GETTING STARTED WITH FLASK – A BEGINNER’S TOOLKIT FOR WEB DEVELOPMENT

Goal

Build a minimal Quote Generator Web App using Flask and Python, where a button click displays a new random quote each time.

---

1. Quick Summary of the Technology

Flask is a lightweight, open-source Python web framework designed for building web applications quickly and flexibly. It follows a minimalistic approach, allowing developers to structure their projects however they prefer.

Why Flask?

- Very easy for Python beginners to understand
- Supports routing, templates, form handling, and extensions
- Perfect for small–medium web apps, APIs, and prototypes
- Ideal for learning core web development concepts (requests, responses, templates, sessions)

Real-World Usage

- Microservices for Netflix
- Internal tooling at Airbnb
- APIs for Reddit

For Your Project

You’ll build a simple, functional Quote Generator App using:

- Flask routes
- A template with a button
- A Python list of quotes
- A random quote generator function

---

2. System Requirements

Operating Systems

- Windows 10/11
- macOS
- Linux (Ubuntu, Debian, etc.)

Software Requirements

- Python 3.9 or later
- Virtual environment tool (`venv`)
- Text editor (VS Code or PyCharm recommended)

Dependencies

- Flask (installed via pip)
- No database required for the minimal quote generator
- Optional: Browser for viewing the app

---

3. Installation & Setup Instructions

1. **Install Python**
   - Download from python.org
2. **Create a project folder**

```
mkdir flask_toolkit
cd flask_toolkit
```

3. **Create a virtual environment**

```
python -m venv venv
```

4. **Activate the virtual environment**

- Windows: `venv\Scripts\activate`
- Mac/Linux: `source venv/bin/activate`

5. **Install Flask**

```
pip install flask
```

6. **Create main application file**

- `app.py`

7. **Create templates folder**

```
mkdir templates
```

8. **Run the app**

```
python app.py
```

9. **Open the browser**

- Visit: `http://127.0.0.1:5000/`

---

4. Minimal Working Example

app.py

```
from flask import Flask, render_template
import random

app = Flask(__name__)

quotes = [
    "The only limit to our realization of tomorrow is our doubts of today.",
    "You miss 100% of the shots you don't take.",
    "In the middle of difficulty lies opportunity.",
    "It always seems impossible until it's done.",
]

@app.route("/")
def index():
    return render_template("index.html", quote=random.choice(quotes))

if __name__ == "__main__":
    app.run(debug=True)
```

templates/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Quote Generator</title>
</head>
<body style="font-family: Arial; text-align: center; margin-top: 100px;">
    <h2>{{ quote }}</h2>
    <br/>
    <form method="GET">
```

```
        <button type="submit" style="padding: 10px 20px; font-size: 16px;">
            Show Another Quote
        </button>
    </form>
</body>
</html>
```

Expected Output: Each time you refresh the page or click the button, a new random quote appears.

---

5. AI Prompt Journal

Prompt 1 — Flask Learning Roadmap

- **Prompt Used:** "Can you create a structured 4-phase plan to learn Flask?"
- **AI Response Summary:** Provided a structured four-phase learning journey:
  1. Fundamentals & Setup (Hello World, virtual environment, routes, request/response)
  2. Routing, Templates & Forms (Dynamic routes, render_template, form handling)
  3. Data Persistence & Extensions (Flask-SQLAlchemy CRUD, migrations, other extensions)
  4. User Authentication & Mini-Project (Registration, login/logout, sessions, integrate previous skills)
- **Helpful Part:** Clear step-by-step progression for beginners to intermediate.
- **Evaluation:** Extremely helpful; easy to follow.

Prompt 2 — Explaining Flask Philosophy

- **Prompt Used:** "Explain the philosophical differences between Python scripts and Flask apps."
- **AI Response Summary:** Clarified web app concepts:
  - Event-driven request–response model
  - Stateless nature of Flask; persistence via sessions or database
  - Routes map to functions rather than direct calls
  - Multi-user and concurrency considerations
- **Helpful Part:** Provided foundational mental models for web development.
- **Evaluation:** Very helpful for approaching Flask development with correct mindset.

Prompt 3 — README Creation for Quote Generator

- **Prompt Used:** "Create a comprehensive README for my Quote Generator."
- **AI Response Summary:** Full README structure covering installation, usage, features, configuration, troubleshooting, and code structure.
- **Helpful Part:** Ready-to-use documentation framework.
- **Evaluation:** Very helpful; minimal edits required.

6. Common Issues & Fixes

1. **App not starting:** `pip show flask` to confirm installation.
2. **Port already in use:** Run `python app.py --port=5001`
3. **Template not found:** Ensure folder is `templates` and file is `index.html` (case-sensitive).
4. **Virtual environment not activating (Windows):** `Set-ExecutionPolicy Unrestricted -Scope Process`, then activate again.
5. **Browser not updating quote:** Open in Incognito Mode to bypass cache.
6. **Debug mode not reloading:** Delete `.pyc` files and ensure `app.run(debug=True)`.
7. **Jinja errors:** Check `{{ quote }}`, proper HTML nesting, and closing tags.

7. References

- Flask Official Documentation
- Python Docs
- DigitalOcean Flask Tutorials
- freeCodeCamp Web Development Tutorials
- StackOverflow — Flask tag