2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #52440

Teaching with AI - a few examples

Dr. Shilpa Gupta, San Jose State University

Shilpa Gupta is an Adjunct Professor in the Industrial and Systems Engineering Department at San Jose State University, where she teaches graduate and undergraduate courses in Data Science and Operational Analytics. She brings more than a decade of industry experience as a professional data scientist, having tackled complex business problems and delivered high-impact, customer-facing solutions across a range of organizations. Her work spans innovative applications of data science that bridge rigorous methodology with real-world outcomes — an approach that now shapes her deep commitment to data science pedagogy. Dr. Gupta holds a doctorate in Industrial Engineering from Arizona State University

Teaching with AI - a few examples

***Abstract*—In a data science classroom, certain skills are immutable: reading a technical journal article, critically analyzing text or code not written by oneself, communicating in the discipline, and metacognition. Additional skills are becoming increasingly relevant with the proliferation of large language model (LLM)-based tools, including problem framing, prompt engineering, and systems thinking. This practice report presents three classroom activities that instructors may use as-is or adapt to design AI-centered tasks targeting these skills. Each activity is presented in a lesson-plan format with learning objectives, setup, implementation details, and assessment criteria. These activities were piloted across three graduate-level engineering and data science courses over two semesters. Observations reported here are informal and reflective in nature; a formal empirical study with IRB approval is planned for a future semester. Our goal is to reduce instructor preparation time and provide a replicable template for further instructional design.**

***Index Terms*—LLM; critical thinking; reading; systems thinking; instructional design; practice report**

Introduction

Generative Artificial Intelligence (GenAI) systems have rapidly become embedded in professional and educational contexts, reshaping how knowledge is produced, interpreted, and applied. In higher education, these technologies present both opportunities and challenges. While GenAI tools can generate text, code, and analytical outputs with minimal prompting, their ease of use raises concerns regarding over-reliance, superficial learning, and erosion of critical thinking skills Bastani et al. (2024). Consequently, educators are increasingly tasked with designing learning experiences that leverage AI's capabilities while preserving and strengthening human judgment.

Recent scholarship highlights the importance of AI literacy and critical engagement with AI-generated outputs Walter (2024). In disciplines such as engineering, data science, and machine learning, students must not only understand how AI systems function but also how to evaluate their outputs for correctness, logic, and ethical implications. Cognitive research in programming further emphasizes that reading and understanding code is a distinct and essential skill that requires deliberate practice Hermans (2021).

This practice report contributes a set of instructional examples that integrate GenAI into coursework with the explicit goal of fostering critical reading, thinking, and systems thinking in the discipline. Rather than asking students to generate solutions using AI, the activities require students to analyze, critique, and refine AI-generated content. By situating GenAI within authentic disciplinary tasks, such as reading journal articles, debugging machine learning code, and designing AI systems with ethical constraints, this approach reframes AI as a partner in learning rather than a replacement for human reasoning.

Background

Ubiquitous availability of LLMs has spurred debate among educators. On one side, educators such as Bastani et al. (2024), Cotton, Cotton, and Shipway (2024) and Rudolph, Tan, and Tan (2023) highlight the negative impact of such tools, chief among them are weakening of academic integrity and decrease in independent thinking leading to propagation of biases in written content. On the other side, educators such as Ellis and Slade (2023), Xing (2024), Pankiewicz et al. (2024), Adeyele and Ramnarain (2024), are more optimistic and encourage incorporating LLMs in classrooms for improved enthusiasm for learning among the students. Bray (2023) support their claim by showing minimal impact on the grade distribution among the students.

Mollick and Mollick (2023) have categorized the ways students can leverage LLMs into seven roles, such as coach, tutor, and teammate among others. Tu et al. (2023) provide great examples of LLMs as a curriculum generators personal tutor and quiz generators, and illustrate how LLMs can be incorporated into data science pipelines. However, Tu et al. (2023) do not provide example of how LLM output could be used for critical thinking, a gap this practice report addresses.

