Project Title:

Health Calculator Microservice- Part 1

Objective:

To develop a Python-based microservice that calculates health metrics (BMI and BMR) using a REST API. The project will be containerized with Docker, managed with Makefile, and deployed to Azure using GitHub Actions for CI/CD. It will be developed in two steps, the first step is to develop and build the microservice locally with testing tool, and the second step is to create the CI/CD pipelines and deploy it on Azure

Mathematical Equations for Health Calculations

1. **Body Mass Index (BMI)**:

$$ \text{BMI} = \frac{\text{weight (kg)}}{(\text{height (m)})^2}$$

2. **Basal Metabolic Rate (BMR)** (Harris-Benedict Equation):
    - For **males**:

    $$ \text{BMR} = 88.362 + (13.397 \times \text{weight (kg)}) + (4.799 \times \text{height (cm)}) - (5.677 \times \text{age (years)})$$

    - For **females**: $$ \text{BMR} = 447.593 + (9.247 \times \text{weight (kg)}) + (3.098 \times \text{height (cm)}) - (4.330 \times \text{age (years)}) $$

Project Requirements:

1. **Python Microservice**:
    - Develop a Flask REST API with endpoints:
        - `/bmi` : Calculates BMI using `height` (meters) and `weight` (kg).
        - `/bmr` : Calculates BMR using `height` (cm), `weight` (kg), `age` , and `gender` .
2. **Containerization with Docker**:

- Create a `Dockerfile` to containerize the application.

3. **Dependency Management**:
   - Manage dependencies in `requirements.txt`.
4. **Testing**:
   - Write unit tests to validate the BMI and BMR calculations and API endpoints.

Detailed Project Instructions

1. Microservice Development

1. **Create a directory** named `health-calculator-service`.
2. Inside `health-calculator-service`, create the following files:
   - **`app.py`**:
     - Define the Flask API with two endpoints (`/bmi` and `/bmr`).
   - **`health_utils.py`**:
     - Define utility functions `calculate_bmi` and `calculate_bmr`.

Example Code:

- **app.py**

```
app = Flask(__name__)

@app.route('/bmi', methods=['POST'])
def bmi():
    # here goes the code

@app.route('/bmr', methods=['POST'])
def bmr():
    # here goes the code

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

- **health_utils.py**

```
def calculate_bmi(height, weight):
    """Calculate Body Mass Index (BMI) given height in meters and weight in kil
    return BMI

def calculate_bmr(height, weight, age, gender):
    """Calculate Basal Metabolic Rate (BMR) using the Harris-Benedict equation.'
```

2. Containerization with Docker

- Create a **Dockerfile** in the `health-calculator-service` directory to containerize the application.

Example Dockerfile:

```
FROM python:3.9-slim

WORKDIR /app

COPY . /app

RUN pip install -r requirements.txt

EXPOSE 5000

CMD ["python", "app.py"]
```

3. Dependency Management

- Create a **requirements.txt** file with dependencies.

Example requirements.txt:

```
Flask==2.0.2
```

4. Testing

- Create a **test.py** file to validate BMI and BMR calculations.

Example test.py:

```
import unittest
from health_utils import calculate_bmi

class TestHealthUtils(unittest.TestCase):

    def test_calculate_bmi(self):
        self.assertAlmostEqual(calculate_bmi(1.75, 70), 22.86, places=2)


if __name__ == '__main__':
    unittest.main()
```

Expected Deliverables:

1. A GitHub repository with:
    - The Flask microservice code (`app.py` , `health_utils.py`).
    - Unit tests (`test.py`).
    - Dockerfile.
    - requirements.txt.