⭐ ELIAS – Day 15 (Tuesday) Official Work Plan

(System Polishing, Stress Testing & Minor Features Sprint)

Workload: 2.5 – 3 hours per member

Goal: Make ELIAS stable, smooth, and presentation-ready

🧩 Team Roles (Reference)

(Same roles as before — consistency maintained)

Member	Role	Focus Today
M S	Project Lead	UI polishing + small features
S S	NLP Lead	Accuracy polishing
A K	Vision Engineer	Reliability & lighting handling
R A	Plugin Dev	Polishing all plugin behaviors
M A	Documentation & QA	Report writing + Test validation

🎯 Day 15 Theme:

“Make it feel professional.”

Everything works — now it must *feel* finished.

🟦 1. M S — UI Polishing & Minor Integrations

🔵 Task 1 — Add “History Log Viewer” in UI

In Streamlit UI:

- Show last 10 log lines live
- Book-like formatting
- Add filter (info / error / plugin)

🔵 Task 2 — Add Voice Input Button (Prototype)

Not full recognition — just integration:

- Add a microphone button

- When clicked → record 3 seconds → save file
- Later team will use this for voice recognition
  (For now: indicator only)

🔵 Task 3 — Add “Quick Actions” Buttons

On UI add:

- “Open Google”
- “Open YouTube”
- “Create Note”
- “Tell a Joke”

These should directly call route_intent().

Deliverables:

✓ Log viewer
✓ Mic prototype
✓ Quick action buttons

🟩 2. S S — NLP Polishing & Mini-Model Rules

🟢 Task 1 — Improve Sentence Ambiguity Handling

Examples to handle:

- “google something for me” → interpret as search
- “go youtube and search xyz” → two intents → run second

Implement simple **multi-intent detection** (even partial).

🟢 Task 2 — Add synonyms dictionary

Include synonyms:

- open → launch, access, visit
- search → look up, find, get info on
- make note → write down, jot

Synonym mapping → more robust detection.

🟢 Task 3 — Add “confidence score” per intent

Very simple formula:

- Weighted overlap
- Regex match boost
- Token similarity

Return:

```
{
 "intent": "open_website",
 "confidence": 0.86,
 "entities": {...}
}
```

Deliverables:

✓ Multi-intent

✓ Synonyms

✓ Confidence-based parser

🟪 3. A K — Vision Reliability Enhancement

🟣 Task 1 — Improve Low-Light Detection

Add stability logic:

- If brightness < threshold → auto-enhance image
- Apply histogram equalization
- Use blurred+sharpened composite

🟣 Task 2 — Add 3-step confirmation

Prevent false positives:

Face must match *again* in:

1. Frame 1
2. Frame 2

3. Frame 3 (within 1.2 seconds)

Only then → authenticated.

🟣 Task 3 — Add vision status in UI

Provide:

- “Face detected”
- “Authenticating...”
- “Authenticated (X ms)”
- “Lighting too low”

Deliverables:

✓ More stable recognition
✓ 3-frame validation
✓ UI status integrated

🟧 4. R A — Plugin Refinement & Uniform Standardization

🟠 Task 1 — Standardize All Plugin Return Formats

Every plugin should return:

```
{
 "ok": True/False,
 "message": "...",
 "data": {...}
}
```

🟠 Task 2 — Add “search_web” plugin

Use:

```
import webbrowser
webbrowser.open(f"https://www.google.com/search?q={query}")
```

Handles commands like:

- “search web for python”

- “google AI notes”

🟠 Task 3 — Add “time_date” plugin

Returns:

- Current time
- Current date
- Day of week

Deliverables:

✓ Unified plugin output
✓ Web search plugin
✓ Date/time plugin

🟨 5. M A — Documentation, QA & Report Progress

🟡 Task 1 — Final Report Section Writing

Add these chapters:

- “UI & UX Design Decisions”
- “Vision Authentication Pipeline”
- “Hybrid NLP Model Explanation”
- “Plugin Architecture & Extensibility”
- “Testing Strategy”

🟡 Task 2 — Create a Full Test Matrix

Include:

- Functional tests
- UI tests
- NLP accuracy tests
- Vision reliability tests
- Plugin stability tests

🟡 Task 3 — Prepare Demo Script (v1)

Include:

1. Greeting → ELIAS replies
2. Open website → works
3. Create note → works
4. Search YouTube → works
5. Play music → works
6. Tell a joke
7. Vision authentication check
8. Closing summary

Deliverables:

✓ Major report sections
✓ Complete test matrix
✓ Demo flow ready

🏁 Day 15 End-of-Day Outcome

By tonight:

✓ UI becomes professional
✓ NLP becomes smarter
✓ Vision becomes reliable
✓ Plugins behave uniformly
✓ Documentation becomes structured
✓ Full test coverage exists

Meaning:

🎉 **The development phase is DONE.**
Next = polish, rehearse, finalize report, finalize demo.