

Developing an API for a Distributed Environment


Question 1

```
○ george.koridze@MBP-GK-QQXJPGK7P4 testing-with-python % /usr/local/bin/python3 /Users/george.koridze/Desktop/Essex/SSD/ePortfolio/Unit7/API.py
* Serving Flask app 'API'
* Debug mode: on
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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 672-162-057
```

Question 2

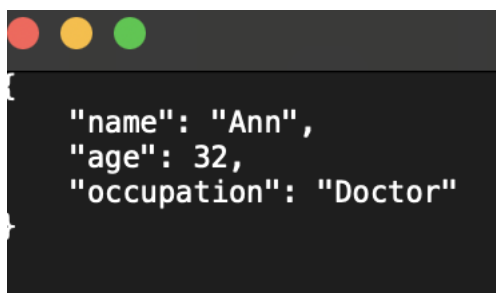
Run: w3m <http://127.0.0.1:5000/user/Ann>

Output:



```
SSD — w3m http://127.0.0.1:5000/user/Ann — 91x40
Download List Panel

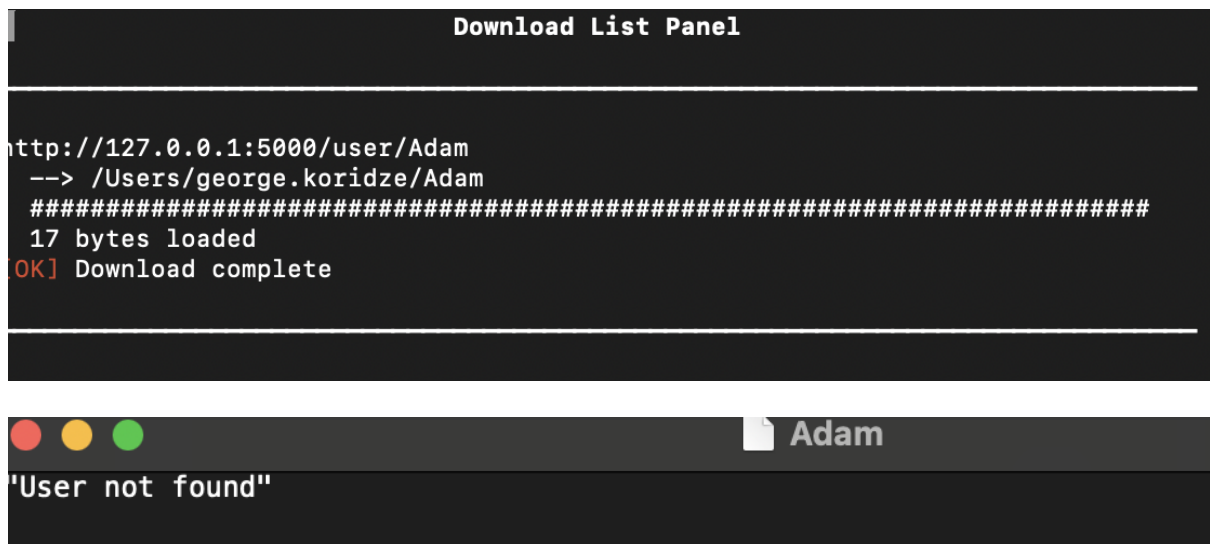
http://127.0.0.1:5000/user/Ann
--> /Users/george.koridze/Desktop/Essex/SSD/q
#####
65 bytes loaded
OK] Download complete
```



```
{
  "name": "Ann",
  "age": 32,
  "occupation": "Doctor"
}
```

Question 3

Run: w3m <http://127.0.0.1:5000/user/Adam>



Question 4

1. **Routing:** Define endpoints and map them to Python functions (e.g., /user/<string:name>).
2. **Request Parsing:** Handle GET, POST, PUT, and DELETE HTTP methods with ease.
3. **Debugging:** Provides an integrated development server with debugging features.
4. **Extensibility:** Supports additional libraries like Flask-RESTful to simplify API creation.
5. **Template Rendering:** Offers tools to render HTML templates for full-stack web development.
6. **Scalability:** Allows for the development of simple APIs and complex web applications alike.

This makes Flask an excellent choice for developing APIs and small-scale web applications.