INF385T: Special Topics in Information Science: Language Models and Agentic Workflows

Fall 2026 Syllabus

Dr. Abhijit Mishra

The University of Texas at Austin, School of Information

Contents

Semester: Fall 2026
Time / Venue: Thursday, 3:30 PM–6:30 PM, UTA 1.210A

Role	Name	Email
Instructor	Dr. Abhijit Mishra (he/his)	abhijitmishra@utexas.edu

Canvas: link
Instructor Office Hours: Thursday 6:30 PM–8:00 PM, immediately after class, via Zoom (link), or by appointment.

Communication and Asking for Help

Please ask all questions that are applicable to the entire class on Canvas so that others may benefit from the discussion. Use email for questions unique to individual circumstances; in those cases, please contact the instructor at abhijitmishra@utexas.edu.

University Policies and Resources for Students

In addition to the policies in this syllabus, please review the University's Policies and Resources for Students Canvas page, which lists university-wide resources relevant to every course you take at UT Austin, including academic integrity, disability accommodations, mental health, and Title IX resources.

Course Description

This course examines the design, implementation, and evaluation of agentic AI systems built on large language models. Moving beyond prompt engineering, students study structured reasoning strategies (e.g., chain-of-thought and tree-of-thought), retrieval-augmented generation (RAG), and multi-agent workflows. Emphasis is placed on system architecture, orchestration, and the integration of language models with external tools, memory, and data sources. The course also addresses human-centered design principles, transparency, reliability, and evaluation methodologies for real-world deployment. Through hands-on Python laboratories and two group projects, students design and implement industry-relevant AI systems while critically assessing performance, robustness, and governance considerations. The course is suitable for advanced master's and doctoral students interested in intelligent systems, human-centered AI, and applied language model architectures.

Rationale and Audience

This graduate-level topics course addresses the growing need for training in the design, evaluation, and deployment of agentic AI systems built on large language models (LLMs). As language-model-based systems become central to industry practice across consulting, healthcare, finance, public policy, and technology, there is increasing demand for professionals who understand not only prompting but also the architectural principles underlying retrieval-augmented generation, structured reasoning workflows, multi-step agent systems, and human-in-the-loop design.

Within the MSIS program, students are already encountering generative AI tools in internships and industry roles; however, there is a need for an advanced course that systematically teaches how to design, build, and evaluate these systems. This course fills that gap by combining foundational understanding—reasoning strategies such as chain-of-thought and tree-of-thought prompting—with applied system design, including RAG pipelines, agent orchestration, evaluation frameworks, and deployment considerations.

The course is also appropriate for doctoral students interested in human-centered AI, intelligent systems, information retrieval, and applied machine learning, offering opportunities for research-oriented projects and experimentation with emerging agentic architectures.

Course Goals and Learning Objectives

The objectives of the course are to:

- Provide students with a systems-level understanding of large language models, agentic AI architectures, and retrieval-augmented workflows.
- Develop conceptual clarity around structured reasoning strategies, multi-agent coordination, and orchestration mechanisms.
- Introduce human-centered principles for designing reliable, transparent, and responsible AI systems.
- Cultivate practical, project-based experience in building and evaluating agentic AI systems using Python.
- Strengthen students' ability to communicate system architecture, design trade-offs, and evaluation results in professional and academic contexts.

By the end of the course, students will be able to:

- Design and implement retrieval-augmented and agentic AI systems integrating language models, external tools, and memory components.
- Evaluate system performance using structured reasoning analysis, grounding metrics, and robustness testing.
- Architect multi-step and multi-agent workflows appropriate for real-world applications.
- Incorporate human-in-the-loop mechanisms and explainability features into system design.
- Present and defend a complete agentic AI system, including architectural documentation and empirical evaluation.

Prerequisites

Proficiency in Python programming is assumed. Prior coursework in Natural Language Processing, Machine Learning, or Deep Learning is recommended and will position students for stronger engagement with the technical components of the course. Students who are unsure whether their preparation is sufficient should contact the instructor before enrolling.

Instruction Modality and Class Format

Class meetings are held in person in a computer-equipped classroom. Except for presentation and workshop meetings, each three-hour class is divided into approximately **1.5 hours of theory** and **1.5 hours of hands-on Python laboratory work**. Active participation in both portions is

expected. Lectures are not recorded by default; if this changes, alternative video resources will be made available where possible.

Required Materials

There is no required textbook. Assigned readings will be available online at no cost or through UT Libraries and will be posted on Canvas. Slides, lecture notes, notebooks, and supplemental resources will normally be provided before the relevant class.

Required Devices and Computing Resources

Students must bring a laptop computer capable of running Python and Jupyter notebooks. The course is device agnostic, although a PC or Mac is preferable; contact the instructor in advance if using customized hardware or an operating system that may require accommodation. Students will use Python, version control, language-model APIs, open-source models, vector databases, and agent-orchestration libraries. For resource-intensive exercises, the class may use Google Colaboratory or available University computing resources.

For required agentic interactions, the instructor will provide access to an **instructor-managed mini language model** with a capped course credit. The instructor will manage the system, credentials, access controls, and usage limits. The course and the School of Information will not sponsor additional commercial AI subscriptions, premium accounts, or API expenses. Students are not required to purchase an AI subscription. Any optional use of another paid service is at the student's expense, must comply with the activity's AI tier and data rules, and may not provide an advantage required to complete assessed work.

Generative AI Tools Policy

Course AI Policy Statement (required University language): In accordance with the University's Institutional Rules on Student Services and Activities, Chapter 11, students accept the responsibility to always uphold academic integrity and an honor code reflective of a scholarly community devoted to academic and personal success. All members of the University community are fully accountable and responsible for any output they produce as part of academic work. They are also responsible for following the guidance specified in the Texas Statement on Academic Integrity and avoiding prohibited uses of generative AI tools outlined in the Information Security Office's guidance on Acceptable Use of Generative AI Tools.

Because language models are the subject and primary laboratory material of this course, authorized use of generative AI is expected in many activities. Permission depends on the tier assigned to each activity:

Tier	Permitted use	Where it applies
Tier 0 — No AI	No generative AI tool, AI assistance, AI-enabled search, or AI code completion may be used. Work must be completed independently unless collaboration is explicitly permitted.	The paper-based quiz, designated conceptual checks, and any activity labeled Tier 0
Tier 1 — Directed AI	AI may be used only for the purposes, models, tools, and stages explicitly authorized in the activity instructions. Students must preserve requested prompts, outputs, traces, and disclosures. AI may not replace the student's required analysis or explanation.	In-class laboratories, structured experiments, and the four take-home assignments unless otherwise stated
Tier 2 — Full AI with disclosure	Generative AI and agentic tools may be used throughout system design and implementation, subject to the data, security, documentation, and resource limits in this syllabus. Students must disclose models, prompts, tools, generated code or content, and substantive human decisions.	Both group projects and activities explicitly labeled Tier 2

Using AI beyond the tier authorized for an activity, concealing required disclosures, fabricating logs or results, or misrepresenting generated work as independently produced may constitute academic

misconduct. The instructor may request a demonstration, code walkthrough, prompt history, design rationale, or oral explanation of submitted work.

Tool Access, Credits, and Student Costs

For authorized agentic interactions, students will use the instructor-managed mini-model service whenever the activity requires hosted model access. Access will be subject to per-student or per-team capped credits, rate limits, logging, and acceptable-use controls. Students must design experiments within the assigned budget and may not share credentials, evade limits, or access another student's allocation.

Neither the course nor the School will sponsor optional generative-AI products, additional API credits, premium plans, or individual subscriptions. Free, local, or student-funded tools may be used only when the assignment permits them. Purchasing an optional service will never be necessary to satisfy a course requirement, and grading will not reward access to a more expensive model.

Data, Privacy, and Course Materials

Do not submit confidential, proprietary, personally identifiable, export-controlled, or otherwise restricted data to an AI system. Use only data and services authorized for the activity, comply with provider and University policies, and minimize the information transmitted to third-party services.

Course materials—including slides, assessments, notebooks, datasets, starter code, and student work—may not be uploaded to an external AI tool unless the instructor explicitly authorizes that material and tool for the activity. Authorization to use an AI tool does not automatically authorize uploading all course content. AI-generated, fabricated, or synthetic data may be used only when the assignment explicitly permits it and requires clear labeling and documentation.

Academic Integrity

Students who violate University rules on academic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Since such dishonesty harms the individual, all students, and the integrity of the University, policies on academic dishonesty will be strictly enforced. For further information, visit the Student Conduct and Academic Integrity website.

Classroom Recording Policy

Per HOP 2-9970, students are prohibited from recording class instruction (audio or video) unless the student obtains the instructor's permission or Disability and Access (D&A) has approved audio recording as an accommodation.

Course Material Sharing Policy

No materials used in this class, including lecture handouts, slides, videos, assessments, project materials, in-class materials, notebooks, and review materials, may be shared online or with anyone outside the class without the instructor's explicit written permission. Unauthorized sharing may

facilitate academic misconduct and may be reported to Student Conduct and Academic Integrity. If a class session is recorded, recordings are reserved for enrolled students for educational purposes only, are protected under FERPA, and must not be shared outside the class.

Class Participation

The course depends on sustained engagement with theory, technical practice, and peer critique. Participation includes preparation for class, contribution to discussion, completion of in-class laboratory work, constructive project feedback, and short conceptual spot-checks or quizzes. Laboratory artifacts or brief reflections may be due by 11:59 PM on the day of class unless otherwise stated.

There is no traditional final examination. The course includes one individual, closed-device, paper-based quiz held during the first 1.5 hours of Week 8. Assessment otherwise emphasizes applied assignments, laboratory work, working systems, evaluation evidence, technical reports, presentations, and the ability to explain design decisions and results.

Assignments and Course Projects

1. In-Class Laboratories and Participation — Tier 1 unless otherwise noted

Laboratories translate each week's theory into working Python systems or controlled experiments. Students should expect to submit selected notebooks, traces, evaluation artifacts, design diagrams, or short reflections. Some conceptual checks may be designated Tier 0.

2. Four Take-Home Assignments — Tier 1

Four individual take-home assignments will reinforce structured model interfaces and tool use, context engineering and grounding, single-agent control and recovery, and agent planning and evaluation. All four assignments will be completed during the first half of the course before the mid-October quiz. Each assignment will specify the authorized AI operations, required documentation, collaboration rules, and submission format. Together, the assignments constitute 20% of the course grade.

3. Paper-Based Quiz — Tier 0

The individual paper-based quiz will be administered during the first half of Week 8 on Thursday, October 15, and will last 1.5 hours. It assesses conceptual understanding of workflows and agents, context and state, tool interfaces, control loops and termination, evaluation and tracing, planning, and verification. No electronic devices, generative AI tools, or AI-enabled assistance may be used.

4. Group Project 1: Reliable Tool-Using Workflow — Tier 2

Student teams will select a project from an instructor-provided pool of defined agentic-workflow problems. Each option will specify the problem context, minimum system capabilities, required evaluation criteria, and expected deliverables. Teams will design, implement, evaluate, and present a reliable tool-using workflow that includes structured model outputs, retrieval or another context source, at least one external tool, explicit control flow, trace inspection, and an evaluation harness. The working system, presentation, demonstration, and written technical report are all due Thursday, November 5. The project is graded through the system and evaluation artifacts (8%), presentation and demonstration (5%), and report (7%).

5. Group Project 2: Production-Oriented Agentic System — Tier 2

Project 2 is an open-ended team project. The call for proposals will be issued Thursday, October 29. Teams will identify a meaningful problem, justify why an agentic architecture is appropriate, and propose a complete system integrating language models with external tools, memory, data

sources, or appropriately justified multiple agents. The proposal must define the system architecture, evaluation strategy, human-oversight approach, security boundaries, and planned deliverables. The completed system must include state management, tool permissions, recovery behavior, termination conditions, human oversight, security testing, observability, and empirical evaluation. Proposals are due Thursday, November 12. The working system and presentation are due Thursday, December 3. The written final report is due Thursday, December 10, allowing teams to incorporate presentation feedback. The project is graded through the system and evaluation artifacts (12%), presentation and architecture defense (7%), and final report (11%).

