

INTEREST CALCULATOR

1. What Are We Building?

We are building an Interest Calculator Web Application.

This application helps users calculate:

- Simple Interest (SI)
- Compound Interest (CI)

The user enters the principal amount, rate of interest, and time period. The application then calculates and displays the interest value instantly.

2. Technologies Used

Technology	What it is used for
Python	Main programming language
Flask	Helps create the website
HTML	Creates structure of webpage
CSS	Makes the page look beautiful
Jinja2	Display dynamic results

3. Before Starting (Setup)

Step 3.1: Install Python

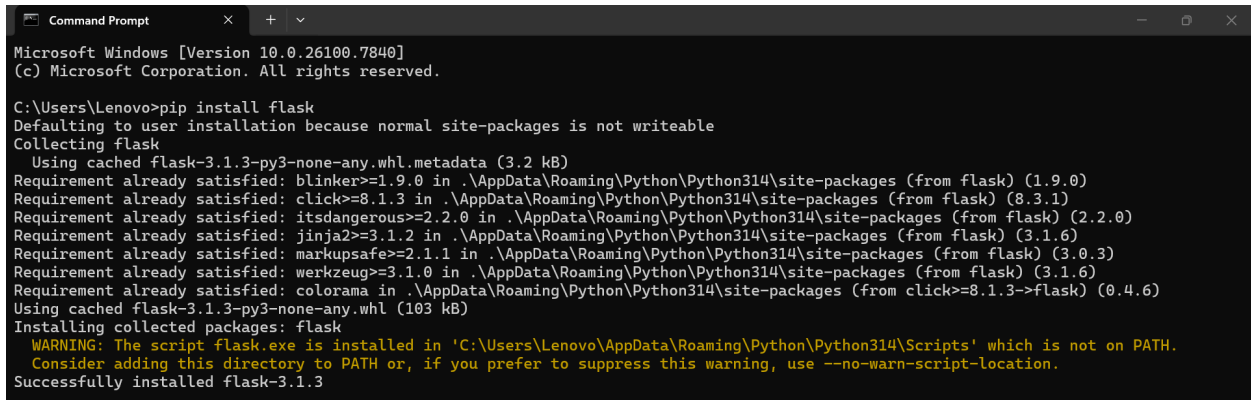
Download from: <https://www.python.org>

While installing → check “Add Python to PATH”

Step 3.2: Install Flask

Open terminal / command prompt and type:

```
pip install flask
```



```
Microsoft Windows [Version 10.0.26100.7840]
(c) Microsoft Corporation. All rights reserved.

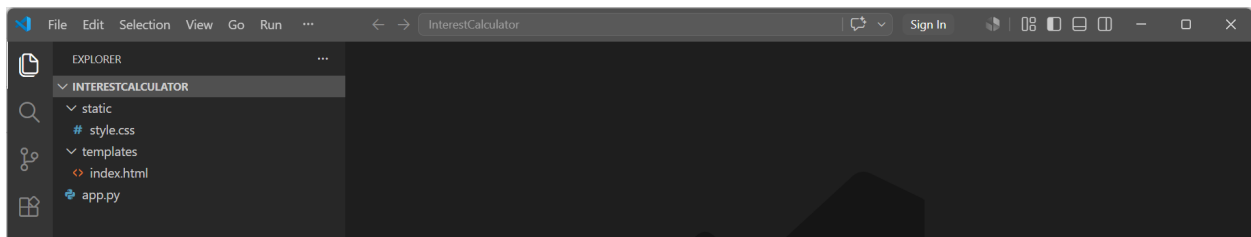
C:\Users\Lenovo>pip install flask
Defaulting to user installation because normal site-packages is not writeable
Collecting flask
  Using cached flask-3.1.3-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: blinker>=1.9.0 in .\AppData\Roaming\Python\Python314\site-packages (from flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in .\AppData\Roaming\Python\Python314\site-packages (from flask) (8.3.1)
Requirement already satisfied: itsdangerous>=2.2.0 in .\AppData\Roaming\Python\Python314\site-packages (from flask) (2.2.0)
Requirement already satisfied: Jinja2>=3.1.2 in .\AppData\Roaming\Python\Python314\site-packages (from flask) (3.1.6)
Requirement already satisfied: MarkupSafe>=2.1.1 in .\AppData\Roaming\Python\Python314\site-packages (from flask) (3.0.3)
Requirement already satisfied: Werkzeug>=3.1.0 in .\AppData\Roaming\Python\Python314\site-packages (from flask) (3.1.6)
Requirement already satisfied: colorama in .\AppData\Roaming\Python\Python314\site-packages (from click>=8.1.3->flask) (0.4.6)
Using cached flask-3.1.3-py3-none-any.whl (103 kB)
Installing collected packages: flask
  WARNING: The script flask.exe is installed in 'C:\Users\Lenovo\AppData\Roaming\Python\Python314\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed flask-3.1.3
```

This installs Flask, which is required to run the app.