Development of critical thinking skills has become crucial since LLMs often hallucinate, as argued in Xie, Wu, and Chakravarty (2023). This creates the need for critique-based assignments. The definition of critical thinking used here is from Paul (2018): "disciplined, self-directed thinking that exemplifies the perfection of thinking appropriate to a particular mode or domain of thought." Furthermore, Bien and Mukherjee (2024) document variability in LLM responses and conclude that by encouraging instructors to teach students to 'check for reasonable of outputs,' directly supporting the approach taken here.

In our view, data science literacy is the ability to read, apply critical thinking, understand and communicate statistical and computational information. It is the first rung in the hierarchical competencies and is based on the statistical literacy defined by Ziegler and Garfield (2018). In addition to reading articles and reports written by other, students in data science also need to engage in reading code written by others. Code reading, unlike reading a text, requires active participation from the user for understanding the flow of logic, by either running the code partially or stepping through it (Hermans (2021)).

Instructional Context

The three instructional activities described in this practice report were piloted in three graduate-level courses at a research university: Math Foundations for Data Science, Data Ethics and AI Tools and Processes. All these courses serve upper-division undergraduate and graduate students in engineering and data science disciplines who have foundational programming experience and exposure to statistical methods. Collectively, the activities have been used with cohorts of approximately 100 students across the three courses per semester.

The goal of this work is not to measure learning gains statistically, but to document replicable instructional designs that operationalize critical thinking, especially with AI-generated content. Observations reported here are informal and based on instructor reflection and in-class discussion, not systematic data collection. Student quotes and work samples were not collected or analyzed for this report. A formal empirical study, including pre/post assessments, structured reflection prompts, and rubric-based coding of student work — is planned for a future semester pending IRB approval.

Each activity required minimal additional technological infrastructure beyond access to AI tools and Python-based computing environments, making the designs broadly replicable across institutional contexts.

Instructional Design

The pedagogical design draws on established theories of critical thinking and literacy. Critical thinking is defined as disciplined, self-directed thinking appropriate to a particular domain of knowledge Paul (2018). Data science literacy is understood as the ability to read, interpret, critique, and communicate statistical and computational information. Within this framework, GenAI-generated content serve as instructional material that students must actively interact with.

Each activity followed a structured progression that included individual analysis, collaborative discussion, and reflection. This structure ensured that students engaged deeply with the material and articulated their reasoning processes rather than relying on AI-generated explanations.

Critical Thinking and Reading in the Discipline

The first instructional example focuses on critical reading within a disciplinary context. Students were provided with an AI-generated summary or explanation of an academic work, specifically Strang (2018) article on multiplying and factoring matrices, and asked to read it alongside the original source material to identify conceptual inaccuracies, omissions, or misleading explanations.

This activity is grounded in the reading-for-understanding framework proposed by Schoenbach et al. (1999), which emphasizes active engagement with texts through questioning, annotation, and verification. By introducing AI-generated summary into this process, the activity highlights the necessity of disciplinary understanding when evaluating AI outputs. This module reframes AI as a fallible partner in the learning process. Students observe that while AI can generate fluent and plausible explanations quickly, disciplinary understanding is essential for assessing accuracy and depth.

Critical Thinking and Reading Code Written by Others

The second instructional example addresses critical thinking in the context of code comprehension. Drawing on principles from Hermans (2021), students were given an executable Python script generated by an LLM that purported to compare stochastic gradient descent and batch gradient descent on the Iris dataset. Although the code executed without syntax errors, it contained a logical flaw that undermined the validity of the comparison.

Students were tasked with reading the code, articulating its intended purpose, and identifying the source of the logical error. This required understanding both the mechanics of the code and the underlying optimization concepts. After identifying the issue, students proposed and implemented corrections, then evaluated whether the revised output aligned with theoretical expectations. This activity emphasized that executable code is not necessarily logically correct code.

Systems Thinking Using Prompt Engineering

The third instructional example focuses on systems thinking and ethical reasoning through prompt engineering. Students were asked to design a conceptual AI system for a domain such as healthcare using structured prompts. After defining the system's functionality, students identified at least three ethical dilemmas associated with its deployment and analyzed the corresponding regulatory implications. Following Walter (2024), this activity emphasized that AI literacy includes understanding societal impacts and ethical constraints. A reflection completed without AI assistance asked students how considering ethical dilemmas altered their perception of the system's design and feasibility.

