# Summary of today's edits

## 1. Web-Compliance folder

### Created

- web-compliance/ folder with a dark-themed web interface
- api_server.py — FastAPI server (port 8001) for compliance checking
- compliance_checker.py — compliance checking logic
- index.html — dark-themed web interface with 7 tabs:
  - Overview, Parameters, Compliance, Rooms, Luminaires, Scenes, Raw Data
- Documentation:
  - README.md, SETUP.md, API_USAGE.md
  - CLASSIFICATION_GUIDE.md, CLASSIFICATION_SUMMARY.md
- requirements.txt — Python dependencies
- start_server.bat / start_server.sh — startup scripts

### Features

- Dark theme UI
- Displays all extracted parameters
- Displays all compliance results
- Old version preserved (no files deleted)

---

## 2. LuxSCale interface folder

### Created

- LuxSCale interface/ folder with a standards-first approach
- index.html — redesigned two-step interface:
  - Step 1: Enter room dimensions + type → get standard requirements
  - Step 2: Configure luminaires based on requirements → generate report
- api_server.py — FastAPI server (port 8001) with new endpoint:
  - GET /get-standard-requirements — returns standard requirements for room type
- standards_lookup.py — new module to look up standards by room type
- report_generator.py — generates Dialux-like reports
- compliance_integration.py — integrates compliance checking
- Documentation:
  - README.md, DOCUMENTATION.md, API_DOCUMENTATION.md
  - QUICK_START.md, CONCEPT.md

- start_service.bat — starts both API and web server

### Concept

- Standards-first: user sees requirements before configuring luminaires
- Minimal questions: only essential inputs
- Two-step workflow: requirements → configuration → report

### Changes

- Removed "💡 Luminaire Information" section header
- Integrated luminaire fields into Room Information section
- Added step indicator showing progress
- Added requirements display card showing standard parameters

---

## 3. Bug fixes and improvements

### Fixed aliases.json path issue

- Updated compliance_checker.py to correctly locate aliases.json
- Added multiple path resolution methods
- Now loads from: E:\AI_projects\final_comparator\report_export\extractors\aliases.json

### Fixed utilisation_profile extraction

- Enhanced inference logic in compliance_checker.py
- Now infers from:
  1. Room data
  2. Scene names (factory, industrial keywords)
  3. Room names (office, corridor, storage keywords)
  4. Project context
- Improved standard matching accuracy

---

## 4. Documentation created

### Web-Compliance

- CLASSIFICATION_GUIDE.md — how reports are classified
- CLASSIFICATION_SUMMARY.md — quick reference
- API_USAGE.md — API usage guide
- SETUP.md — setup instructions

### LuxSCale Interface

- CONCEPT.md — concept and workflow explanation
- DOCUMENTATION.md — code documentation
- API_DOCUMENTATION.md — API reference
- QUICK_START.md — quick start guide

---

## 5. File structure

### New folders

```
final_comparator/
├── web-compliance/        # NEW - Dark-themed compliance checker
│   ├── api_server.py
│   ├── compliance_checker.py
│   ├── index.html
│   └── [documentation files]
│
└── LuxSCale interface/     # NEW - Standards-first design tool
    ├── api_server.py
    ├── standards_lookup.py
    ├── report_generator.py
    ├── compliance_integration.py
    ├── index.html
    └── [documentation files]
```

### Existing files

- All original files preserved
- No deletions
- Both systems use existing compliance checker and standards

---

## 6. Key features added

### Web-Compliance

- Dark theme UI
- Complete parameter display
- Complete compliance display
- 7-tab interface for different views
- Standalone system

### LuxSCale Interface

- Standards-first approach
- Two-step workflow
- Requirements display before configuration
- Minimal questions
- Automatic compliance checking
- Batch file for easy startup

---

## 7. API endpoints

### Web-Compliance (Port 8000)

- POST /check-compliance-detailed — detailed compliance check
- GET /health — health check
- GET /standards-info — standards information

### LuxSCale Interface (Port 8001)

- GET /get-standard-requirements — get requirements for room type
- POST /generate-report — generate lighting design report
- GET /download-report/{id} — download generated report
- GET /health — health check

---

## 8. Workflow improvements

### Before

- Single-step form with all fields
- User had to know all parameters
- No visibility into standard requirements

### After

- Two-step process
- User sees standard requirements first
- Guided configuration based on requirements
- Clearer workflow

---

## Summary

Today's work:

1. Created two new systems (web-compliance and LuxSCale)
2. Fixed path and extraction issues
3. Added documentation
4. Implemented a standards-first design approach
5. Preserved all existing files
6. Added batch files for easy startup

Both systems are ready to use and integrate with the existing compliance checker and standards structure.