

assignment=15

July 27, 2023

1 Q1. What is Flask Framework? What are the advantages of Flask Framework?

Flask is a popular web framework for building web applications using Python. It is a lightweight and modular framework that is easy to use and provides flexibility for developers. Some of the advantages of using Flask are:

Lightweight: Flask is a lightweight framework that does not come with any pre-built components, allowing developers to choose the components they need and avoiding any unnecessary bloat in their application.

Easy to use: Flask is easy to learn and use, making it an excellent choice for beginners. It also has a simple and intuitive API that allows developers to quickly build web applications.

Flexible: Flask allows developers to customize and extend the framework according to their needs. It does not impose any specific architecture or directory structure, giving developers more freedom in building their applications.

Built-in development server: Flask comes with a built-in development server, making it easy to test and debug applications without the need for additional tools.

Large community: Flask has a large community of developers who contribute to the framework's development and provide support to other developers. This community provides a wealth of resources, such as plugins, extensions, and documentation, that can help developers to build applications more quickly and efficiently.

Overall, Flask is a powerful framework that is easy to use, flexible, and customizable. It is an excellent choice for building web applications, particularly for smaller projects or prototypes.

2 Q2. Create a simple Flask application to display 'Hello World!!'. Attach the screenshot of the output in Jupyter Notebook.

```
[ ]: from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return '<h1>Hello World!!</h1>'
```

```
if __name__=="__main__":  
    app.run(host="0.0.0.0")
```

Hello World!!

3 Q3. What is App routing in Flask? Why do we use app routes?

App routing in Flask refers to the process of mapping URLs to functions in a Flask application. When a user makes a request to a URL in a Flask application, Flask uses the app routing mechanism to determine which function should be executed to handle the request.

App routes are essential in Flask because they enable developers to create multiple pages or endpoints for their web application. Each app route defines a unique URL, and when a user visits that URL, Flask calls the function that is associated with the route to generate the HTML content to be displayed on the page.

For example, consider a Flask application that has two pages, one to display a login form and another to display the user's profile information. In this case, the login page might have the URL '/login', and the profile page might have the URL '/profile'. By using app routes, the developer can define functions that will handle requests made to these URLs, making it easy to manage the logic for each page in a separate function.

App routes can also be used to define dynamic URLs that can accept parameters, such as user IDs or product names. By using dynamic routes, a developer can create a single function to handle requests to multiple URLs, reducing the amount of duplicated code and making the application more maintainable.

Overall, app routing is a crucial part of building web applications in Flask, as it enables developers to create flexible and dynamic web applications that can handle multiple URLs and serve content tailored to the user's request.

4 Q4. Create a “/welcome” route to display the welcome message “Welcome to ABC Corporation” and a “/” route to show the following details:

Company Name: JK Corporation

Location: India

Contact Detail: 999-999-9999

```
[ ]: from flask import Flask  
  
app = Flask(__name__)  
  
@app.route("/")  
def company_details():
```

```

    return "<h2>Company Name: JK Corporation</h2><br><h2>Location: India</h2><br><h2>Contact Detail: 999-999-9999</h2>"

@app.route("/welcome")
def welcome_message():
    return "<h1>Welcome to JK Corporation</h1>"

if __name__ == "__main__":
    app.run(host = "0.0.0.0")

```

company name: JK CORPORATION

LOCATION: India

CONTACT Number: 9875655454

<https://brown-analyst-tdtgg.pwskills.app:5000/>

Welcome to JK Corporation

<https://brown-analyst-tdtgg.pwskills.app:5000/welcome>

5 Q5. What function is used in Flask for URL Building? Write a Python code to demonstrate the working of the url_for() function.

In Flask, the url_for() function is used for URL building. This function takes the endpoint name and any arguments and returns a URL to that endpoint. The url_for() function is useful because it generates URLs based on the routes defined in the application, which ensures that the URLs are consistent and will continue to work even if the URL scheme changes.

Here's an example of how to use the url_for() function in Flask:

```

[ ]: from flask import Flask, url_for

app = Flask(__name__)

@app.route('/')
def index():
    return 'This is the homepage.'

@app.route('/user/<username>')
def profile(username):
    return f'This is the profile page for {username}.'

@app.route('/login')
def login():
    return 'Please log in.'

```

```
with app.test_request_context():
    print(url_for('index')) # output: '/'
    print(url_for('login')) # output: '/login'
    print(url_for('login', next='/')) # output: '/login?next=%2F'
    print(url_for('profile', username='Alice')) # output: '/user/Alice'
```

[]:

[]:

[]:

[]: