ScholarLens
Making Research Intuitive, Interactive, and Insightful

Project Kickoff Report – Iteration 2

Team: Hasita Chowdary Meka (Database & NLP) · Akshiya Vaibhavi Saravanan (Backend & DevOps) · Yashaswi Aryan (Frontend & UI/UX)

Timeline: 5 Weeks · **Date:** October 26, 2025

1 Project Overview and Scope

1.1 Core Concept

ScholarLens is an AI-powered web application that ingests research papers from open-access sources (arXiv, PubMed Central), builds dynamic knowledge graphs of methods, datasets, authors, and institutions, and provides multi-audience explanations, trend analysis, and interactive exploration tools.

1.2 Project Goals and Objectives

- Develop intelligent corpus management with semantic search using NLP entity extraction
- Create interactive knowledge graph visualizations mapping research relationships
- Implement evidence-backed Q&A using Retrieval-Augmented Generation (RAG)
- Build analytics dashboards tracking trending methods
- Design multi-audience summaries for experts, students, and policymakers

1.3 Project Scope

In Scope (5-week MVP): FastAPI backend; React frontend; Neo4j graph database; PostgreSQL for metadata; basic NLP pipeline with spaCy; HuggingFace Transformers for summarization; FAISS vector database; D3.js visualization; 8+ SQL queries; Docker deployment.

Out of Scope: Advanced trend forecasting; patent databases; mobile apps; real-time collaboration.

1.4 Key Deliverables by Phase

- **Week 1 – Initiation & Design:** Project Charter, Skills Matrix, Database Schemas (Neo4j & PostgreSQL), API Specs, UI Wireframes
- **Weeks 2-3 – Core Backend & Database:** FastAPI Backend, Neo4j & PostgreSQL Databases, PDF Parsing, Basic Entity Extraction, Core API Endpoints
- **Weeks 3-4 – Frontend & Advanced Features:** React App, Basic UI Components, Simple Graph Visualization, Search Interface, Basic RAG Q&A
- **Week 5 – Integration & Documentation:** Integration Tests, 8+ SQL Queries, Technical Documentation, User Manual, Docker Deployment, Final Report

1.5 Major Milestones

- **Week 1:** Kickoff, Requirements, Database & API Design Complete
- **Week 2:** Backend & Database Implementation Complete
- **Week 3:** Frontend Setup & Backend-Frontend Integration Started
- **Week 4:** Graph Visualization & Core Features Complete
- **Week 5:** System Integration, Testing, Documentation & Deployment Complete

2 Team Structure and Responsibilities

2.1 Skills Assessment Summary

Comprehensive skills assessment (scale 0-4) revealed **no critical gaps**. Strengths: Database & NLP (Neo4j, PostgreSQL, spaCy: 4), Backend & DevOps (FastAPI, Docker, Cloud: 4), Frontend & UI/UX (React, D3.js: 4).

2.2 Role Assignments

Hasita Chowdary Meka: Database & NLP Lead (9 tasks)

- *Responsibilities:* Neo4j & PostgreSQL design; NLP pipeline; entity extraction; vector database; HuggingFace integration; 8+ SQL queries
- *Technologies:* Neo4j, PostgreSQL, Cypher, Python, spaCy, HuggingFace (BART, LongT5), FAISS, Sentence Transformers

Akshiya Vaibhavi Saravanan: Backend & DevOps Lead (9 tasks)

- *Responsibilities:* FastAPI backend; PDF parsing; RAG system; trend analysis; Docker; CI/CD; deployment
- *Technologies:* Python, FastAPI, Uvicorn, Docker, Docker Compose, Git, AWS/GCP, Nginx, PyMuPDF

Yashaswi Aryan: Frontend & UI/UX Lead (9 tasks)

- *Responsibilities:* React architecture; D3.js visualizations; UI design; dashboard; user interfaces
- *Technologies:* React, TypeScript, Vite, Chakra UI, D3.js, Axios, CSS

2.3 Collaboration Strategy

Daily 15-min stand-ups; weekly sprint reviews; pair programming for integration; shared GitHub repository with feature branching; bi-weekly code reviews.

3 Technology Stack and Tools

3.1 Backend Technologies

- **Core:** Python 3.13+, FastAPI 0.115+, Uvicorn
- **NLP:** spaCy, HuggingFace Transformers (BART, LongT5), Sentence Transformers
- **Databases:** Neo4j (graph), PostgreSQL (relational), FAISS (vector)

3.2 Frontend Technologies

- **Core:** React 19.0+, Vite 6.1+, TypeScript
- **UI:** Chakra UI, D3.js, Axios

3.3 DevOps and Infrastructure

- **Version Control:** Git, GitHub
- **Deployment:** Docker, Docker Compose, Nginx, AWS/GCP

4 Data Sources and Management

4.1 Open-Access Data Sources

- **arXiv.org** – 2.4M+ preprints (physics, math, CS) – API: `https://arxiv.org/help/api`
- **PubMed Central** – 5.7M+ biomedical articles – API: `https://www.ncbi.nlm.nih.gov/books/NBK25501/`

License Compliance: CC BY, CC BY-SA, CC BY-NC, CC0 licenses only.

