**Step 2: Place the AI File**

1. **Folder Structure**: Place the ai\_service.py file in the **project root directory** alongside your index.php and dashboard.php files.

arduino

Copy code

project\_root/

├── ai\_service.py

├── public/

│ ├── index.php

│ ├── dashboard.php

├── assets/

├── config/

├── controllers/

├── models/

1. **Install Required Dependencies**: Open the terminal and install the required Python libraries:

**bash**

**Copy**

***pip install flask numpy joblib scikit-learn***

***pip install mysql-connector-python***

***pip3 install mysql-connector-python***

***python -m pip install --upgrade pip***

***pip install flask\_cors***

***pip3 install flask\_cors***

***pip show flask\_cors***

***python ai\_service.py***

To integrate **AI** into your Environmental Monitoring and Reporting System (EMRS), we can customize and use the provided **Flask AI code** for specific functionalities. Here's a step-by-step guide on what the AI code does, where to place it, and how it will enhance your project.

**AI Integration for EMRS Project**

The AI model can assist in analyzing **environmental data**, predicting trends, and generating insights. Below are the enhancements we can achieve with AI:

1. **Climate Visualization**:
   * Use the AI model to predict and visualize climate trends, such as:
     + Pollution hotspots
     + Deforestation patterns
     + Climate impact over time
   * Visualize the predictions using interactive charts and maps.
   * Input data from reports or uploaded environmental metrics.
2. **Community Engagement**:
   * Analyze data from user-submitted reports and success stories.
   * Identify high-impact areas where community intervention can be helpful.
   * Use AI to summarize success stories or predict areas of community improvement.
3. **Real-Time Alerts Map**:
   * Integrate AI to analyze patterns of reported incidents (e.g., pollution).
   * Use **real-time data** to identify emerging problem areas and alert authorities.
   * Predict future occurrences or areas at high risk.

**Steps to Create and Use the AI File**

**Step 1: Create the AI Code (Flask API)**

Save the provided AI code (with minor enhancements) into a new file named **ai\_service.py**. Here is an updated version of the provided code:

**ai\_service.py**

python

Copy code

# Import required libraries

from flask import Flask, request, jsonify

import numpy as np

import joblib

import os

from sklearn.linear\_model import LogisticRegression

# ---------------------------

# MODEL SETUP AND FILE CHECK

# ---------------------------

MODEL\_FILE = "model.joblib"

if not os.path.exists(MODEL\_FILE):

# Create and save a dummy model

dummy\_model = LogisticRegression()

X\_dummy = [[0, 1], [1, 0], [2, 1], [3, 1]] # Example features

y\_dummy = [0, 1, 1, 0] # Example labels

dummy\_model.fit(X\_dummy, y\_dummy)

joblib.dump(dummy\_model, MODEL\_FILE)

print(f"Model created and saved as {MODEL\_FILE}")

model = joblib.load(MODEL\_FILE)

# ---------------------------

# FLASK APP SETUP

# ---------------------------

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

@app.route("/")

def home():

return jsonify({"message": "AI Service for EMRS is running!"})

@app.route("/predict", methods=["POST"])

def predict():

"""

API endpoint for predictions.

"""

try:

data = request.json

if not data or "features" not in data:

return jsonify({"error": "Invalid input data. Provide 'features' key."}), 400

features = np.array(data["features"]).reshape(1, -1)

prediction = model.predict(features)

return jsonify({"prediction": prediction.tolist()})

except Exception as e:

return jsonify({"error": str(e)}), 500

# ---------------------------

# RUN FLASK SERVER

# ---------------------------

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

print("Starting AI Service...")

app.run(debug=True, port=8000)

**Step 2: Place the AI File**

1. **Folder Structure**: Place the ai\_service.py file in the **project root directory** alongside your index.php and dashboard.php files.

arduino

Copy code

project\_root/

├── ai\_service.py

├── public/

│ ├── index.php

│ ├── dashboard.php

├── assets/

├── config/

├── controllers/

├── models/

1. **Install Required Dependencies**: Open the terminal and install the required Python libraries:

bash

Copy code

pip install flask numpy joblib scikit-learn

**Step 3: Run the AI File**

1. Start the Flask server:
   * Open the terminal or command prompt.
   * Navigate to the project root directory where ai\_service.py is located.
   * Run:

bash

Copy code

python ai\_service.py

1. You should see the message:

csharp

Copy code

Starting AI Service...

\* Running on http://127.0.0.1:8000/