System Development Lab
MicroWebServices

Mugilan E.S.
2019202033

To-Do Application

Overview:

This is a Python MicroWebService using Flask Framework with JSON DB as the Database. This application has two services namely, Users & Todos.

User-Service:

The user service provides a RESTful endpoint to list the users in our application and also allows to query the user lists based on their usernames. This service currently runs on port 5000 of our server.

To-Do-Service:

The ToDo service provides a RESTful endpoint to list all the lists aswell as providing the list of projects filtered on the basis of usernames. This service runs on port 5001 of our server.

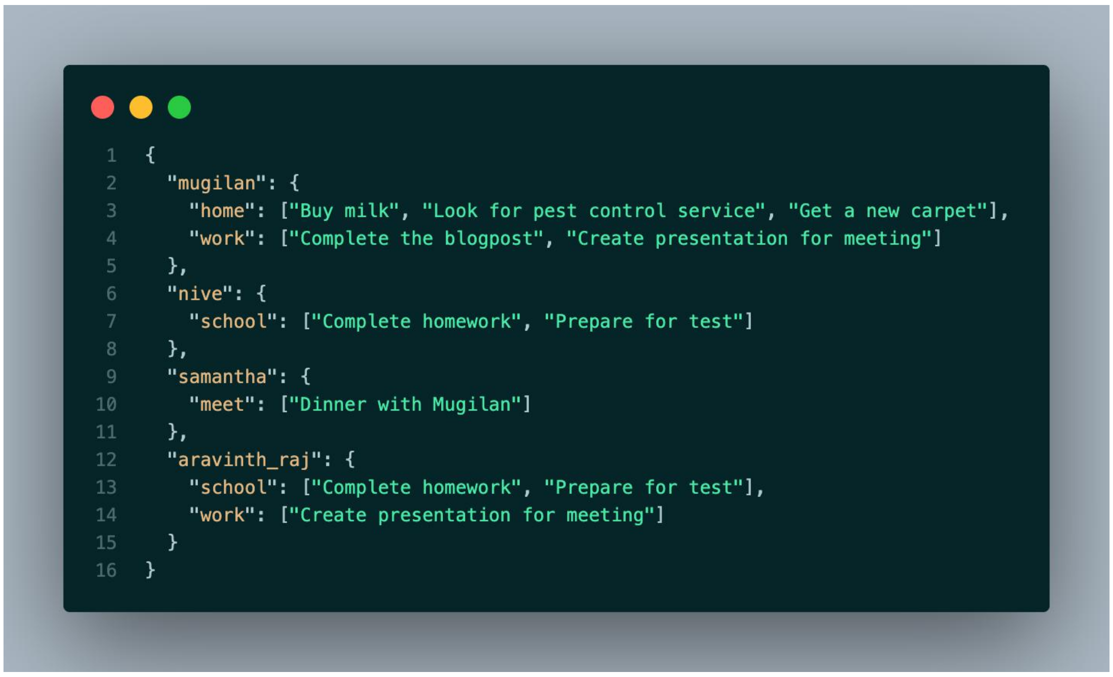
Create The Project:

