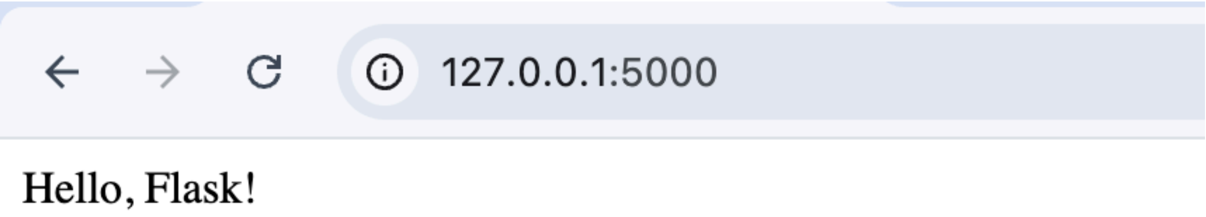
Weekly Assignment 27-Oct-24 : Python with Flask

1. Hello Flask Website

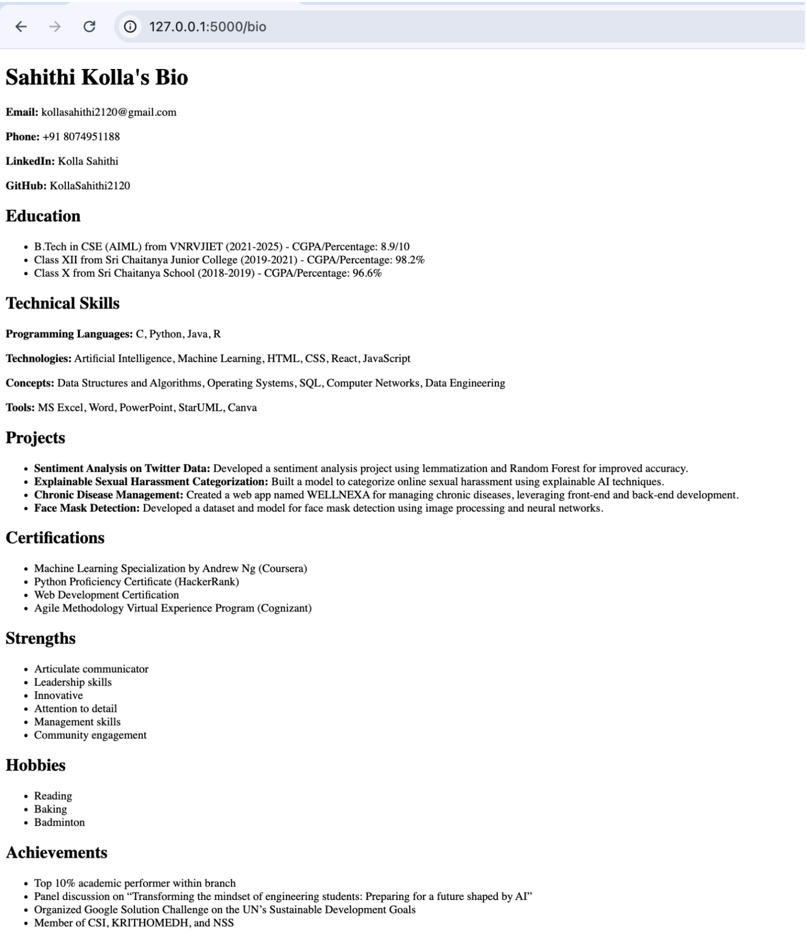
- **Task**: Create a simple "Hello, World!" Flask application.
- **Requirements**: Make a single route (/) that displays “Hello, Flask!” on a web page.
- **Hint**: Start by setting up a basic Flask app with a single route. Use app.route() to set the URL path.



Hello, Flask!

2. Personal Bio Page

- **Task**: Design a simple personal bio page.
- **Requirements**: Add a route (/bio) with basic information like name, age, and hobbies displayed in HTML.
- **Hint**: Use Flask's render_template function and create a basic HTML file to display personal details.



