Q1

GET method is used to request the server to GET data or information.It should not be used with the requesting sensitive data,and there are certain limits to the requesting data.GET requests to remain in the browser history, and it can also be bookmarked. GET requests can be cached.

Post method is used to send the data to the server.It is generally used when we want to upload data or any file to the server like filling out the form.POST requests cannot be bookmarked, and it does not remain in the browser history. It cannot be cached.

Q2 The Request, in Flask, is an object that contains all the data sent from the Client to Server. This data can be recovered using the GET/POST Methods. When the Flask application handles a request, it creates a Request object based on the environment it received from the WSGI server. Because a worker (thread, process, or coroutine depending on the server) handles only one request at a time, the request data can be considered global to that worker during that request.

Q3 A redirect is used in the Flask class to send the user to a particular URL with the status code. conversely, this status code additionally identifies the issue The redirect function allows you to redirect the user to another URL. redirect can be used in Server Components, Client Components, Route Handlers, and Server Actions

Q4 Templates are files that contain static data as well as placeholders for dynamic data. A template is rendered with specific data to produce a final document. Flask uses the Jinja template library to render templates. In your application, you will use templates to render HTML which will display in the user's browser.

the render_template() function to render a template file called index.html. Next, you'll have to create the index. html template file in a directory called templates inside your flask_app directory.

Q5

from flask import Flask,request,jsonify

app = Flask(**name**)

@app.route("/get-user/user_id>") def get_user(user_id): user_data={ "user_id":user_id, "name" : "John doe", "email": "john.doe@example.com" }

extra=request.args.get("extra") if extra: user_data["extra"]=extra

return jsonify(user_data), 200

if **name**=="**main**": app.run(debug=True)