4. Create Project Structure

Create a folder named `InterestCalculator`

Inside it, create:



5. Create `app.py` File

`app.py` is the brain of the application. This file contains the backend logic.

Main Tasks Performed by `app.py` -

- Creates Flask application
- Handles GET and POST requests
- Receives user input from HTML form
- Calculates interest

- Sends result back to webpage

Step 5.1: Import Required Modules

```
from flask import Flask, render_template, request
```

These help:

- Create Flask application
- Handle forms
- Display webpages

Step 5.2: Create Flask Application

```
app = Flask(__name__)
```

Explanation:

- Creates Flask app
- Initializes backend server

Step 5.3: Create Homepage Route

```
@app.route("/", methods=["GET", "POST"])
```

Explanation:

- Opens homepage
- Accepts user form data

Step 5.4: Create Flask Application

```
interest_type = request.form["interest_type"]  
p = float(request.form["principal"])  
r = float(request.form["rate"])  
t = float(request.form["time"])
```

Explanation:

Gets:

- Interest type
- Principal amount
- Rate of interest
- Time period

Step 5.5: Calculate Simple Interest

```
interest = (p * r * t) / 100
```

Explanation:

Calculates simple interest using the SI formula.

Step 5.6: Calculate Compound Interest

```
amount = p * (1 + r/100) ** t  
interest = amount - p
```

Explanation:

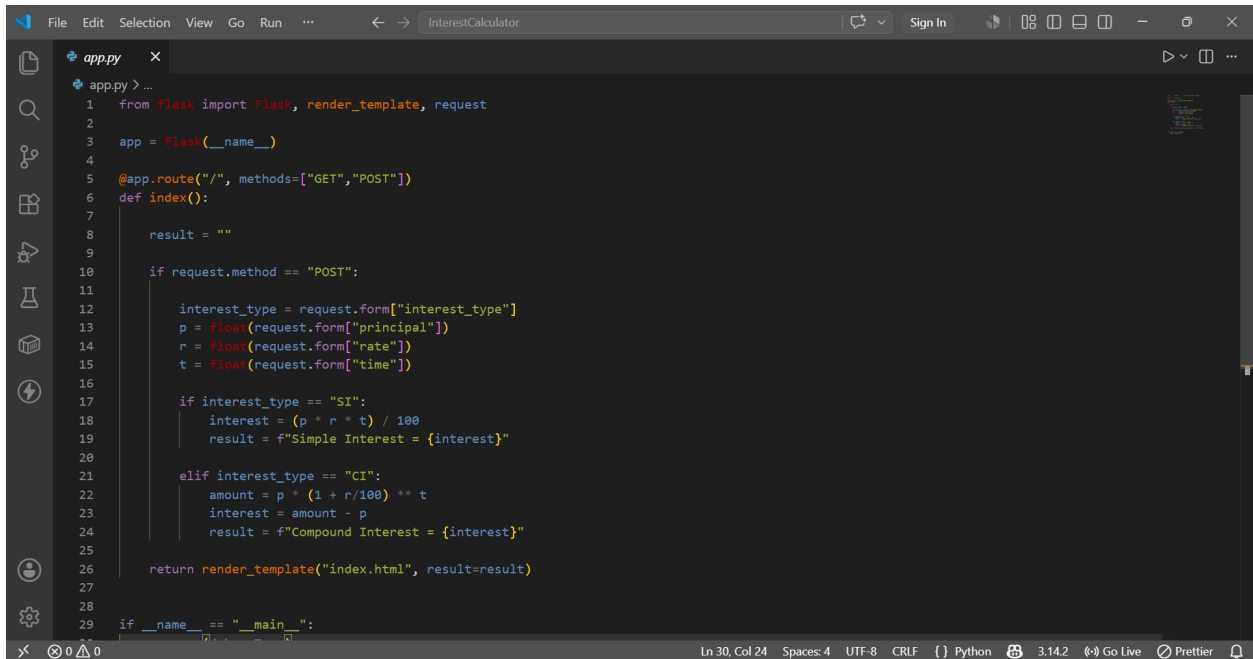
Calculates compound interest using the CI formula.

