Excellent.
Below is the **complete, detailed execution document for Person 2**, written to the **same standard, structure, and rigor** as Person 1's document.

This document is:

- ✅ Fully aligned with `context.md`
- ✅ Consistent with Person 1 and Person 3 responsibilities
- ✅ Free of overlap or role creep
- ✅ Detailed enough to execute independently
- ✅ Explicit about inputs, outputs, checks, and integration points

---

📄 Person 2 Execution Document

Role: NLP Extraction & Ontology Mapping Engineer

---

1. Role Overview

Primary Responsibility

You are responsible for converting **free-text student input** into **structured, ontology-aligned symbolic evidence**.

Your work forms the **grounding layer** between:

- natural language (what the student says)
- symbolic reasoning (what the system reasons about)

You do **not** perform reasoning, inference, ranking, or explanation generation.
You do **not** define ontology structure or rules.

Your output must be **clean, structured, confidence-aware**, and **ontology-compatible**.

---

2. What You Are Building (High-Level)

You will deliver:

1. An **NLP preprocessing pipeline**
2. An **emotion, symptom, and trigger extraction module**
3. A **semantic mapping layer** that maps text → ontology URIs
4. **Extraction confidence scores**
5. Session-aware **evidence packaging** for the knowledge graph
6. Documentation and test cases

Your work ensures that symbolic reasoning operates on **accurate and meaningful evidence**.

---

3. Inputs You Need Before Starting

Required Inputs

- Final `context.md`
- Ontology files (`.owl` / `.ttl`) from **Person 1**
- List of ontology URIs and class/property semantics

Helpful Datasets to use

- Dataset samples (Figshare, Kaggle)
- Initial keyword lists (if any)

You should **not** start ontology mapping until Person 1 provides a **stable ontology draft**.

---

4. Phase-by-Phase Task Plan

PHASE 1 — NLP Foundations

Objective

Prepare raw text for reliable extraction without over-engineering.

Tasks

1. Text Preprocessing

Implement:

- tokenization
- lowercasing
- punctuation handling
- lemmatization
- stopword removal (carefully, do not remove negations)

Use **NLTK**.

---

2. Sentence Segmentation

- Split multi-sentence input
- Preserve sentence boundaries for evidence granularity

Checks

- Negations are preserved ("not sleeping" ≠ "sleeping")
- No aggressive cleaning that removes meaning

PHASE 2 — Emotion, Symptom & Trigger Extraction

Objective

Identify candidate concepts mentioned in text.

Tasks

3. Keyword & Phrase Dictionaries

Create dictionaries for:

- emotions (stress, anxious, worried, etc.)
- symptoms (insomnia, fatigue, panic, etc.)
- triggers (exam, workload, deadline, etc.)

Use dataset-informed vocabulary.

---

4. Pattern-Based Extraction

Implement simple patterns:

- “can’t sleep”
- “feeling anxious”
- “stressed because of exams”

Avoid deep ML models.

---

5. Unified Extraction API

Implement:

```
def extract_all(text):
    return {
        "emotions": [...],
        "symptoms": [...],
        "triggers": [...],
        "raw_phrases": [...],
        "confidence": {...}
    }
```

Checks

- Each extracted item has:
  - surface phrase
  - category (emotion/symptom/trigger)
- No ontology URIs yet (next phase)

PHASE 3 — Semantic Mapping to Ontology

Objective

Map extracted phrases to **ontology concepts** reliably.

Tasks

6. URI Mapping File

Create a mapping:

- phrase → ontology URI
- class type

---

7. Semantic Similarity Fallback

Use **sentence-transformers**:

- encode phrase
- compute similarity with ontology labels/comments
- apply threshold (e.g., ≥ 0.7)

---

8. Ontology-Aligned Output

Each extracted concept must output:

```
{
  "uri": "mh:Insomnia",
  "label": "Insomnia",
  "confidence": 0.92,
  "method": "keyword|semantic",
  "turnIndex": 3
}
```

Checks

- No mapping to wrong class types
- Semantic mapping is explainable
- Thresholds documented

PHASE 4 — Confidence & Persistence Support

Objective

Provide evidence quality signals for downstream reasoning.

Tasks

9. Confidence Assignment

Combine:

- keyword certainty
- semantic similarity score
- linguistic strength

Normalize confidence to [0,1].

---

10. Persistence Metadata

Attach:

- timestamp
- turn index
- repetition count (if seen before)

This supports **continuous context reasoning**.

Checks

- Confidence reflects extraction quality, not inference
- Persistence does not imply causality

PHASE 5 — Session Evidence Packaging

Objective

Deliver clean evidence to the knowledge graph layer.

Tasks

11. Evidence Packaging Format

Package outputs in a format expected by `graph_manager.py`:

```
{
  "session_id": "...",
```

```
  "evidence": [...],
  "confidence_summary": {...}
}
```

---

12. KG Insertion Contract

Coordinate with **Person 1** to ensure:

- correct property usage
- correct class assignment

Checks

- No reasoning performed
- No ranking or filtering based on severity

PHASE 6 — Testing & Validation

Objective

Ensure robustness before integration.

Tasks

13. Test on Sample Inputs

- 20–25 realistic student messages
- mixed emotions and symptoms
- ambiguous phrasing

---

14. Error Analysis

- false positives
- missed concepts
- mapping errors

Iteratively refine dictionaries and thresholds.

---

15. Documentation

Document:

- extraction logic
- confidence logic
- known limitations

---

5. Integration With Other Members

What You Provide

To **Person 1**:

- missing vocabulary feedback
- synonym suggestions

To **Person 3**:

- clean symbolic evidence
- confidence and persistence metadata

What You Need From Others

From **Person 1**:

- final ontology URIs
- property semantics

From **Person 3**:

- expected evidence fields for rules

---

6. Project Flow — Your Involvement

You are active in:

1. Ontology grounding phase
2. Evidence pipeline construction
3. Integration & testing
4. Final case-study preparation

Your work is **iterative** — refinement is expected.

---

7. Final Deliverables Checklist

- [ ] NLP preprocessing module
- [ ] Extraction pipeline
- [ ] Ontology mapping logic
- [ ] Confidence & persistence metadata
- [ ] Test cases + documentation

---

8. Success Criteria

Your work is successful if:

- Free text is reliably grounded to ontology
- Confidence values are meaningful
- Reasoning engine receives clean evidence
- Explanations remain faithful to user input

---

9. Important Boundaries (Do Not Cross)

❌ Do not write SWRL rules
❌ Do not infer mental states
❌ Do not rank interventions
❌ Do not implement safety escalation

Your role ends at **structured evidence generation**.

Final Note

If this layer is noisy, **everything downstream fails**.
If this layer is clean, the system becomes explainable, trustworthy, and defensible.
