🚀 Deployment of Python Flask + MySQL on PythonAnywhere

This document provides a **step-by-step guide** to deploy a **Python Flask backend application** with a **MySQL database** on **PythonAnywhere**.

📌 Prerequisites

- Basic understanding of Python & Flask
- PythonAnywhere free account
- Flask project files ready
- Basic MySQL knowledge

1️⃣ Create PythonAnywhere Account

1. Visit 👉 https://www.pythonanywhere.com
2. Click **Sign Up**
3. Create a **Free Account**
4. Login and open the **Dashboard**

2️⃣ Create MySQL Database (PythonAnywhere)

Database Setup Steps

1. Go to the **Databases** tab

2. Create a **MySQL password**

3. Note the database credentials carefully:

    - **DB Host**: `yourusername.mysql.pythonanywhere-services.com`
    - **DB Name**: `yourusername$demo_db`
    - **DB User**: `yourusername`
    - **DB Password**: `********`

⚠️ These details are required in `app.py`.

3️⃣ Create Table in MySQL Database

1. Open **MySQL Console**
2. Execute the following SQL commands:

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    email VARCHAR(50)
);

INSERT INTO users (name, email)
VALUES ('Yogesh', 'yogesh@gmail.com');
```

4️⃣ Upload Flask Project Files

Folder Structure

```
backend-flask/
├─ app.py
└─ requirements.txt
```

Upload Steps

1. Go to **Files** tab on PythonAnywhere
2. Create a new folder named: `backend-flask`
3. Upload the following files into this folder:
   - `app.py`
   - `requirements.txt`

5️⃣ Update `app.py` (IMPORTANT)

Edit `app.py` as shown below:

```
from flask import Flask, jsonify
from flask_cors import CORS
import mysql.connector

app = Flask(__name__)
CORS(app)

db = mysql.connector.connect(
    host="yourusername.mysql.pythonanywhere-services.com",
    user="yourusername",
    password="your_mysql_password",
    database="yourusername$demo_db"
)

@app.route("/users")
def get_users():
    cursor = db.cursor(dictionary=True)
    cursor.execute("SELECT * FROM users")
    data = cursor.fetchall()
    return jsonify(data)
```

⚠ **Do NOT use** `app.run()` - PythonAnywhere runs Flask applications using WSGI, not the Flask development server.

6️⃣ Create Virtual Environment

Virtual Environment Setup Steps

1. Go to **Consoles**
2. Start a **Bash console**
3. Navigate to the project directory:

   ```
   cd backend-flask
   ```

4. Create a virtual environment:

   ```
   python3.10 -m venv venv
   ```

5. Activate the virtual environment:

   ```
   source venv/bin/activate
   ```

6. Install dependencies:

   ```
   pip install -r requirements.txt
   ```

✔ Virtual environment is now ready.

7️⃣ Configure Flask Web App

1. Go to the **Web** tab
2. Click **Add a new web app**
3. Choose:
   - **Manual configuration**
   - **Python 3.10**

✔ Web application entry is created.

8️⃣ Configure WSGI File (MOST IMPORTANT)

WSGI Configuration Steps

1. In the **Web** tab, open the **WSGI configuration file**
2. Remove the existing content
3. Add the following code:

```
import sys

path = '/home/yourusername/backend-flask'
if path not in sys.path:
    sys.path.insert(0, path)

from app import app as application
```

📌 Replace `yourusername` with your PythonAnywhere username.

⚠️ Ignore any "imported but unused" warnings — this is expected in WSGI files.

9️⃣ Set Virtual Environment for Web App

1. In **Web** → **Virtualenv** section

2. Enter the following path:

   ```
   /home/yourusername/backend-flask/venv
   ```

3. Click **Save**

🔄 Reload Web App

1. Click the **Reload** button in the **Web** tab
2. This restarts the web application with updated settings

✅ Test the API

Browser / Postman

```
https://yourusername.pythonanywhere.com/users
```

Expected Output

```
[
  {
    "id": 1,
    "name": "Yogesh",
    "email": "yogesh@gmail.com"
  }
]
```

✔ Flask + MySQL deployment successful!

🎯 Final Deployment Flow

```
React / Postman
      ↓
Flask API (PythonAnywhere)
      ↓
MySQL Database (PythonAnywhere)
```