Late Work and Missed Work

A grace policy of two days applies to this course. You do not have to use this policy, but if unexpected personal events arise, email the instructor as soon as possible and preferably before the due date. You may use either one two-day grace period for one assignment or two separate one-day extensions for two different assignments. The grace policy does not automatically apply to presentations, in-class work, or team milestones where an extension would affect other students.

The only excused absences are for religious holidays or extenuating emergency circumstances. If you plan to miss class for a religious holiday, notify the instructor at least two weeks in advance. You will not be penalized, though you remain responsible for missed work. For unexcused absences, make-up opportunities for missed in-class work are not guaranteed but may be granted; check with the instructor.

Grading Policy

This course uses plus/minus grading for the final class grade.

Grade Component	Percentage
In-class laboratories and participation	20%
Four take-home assignments	20%
Paper-based quiz	10%
Group Project 1 — overall	**20%**
System and evaluation artifacts	8%
Presentation and demonstration	5%
Technical report	7%
Group Project 2 — overall	**30%**
System and evaluation artifacts	12%
Presentation and architecture defense	7%
Final technical report	11%
Total	**100%**

Final Exam: This course has no traditional final examination. In lieu of a final exam, students complete the final project presentation and report described in the Course Outline. The final report is due Thursday, December 10, 2026, unless Canvas announces a revised deadline.

Grade Cutoffs

Grade	Cutoff	Grade	Cutoff
A	94%	C+	77%
A-	90%	C	74%
B+	87%	C-	70%
B	84%	D+	67%
B-	80%	D	64%
		D-	60%
		F	$< 60\%$

Course Outline

All instructions, assignments, readings, and essential information will be on Canvas. Check the site regularly. Changes to the schedule may be made at the instructor's discretion. Dates follow the Fall 2026 Thursday meeting pattern. Each standard meeting combines approximately 1.5 hours of theory with 1.5 hours of Python laboratory work.

Core open-access reference: Roitman, H. (2026). *The Hitchhiker's Guide to Agentic AI: From Foundations to Systems* (Version 1.3, arXiv:2606.24937). Specific chapters are assigned below; a copy is also provided with the course materials.

Week 1 — From Language Models to Agentic Systems (Thu Aug 27)

Theory: Evolution from NLP and LLM applications to agentic systems; models, workflows, agents, and autonomous systems; components of an agentic system; when to use—and not use—agents; human-centered framing of agent autonomy.

Lab: Access proprietary and open-source models through APIs; compare unstructured and structured responses; establish a reproducible experimentation environment; record model, prompt, parameter, latency, and cost information.

References:

1. Roitman, *The Hitchhiker's Guide to Agentic AI*, Chapter 15, "Introduction to Agentic AI."
2. Anthropic. Building Effective Agents. Focus on the distinction between workflows and agents and the guidance on when to use agents.

Assignment (not graded): Pre-course survey and computing-environment check

Week 2 — Structured Model Interfaces and Tool Use (Thu Sept 3)

Theory: Transformer and in-context learning recap; structured outputs and schemas; function calling and tool-use interfaces; model limitations and failure modes; tool descriptions as interface contracts.

Lab: Generate schema-constrained outputs; define and validate callable tools; build a model–tool–observation interaction; diagnose invalid arguments and tool-selection errors.

References:

1. Roitman, Chapter 1.12.11, "Constrained Decoding," Chapter 1.13.5, "Structured Output Prompts," and Sections 18.3.4–18.4.3 on tool descriptions and execution.
2. OpenAI. Function Calling in the OpenAI API.
3. OpenAI. Introducing Structured Outputs in the API.

Take-Home Assignment 1: Structured outputs, tool contracts, and validated model–tool interaction

Week 3 — Context Engineering: Retrieval, Memory, and State (Thu Sept 10)

Theory: Context engineering versus prompt engineering; retrieval as an agent capability; dense, sparse, and hybrid retrieval; working memory, episodic memory, and persistent state; context selection, compression, and provenance.

Lab: Build a compact retrieval pipeline; add short-term and persistent memory; inspect retrieved evidence and citations; evaluate retrieval and grounding failures.