Reflective Observations

The following observations are based on informal instructor reflection and in-class discussion across two semesters of implementation. They are intended to illustrate how students typically engaged with each activity and to highlight design choices that appeared productive. These observations are qualitative and anecdotal; they do not constitute systematic research findings.

Critical Reading and Evaluation of AI-Generated Text

In the critical reading activity, many students initially approached the AI-generated summary as largely reliable, focusing on surface-level clarity rather than conceptual accuracy. However, when asked to explicitly compare the AI output with the original academic article, students began to identify missing assumptions, oversimplified explanations, and imprecise use of disciplinary language. This comparison prompted closer reading of the source material than is typically observed when students read technical articles in isolation.

As discussions progressed, students increasingly justified their critiques by referencing specific passages, definitions, or mathematical arguments from the original text rather than relying on intuition or general impressions. Instructor observation suggests that students became more aware of how fluent language can obscure conceptual gaps, reinforcing the importance of disciplinary knowledge when evaluating AI-generated summaries.

Code Comprehension and Logical Reasoning

The code analysis activity revealed a progression from surface-level acceptance to deeper logical scrutiny. Although the AI-generated Python script executed without syntax errors, many students initially assumed that a working program implied a valid comparison of optimization algorithms. Through guided code reading and discussion, students identified inconsistencies between the stated goal of the script and its implementation.

Students demonstrated reasoning by articulating why the code's output was misleading, proposing targeted corrections, and evaluating revised results against theoretical expectations discussed in class. Collaborative discussion supported this process, as students compared interpretations and refined their understanding collectively.

Systems-Level and Ethical Awareness

In the systems thinking activity, students initially focused on technical feasibility and performance-oriented design choices when prompted to conceptualize an AI system. When asked to identify ethical dilemmas and regulatory implications, students noted a noticeable shift in perspective, recognizing how data sources and modeling decisions introduce biases that are not apparent in purely technical solutions.

The reflection component, completed without AI assistance, indicated that students increasingly framed system design as a set of interconnected decisions rather than isolated technical steps. Many students noted that considering ethical and regulatory requirements altered their assessment of system feasibility and improved their awareness of trade-offs between accuracy, transparency, and accountability.

In summary, across all three instructional examples, instructor observation suggests that positioning AI-generated content as an object of critique, rather than a source of answers, can meaningfully support the development of critical thinking, code literacy, and systems-level reasoning in data-centric disciplines. A formal study is needed to substantiate these observations empirically.

Discussion

The instructional experience from these examples suggests that the educational value of GenAI in the classroom is strongly shaped by how it is positioned within learning activities. When AI tools are framed as sources of critique rather than answer generators, they can support higher-order cognitive skills that are central to data science and engineering education, including critical reading, code comprehension, and systems-level reasoning.

A key implication of this work is that concerns about over reliance on AI and erosion of critical thinking are not inherent to the tools themselves, but to instructional design choices. In the activities presented, students were not rewarded for producing polished outputs with AI assistance; instead, they were required to interrogate the assumptions, logic, and implications of AI-generated content. This shift re-centers learning around disciplinary judgment.

The examples also highlight the pedagogical value of AI-generated content precisely because it is often fluent yet imperfect. Unlike expert-written textbook examples, AI outputs frequently contain subtle conceptual gaps or logical inconsistencies that are difficult to detect without domain knowledge, making them effective materials for practicing evaluation and verification skills.

Another important consideration is the role of instructor mediation. While the activities were designed to require minimal additional infrastructure, their effectiveness appeared to depend on structured prompts, guided discussion, and explicit expectations about student reasoning. This suggests that successful integration of GenAI into coursework does not eliminate the need for instructional scaffolding; rather, it shifts the instructor's role toward designing tasks that make reasoning visible.