1. Make a new folder for the project files.

2. Create a Virtual Environment using the built-in venv module. Activate it.

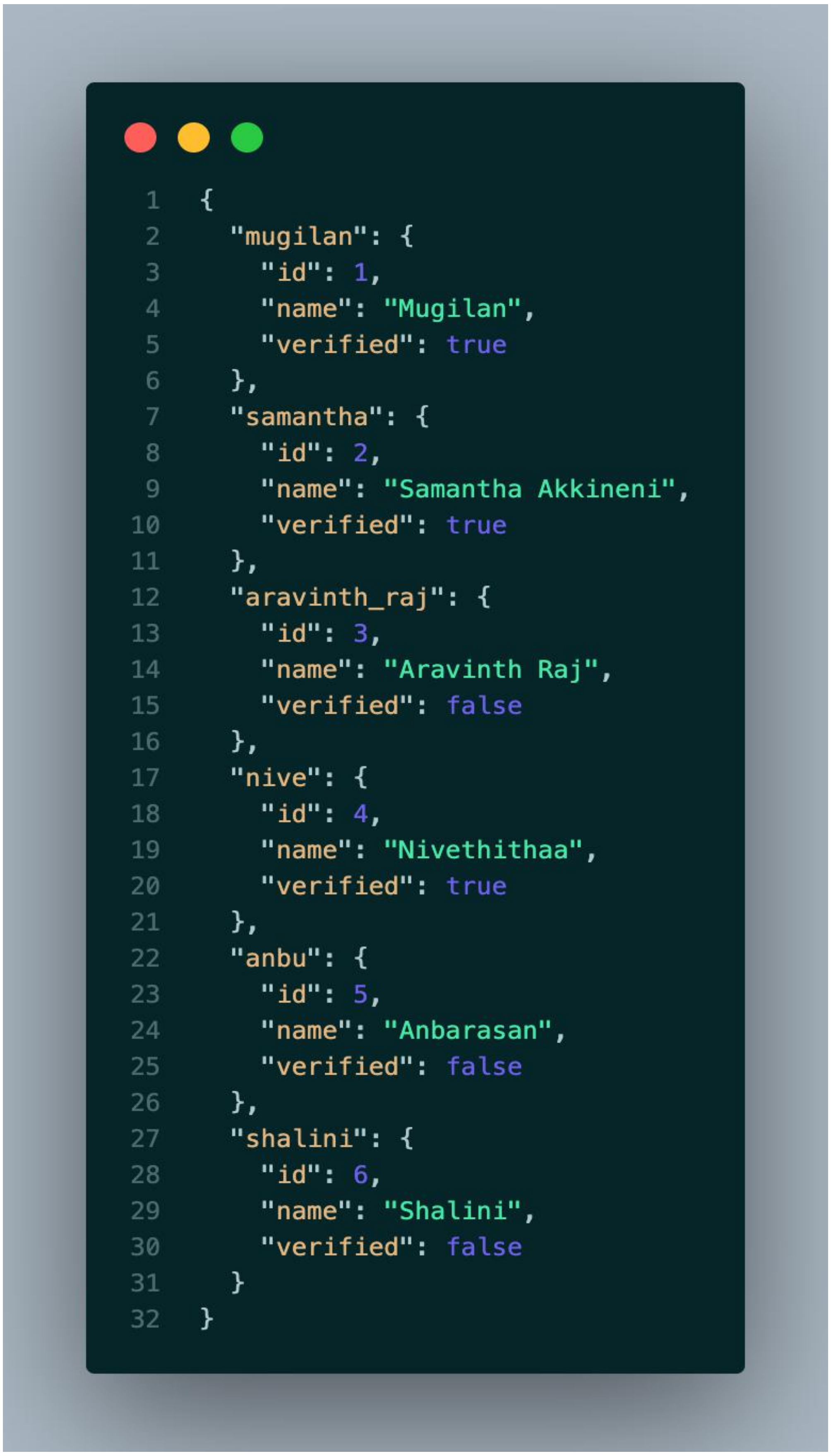
   `python -m venv .venv`

database/todos.json



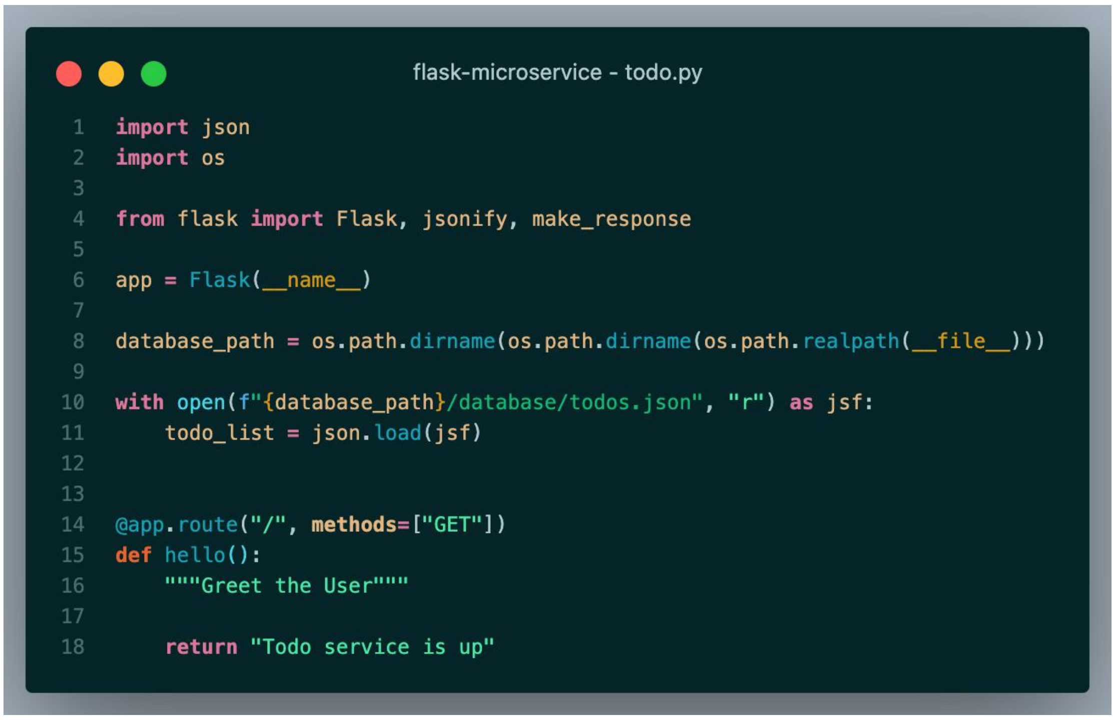
```json
{
  "mugilan": {
    "home": ["Buy milk", "Look for pest control service", "Get a new carpet"],
    "work": ["Complete the blogpost", "Create presentation for meeting"]
  },
  "nive": {
    "school": ["Complete homework", "Prepare for test"]
  },
  "samantha": {
    "meet": ["Dinner with Mugilan"]
  },
  "aravinth_raj": {
    "school": ["Complete homework", "Prepare for test"],