Sahithi Kolla's Bio

Email: kollasahithi2120@gmail.com

Phone: +91 8074951188

LinkedIn: Kolla Sahithi

GitHub: KollaSahithi2120

Education

- B.Tech in CSE (AIML) from VNRVJIET (2021-2025) - CGPA/Percentage: 8.9/10
- Class XII from Sri Chaitanya Junior College (2019-2021) - CGPA/Percentage: 98.2%
- Class X from Sri Chaitanya School (2018-2019) - CGPA/Percentage: 96.6%

Technical Skills

Programming Languages: C, Python, Java, R

Technologies: Artificial Intelligence, Machine Learning, HTML, CSS, React, JavaScript

Concepts: Data Structures and Algorithms, Operating Systems, SQL, Computer Networks, Data Engineering

Tools: MS Excel, Word, PowerPoint, StarUML, Canva

Projects

- **Sentiment Analysis on Twitter Data:** Developed a sentiment analysis project using lemmatization and Random Forest for improved accuracy.
- **Explainable Sexual Harassment Categorization:** Built a model to categorize online sexual harassment using explainable AI techniques.
- **Chronic Disease Management:** Created a web app named WELLNEXA for managing chronic diseases, leveraging front-end and back-end development.
- **Face Mask Detection:** Developed a dataset and model for face mask detection using image processing and neural networks.

Certifications

- Machine Learning Specialization by Andrew Ng (Coursera)
- Python Proficiency Certificate (HackerRank)
- Web Development Certification
- Agile Methodology Virtual Experience Program (Cognizant)

Strengths

- Articulate communicator
- Leadership skills
- Innovative
- Attention to detail
- Management skills
- Community engagement

Hobbies

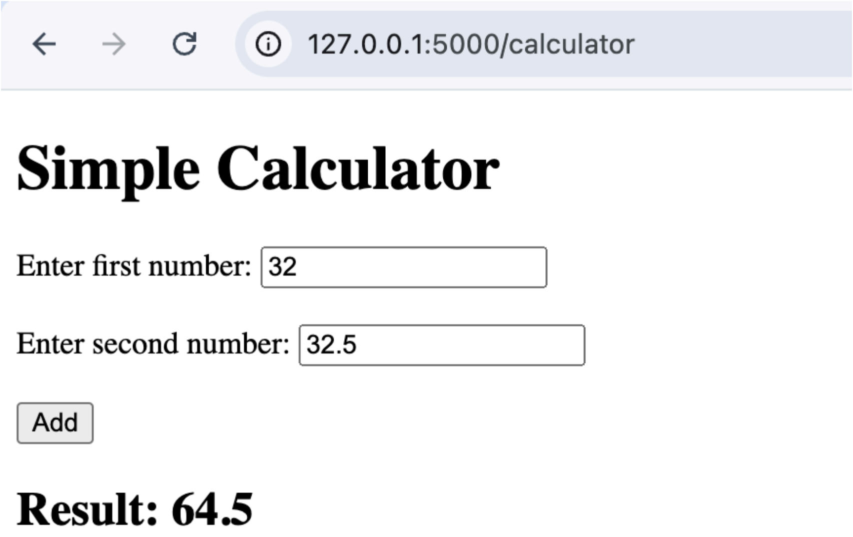
- Reading
- Baking
- Badminton

Achievements

- Top 10% academic performer within branch
- Panel discussion on “Transforming the mindset of engineering students: Preparing for a future shaped by AI”
- Organized Google Solution Challenge on the UN’s Sustainable Development Goals
- Member of CSI, KRITHOMEDH, and NSS

3. Calculator App

- **Task**: Build a simple calculator that can add two numbers.
- **Requirements**: Create a form where users enter two numbers. Display the result on submission.
- **Hint**: Use HTML forms and handle data in Flask using the request.form method to get the inputs.



