from flask import Flask, request, jsonify

import mimetypes

import os

app = Flask(\_\_name\_\_)

# Helper functions

def is\_prime(num):

"""Check if a number is prime."""

if num < 2:

return False

for i in range(2, int(num\*\*0.5) + 1):

if num % i == 0:

return False

return True

def get\_highest\_lowercase(alphabet\_array):

"""Get the highest lowercase alphabet from an array."""

lowercase\_letters = [ch for ch in alphabet\_array if ch.islower()]

return max(lowercase\_letters) if lowercase\_letters else None

# POST endpoint

@app.route('/process', methods=['POST', 'GET'])

def process\_data():

if request.method == 'POST':

try:

# Extract JSON data

data = request.get\_json()

# Required fields from JSON

user\_id = data.get("user\_id")

college\_email = data.get("college\_email")

college\_roll = data.get("college\_roll")

numbers = data.get("numbers", [])

alphabets = data.get("alphabets", [])

file = request.files.get("file")

# Calculations

highest\_lowercase = get\_highest\_lowercase(alphabets)

prime\_found = any(is\_prime(num) for num in numbers)

# File details

if file:

file\_valid = True

mime\_type = mimetypes.guess\_type(file.filename)[0] or "unknown"

file\_size\_kb = len(file.read()) / 1024

file.seek(0) # Reset file pointer after reading

else:

file\_valid = False

mime\_type = None

file\_size\_kb = None

# Construct response

response = {

"status": "success",

"user\_id": user\_id,

"college\_email": college\_email,

"college\_roll": college\_roll,

"numbers": numbers,

"alphabets": alphabets,

"highest\_lowercase": highest\_lowercase,

"prime\_found": prime\_found,

"file": {

"valid": file\_valid,

"mime\_type": mime\_type,

"size\_kb": round(file\_size\_kb, 2) if file\_size\_kb else None,

},

}

return jsonify(response), 200

except Exception as e:

return jsonify({"status": "error", "message": str(e)}), 400

elif request.method == 'GET':

operation\_code = "OP12345" # Example static operation code

return jsonify({"operation\_code": operation\_code}), 200

# Main

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)