    "work": ["Create presentation for meeting"]
  }
}
```

database/users.json

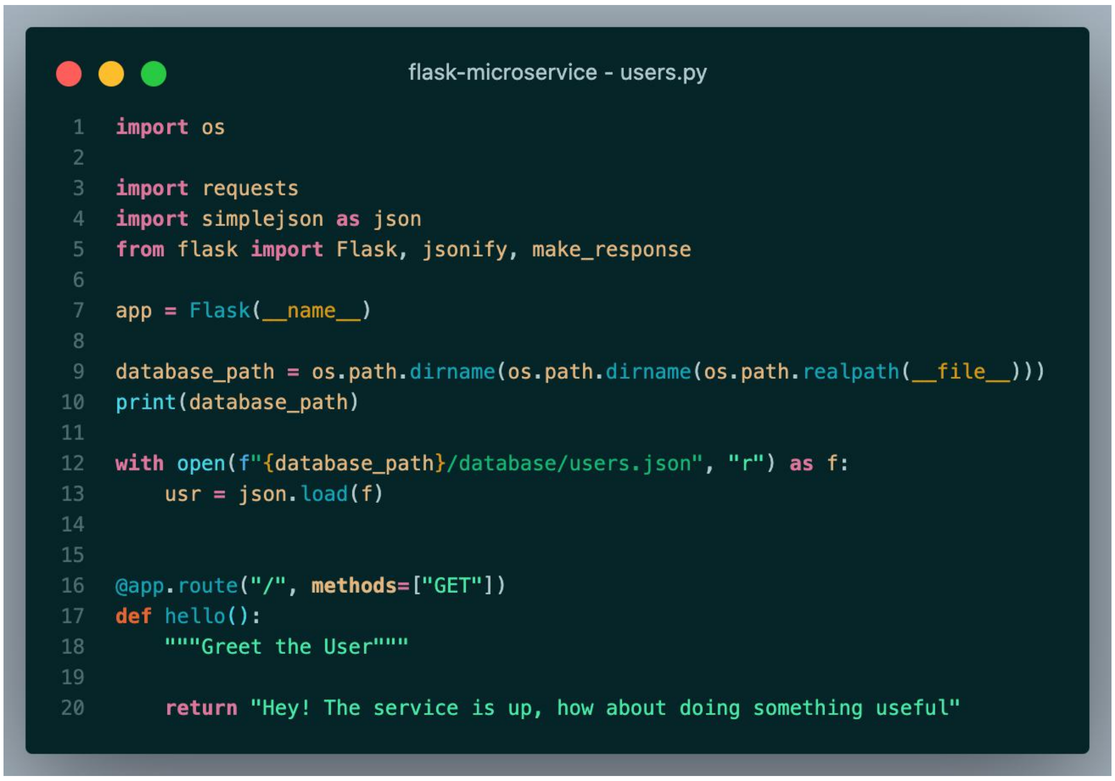


```json
{
  "mugilan": {
    "id": 1,
    "name": "Mugilan",
    "verified": true
  },
  "samantha": {
    "id": 2,
    "name": "Samantha Akkineni",
    "verified": true
  },
  "aravinth_raj": {
    "id": 3,
    "name": "Aravinth Raj",
    "verified": false
  },
  "nive": {
    "id": 4,
    "name": "Nivethithaa",
    "verified": true
  },
  "anbu": {
    "id": 5,
    "name": "Anbarasan",
    "verified": false
  },
  "shalini": {
    "id": 6,
    "name": "Shalini",
    "verified": false
  }
}
```

