**Developing an API for a Distributed Environment:**

**api.py**

```
from flask import Flask
from flask_restful import Api, Resource, reqparse

app = Flask(__name__)
api = Api(app)

users = [
    {
        "name": "James",
        "age": 30,
        "occupation": "Network Engineer"
    },
    {
        "name": "Ann",
        "age": 32,
        "occupation": "Doctor"
    },
    {
        "name": "Jason",
        "age": 22,
        "occupation": "Web Developer"
    }
]


class User(Resource):
    def get(self, name):
        for user in users:
            if (name == user["name"]):
                return user, 200
        return "User not found", 404

    def post(self, name):
        parser = reqparse.RequestParser()
        parser.add_argument("age")
        parser.add_argument("occupation")
        args = parser.parse_args()

        for user in users:
            if (name == user["name"]):
                return "User with name {} already exists".format(name), 400

        user = {
            "name": name,
            "age": args["age"],
            "occupation": args["occupation"]
        }
        users.append(user)
        return user, 201

    def put(self, name):
        parser = reqparse.RequestParser()
        parser.add_argument("age")
        parser.add_argument("occupation")
        args = parser.parse_args()
```

```
        for user in users:
            if (name == user["name"]):
                user["age"] = args["age"]
                user["occupation"] = args["occupation"]
                return user, 200

        user = {
            "name": name,
            "age": args["age"],
            "occupation": args["occupation"]
        }
        users.append(user)
        return user, 201

    def delete(self, name):
        global users
        users = [user for user in users if user["name"] != name]
        return "{} is deleted.".format(name), 200


api.add_resource(User, "/user/<string:name>")

app.run(debug=True)
```

**Question 1:**

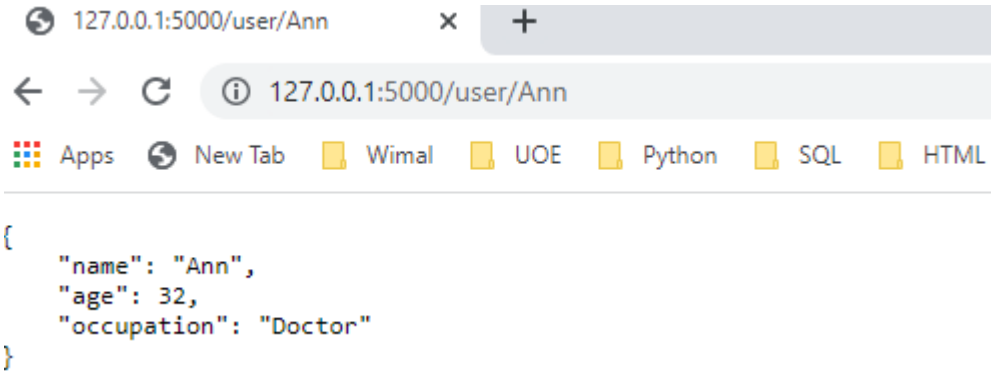
Run the API.py code. Take a screenshot of the terminal output. What command did you use to compile and run the code?

```
* Serving Flask app 'api' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 557-066-859
```

The command used: python api.py

**Question 2:**

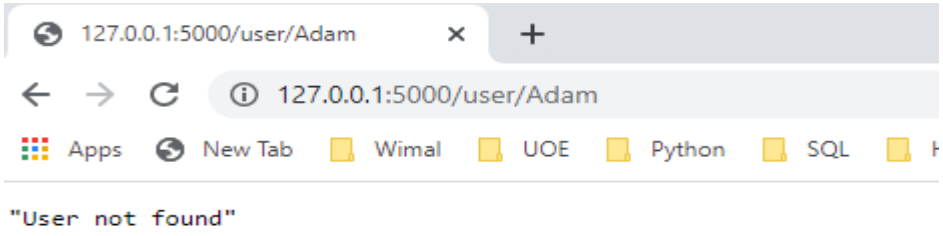
Run the following command at the terminal prompt: w3m **http://127.0.0.1:5000/user/Ann**
What happens when this command is run, and why?




**http://127.0.0.1:5000/user/Ann** calls the endpoint user with Ann as the parameter. Since the method is GET it returns the above result

**Question 3:**

Run the following command at the terminal prompt: w3m **http://127.0.0.1:5000/user/Adam**
What happens when this command is run, and why?




**http://127.0.0.1:5000/user/Adam** follows the same method as above but returns an error because there is no “Adam” in users.

**Question 4:**

What capability is achieved by the flask library?
Libraries allow you to run a web application