Step 5.7: Display Result

```
return render_template("index.html", result=result)
```

Explanation:

Sends calculated results to the HTML page.

A screenshot of a code editor window titled 'InterestCalculator'. The editor shows a Python file named 'app.py' with the following code:

```
1 from flask import Flask, render_template, request
2
3 app = Flask(__name__)
4
5 @app.route("/", methods=["GET", "POST"])
6 def index():
7
8     result = ""
9
10    if request.method == "POST":
11
12        interest_type = request.form["interest_type"]
13        p = float(request.form["principal"])
14        r = float(request.form["rate"])
15        t = float(request.form["time"])
16
17        if interest_type == "SI":
18            interest = (p * r * t) / 100
19            result = f"Simple Interest = {interest}"
20
21        elif interest_type == "CI":
22            amount = p * (1 + r/100) ** t
23            interest = amount - p
24            result = f"Compound Interest = {interest}"
25
26    return render_template("index.html", result=result)
27
28
29 if __name__ == "__main__":
```

The editor interface includes a sidebar with icons for file explorer, search, and other tools. The status bar at the bottom shows 'Ln 30, Col 24', 'Spaces: 4', 'UTF-8', 'CRLF', and other configuration details.

6. Frontend Development

Step 6.1: Create HTML File

Create:

templates/index.html

Explanation:

Creates a webpage interface.

Step 6.2: Add Form Inputs

```
<input type="number" name="principal">
```

Explanation:

Takes numeric input from the user.

Step 6.3: Add Interest Type Dropdown

```
<select name="interest_type">
```

Explanation:

Allows user to choose:

- Simple Interest
- Compound Interest

Step 6.4: Display Result

```
<h2>{{ result }}</h2>
```

Explanation:

Shows calculated interest dynamically.

Step 6.5: Create CSS File

Create:

```
static/style.css
```

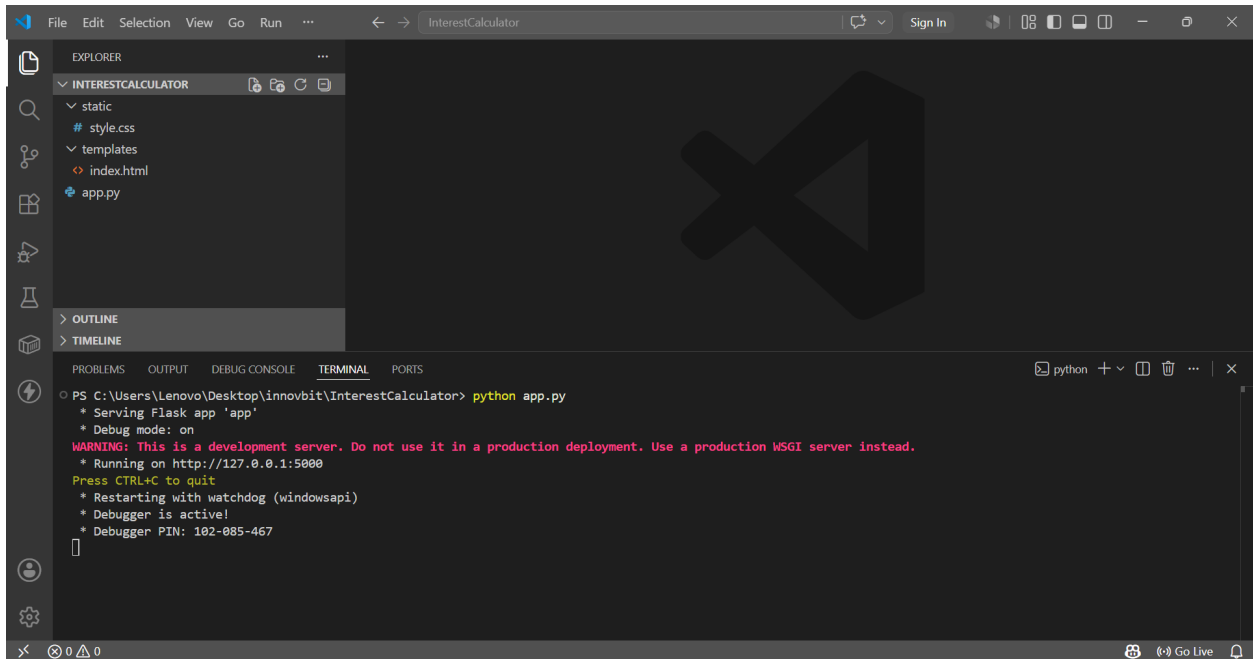
Explanation:

Improves webpage design.