Then Create Services in services folder. This is a sample part


flask-microservice - todo.py

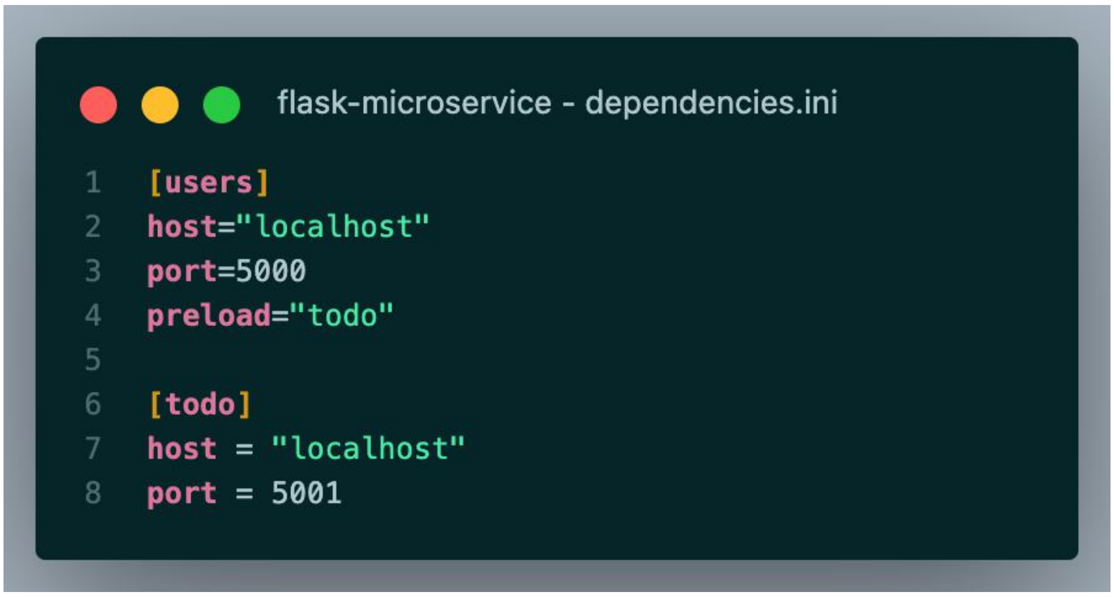
```python
import json
import os

from flask import Flask, jsonify, make_response

app = Flask(__name__)

database_path = os.path.dirname(os.path.dirname(os.path.realpath(__file__)))

with open(f"{database_path}/database/todos.json", "r") as jsf:
    todo_list = json.load(jsf)


@app.route("/", methods=["GET"])
def hello():
    """Greet the User"""

    return "Todo service is up"
```


flask-microservice - users.py

```python
import os

import requests
import simplejson as json
from flask import Flask, jsonify, make_response

app = Flask(__name__)

database_path = os.path.dirname(os.path.dirname(os.path.realpath(__file__)))
print(database_path)

with open(f"{database_path}/database/users.json", "r") as f:
    usr = json.load(f)


@app.route("/", methods=["GET"])
def hello():
    """Greet the User"""

    return "Hey! The service is up, how about doing something useful"
```

Create a empty `__init__.py` file to make the folder as a module

Declare dependencies in the root of the project



```ini
[users]
host="localhost"
port=5000
preload="todo"

[todo]
host = "localhost"
port = 5001
```

