PYTHON

Assignment

1. **Basic Logging Setup: Write a program that configures a basic logging setup. Log a message at each log level (DEBUG, INFO, WARNING, ERROR, CRITICAL).**
2. **Logging to a File: Modify the above program to log all messages to a file instead of the console.**
3. **Log Formatting: Write a script that uses a custom log format (include the date, time, log level, and message) in the log messages.**
4. **Logging Variables: Create a Python program that logs variables or function outputs. Make sure to log input and output values at each function call.**
5. **Logging Exceptions: Write a Python script that handles exceptions and logs them to a file using the logging.exception() method.**
6. **Rotating Log Files: Implement rotating log files, where after the log file exceeds a certain size, it rotates into a new file. Use the RotatingFileHandler from the logging module.**
7. **Logging Different Modules: Create a program with two different modules (e.g., module1.py and module2.py). Configure logging such that each module logs messages to the same log file but with a distinct prefix or identifier.**
8. **Log Filter: Write a program that implements a log filter to allow only messages with a certain log level or above to be logged (e.g., only WARNING and ERROR messages should appear).**
9. **Logging Configuration via File: Configure logging via an external configuration file (e.g., a .conf or .ini file) and set up different loggers for different parts of the program.**
10. **Timed Rotating Log Files: Implement timed rotating log files, where the log file rotates after a certain time interval (e.g., every midnight). Use the TimedRotatingFileHandler from the logging module.**

Tasks to Explore

1. Task: Web Scraping Application

Scenario: You are scraping data from a website and need to log when the scraping starts, if any pages fail to load, and the number of data entries successfully scraped.

Logging Tasks:

- **INFO**: Log when the web scraping starts and ends.
- **ERROR**: Log if a page fails to load or the server returns an error (e.g., 404, 500).
- **WARNING**: Log if some expected data (e.g., product price) is missing on a page.
- **DEBUG**: Log the number of entries scraped from each page for detailed tracking.

2. API Request Handling

Scenario: You're building a service that interacts with a third-party API. You need to log when requests are made, if the response times out, or if there are any errors in the API response.

Logging Tasks:

- **INFO**: Log each API request being made (with URL and parameters).
- **ERROR**: Log any timeout or connection errors.
- **WARNING**: Log any unusual or unexpected status codes (like 3xx redirects or 4xx errors).
- **DEBUG**: Log the response time for each request to track performance.

3. User Authentication System

Scenario: You're developing a user authentication system for a web app. You need to log login attempts, successful logins, and failed login attempts due to incorrect credentials or locked accounts.

Logging Tasks:

- **INFO**: Log successful login attempts with the user's ID or username.
- **WARNING**: Log failed login attempts due to incorrect credentials.
- **ERROR**: Log failed login attempts if the account is locked or suspended.
- **CRITICAL**: Log multiple consecutive failed login attempts (potential security risk)

4. Database Backup Script

Scenario: You're running an automated database backup script. You need to log the start and end of the backup, any errors during the backup process, and ensure the size of the backup file is logged for verification.

Logging Tasks:

- **INFO**: Log when the backup process starts and ends.

- **ERROR**: Log any issues that occur during the backup (e.g., unable to connect to the database).
- **WARNING**: Log if the backup file is smaller than expected (indicating potential data loss).
- **DEBUG**: Log the time taken for the backup process and the size of the backup file

5. Email Notification System

Scenario: You are developing an automated email notification system that sends out alerts for various events. You need to log when emails are sent, if an email fails to send, or if there are missing recipients.

Logging Tasks:

- **INFO**: Log when an email is successfully sent (to which recipient).
- **ERROR**: Log if the email fails to send due to SMTP or network errors.
- **WARNING**: Log if any expected recipient email address is missing.
- **DEBUG**: Log the content of the email for debugging purposes (optional).