This work also has implications for academic integrity and assessment. By emphasizing critique, justification, and reflection — particularly components completed without AI assistance — the activities reduce incentives for uncritical AI use while maintaining alignment with learning objectives.

Limitations and Future Work

Several limitations should be acknowledged. The observations reported here are informal and drawn from a limited set of courses within data-centric disciplines. They do not claim generalizability or measure learning gains quantitatively. Student experiences may vary based on prior preparation, familiarity with AI tools, and disciplinary context. The absence of systematic data collection — including student work samples, structured reflection prompts, or pre/post assessments implies that the observations cannot support causal claims about student learning.

A formal empirical study with IRB approval is planned for the later semester. This study will include structured data collection methods such as coded analysis of student reflections, pre/post concept assessments, and rubric-scored artifacts. The goal will be to move from the descriptive, practice-oriented claims made here toward evidence-based conclusions about the instructional effectiveness of critique-centered AI integration.

Conclusion

This practice report presented a set of GenAI-based instructional activities designed to foster critical thinking and systems thinking in higher education. The activities illustrate how AI can be integrated into instruction not as an answer generator, but as a tool for reflection, critique, and learning. For instructors considering how to respond to the increasing presence of generative AI in their classrooms, this work suggests a shift in emphasis rather than a wholesale change in curriculum.

Designing activities in which students are asked to read, evaluate, and correct AI-generated explanations, code, or system designs allows instructors to foreground disciplinary judgment, metacognition, and ethical reasoning without requiring extensive new tooling or course redesign. This framing reduces the need for constant policing of AI use and instead re-centers the classroom around skills that remain fundamentally human: critical evaluation, contextual understanding, and responsible decision-making. Future empirical work will assess learning outcomes more rigorously across disciplines and explore how similar approaches might be adapted for other fields.

References

Adeyele, Victoria Olubola, and Umesh Ramnarain. 2024. "Exploring the Integration of ChatGPT in Inquiry-Based Learning: Teacher Perspectives." *International Journal of Technology in Education* 7 (2): 200–217.

Bastani, Hamsa, Osbert Bastani, Alp Sungu, Haosen Ge, Ozge Kabakcı, and Rei Mariman. 2024. "Generative Ai Can Harm Learning." *Available at SSRN* 4895486.

Bien, Jacob, and Gourab Mukherjee. 2024. "Generative AI for Data Science 101: Coding Without Learning to Code." *Journal of Statistics and Data Science Education*, no. just-accepted: 1–12.

Bray, Robert. 2023. "Lessons Learned When Teaching Data Analytics with ChatGPT to MBAs in Spring 2023."

Cotton, Debby RE, Peter A Cotton, and J Reuben Shipway. 2024. "Chatting and Cheating: Ensuring Academic Integrity in the Era of ChatGPT." *Innovations in Education and Teaching International* 61 (2): 228–39.

Dugan, Kelley E, Erika A Mosyjowski, Shanna R Daly, and Lisa R Lattuca. 2022. "Systems Thinking Assessments in Engineering: A Systematic Literature Review." *Systems Research and Behavioral Science* 39 (4): 840–66.

Ellis, Amanda R, and Emily Slade. 2023. "A New Era of Learning: Considerations for ChatGPT as a Tool to Enhance Statistics and Data Science Education." *Journal of Statistics and Data Science Education* 31 (2): 128–33.

Hermans, Felienne. 2021. *The Programmer's Brain: What Every Programmer Needs to Know about Cognition*. Simon; Schuster.

Koral Kordova, Sigal, Moti Frank, and Anat Nissel Miller. 2018. "Systems Thinking Education—Seeing the Forest Through the Trees." *Systems* 6 (3): 29.

Mollick, Ethan, and Lilach Mollick. 2023. "Assigning AI: Seven Approaches for Students, with Prompts." *arXiv Preprint arXiv:2306.10052*.

Pankiewicz, Maciej, Andres Felipe Zambrano, Amanda Barany, and Ryan S Baker. 2024. "How We Code Code: Leveraging GPT and Ordered Networks for Understanding Introductory Programming Education." In *International Conference on Quantitative Ethnography*, 225–40. Springer.