127.0.0.1:5000/calculator

Simple Calculator

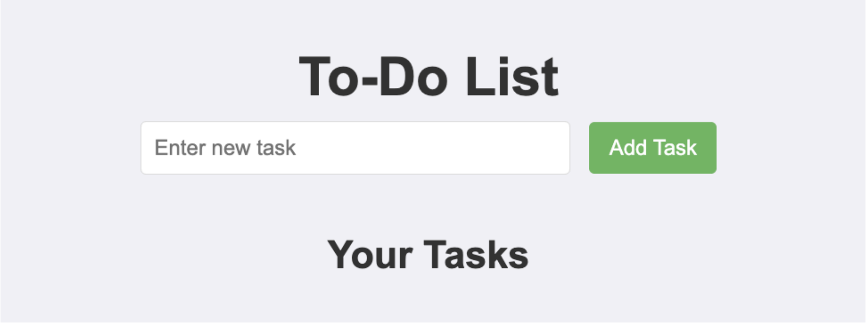
Enter first number: 32

Enter second number: 32.5

Result: 64.5

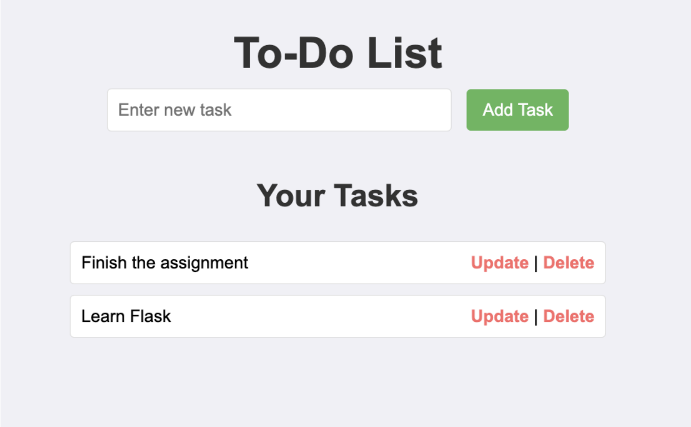
4. Mini To-Do List

- **Task**: Create a basic to-do list web app where users can add tasks.
- **Requirements**: Implement a form to add tasks and display the tasks on the same page.
- **Hint**: Use a list to store tasks temporarily and a POST method to add new tasks to the list.