References:

1. Roitman, Chapter 16, "Retrieval-Augmented Generation," and Chapter 17, "Agentic Memory Systems."
2. Roitman, Sections 18.2 and 18.6 on context-window and state management.
3. LangChain. Persistence. Focus on checkpoints, stores, short-term memory, and durable state.

Take-Home Assignment 2: Context engineering and grounded tool use; due Thursday, September 17

Week 4 — Deterministic Agentic Workflows (Thu Sept 17)

Theory: Workflows versus autonomous agents; prompt chaining and conditional routing; parallelization and aggregation; evaluator–optimizer patterns; state machines and task graphs; deterministic control around nondeterministic models.

Lab: Implement a stateful workflow; add routing and parallel branches; introduce validation gates and retries; visualize workflow execution.

References:

1. Roitman, Section 18.5, "Orchestration Patterns," and Chapter 20, "Agent Design Patterns."
2. Anthropic. Building Effective Agents. Focus on prompt chaining, routing, parallelization, orchestrator–worker, and evaluator–optimizer workflows.

Group Project 1 project pool released: Teams select one instructor-defined reliable tool-using workflow problem.

Week 5 — Single-Agent Loops and Adaptive Control (Thu Sept 24)

Theory: The plan–act–observe loop; ReAct and related execution patterns; dynamic tool selection; planning versus reacting; stop conditions and completion criteria; error recovery and bounded autonomy.

Lab: Build a single-agent execution loop; add multiple tools and dynamic selection; implement iteration and cost limits; handle tool failures and incomplete tasks.

References:

1. Roitman, Sections 18.4–18.7 on tool integration, orchestration, state, and recovery.
2. Roitman, Chapter 19, "Loop Engineering."
3. Yao et al. (2022). ReAct: Synergizing Reasoning and Acting in Language Models.

Take-Home Assignment 3: Single-agent control, tool use, and recovery; due Thursday, October 1

Week 6 — Agent Evaluation, Tracing, and Reliability (Thu Oct 1)

Theory: Why final-answer accuracy is insufficient; evaluating trajectories and tool calls; task completion, correctness, grounding, and efficiency; trace inspection and failure taxonomies; deterministic, model-based, and human graders; regression testing.

Lab: Instrument an agent with structured traces; construct an evaluation dataset; implement task- and trajectory-level graders; compare workflow variants; Project 1 testing workshop.

References:

1. Roitman, Chapter 14, "LLM Evaluation," and Chapter 21, "Agentic Environments and Benchmarks."
2. Anthropic. Demystifying Evals for AI Agents.
3. Roitman, Sections 18.7.4 and 18.8.4 on observability and evaluation in production.

Take-Home Assignment 4: Agent planning, trace analysis, and evaluation

Week 7 — Planning, Reflection, and Verification (Thu Oct 8)

Theory: Task decomposition and plan generation; plan execution and replanning; reflection and self-critique; verifier and critic components; agent-as-judge limitations; reliable recovery behavior.

Lab: Add planning to a single-agent system; implement a verifier or critic; trigger replanning after failure; compare simple and reflective agents; measure quality, latency, and cost trade-offs.

References:

1. Roitman, Sections 18.5.1–18.5.2 on ReAct and plan-and-execute orchestration.
2. Roitman, Sections 19.5–19.8 on loop patterns, verification, termination, and failure modes.
3. Shinn et al. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning.

Week 8 — Paper-Based Quiz and Agent Interoperability (Thu Oct 15)

First half — Paper-based quiz (1.5 hours, Tier 0): Workflows versus agents; context, retrieval, memory, and state; tool interfaces; agent loops and termination; evaluation, tracing, planning, and verification.

Second half — Interoperability lab: Model Context Protocol concepts; tools, resources, prompts, clients, and servers; agent-to-agent communication; capability discovery and agent cards; connection to an interoperable tool service.

References:

1. Roitman, Chapter 22, "Model Context Protocol," and Chapter 24, "Agent-to-Agent Communication."
2. Model Context Protocol. Architecture Overview.
3. Google Developers. Developer's Guide to AI Agent Protocols.

