Web Development

using

Python Flask

Dan Yee

What is Flask?

- Flask is a web application framework written in Python.
- It allows programmers to develop and deploy server-backed web applications using Python.

- Examples:
  - Login pages
  - Account dashboards

Why is this important?

- Vanilla web development using just HTML, CSS and JavaScript is boring. You can already do a lot with just JavaScript on a website.
- What if you could do even more?
  - What if you wanted to access a database to perform login verification?
  - What if you wanted to write a website that tells you the different times in different locations?

Server-Backed Applications

- It’s a common misconception that using JavaScript to add website interactivity is a server implementation.
  - JavaScript on a website is considered Client-Side.
  - If you want a server-backed web application using JavaScript, you need to use a web framework such as Express.js
- With a server running on the backend with Flask, you can run any console application written in Python on a website.

Who Uses This?

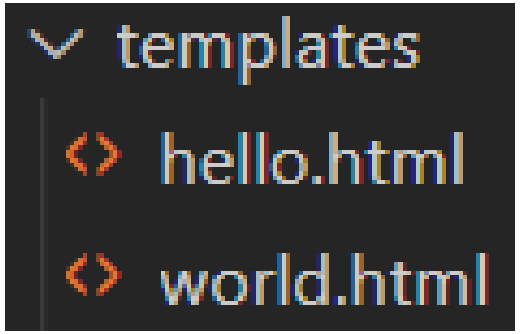
- In the professional world, when it comes to web development, many turn to frameworks like Flask and Express.js

- Personally, my professional experience with this is for a website I'm developing for a research lab.
  - The website uses Python's Request library to send an HTTP POST request to an API and return the data to be displayed on a webpage.

Getting Started with Flask

- The file structure of your Flask application will be slightly different.

- For all your HTML files, you must store them in a folder named “templates”. This is the default folder Flask will look in when you load an HTML file.
  - You can change this, but it’s not recommended.



Getting Started with Flask

- Before we can start making websites with Flask, we need a few pre-requisites:

- A package manager for Python modules – we will use Pip
- The Flask package
  - In your terminal, use “pip install flask” to install the Flask module to your development environment

Creating Your First Flask App

- Create a new Python source file.
- Import the Flask module and create a new Application using the Flask constructor:

```
1    import flask
2
3    App = flask.Flask(__name__)
```

This creates and initializes a Flask application

What Is A Route?

- Have you written basic websites before and saw your extensions as “website.com/page.html”?
- Do you want to get rid of that .html file extension? You can do so using a route!

A route in your Flask application allows your URL to go from “website.com/page.html” to “website.com/page”

Creating a Route

- Before we get to creating a route, we need go over one thing:

render_template

- This does what you might think it does. It “renders” the HTML “template” and displays it as it would it you just opened the HTML file in your browser.
  - This function call, by default, searches for the specified HTML file in a folder named “templates”

Creating a Route

- To create a route, we first use a decorator:

  @App.route(“/route_name”)

  - This decorator defines a route named “route” that allows you to (eventually) visit a specific website using the URL: “website.com/route”

- Example:

```
import flask

App = flask.Flask(__name__)

@App.route("/home")
```

Creating a Route

- The decorator also declares that the function defined right below it will be executed when this route is visited on a browser.

- Example:

```
@App.route("/hello")
def home():
    return flask.render_template("hello.html")
```

Running Your Application

- So how do you run your application and preview your work so far?
- You call the “run” function on your application.
  - By default, the application runs in development mode on your local machine on port 5000. You can visit the page at “localhost:5000” or “127.0.0.1:5000”
  - Execute the Python file.

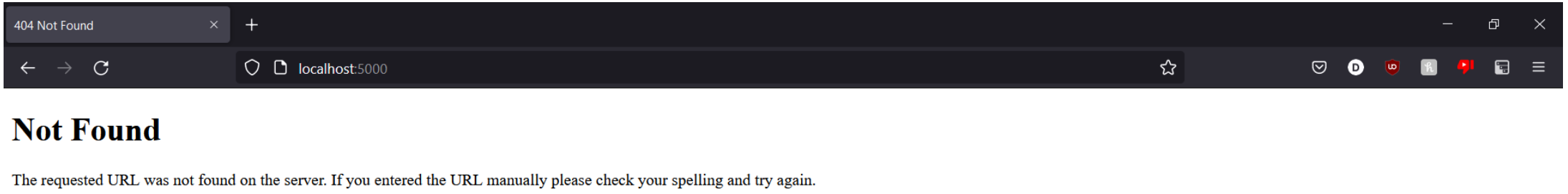
- Example:

```
App.run()
```

```
 * Serving Flask app 'main'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Running Your Application

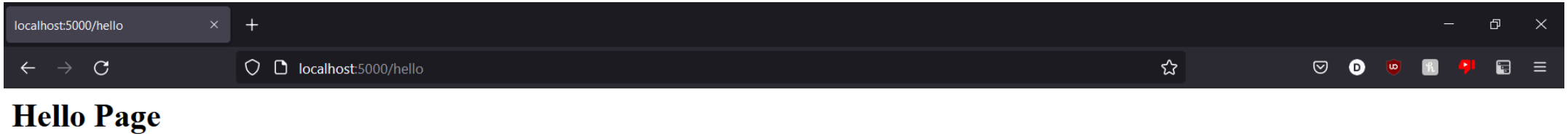
- When you first run your application and visit the localhost page on port 5000, you might see this:



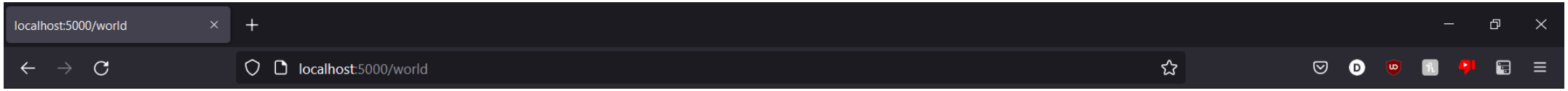
- This is because we never defined a route for just “/” which is the default loaded route.

Running Your Application

- However, if you edit the URL to include “/route_name”, you will see your HTML file rendered if it exists.



Hello Page



World Page

In the above images, I created two HTML files, “hello.html” and “world.html”, that simply display the name of the page in a header tag.

Summary

- In this presentation, we touched on:
  - What Flask is.
  - Why it’s important and what it’s used for.
  - The process of setting up a basic Flask web application:
    - Creating the Flask App
    - Creating page routes
    - Rendering HTML files
    - Running the application