# Role: Backend Developer

# Module: VeriHarvest Backend API (FastAPI + PostgreSQL) - to Store and Manage the Data

from fastapi import FastAPI, HTTPException

from pydantic import BaseModel

import psycopg2

from psycopg2.extras import RealDictCursor

import os

# Initialize FastAPI App

app = FastAPI()

# Database Connection

DATABASE\_URL = "postgresql://user:password@localhost:5432/veriharvest\_db"

conn = psycopg2.connect(DATABASE\_URL, cursor\_factory=RealDictCursor)

cursor = conn.cursor()

# Define Models

class FoodBatch(BaseModel):

batch\_id: int

product\_name: str

supplier: str

freshness\_score: int

blockchain\_verified: bool

status: str

# API Endpoints

@app.post("/add\_batch/")

def add\_batch(batch: FoodBatch):

try:

cursor.execute("INSERT INTO food\_batches (batch\_id, product\_name, supplier, freshness\_score, blockchain\_verified, status) VALUES (%s, %s, %s, %s, %s, %s)",

(batch.batch\_id, batch.product\_name, batch.supplier, batch.freshness\_score, batch.blockchain\_verified, batch.status))

conn.commit()

return {"message": "Batch added successfully"}

except Exception as e:

raise HTTPException(status\_code=500, detail=str(e))

@app.get("/get\_batch/{batch\_id}")

def get\_batch(batch\_id: int):

cursor.execute("SELECT \* FROM food\_batches WHERE batch\_id = %s", (batch\_id,))

batch = cursor.fetchone()

if not batch:

raise HTTPException(status\_code=404, detail="Batch not found")

return batch

@app.get("/get\_all\_batches/")

def get\_all\_batches():

cursor.execute("SELECT \* FROM food\_batches")

batches = cursor.fetchall()

return batches

# Run API Server

if \_\_name\_\_ == "\_\_main\_\_":

import uvicorn

uvicorn.run(app, host="0.0.0.0", port=8000)