Install dependencies for the project to work using pip

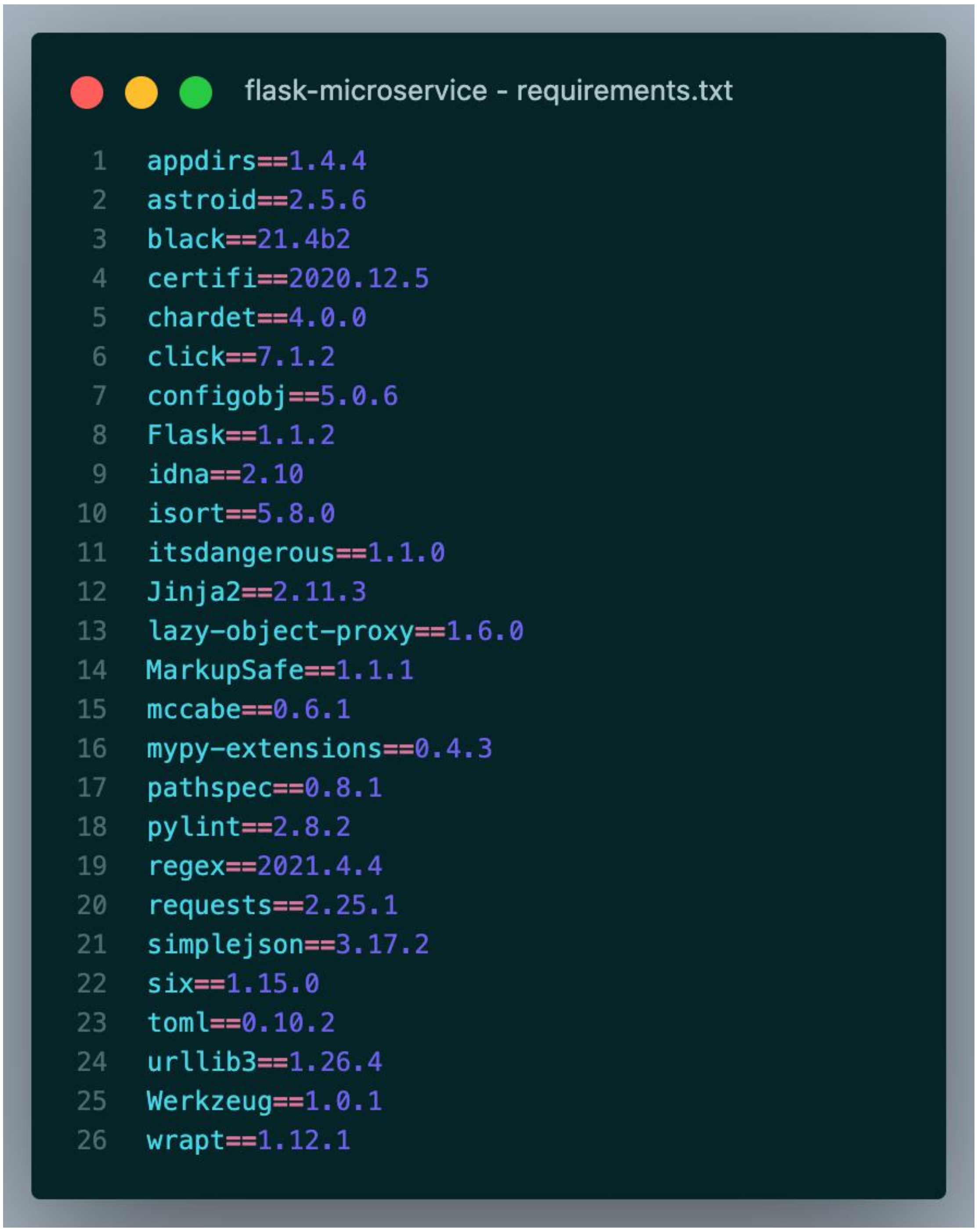
1. flask
2. requests
3. simplejson
4. configobj

You have to activate the Virtual Environment before installing these packages.