To-Do List

Your Tasks

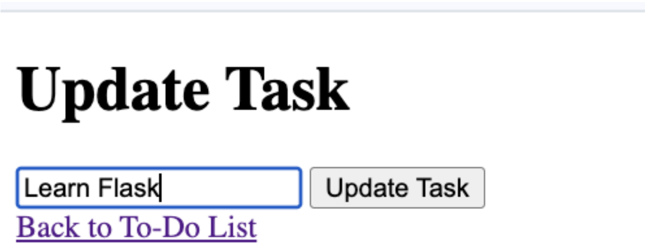


To-Do List

Your Tasks

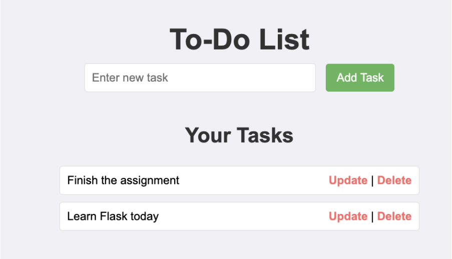
Finish the assignment Update | Delete

Learn Flask Update | Delete



Update Task

Learn Flask



To-Do List

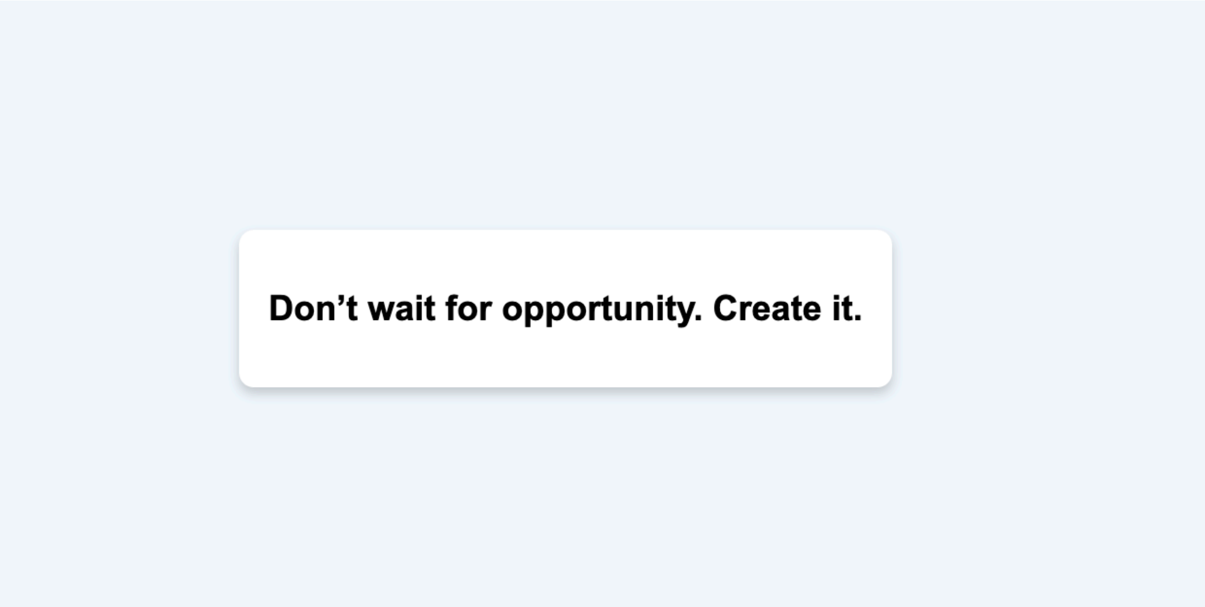
Your Tasks

Finish the assignment Update | Delete

Learn Flask today Update | Delete

5. Random Quote Generator

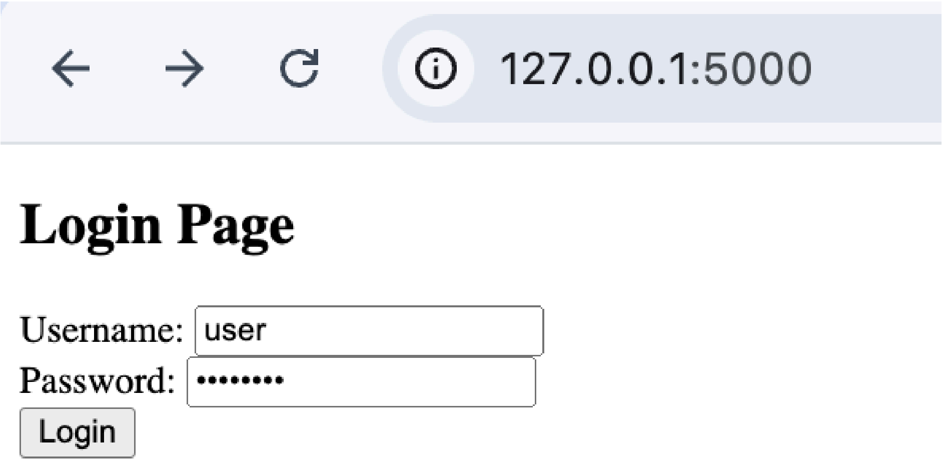
- **Task**: Create an app that displays a random motivational quote from a predefined list each time the page is refreshed.
- **Requirements**: Display one random quote from a list of quotes every time the user visits the /quote page.
- **Hint**: Use Python's random.choice() to select a quote from a list and display it using HTML.



Don’t wait for opportunity. Create it.

6. Simple Login Page

- **Task**: Build a basic login form with username and password fields.
- **Requirements**: Display a welcome message if the username is “user” and the password is “password.”
- **Hint**: Use POST requests and if statements to check login credentials.

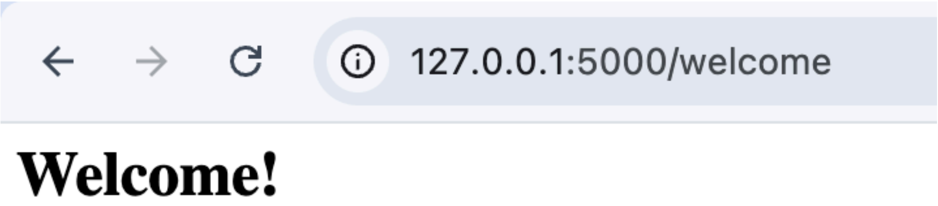


127.0.0.1:5000

Login Page

Username: user

Password: ••••••••

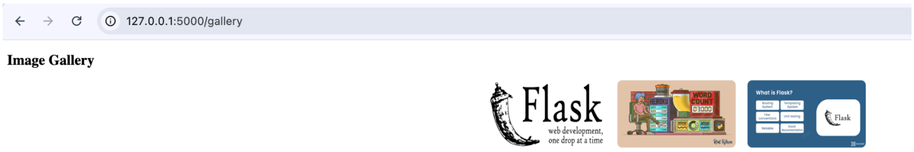


127.0.0.1:5000/welcome

Welcome!

