- **Flask**: A Python web framework for the application.
- **PostgreSQL**: A relational database for storing form data.
- **Docker**: To containerize the application and database.
- **Docker Compose**: To manage multi-container Docker applications.

Step 1: Set Up the Application

1.1 Create the Project Directory

```
mkdir form-app
cd form-app
```

1.2 Write the Flask Application

```
gedit app.py
```

```
from flask import Flask, render_template, request, redirect, url_for
import psycopg2

app = Flask(__name__)

# Database configuration
DB_HOST = "db"
DB_NAME = "mydb"
DB_USER = "user"
DB_PASSWORD = "password"

def get_db_connection():
        conn = psycopg2.connect(
        host=DB_HOST,
        database=DB_NAME,
        user=DB_USER,
        password=DB_PASSWORD
        )
        return conn

@app.route("/")
def home():
        return render_template("index.html")

@app.route("/submit", methods=["POST"])
def submit():
        name = request.form["name"]
        email = request.form["email"]

        # Insert data into the database
        conn = get_db_connection()
        cur = conn.cursor()
```

```
        cur.execute("INSERT INTO users (name, email) VALUES (%s, %s)", (name, email))
        conn.commit()
        cur.close()
        conn.close()

        return redirect(url_for("home"))

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

1.3 Create the HTML Form

```
mkdir templates
gedit templates/index.html
```

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Form Application</title>
</head>
<body>
        <h1>Submit Your Details</h1>
        <form action="/submit" method="POST">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <br>
        <button type="submit">Submit</button>
        </form>
</body>
</html>
```

1.4 Create the `requirements.txt` File

```
gedit  requirements.txt
```

```
Flask==2.0.1
psycopg2-binary==2.9.1
```

Step 2: Set Up the Database

2.1 Create a `docker-compose.yml` File

gedit docker-compose.yml

```
version: '3.8'

services:
  db:
        image: postgres:13
        environment:
        POSTGRES_USER: user
        POSTGRES_PASSWORD: password
        POSTGRES_DB: mydb
        ports:
        - "5432:5432"
        volumes:
        - postgres_data:/var/lib/postgresql/data

  web:
        build: .
        ports:
        - "5000:5000"
        depends_on:
        - db
        environment:
        - DB_HOST=db
        - DB_NAME=mydb
        - DB_USER=user
        - DB_PASSWORD=password

volumes:
  postgres_data:
```

2.2 Create the Dockerfile

gedit Dockerfile

```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 5000
CMD ["python", "app.py"]
```

Step 3: Initialize the Database

First check if you have docker-compose

docker-compose –version

Install PostgreSQL Client

sudo apt update
sudo apt install postgresql-client

Check the version of the PostgreSQL client:
psql --version

Run the Database Container:
docker-compose up db

Open new tab and Connect to the PostgreSQL database:
psql -h localhost -U user -d mydb

*** password will not be popos, you need to provide the password you have mentioned in yaml file**********

Create the `users` Table:

```
CREATE TABLE users (
        id SERIAL PRIMARY KEY,
        name VARCHAR(100) NOT NULL,
        email VARCHAR(100) NOT NULL
);
```

```
mydb=# CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL
);
```

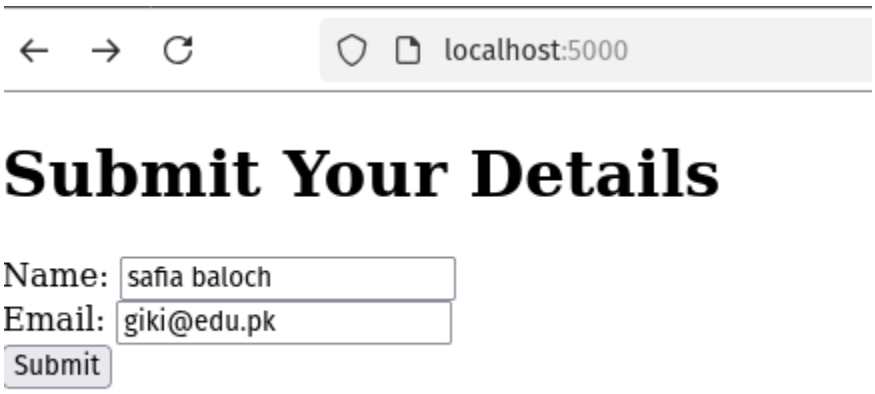
Stop the Database Container:
Ctrl C

Step 4: Run the Application

Build and Run the Application:
docker-compose up --build

1. **Access the Application**:
    - Open `http://localhost:5000` in your browser.

- Fill out the form and submit it. The data will be stored in the PostgreSQL database.



Step 5: Verify the Data

Connect to the Database:
psql -h localhost -U user -d mydb

Query the `users` Table:
SELECT * FROM users;