`source .venv/bin/activate`

You can freeze the requirements needed for the project in a .txt file

`pip freeze > requirements.txt`

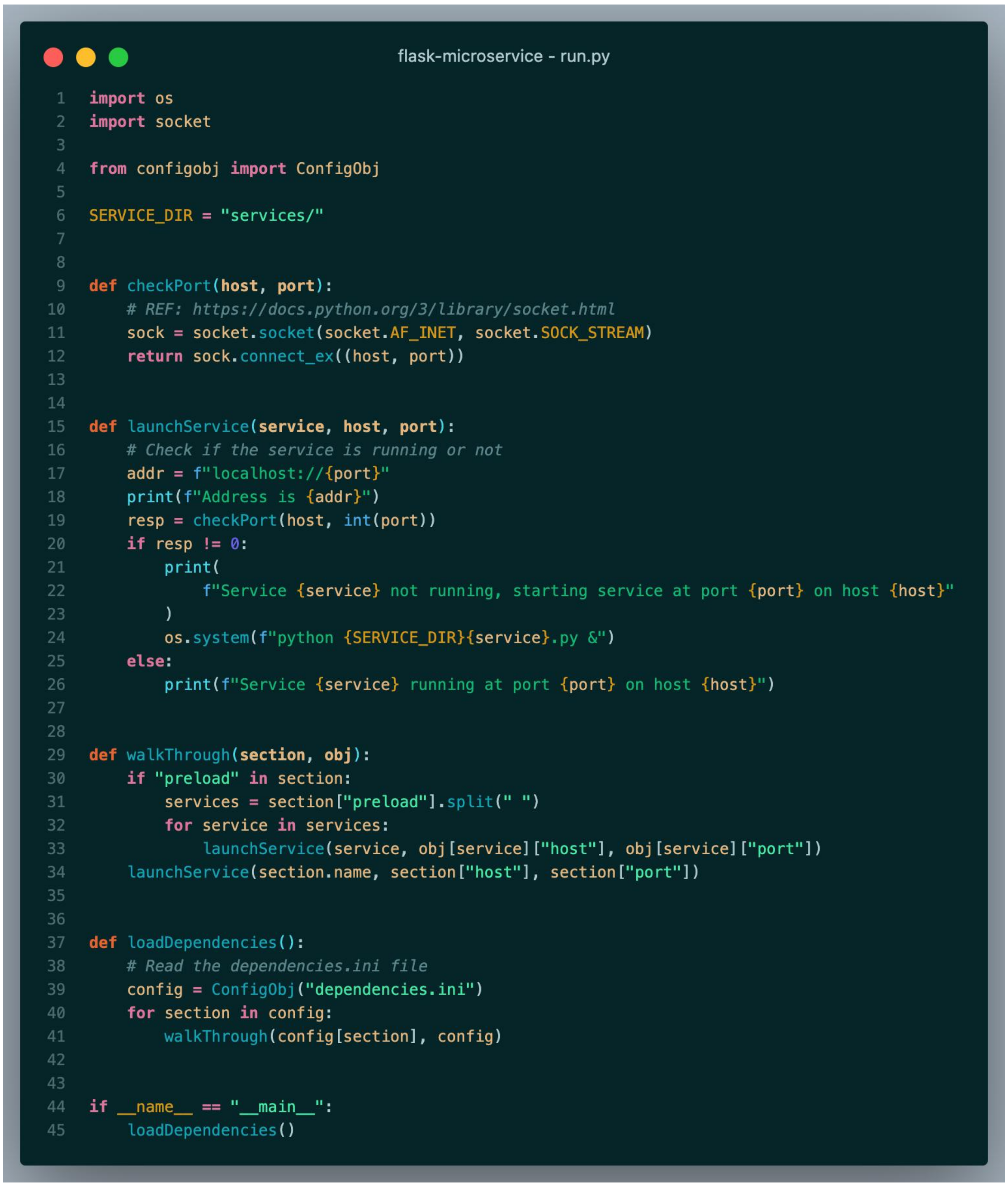


```
appdirs==1.4.4
astroid==2.5.6
black==21.4b2
certifi==2020.12.5
chardet==4.0.0
click==7.1.2
configobj==5.0.6
Flask==1.1.2
idna==2.10
isort==5.8.0
itsdangerous==1.1.0
Jinja2==2.11.3
lazy-object-proxy==1.6.0
MarkupSafe==1.1.1
mccabe==0.6.1
mypy-extensions==0.4.3
pathspec==0.8.1
pylint==2.8.2
regex==2021.4.4
requests==2.25.1
simplejson==3.17.2
six==1.15.0
toml==0.10.2
urllib3==1.26.4
Werkzeug==1.0.1
wrapt==1.12.1
```