7. Run the App

Open terminal

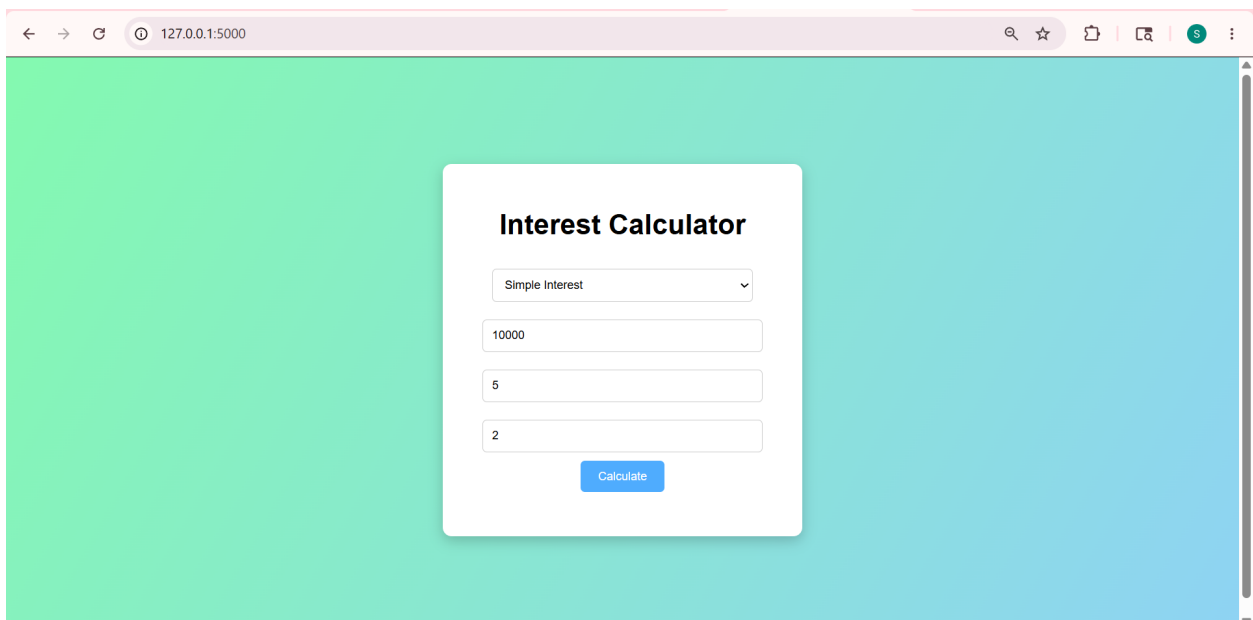
```
python app.py
```



Then open browser:

`http://127.0.0.1:5000`

The Interest Calculator webpage opens automatically.



8. How Everything Works Together (Flow)

- User opens application
- User enters Principal amount, Rate, Time
- User selects SI or CI
- Flask receives form data
- Python calculates interest
- Result displayed on webpage

9. Common Mistakes (Very Important)

9.1 Flask Not Installed

Error:

```
No module named flask
```

Fix:

```
pip install flask
```

9.2 HTML File Not Found

Error:

```
TemplateNotFound
```

Fix:

Keep HTML file inside:

```
templates/
```

folder.

9.3 CSS Not Working

Fix:

Keep CSS file inside:

```
static/
```

folder.

9.4 Wrong Flask Run Command

Correct command:

```
python app.py
```

11. Tips & Tricks

11.1 Add More Features

- EMI calculator
- Loan calculator
- GST calculator
- Currency converter
- Tax calculator
- Savings planner

11.2 Improve UI

- Animations
- Icons
- Hover effects
- Dark mode
- Attractive buttons

11.3 Add Input Validation

- Negative numbers
- Empty inputs
- Invalid values
- Zero values
- Special characters
- Large number limits

11.4 Use Database

- Previous calculations
- User history
- Saved reports
- Calculation timestamps
- Frequently used values
- User profiles

12. Final Summary

This project teaches:

- Flask basics
- HTML form handling
- Backend calculations
- Frontend and backend integration
- Web development concepts
- Financial calculation logic

The Interest Calculator Application is a beginner-friendly project that helps students learn Python Flask web development in a simple and practical way.