7. Image Gallery

- **Task**: Create a simple gallery page that displays three static images.
- **Requirements**: Display three images side by side on a page.
- **Hint**: Use HTML <img> tags in your HTML template and store the images in a /static folder in the project.

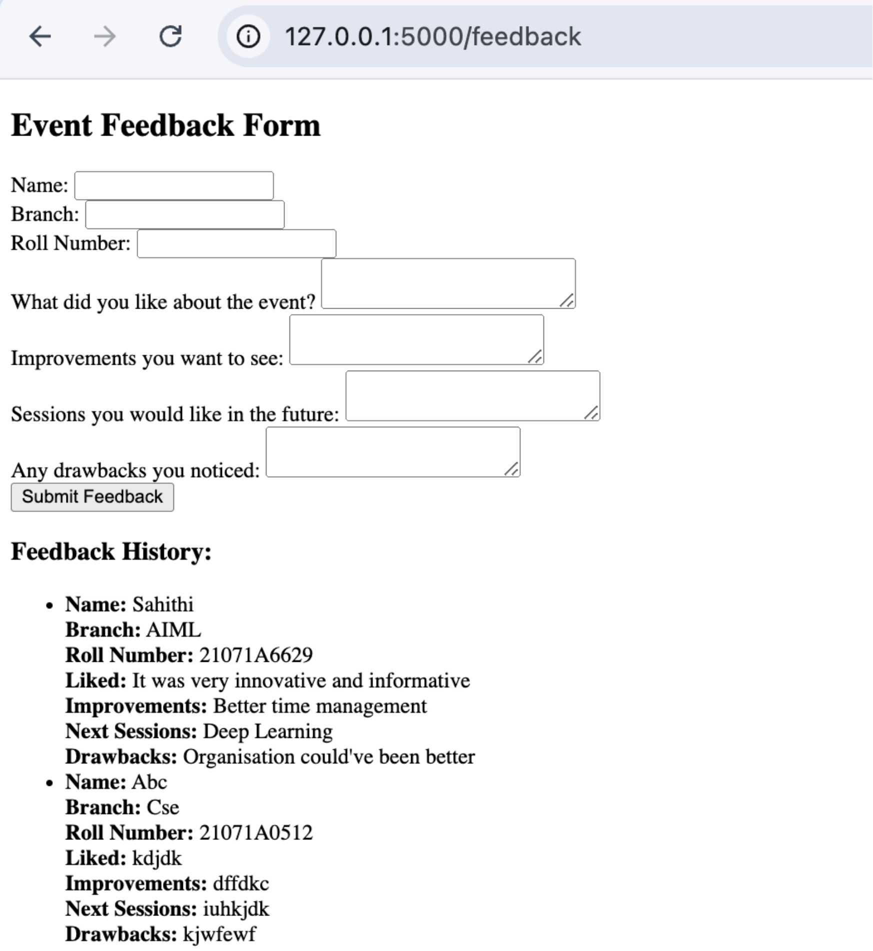


127.0.0.1:5000/gallery

Image Gallery

8. Feedback Form

- **Task**: Make a feedback form that saves users' names and feedback temporarily.
- **Requirements**: Save the submitted feedback to a list and display the list on the same page.
- **Hint**: Use Python lists or dictionaries to store each feedback entry, and display feedback history at the bottom of the page.



