Milestone 8: Logging and Monitoring

# Objective

To implement robust logging for the PM2.5 prediction API, enabling observability, debugging, and traceability of requests, predictions, and security-related events.

# Implementation

- Created `core/logger.py` to configure application-wide logging.

- Integrated logging into:

* • API key verification
* • Prediction request lifecycle
* • Error handling during model execution

# Configuration

- Log output is written to `logs/api.log`.

- Logging configuration (log directory, app name) is centralized in `config.py`.

# Logged Information

- API key authentication success/failure

- Incoming request data

- Successful prediction values

- Model errors (with stack traces)

# Log Directory Location

The `logs/` directory is located in the project root (`dscp/logs`). It is automatically created at runtime if it does not exist. This keeps logs separate from source code and suitable for mounting or syncing in containerized or cloud environments.

# Example Log Output

2024-05-21 15:00:22 | INFO | New /predict request received.  
2024-05-21 15:00:22 | WARNING | Unauthorized access attempt with API key: seckey

2024-05-21 15:00:22 | INFO | API key verified successfully  
2024-05-21 15:00:22 | INFO | Prediction completed successfully. PM2.5 = 8.13

# Conclusion

Logging has been fully integrated into the API lifecycle. This enables proactive monitoring, auditability, and simplifies debugging when needed. All logs are written to both the terminal and the file `logs/api.log`.