📁 PROJECT_STRUCTURE.md

Purpose of This Document

This document defines the **official directory structure and file organization** for the project:

> **Ontology-Driven Mental Health Knowledge Graph with Explainable Causal Reasoning**

Its goals are to:

- eliminate ambiguity about file placement
- ensure smooth integration between team members
- keep ontology, NLP, reasoning, and logging cleanly separated
- support traceability, auditing, and evaluation

This structure is **mandatory** for all contributors.

Top-Level Directory Structure

```
/project-root
│
├── ontology/
├── data/
├── nlp/
├── session/
├── reasoning/
├── backend/
├── logs/
├── tests/
├── docs/
└── README.md
```

Each directory has a **single, well-defined responsibility**.

---

1. ontology/

Purpose

Contains all **formal knowledge representations**.

Ownership

Primary: **Person 1**
Read-only usage by others.

Structure

```
ontology/
├── mental_health.owl
├── mental_health.ttl
├── base_graph.ttl
└── ontology_versions/
    └── mental_health_v1.owl
```

Contents

- OWL ontology (Protégé-authored)
- RDF/Turtle exports
- Base (non-session) knowledge graph
- Versioned backups

Rules

- No runtime session data here
- No reasoning outputs stored here
- Ontology edits must be versioned

---

2. data/

Purpose

Stores **datasets and dataset-derived resources**.

Ownership

Shared, coordinated by **Person 1**

Structure

```
data/
├── datasets/
|    ├── emotion_lexicon.csv
|    ├── symptom_mapping.csv
|    └── trigger_phrases.csv
|
├── annotations/
|    ├── causal_strengths.json
|    └── ontology_annotations.ttl
|
└── README.md
```

Contents

- Curated datasets
- Lexicons
- Statistical annotations (non-decision-making)
- Mapping tables

Rules

- Datasets **do not perform inference**
- Statistical values are **supportive only**
- No ML model training here

---

3. nlp/

Purpose

All **language processing and semantic extraction** logic.

Ownership

Primary: **Person 2**

Structure

```
nlp/
├── extractor.py
├── concept_mapper.py
├── confidence_estimator.py
├── patterns/
│   ├── emotion_patterns.json
│   ├── symptom_patterns.json
│   └── trigger_patterns.json
└── README.md
```

Responsibilities

- Extract emotions, symptoms, triggers
- Map text to ontology concepts
- Produce structured evidence objects
- Assign extraction confidence

Rules

- No reasoning logic here
- No ontology modification
- Output must conform to `EVIDENCE_SCHEMA.md`

---

4. session/

Purpose

Manages **continuous conversational context**.

Ownership

Shared (Person 2 → input, Person 3 → consumption)

Structure

```
session/
├── graph_manager.py
├── context_store.py
└── README.md
```

Responsibilities

- Maintain per-user session graphs
- Insert extracted evidence as RDF
- Track temporal persistence
- Reset sessions safely

---

5. reasoning/

Purpose

All **symbolic reasoning, causal inference, and explanations**.

Ownership

Primary: **Person 3**

Structure

```
reasoning/
├── rules/
│   ├── swrl_rules.owl
│   └── rule_catalog.md
│
├── sparql/
│   ├── materialization.sparql
│   ├── explanation_queries.sparql
│   └── escalation_checks.sparql
│
├── explainer.py
├── ranking.py
└── README.md
```

Responsibilities

- SWRL rule definitions

- SPARQL materialization
- Risk inference
- Intervention mapping
- Causal explanation generation
- Ranking & confidence aggregation
- Safety escalation logic

Rules

- SWRL = inference
- SPARQL = graph operations
- No NLP logic here

---

6. backend/

Purpose

Local API layer and orchestration.

Ownership

Shared, already implemented

Structure

```
backend/
├── app.py
├── routes/
│   ├── chat.py
│   └── health.py
└── README.md
```

Responsibilities

- Accept chat messages
- Call NLP → session → reasoning pipeline
- Return final response + explanation

---

7. logs/

Purpose

Auditability and safety tracking.

Ownership

Primary: **Person 3**

Structure

```
logs/
├── audit/
│   ├── escalation_events.log
│   └── rule_traces.log
└── README.md
```

Contents

- Safety escalation events
- Rule firing traces
- Explanation provenance

---

8. tests/

Purpose

Verification and validation.

Ownership

Shared

Structure

```
tests/
├── ontology_tests.py
├── nlp_tests.py
├── reasoning_tests.py
└── integration_tests.py
```

Responsibilities

- Validate ontology consistency
- Test extraction accuracy
- Test reasoning correctness
- Test end-to-end flow

---

9. docs/

Purpose

All **project documentation**.

Structure

```
docs/
├── context.md
├── PROJECT_STRUCTURE.md
├── INTEGRATION_CONTRACT.md
├── EVIDENCE_SCHEMA.md
├── RULE_CATALOG.md
└── member_plans/
```

Rules

- Docs are authoritative
- Code must align with docs
- Docs updated when logic changes

---

10. README.md

Purpose

Entry point for evaluators.

Contents

- Project summary
- Architecture overview
- How to run locally
- Ethical disclaimer

Final Enforcement Rules

- Every file has **one owner**
- Every directory has **one responsibility**
- No circular dependencies
- No logic duplication across layers
- Ontology, NLP, reasoning, and safety remain **strictly separated**

Why This Structure Works

✓ KRR-aligned
✓ Examiner-friendly
✓ Easy to integrate
✓ Scalable without scope creep
✓ Supports explainability & audit trails
