
a) Homepage (/)



This screenshot displays the output of the homepage when accessed without any query parameters. The default welcome message, "**Welcome to the homepage!**", is shown, along with a hyperlink labeled "**Go to the contact form**" that redirects users to the contact page. This demonstrates the basic functionality

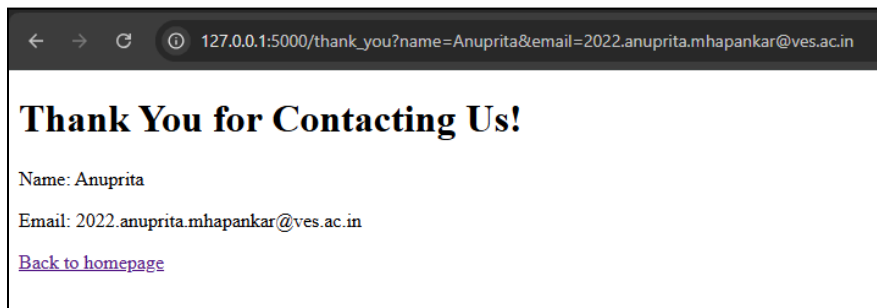
of the homepage and how it serves static content in the absence of user-specific input.

b) Contact Page (/contact)

A screenshot of a web browser showing a contact form. The browser's address bar displays '127.0.0.1:5000/contact'. The page has a title 'Contact Us'. Below the title, there are two input fields: 'Name:' with the value 'Anuprita' and 'Email:' with the value '2022.anuprita.mhapankar@v'. A 'Submit' button is located below the email field. At the bottom of the form, there is a link labeled 'Back to homepage'.

This screenshot shows the contact form page where users can enter their **name** and **email**. The form includes labeled input fields and a submit button. When the user enters their details and submits the form, the data is processed via the **POST** method and redirected to the **thank-you page**. This highlights Flask's ability to handle user input and form submission securely.

c) After Form Submission (/thank_you)

A screenshot of a web browser showing a 'Thank You' message. The browser's address bar displays '127.0.0.1:5000/thank_you?name=Anuprita&email=2022.anuprita.mhapankar@ves.ac.in'. The page has a title 'Thank You for Contacting Us!'. Below the title, it displays the submitted information: 'Name: Anuprita' and 'Email: 2022.anuprita.mhapankar@ves.ac.in'. At the bottom, there is a link labeled 'Back to homepage'.

This screenshot represents the output after successfully submitting the contact form. The **thank-you page** displays a message, "**Thank You for Contacting Us!**", along with the user's submitted **name and email**. Additionally, a "**Back to homepage**" link is provided for navigation. This demonstrates how Flask processes form data, passes it via the URL query string, and renders it dynamically on the response page.

CONCLUSION

A Flask web application was developed with multiple routes to handle **GET** and **POST** requests. It included a **homepage (/)** with a dynamic welcome message, a **contact page (/contact)** for form submission, and a **thank-you page (/thank_you)** to display submitted data. The experiment demonstrated Flask's routing, request handling, form submission, and template rendering capabilities.