If you want to install the packages ftom the requirements.txt

`pip install -r requirements.txt`

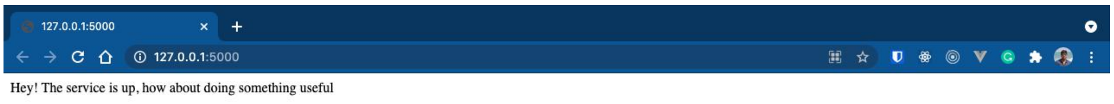
Create a main `run.py` file the root where the microservices will be started up.


flask-microservice - run.py

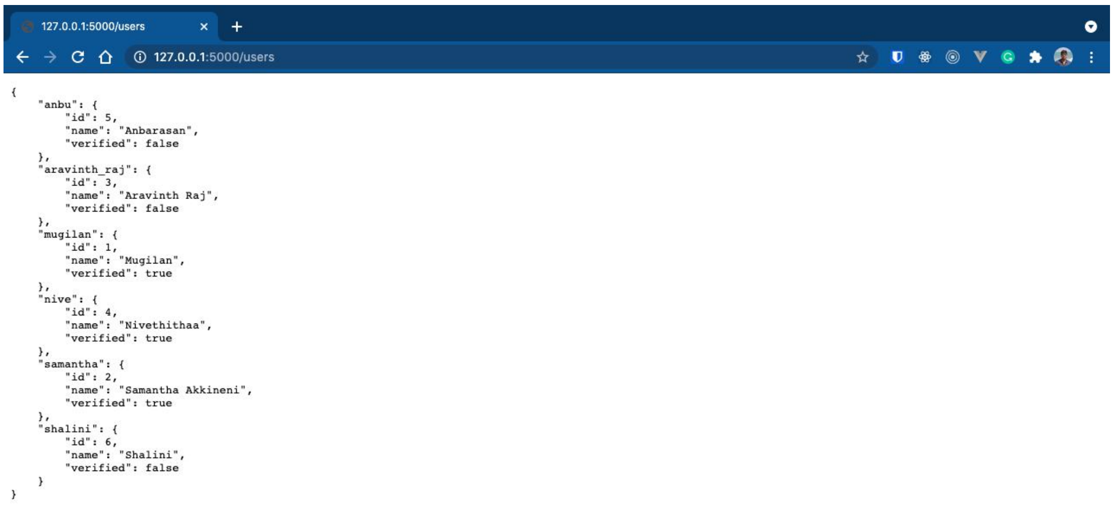
```python
import os
import socket

from configobj import ConfigObj

SERVICE_DIR = "services/"


def checkPort(host, port):
    # REF: https://docs.python.org/3/library/socket.html
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    return sock.connect_ex((host, port))


def launchService(service, host, port):
    # Check if the service is running or not
    addr = f"localhost://{port}"
    print(f"Address is {addr}")
    resp = checkPort(host, int(port))
    if resp != 0:
        print(
            f"Service {service} not running, starting service at port {port} on host {host}"
        )
        os.system(f"python {SERVICE_DIR}{service}.py &")
    else:
        print(f"Service {service} running at port {port} on host {host}")


def walkThrough(section, obj):
    if "preload" in section:
        services = section["preload"].split(" ")
        for service in services:
            launchService(service, obj[service]["host"], obj[service]["port"])
    launchService(section.name, section["host"], section["port"])


def loadDependencies():
    # Read the dependencies.ini file
    config = ConfigObj("dependencies.ini")
    for section in config:
        walkThrough(config[section], config)


if __name__ == "__main__":
    loadDependencies()
```

