AI Course Project Ideas (Application-Focused, 8-Week Timeline)

The following project ideas are designed for an **upper-division AI course** and emphasize **building AI systems and applications**, not training models from scratch. Projects focus on **AI as system integration**—combining LLMs, retrieval, planning, reasoning, and tools—consistent with a modern interpretation of *Russell & Norvig's agent-based view of AI*.

Each project includes:

- **What to build**
- **Core AI concepts**
- **Implementation guidance**
- **Optional stretch goals**

---

1. Course-Specific AI Tutor (RAG-Based Chatbot)

What to build
A chatbot that answers questions for a specific course (e.g., AI, OS, Networks) using lecture notes, slides, assignments, and policies.

Core AI concepts

- Knowledge-based agents
- Information retrieval
- Reasoning with external memory
- Prompt engineering

Implementation guidance

- Use a hosted LLM (OpenAI, Anthropic, or Ollama).
- Ingest PDFs/Markdown → chunk → embed → store in a vector DB (FAISS, Chroma, Weaviate).
- Implement Retrieval-Augmented Generation (RAG) with citations.
- Add guardrails for unknown or missing information.
- Evaluate using a predefined Q&A benchmark.

Stretch goals

- Role-based responses (student vs. TA).
- Quiz or practice-question generation.

---

2. AI Study Planner / Academic Coach Agent

What to build
An agent that helps students plan study schedules based on deadlines, difficulty, and available time.

Core AI concepts

- Goal-based agents
- Planning and scheduling
- Constraint satisfaction
- Human-in-the-loop reasoning

Implementation guidance

- Accept user constraints (time blocks, priorities).
- Use planner-style prompts or tool-calling LLMs.
- Integrate with a calendar API or mock schedule.
- Require explicit reasoning for decisions.

Stretch goals

- Adaptive replanning when tasks are missed.
- Multi-week optimization.

---

3. Intelligent Document Q&A System for Technical Manuals

What to build
A system that answers questions about complex documentation (e.g., Linux man pages, API docs, hardware manuals).

Core AI concepts

- Knowledge representation
- Semantic search
- Explanation generation

Implementation guidance

- Clean and normalize documents before embedding.
- Compare naive RAG vs. hierarchical retrieval.
- Emphasize factual accuracy and citations.

Stretch goals

- Query rewriting.
- Confidence or uncertainty scoring.

---

4. AI Codebase Navigator & Explainer

What to build
A tool that helps developers understand a large codebase by answering questions like *“Where is X implemented?”* or *“How does Y work?”*

Core AI concepts

- Symbolic + neural reasoning
- Abstraction and explanation

- Search over structured data

Implementation guidance

- Parse code into logical chunks (files, functions).
- Store embeddings with metadata (language, module).
- Use retrieved snippets to ground explanations.
- Enforce “no hallucinated APIs” constraints.

Stretch goals

- Dependency graph visualization.
- Change-impact analysis.

---

5. Domain-Specific Research Assistant

What to build
An AI assistant specialized in summarizing and answering questions about a narrow research domain (e.g., AI alignment, cybersecurity, robotics).

Core AI concepts

- Knowledge-based agents
- Information synthesis
- Multi-document reasoning

Implementation guidance

- Curate a corpus of 20–50 papers/articles.
- Implement RAG with structured citations.
- Compare abstractive vs. extractive summaries.

Stretch goals

- Trend detection.
- Identification of open research problems.

---

6. AI Policy or Compliance Assistant

What to build
An assistant that helps users interpret policies (academic integrity, HR rules, privacy policies).

Core AI concepts

- Logic-based reasoning
- Rule following
- Natural language interpretation

Implementation guidance

- Encode policies as text plus structured rules.

- Combine RAG with constraint-based prompting.
- Require explanations that reference policy sections.

Stretch goals

- Conflict detection between policies.
- Hypothetical scenario analysis.

---

7. Multi-Agent Debate or Decision System

What to build
A system where multiple AI agents represent different viewpoints and debate a decision (e.g., ethical tradeoffs, system design).

Core AI concepts

- Multi-agent systems
- Game-theoretic reasoning (lightweight)
- Coordination and conflict

Implementation guidance

- Define agent roles, goals, and constraints.
- Implement turn-based prompting.
- Add a “judge” or aggregator agent to summarize outcomes.

Stretch goals

- Voting or negotiation protocols.
- Detection of convergence or deadlock.

---

8. Intelligent Search Engine for Campus Resources

What to build
A smart assistant that helps users find campus services, offices, forms, and procedures.

Core AI concepts

- Search and retrieval
- Knowledge organization
- User modeling

Implementation guidance

- Scrape or mock campus information pages.
- Build hybrid keyword + semantic search.
- Optimize for precision and trustworthiness.

Stretch goals

- Personalization by user role.

- Multilingual support.

---

9. AI-Powered Debugging Assistant

What to build
A tool that explains error messages and suggests fixes for beginner or intermediate programmers.

Core AI concepts

- Diagnosis
- Explanation generation
- Case-based reasoning

Implementation guidance

- Curate a dataset of common errors and explanations.
- Use retrieval to ground responses.
- Limit suggestions to safe, well-scoped advice.

Stretch goals

- Step-by-step debugging dialogue.
- IDE integration mockup.

---

10. Autonomous Task-Oriented Agent (Tool-Using)

What to build
An agent that completes a bounded real-world task (e.g., planning a trip, organizing an event, generating a study guide).

Core AI concepts

- Rational agents
- Planning and execution
- Tool use and feedback loops

Implementation guidance

- Define strict task boundaries.
- Implement tool calling (search, calculator, database).
- Log agent decisions for analysis.

Stretch goals

- Failure recovery.
- Cost and efficiency optimization.
