CS4063 – Natural Language Processing: Project Statement

November 22, 2025

1. Objective

This project is a major component of the course and will constitute a significcant portion of your final grade. Unlike the RD project track, the Development Track focuses on building a production-quality, market-Fit AI product using modern software engineering practices and state-of-the-art agentic AI frameworks.

Your goal is to design, architect, and implement a functional AI system that solves a real-world problem and demonstrates:

- Strong product thinking (clear users, value, and market gap)
- Use of modern LLM/agent frameworks (LangChain, LlamaIndex, DSPy, OpenAI Assistants API, Swarm, Haystack)
- Scalable and reliable backend engineering
- API design and integration
- Testing, CI/CD, logging, and monitoring
- Deployment in a realistic environment (local Docker, cloud optional)

Teams may use any modern freely available frameworks, libraries and models (Transformers, spaCy, PyTorch, OpenAI, Hugging Face, LLaMA 3, Mistral, LangChain, LlamaIndex, Haystack, FAISS, ChromaDB, Whisper, etc.).

2. Project Requirements

Each group will choose a problem area and develop a production-grade AI system. You must produce a system that demonstrates:

2.1 Problem Deifinition

- Clear articulation of the problem and target user journey, personas, and pain points
- Market-Fit justification (why this product matters)

2.2 System Architecture

Your system must incorporate:

- A modern agentic pipeline (LangChain / LlamaIndex / DSPy / Swarm)
- Retrieval Augmented Generation (RAG) OR multi-agent work ow 1
- A vector store (FAISS, Milvus, Weaviate, ChromaDB)
- Model integration (OpenAI / LLaMA / Mistral / custom Fine-tuned model)
- External tool or API calls (search, code-execution, database, scraper, etc.)
- Modular code design that follows engineering best practices

2.3 Engineering Requirements

Your project must also demonstrate:

- REST API or gRPC service (FastAPI preferred)
- Dockerization (containerization)
- Automated testing (unit + integration)
- Logging + monitoring (Prometheus/Grafana optional)
- Exception handling and recovery logic
- Configuration management (config files, env vars)
- Git work ow: branching, PRs, code reviews

2.4 Deployment

At minimum:

- Deploy locally via Docker Compose

Project Proposal (Required Early Submission)

You will present the idea in class in a short pitch. No slides required, but a very clear pitch and value generation idea.

- what you are building and for whom
- market need, user pain points
- high-level pipeline + agents
- how will you measure success?
- feasibility analysis

5. Possible Product Areas (Suggested Topics)

Projects can focus on—but are not limited to—the following AI product categories:

- AI Assistants—Multi-agent assistants for education, business work ows, scheduling, research, software development, or legal/medical tasks.
- RAG-Based Knowledge Systems — Enterprise knowledge bases; Urdu-domain knowledge assistants; question answering over custom corpora.
- Speech+NLP Systems — Urdu speech-to-text pipelines using Whisper + agentic post-processing.
- Content Automation Products — Auto-report generation, blog writers, social media automation, or compliance automation.
- Information Extraction Tools — Multi-agent pipeline extracting structured data from documents, PDFs, or audio.
- AI Tutoring Systems — Personalized learning agents that generate quizzes, summaries, explanations.

- Financial or Analytics Agents — Portfolio analysis, sentiment-based trading insights, auto-news sum- marization.
- Marketing Bots — Customer service agents, lead generation agents, personalization engines.

6. Final Deliverables

Each group must submit:

- A GitHub repository with full code, documentation, and tests
- A system architecture diagram
- A Dockerfile + Docker Compose file
- API documentation (Swagger or manual)
- A 23 page short report (engineering-focused)
- A 10-minute live presentation + demo