⭐ ELIAS – Day 3 (Thursday) Official Work Plan

(8 PM – 10 PM Work Session)

Theme of the day: Alignment + Light Coding + Preparation for Weekend Development

🧩 Team Roles (Same as Before — Reference Only)

Member	Role	Responsibility
M S	Project Lead & Integrator	Handles integrations, folder structure, system flow, routing
S S	NLP & Intent Logic Developer	Creates & tests regex-based intent patterns
A K	Vision & Authentication Engineer	Handles OpenCV, face-recognition, camera systems
R A	Plugin/Actions Developer	Builds modular action plugins
M A	Documentation & QA	Manages explainers, logs, documentation quality

📅 Thursday Tasks (Final Clean Version)

🟦 1. M S — Project Lead & Integrator

Tasks for Today

✔ Update app.py routing structure to include **new intents** added by S S
✔ Add placeholder routing lines, for example:

```
elif intent == "play_music":
    print("-> would run play music plugin here")
```

✔ Add commented import for vision module (to prepare for Friday):

```
# from vision import detect_face
```

✔ Create **Weekend Development Checklist** inside docs/ folder
(What will be built Friday–Sunday)

✔ Coordinate with all members for correct file placement & folder structure

Expected Output

- Updated routing function in app.py
- System aligned so all modules connect cleanly
- Weekend tasks planned in a structured checklist
- Ready for plugin + vision integration on Friday

🟩 2. S S — NLP Developer

Tasks for Today

✔ Test all existing intent patterns inside app.py

✔ Improve pattern stability (add boundaries, fix cases, refine matching)

✔ Add **one new intent** (very small), e.g.:

- “what is the time”
- “tell date”
- “open notes”

✔ Document final list of working intents

Expected Output

- Clean & stable regex patterns
- New intent added and tested
- NLP system fully ready for plugin binding on Friday

🟪 3. A K — Vision & Authentication Engineer

Tasks for Today

✔ Create models/ folder for storing face-recognition data

✔ Place or download **haarcascade_frontalface_default.xml** (or save link for tomorrow)

✔ Write **vision explanation file** (what detect_face() will do)

✔ Re-test webcam with a simple frame capture (optional)

Expected Output

- models/ folder ready
- Haar cascade path confirmed
- vision explanation documented
- System ready for real face detection on Friday

🟧 4. R A — Plugin / Actions Module Developer

Tasks for Today

✔ Review the placeholder plugin folder
✔ Add comments inside each plugin file describing:

- expected input
- expected output
- brief description of purpose

✔ Plan **3 plugins** for Friday coding
(e.g., open_website, youtube_search, create_note)

✔ Write a short plan/outline and send it to the team

Expected Output

• Plugin skeleton clearly documented
• 3 plugins planned for coding
• Ready for action-module development this weekend

🟨 5. M A — Documentation & QA

Tasks for Today

✔ Create a new documentation file:

docs/README_DAY3.md

✔ Write:

- Summary of all Day 3 tasks
- Updated architecture flow (NLP → App → Plugin → Vision)
- Notes from M S, S S, A K, R A

✔ Check project folders for consistency
✔ Prepare tomorrow's documentation template

Expected Output

- Day 3 documentation completed
- Architecture section updated
- Consistent documentation structure

🎯 **Thursday End-of-Day Expected State (For Entire Team)**

By 10 PM tonight, the team should have:

✅ Updated app.py routing by M S

✅ NLP patterns cleaned, expanded, and tested by S S

✅ Vision models folder prepared by A K

✅ Plugin file documentation created by R A

✅ Day 3 documentation ready by M A

✅ System structure fully ready for **Friday–Sunday implementation**

This means the weekend can now be used efficiently to build:

- Real face detection
- Core plugins
- Streamlit UI
- System integration
- Authentication logic
- Full demo flow