Week 9 — Multi-Agent Systems and Project 1 Workshop (Thu Oct 22)

Theory: When multiple agents are justified; agent-as-tool and handoff patterns; supervisory and hierarchical architectures; role decomposition and task allocation; communication protocols and shared state; coordination failures.

Lab and workshop: Build a supervisor-and-specialist system; implement delegation and handoffs; compare single-agent and multi-agent designs; test Project 1 workflows; review evaluation evidence and reports.

References:

1. Roitman, Chapter 25, "Multi-Agent Systems."
2. Roitman, Section 18.5.3 on multi-agent orchestration and Sections 17.6–17.9 on shared memory and its evaluation.
3. Anthropic. Building Effective Agents. Focus on orchestrator–worker systems and the cautions surrounding unnecessary complexity.

Week 10 — Human-Agent Interaction and Project 2 Call for Proposals (Thu Oct 29)

Theory: Human-in-the-loop and human-on-the-loop systems; approval boundaries and escalation rules; levels of agent autonomy; transparency and explainability; interruptibility, user correction, and trust calibration.

Lab: Add approval gates; implement pause, resume, and cancellation; design risk-based escalation; build a user-facing activity and decision log.

References:

1. Roitman, Section 18.5.4, "Human-in-the-Loop," and Chapter 27, "Agentic UI Frameworks."
2. NIST. AI Risk Management Framework Playbook. Focus on human oversight, defined roles, and operational accountability.
3. OWASP. HITL Dialog Forging (Lies-in-the-Loop).

Group Project 2 call for proposals: Open-ended production-oriented agentic system

Week 11 — Group Project 1 Presentations (Thu Nov 5)

Entire class: Group presentations and demonstrations; architecture explanation; tool-use and workflow demonstration; evaluation findings and failure analysis; peer and instructor critique.

References: Revisit Roitman, Chapters 18–21, and the Week 6 agent-evaluation readings while preparing the architecture defense, demonstration, and report.

Project 1 due: Working system, evaluation artifacts, presentation, demonstration, and written technical report

Week 12 — Agent Security, Safety, and Governance (Thu Nov 12)

Theory: Prompt injection and indirect prompt injection; tool misuse and excessive agency; least-privilege design; data boundaries and secret management; sandboxing and action authorization; red-teaming, accountability, and governance.

Lab: Conduct adversarial testing; implement tool permissions and policy checks; add input, output, and action guardrails; test malicious tool content; document residual risks.

References:

1. Roitman, Section 18.4.4 on sandboxing and safety and Section 19.8 on loop failure modes and anti-patterns.
2. OWASP. AI Agent Security Cheat Sheet.
3. OWASP. LLM06:2025 Excessive Agency.
4. NIST. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile.

Project 2 proposal due: Problem justification, architecture, evaluation strategy, oversight and security plan, and proposed deliverables

Week 13 — Industry Keynote, Deployment, and Project Workshop (Thu Nov 19)

First half — Industry keynote: Real-world agentic AI systems; deployment challenges; reliability and organizational adoption; cost, latency, scaling, and governance; industry Q&A.

Second half — Deployment lab and project workshop: Deconstruct the speaker's system architecture; durable execution and checkpointing; long-running and asynchronous tasks; state persistence, recovery, idempotency, and retries; monitoring, latency, and cost management; add checkpoints and operational metrics; review Project 2 evaluation and security evidence; conduct system stress testing and instructor consultations.

References:

1. Roitman, Sections 18.7–18.10 on recovery, observability, scaling, and a production agent harness.
2. Roitman, Sections 19.9–19.12 on production loops, long-running context, appropriate use, and economics.
3. LangChain. Persistence and Functional API Overview. Focus on checkpoints, resumption, interrupts, and fault tolerance.

Project 2: Mid-review and stress-test evidence

Week 14 — Fall Break / Thanksgiving: No Class Held (Thu Nov 26)

Week 15 — Group Project 2 Final Presentations (Thu Dec 3)

Entire class: Final agentic-system presentations and live demonstrations; architecture and evaluation defense; robustness, security, and human-oversight findings; peer critique and course synthesis.

References: Revisit the readings from Weeks 6, 10, 12, and 13 when preparing the final system's evaluation, oversight, security analysis, production-readiness argument, and architecture defense.

