**Q1: What is Flask Framework? What are the advantages of Flask Framework?**

**Flask Framework**:  
Flask is a lightweight, web application framework in Python. It is designed to make it easy to create web applications and APIs. Flask is based on the WSGI (Web Server Gateway Interface) toolkit and Jinja2 templating engine.

**Advantages of Flask Framework**:

1. **Lightweight**: Flask is minimalistic and allows developers to add extensions as needed, keeping the core framework simple.
2. **Flexibility**: Provides high customization options and supports different development needs.
3. **Ease of Use**: Flask is easy to learn and use, especially for beginners.
4. **Built-in Development Server**: Flask includes a built-in server for testing applications locally during development.
5. **RESTful Support**: Simplifies building RESTful APIs with routing and URL handling.
6. **Modular Design**: Developers can divide an application into multiple independent modules for better organization.
7. **Community Support**: Large and active community, which means plenty of documentation and resources.

**Q2: Create a Simple Flask Application to Display "Hello World!!"**

Below is the code to create a Flask application and display "Hello World!!":

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello\_world():

return 'Hello World!!'

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

To execute and display output in a Jupyter Notebook, you can use the flask command in the terminal or execute Flask in a cell using %run.

**Q3: What is App Routing in Flask? Why Do We Use App Routes?**

**App Routing**:  
App routing in Flask refers to the process of mapping URLs to specific functions in your application. Each route is associated with a function known as a "view function."

**Why Use App Routes**:

1. To define different web pages or APIs.
2. To handle user requests for specific URLs.
3. To organize code for different functionalities or sections of a web application.
4. To enable RESTful API creation by linking routes with HTTP methods (GET, POST, etc.).

Example of app routing:

@app.route('/hello')

def hello():

return 'Hello Flask!'

**Q4: Create “/welcome” and “/” Routes**

**Code Example**:

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return '''

<h1>Company Name: ABC Corporation</h1>

<p>Location: India</p>

<p>Contact Detail: 999-999-9999</p>

'''

@app.route('/welcome')

def welcome():

return 'Welcome to ABC Corporation'

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**Execution in Jupyter Notebook**:  
You can use %run or save the script as a .py file and execute it. Provide a screenshot of the running app in the browser.

**Q5: What Function is Used in Flask for URL Building?**

**Function Used**:  
The url\_for() function in Flask is used for URL building. It generates a URL for a specific function or route using the function's name and optional arguments.

**Code Example**:

from flask import Flask, url\_for

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return 'Home Page'

@app.route('/about')

def about():

return 'About Page'

@app.route('/demo')

def demo():

# Example of URL building

home\_url = url\_for('home')

about\_url = url\_for('about')

return f'Home URL: {home\_url}, About URL: {about\_url}'

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

Output for /demo route:

Home URL: /, About URL: /about