Event Feedback Form

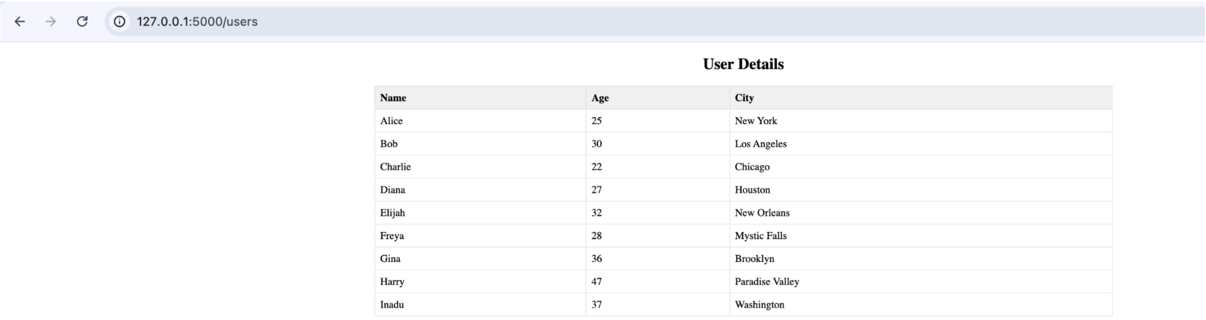
Name:
Branch:
Roll Number:
What did you like about the event?
Improvements you want to see:
Sessions you would like in the future:
Any drawbacks you noticed:

Feedback History:

- **Name:** Sahithi
  Branch: AIML
  Roll Number: 21071A6629
  Liked: It was very innovative and informative
  Improvements: Better time management
  Next Sessions: Deep Learning
  Drawbacks: Organisation could've been better
- **Name:** Abc
  Branch: Cse
  Roll Number: 21071A0512
  Liked: kdjdk
  Improvements: dffdkc
  Next Sessions: iuhkjdk
  Drawbacks: kjwfewf

9. Basic Data Table with Jinja

- **Task**: Display a table of users and their details (name, age, city).
- **Requirements**: Use a predefined list of dictionaries and render it in an HTML table.
- **Hint**: Use Jinja templating with for loops to iterate over the list and display the data in a table format.

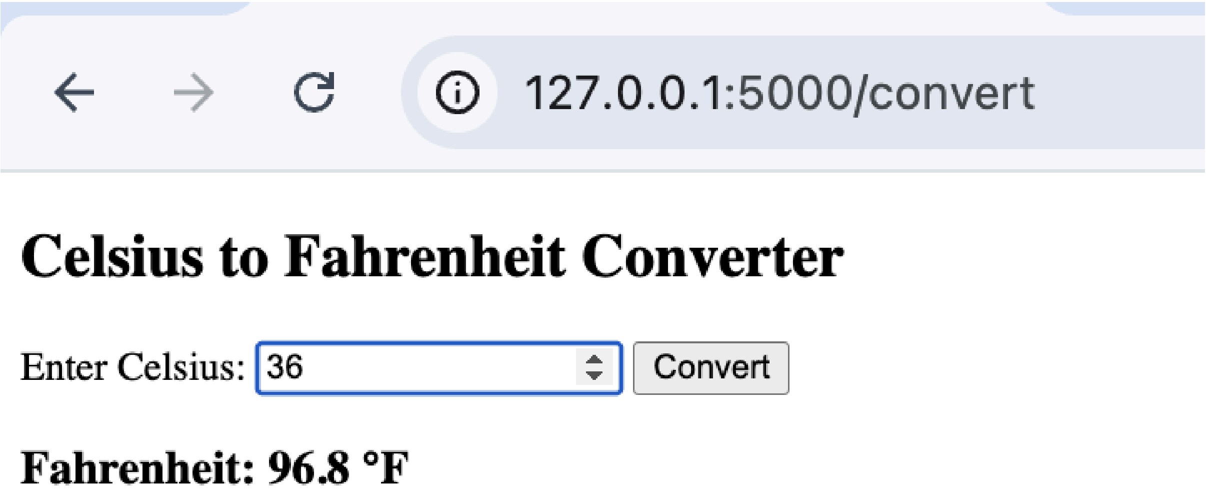


User Details

Name	Age	City
Alice	25	New York
Bob	30	Los Angeles
Charlie	22	Chicago
Diana	27	Houston
Elijah	32	New Orleans
Freya	28	Mystic Falls
Gina	36	Brooklyn
Harry	47	Paradise Valley
Inadu	37	Washington

10. Temperature Converter

- **Task**: Create a simple temperature converter that converts Celsius to Fahrenheit.
- **Requirements**: Allow users to enter a Celsius value and display the converted Fahrenheit value.
- **Hint**: Use forms for input, request.form to get the Celsius value, and simple arithmetic in Python to calculate the Fahrenheit temperature.



Celsius to Fahrenheit Converter

Enter Celsius: 36

Fahrenheit: 96.8 °F