4.2 Database Schema Design

Neo4j: Nodes: Paper, Author, Institution, Method, Dataset, Concept. Relationships: AUTHORED_BY, AFFILIATED_WITH, USES_METHOD, USES_DATASET, CITES, COLLAB-

ORATES_WITH.

PostgreSQL: Users, SavedPapers, Queries, UserWorkspaces.

4.3 Data Pipeline (Simplified for 5 weeks)

1. PDF Ingestion
2. Text Extraction with error handling
3. Basic Entity Extraction – authors, methods, datasets
4. Embedding Generation
5. Graph Population in Neo4j
6. Metadata Storage in PostgreSQL

5 Initial Setup and Version Control

5.1 Version Control

GitHub repository: `backend/`, `frontend/`, `databases/`, `docs/`, `docker-compose.yml`.

Workflow: `main` (stable), `develop` (integration), feature branches, pull requests with reviews.

5.2 Development Environment

Hasita Chowdary Meka: Neo4j AuraDB, PostgreSQL, Python, spaCy, HuggingFace, FAISS.

Akshiya Vaibhavi Saravanan: Python, FastAPI, Docker, AWS/GCP CLI, PDF parsing libraries.

Yashaswi Aryan: Node.js, React, Vite, Chakra UI, D3.js.

6 Risk Assessment and Mitigation

6.1 High Priority Risks

- **Aggressive 5-week timeline (Score: 20)** – *Mitigation:* Focus on MVP features only; daily coordination; parallel development Weeks 3-4; cut non-essential features
- **Integration complexity (Score: 16)** – *Mitigation:* Early API contracts (Week 1); integration starts Week 3; dedicated integration testing Week 5

6.2 Medium Priority Risks

- **NLP pipeline complexity (Score: 12)** – Use pre-trained models; basic entity extraction only; defer advanced features
- **Team availability (Score: 12)** – Flexible daily schedule; cross-training; backup plans
- **Performance issues (Score: 10)** – Optimize queries; pagination; caching where needed

6.3 Mitigation Strategy

Weekly risk reviews; prioritize MVP features; maintain 10% time buffer; escalate blockers immediately.

7 5-Week Timeline Details

7.1 Week 1: Initiation & Design

Focus: Rapid setup and design.

Tasks: Kickoff meeting, scope definition, GitHub setup, database schema design (Neo4j & PostgreSQL), API specification, UI wireframes, tech stack documentation.

Deliverables: Project charter, skills matrix, database schemas, API specs, wireframes.

7.2 Week 2: Backend & Database Core

Focus: Backend infrastructure and data layer.

Tasks: FastAPI setup, Neo4j & PostgreSQL implementation, PDF parsing pipeline, basic NER entity extraction, core API endpoints.

Deliverables: Working backend, databases with schemas, PDF parser, basic entity extraction.

7.3 Week 3: Frontend Start & Integration Begin

Focus: Parallel frontend development and backend completion.

Tasks: React + Vite setup, Chakra UI components, FAISS vector database, simplified RAG system, file upload interface, Cypher queries.

Deliverables: React app structure, basic UI components, vector search, initial frontend-backend connection.

7.4 Week 4: Advanced Features & Integration

Focus: Core features and system integration.

Tasks: D3.js graph visualization, HuggingFace summaries, simple analytics dashboard, search & Q&A interface, frontend-backend API integration, basic trend analysis.

Deliverables: Functional graph visualization, working Q&A, integrated system, basic analytics.

7.5 Week 5: Finalization

Focus: Testing, documentation, deployment.

Tasks: Integration testing, 8+ SQL analytical queries, Docker configuration, bug fixes, technical documentation, user manual, AWS/GCP deployment, final report, presentation.

Deliverables: Tested system, SQL queries, documentation, deployed application, final presentation.

8 Current Iteration Deliverables

8.1 This Report

Three-page project introduction for 5-week timeline summarizing scope, objectives, team contributions, technology stack, and compressed schedule.

8.2 Data Source Confirmation

Dataset: Open-access papers from arXiv (`https://arxiv.org/help/api`) and PubMed Central (`https://www.ncbi.nlm.nih.gov/books/NBK25501/`).

8.3 Progress Tracker

Excel tracker with 36 tasks across 5 weeks:

- **Hasita Chowdary Meka (Database & NLP):** 9 tasks
- **Akshiya Vaibhavi Saravanan (Backend & DevOps):** 9 tasks
- **Yashaswi Aryan (Frontend & UI/UX):** 9 tasks
- **Collaborative:** 7 tasks

9 Success Criteria for 5-Week MVP

9.1 Must-Have Features (Week 5)

- Working PDF ingestion from arXiv/PubMed
- Neo4j graph with basic schema (papers, authors, methods)
- PostgreSQL with user data
- Basic entity extraction (NER)
- Simple RAG-based Q&A
- Interactive graph visualization (D3.js)
- Search interface
- 8+ analytical SQL queries
- Docker deployment

9.2 Nice-to-Have (Time Permitting)

- Advanced trend forecasting
- Multi-audience summaries (all three levels)
- Personal workspace features
- Comprehensive analytics dashboard