**Microplastics Platform Folder Structure Explained**

Here's a **one-line explanation** for each folder/file:

| **File/Folder** | **Purpose** |
| --- | --- |
| **app.py** | Main Streamlit frontend - user interface with all pages |
| **api.py** | FastAPI backend - API endpoints for data, uploads, predictions |
| **requirements.txt** | Python dependencies list - packages needed to run the app |
| **.env.example** | Environment variables template - API keys, database settings |
| **.gitignore** | Git ignore file - excludes temp files, uploads, database |

| **models/** | **Machine Learning models and prediction algorithms** |
| --- | --- |
| \_\_init\_\_.py | Makes folder a Python package |
| **predictor.py** | AI prediction model for microplastic trends |

| **utils/** | **Helper functions for data processing** |
| --- | --- |
| \_\_init\_\_.py | Makes folder a Python package |
| **data\_loader.py** | Loads and processes microplastics data |
| **gps\_extractor.py** | Extracts GPS coordinates from photo EXIF |
| **pdf\_generator.py** | Creates PDF reports for download |
| **alerts.py** | Email/SMS alert system for hotspots |

| **static/** | **Static assets (CSS, images)** |
| --- | --- |
| **style.css** | Custom styling for the web interface |
| **logo.png** | Platform logo/image placeholder |

| **data/** | **Sample datasets and reference files** |
| --- | --- |
| **sample\_data.csv** | Example microplastics pollution data |
| **additives.json** | Polymer additives toxicity information |
| **citizen\_uploads.csv** | Citizen science contribution records |
| **chemical\_library.csv** | Your polymer health impact library |

| **uploads/** | **User-uploaded files (ignored by Git)** |
| --- | --- |
| *(empty)* | Photo uploads from citizen science |

| **database/** | **SQLite database files** |
| --- | --- |
| **microplastics.db** | Main application database |

| **README.md** | **Project documentation and setup instructions** |
| --- | --- |
| **.env.example** | Environment configuration template |
| **requirements.txt** | Package dependencies |

**Quick Setup Summary**

text

# 1. Install packages

pip install -r requirements.txt

# 2. Start API (Terminal 1)

uvicorn api:app --reload --port 8000

# 3. Start Frontend (Terminal 2)

streamlit run app.py --server.port 8501

# 4. Open browser: http://localhost:8501

**Core Flow:** Frontend (app.py) ↔ API (api.py) ↔ Data (data/) ↔ Database (database/)

This structure separates **UI** (app.py), **backend logic** (api.py/utils), **ML** (models), and **data** (data/) for easy maintenance and scaling.