Project 2 presentation deliverables due: Working system, evaluation artifacts, presentation, and demonstration

Week 16 — Offline Report Preparation (Dec 4–10)

- Incorporate presentation feedback.
- Complete technical documentation and evaluation analysis.
- Submit reproducibility materials and the final report.

Project 2 final technical report: Due Thursday, December 10, 2026

Mantra for Student Success

- Aim for full attendance to maximize engagement with lectures, laboratories, demonstrations, and project critiques.
- Complete assigned readings and environment setup before class so that laboratory time can focus on experimentation and analysis.
- Keep complete records of prompts, model and tool versions, system traces, data sources, failures, costs, and evaluation decisions.
- Prioritize transparency by citing tools, resources, models, and data sources and by clearly distinguishing generated output from your own analysis and decisions.
- Treat model output as evidence to inspect rather than truth to accept. Test claims, document failure modes, and design meaningful baselines.
- Seek help early during office hours when programming, system architecture, evaluation, or team coordination becomes challenging.
- Embrace iteration: strong agentic systems emerge from repeated design, testing, critique, documentation, and revision.

Accommodations for Students with Disabilities

The University is committed to creating an accessible and inclusive learning environment for students with disabilities consistent with University policy and federal and state law. Please let me know if you experience any barriers to learning so I can work with you to ensure you have equal opportunity to participate fully in this course.

If you are a student with a disability, or think you may have a disability, and need accommodations, please contact Disability and Access (D&A): http://disability.utexas.edu. If you are already registered with D&A, please share your Accommodation Letter with me as early as possible in the semester so we can discuss how your approved accommodations will be implemented in this course.

Religious Holy Days

By UT Austin policy, you must notify me of your pending absence for a religious holy day as far in advance as possible. If you must miss a class, examination, work assignment, or project to observe a religious holy day, please fill out the required form administered by the Office of the Dean of Students. You will be given an opportunity to complete the missed work within a reasonable time after the absence, as provided by Texas law and University policy.

Names and Pronouns

Professional courtesy and sensitivity are especially important with respect to differences of race, culture, religion, politics, sexual orientation, gender, gender variance, and nationality. Requests to be addressed by a chosen name and pronouns will be honored. Class rosters reflect a student's chosen name if one has been provided.

Basic Needs Security

Any student who faces challenges securing food or housing and believes this may affect their performance in the course is urged to contact the Dean of Students for support. UT maintains the UT Outpost, a free on-campus food pantry and career closet.

Mental Health Support

Students who are struggling for any reason and believe it might affect their performance are encouraged to contact the instructor if comfortable doing so, so that resources or accommodations can be identified. For immediate mental health assistance, call the Counseling and Mental Health Center (CMHC) at 512-471-3515 or contact the iSchool CARE counselor. Outside CMHC business hours, contact the CMHC 24/7 Crisis Line at 512-471-2255.

Title IX Reporting

Title IX protects against sex- and gender-based discrimination, sexual harassment, sexual assault, unprofessional or inappropriate conduct of a sexual nature, dating or domestic violence, and stalking at federally funded educational institutions. Texas law requires University employees, including faculty, to report certain information regarding sexual harassment, sexual assault, dating violence,

and stalking to the Title IX Office. **The instructor is a Responsible Employee and must report Title IX-related incidents disclosed in writing, discussion, or one-on-one conversation.** To seek confidential support, email advocate@austin.utexas.edu. For more information, visit http://www.titleix.utexas.edu/, email titleix@austin.utexas.edu, or call 512-471-0419.

Land Acknowledgement

We acknowledge that we are meeting on the Indigenous lands of Turtle Island, the ancestral name for what is now called North America, and specifically the Alabama-Coushatta, Caddo, Carrizo/Comecrudo, Coahuiltecan, Comanche, Kickapoo, Lipan Apache, Tonkawa, and Ysleta Del Sur Pueblo, and all American Indian and Indigenous Peoples and communities who have been or have become part of these lands and territories in Texas.

This syllabus is subject to change at the instructor's discretion. The most current version and all official announcements will be posted on Canvas.