

# ■ Deep Dive — Week 2: Automation & File Handling

Elite Bootcamp 2025–2026 · System Automation Engineer Path

## ■ ■ Setup Instructions (Debian 12)

- Update system: `sudo apt update && sudo apt install python3 python3-pip python3-venv -y`
- Create and activate virtual environment: `python3 -m venv ~/bootcamp_env && source ~/bootcamp_env/bin/activate`
- Install libraries: `pip install tqdm psutil pytest`
- Create Week 2 workspace: `mkdir -p ~/EliteBootcamp/Week2 && cd ~/EliteBootcamp/Week2`

## ■ Section 1 — File & Directory Handling

Learn to manipulate files and directories safely with `pathlib`, `os`, and `shutil`. Focus on modular, reusable design.

Example:

```
from pathlib import Path import shutil src = Path.home() / 'Downloads' dst = Path.home() / 'Backup' dst.mkdir(exist_ok=True) for file in src.glob('*.txt'): shutil.copy(file, dst / file.name)
```

■ Deep Dive Task: Write a function that deletes temporary files older than 7 days in a directory.

## ■ Section 2 — Working with JSON & CSV

Handle structured data using `json` and `csv` modules. Automate conversions between formats.

Example:

```
import json, csv with open('data.json') as f: data = json.load(f) with open('data.csv', 'w', newline='') as f: writer = csv.DictWriter(f, fieldnames=data[0].keys()) writer.writeheader() writer.writerows(data)
```

■ Deep Dive Task: Write a converter that takes any JSON list and exports it as a formatted CSV.

## ■ Section 3 — Regex for Log Analysis

Use regular expressions to extract data from unstructured logs. Focus on timestamps, IP addresses, and errors.

Example:

```
import re, pathlib log = pathlib.Path('server.log').read_text() pattern = r'(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}).*(ERROR|WARNING)' matches = re.findall(pattern, log) print(matches[:5])
```

■ Engineer Note: Regex-based log parsing is widely used in log monitoring and ETL systems.

## ■ Section 4 — Modular Design

Split functionality into smaller reusable modules. Create a `utils/` folder for file operations.

Example structure:

```
EliteBootcamp/Week2/ ■■■ bulk_tool.py ■■■ utils/ ■ ■■■ __init__.py ■ ■■■  
file_ops.py ■■■ logs/ ■ ■■■ bulk_tool.log ■■■ data/ ■■■ sample_logs.txt ■■■  
summary.json
```

■ Deep Dive Task: Refactor your Week 1 CLI tool to store helper functions in `utils/` modules.

## ■ Section 5 — Logging & Error Handling

Add professional logging and rotating file handlers for reliability.

Example:

```
import logging from logging.handlers import RotatingFileHandler handler =  
RotatingFileHandler('logs/bulk_tool.log', maxBytes=500000, backupCount=3)  
logging.basicConfig(level=logging.INFO, handlers=[handler], format='%(asctime)s]  
%(levelname)s - %(message)s') logging.info('File automation started')
```

■ Tip: Always log errors using `logging.exception()` instead of `print()` for better debugging context.

## ■ Mini Project — Bulk File Renamer & Log Analyzer

Integrate `argparse`, `pathlib`, `regex`, and logging into one CLI tool. Use dry-run mode to preview changes.

Example Run:

```
$ python3 bulk_tool.py --rename --analyze logs/ [INFO] 48 files renamed, 3  
skipped. [INFO] Extracted 120 error lines → logs_summary.json
```

## ■ Deliverables

- `bulk_tool.py` with `argparse` + `regex` + logging
- `utils/file_ops.py` module
- `logs/bulk_tool.log` auto-generated
- `README.md` with usage examples
- Commit message: 'Week 2 – Deep Dive Completed'