Here's the **correct full step-by-step order** for setting up and running your HerbPass prototype (with Oracle + Flask). Save this as your permanent reference:

🌿HerbPass – One-Day Prototype Setup Guide

🐓Environment Setup

1. **Create project folder**

```
mkdir herbpass_demo
cd herbpass_demo
```

2. **Create virtual environment**

```
python -m venv venv
venv\Scripts\activate   # Windows
# or
source venv/bin/activate  # Linux/Mac
```

3. **Install dependencies**

```
pip install flask oracledb qrcode pillow
```

🐕Oracle Database Setup

(a) Connect as SYSTEM

```
sqlplus system/dbms123@localhost:1521/XEPDB1
```

(b) Create dedicated user/schema

```
CREATE USER herbpass IDENTIFIED BY dbms123;
GRANT CONNECT, RESOURCE, CREATE TABLE, CREATE SEQUENCE TO herbpass;
ALTER USER herbpass QUOTA UNLIMITED ON USERS;
```

(c) Connect as herbpass

```
sqlplus herbpass/dbms123@localhost:1521/XEPDB1
```

(d) Create tables

```
CREATE TABLE FARMER_BATCH (
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    batch_code VARCHAR2(100) UNIQUE,
    herb_name VARCHAR2(200),
    farmer_name VARCHAR2(200),
    phone VARCHAR2(50),
    gps_lat VARCHAR2(50),
    gps_lng VARCHAR2(50),
    photo_path VARCHAR2(500),
    qr_path VARCHAR2(500),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE LAB_REPORT (
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    batch_id NUMBER REFERENCES FARMER_BATCH(id),
    file_path VARCHAR2(500),
    sha256_hash VARCHAR2(64),
    uploaded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE PHARMA_STATUS (
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    batch_id NUMBER REFERENCES FARMER_BATCH(id),
    status VARCHAR2(100),
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

🦔 Flask App Setup

(a) Create `app.py`

Inside `herbpass_demo/app.py`:

```
from flask import Flask, request, jsonify
import oracledb, qrcode, os, hashlib
```

```
from datetime import datetime

app = Flask(__name__)

# Oracle connection
ORACLE_USER = "herbpass"
ORACLE_PASSWORD = "dbms123"
ORACLE_DSN = "localhost:1521/XEPDB1"

def get_db():
    return oracledb.connect(user=ORACLE_USER, password=ORACLE_PASSWORD,
dsn=ORACLE_DSN)

@app.route("/farmer", methods=["POST"])
def farmer():
    data = request.json
    conn = get_db()
    cur = conn.cursor()

    batch_code = f"HB-{hashlib.sha256(str(datetime.now()).encode()).hexdigest()
[:10].upper()}"
    qr_file = f"static/{batch_code}.png"

    # Generate QR
    img = qrcode.make(batch_code)
    os.makedirs("static", exist_ok=True)
    img.save(qr_file)

    cur.execute("""
        INSERT INTO FARMER_BATCH (batch_code, herb_name, farmer_name, phone,
gps_lat, gps_lng, photo_path, qr_path)
        VALUES (:1, :2, :3, :4, :5, :6, :7, :8)
    """, (
        batch_code,
        data.get("herb_name"),
        data.get("farmer_name"),
        data.get("phone"),
        data.get("gps_lat"),
        data.get("gps_lng"),
        "uploaded_photo.jpg",  # placeholder
        qr_file
    ))

    conn.commit()
    return jsonify({"message": "Batch created", "batch_code": batch_code, "qr":
qr_file})
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

🐚Running the App

1. **Start Flask**

```
python app.py
```

2. **Send test request (Farmer creates batch)**

```
curl -X POST http://127.0.0.1:5000/farmer
-H "Content-Type: application/json"
-d '{"herb_name": "Ashwagandha", "farmer_name": "Ravi", "phone":
"9876543210", "gps_lat": "16.5", "gps_lng": "80.6"}'
```

✅Response:

```
{
  "message": "Batch created",
  "batch_code": "HB-XXXX",
  "qr": "static/HB-XXXX.png"
}
```

1. **Check batch in database**

```
SELECT id, batch_code, herb_name, created_at
FROM farmer_batch
ORDER BY created_at DESC;
```

🐜After Reboot / Next Day Steps

When you open the project again after days: 1. Open folder:

```
cd herbpass_demo
```

2. Activate venv:

```
venv\Scripts\activate   # Windows
# or
source venv/bin/activate # Linux/Mac
```

3. Start Flask:

```
python app.py
```

4. Connect to Oracle as herbpass:

```
sqlplus herbpass/dbms123@localhost:1521/XEPDB1
```

5. Query latest batches:

```
SELECT id, batch_code, herb_name, created_at FROM farmer_batch ORDER BY
created_at DESC;
```

✅That's your full working prototype cycle (Farmer → QR → Oracle → Flask API).