Start the microservice by running,

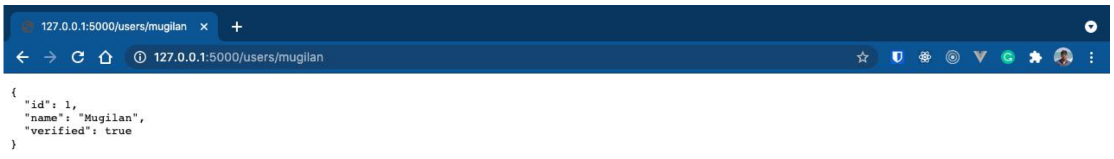
`python run.py`



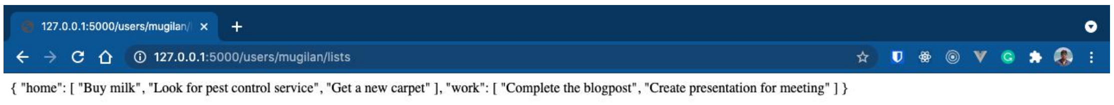
Hey! The service is up, how about doing something useful



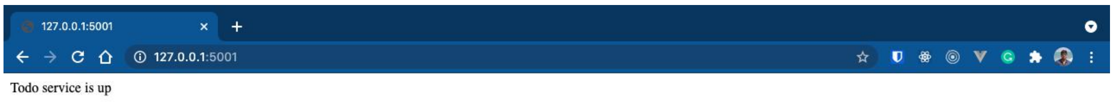
```
{
    "anbu": {
        "id": 5,
        "name": "Anbarasan",
        "verified": false
    },
    "aravinth_raj": {
        "id": 3,
        "name": "Aravinth Raj",
        "verified": false
    },
    "mugilan": {
        "id": 1,
        "name": "Mugilan",
        "verified": true
    },
    "nive": {
        "id": 4,
        "name": "Nivethithaa",
        "verified": true
    },
    "samantha": {
        "id": 2,
        "name": "Samantha Akkineni",
        "verified": true
    },
    "shalini": {
        "id": 6,
        "name": "Shalini",
        "verified": false
    }
}
```



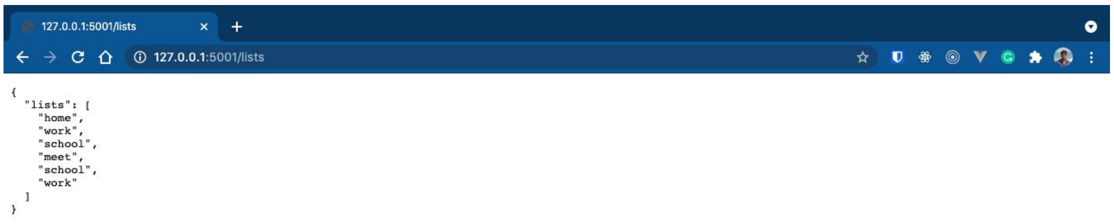
```
{
  "id": 1,
  "name": "Mugilan",
  "verified": true
}
```


127.0.0.1:5000/users/mugilan/
127.0.0.1:5000/users/mugilan/lists

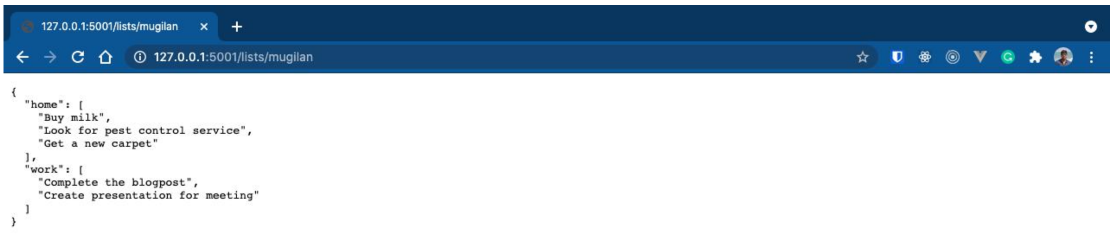
```
{ "home": [ "Buy milk", "Look for pest control service", "Get a new carpet" ], "work": [ "Complete the blogpost", "Create presentation for meeting" ] }
```



Todo service is up



```
{
  "lists": [
    "home",
    "work",
    "school",
    "meet",
    "school",
    "work"
  ]
}
```


127.0.0.1:5001/lists/mugilan

```
{
  "home": [
    "Buy milk",
    "Look for pest control service",
    "Get a new carpet"
  ],
  "work": [
    "Complete the blogpost",
    "Create presentation for meeting"
  ]
}
```

Source Code:

[Mugilan_Flask-Todo-MicroWebService]()