Paul, Richard W. 2018. "Critical Thinking and the Critical Person." In *Thinking*, 373–403. Routledge.

Rudolph, Jürgen, Samson Tan, and Shannon Tan. 2023. "ChatGPT: Bullshit Spewer or the End of Traditional Assessments in Higher Education?" *Journal of Applied Learning and Teaching* 6 (1): 342–63.

Schoenbach, Ruth, Cynthia Greenleaf, Christine Cziko, and Lori Hurwitz. 1999. *Reading for Understanding*. San Francisco: Jossey-Bass.

Strang, Gilbert. 2018. "Multiplying and Factoring Matrices." *The American Mathematical Monthly* 125 (3): 223–30.

Tu, Xinming, James Zou, Weijie J Su, and Linjun Zhang. 2023. "What Should Data Science Education Do with Large Language Models." *arXiv Preprint arXiv:2307.02792* 3.

Walter, Yoshija. 2024. "Embracing the Future of Artificial Intelligence in the Classroom: The Relevance of AI Literacy, Prompt Engineering, and Critical Thinking in Modern Education." *International Journal of Educational Technology in Higher Education* 21 (1): 15.

Xie, Ying, Shaoen Wu, and Sumit Chakravarty. 2023. "AI Meets AI: Artificial Intelligence and Academic Integrity-a Survey on Mitigating AI-Assisted Cheating in Computing Education." In *Proceedings of the 24th Annual Conference on Information Technology Education*, 79–83.

Xing, Yixun. 2024. "Exploring the Use of ChatGPT in Learning and Instructing Statistics and Data Analytics." *Teaching Statistics* 46 (2): 95–104.

Ziegler, Laura, and Joan Garfield. 2018. "Developing a Statistical Literacy Assessment for the Modern Introductory Statistics Course." *Statistics Education Research Journal* 17 (2): 161–78.

APPENDIX I

Critical Thinking and Reading in the Disciplines

Overview of Lesson: The students are tasked with reading a technical article and subsequently reviewing an AI-generated summary of the article to identify if there are errors in the generated content. The intention with the lesson is to encourage students to deepen discipline specific reading skills.

Learning Objectives: After completing this lesson, the student will be able to:

- Improve reading comprehension of technical article in the discipline
- Review the AI-generated summary of the technical article
- Apply critical thinking to evaluate the AI generated summary

Audience: Upper division undergraduate and graduate students interested in developing mathematical understanding behind AI models. Prerequisites: Prior to this lesson, students should be familiar with:

- Foundational mathematical concepts required for the article, example that was implemented in the class was based on Strang (2020),
- Reading strategies for a technical article.

Time Required: The activity would require 60 - 90 minutes. This activity should be preceded by a lesson on foundational concepts required to understand the article. The activity can be broken into two parts interspersed with discussion:

- Reading and understanding article
- Evaluating AI generated summary of the article

Technology and Other Materials: Technology: Computer and access to document reading tools. The NotebookLM-generated overview used in this implementation is available at: https://notebooklm.google.com/notebook/e1ab9595-9503-40c5-817b-31944424e8b0?artifactId=003b4000-942e-4b28-bcd9-857bd55492f3

Lesson Plan: The lesson is designed to assess data science literacy, defined here as the ability to read, apply critical thinking, understand, and communicate statistical and computational information (adapted from Ziegler and Garfield (2018)). Critical thinking is defined per Paul (2018).

First Stage of the Lesson - Read: Students employ technical reading skills (e.g., think-aloud or pair reading) to read the shared article.

Second Stage of the Lesson - Review: Students review the AI-generated content in pairs and critically evaluate it for accuracy, going back and forth between the article and the AI summary.

Third Stage of the Lesson - Share and Reflect: Whole-class discussion on observations and reflections.

Assessment Criteria: Instructors may assess student work using the following criteria

Identification of errors	
Developing	Identifies surface-level errors only (e.g., minor wording issues)
Proficient	Identifies at least one substantive conceptual inaccuracy with a reference to the original text
Exemplary	Identifies multiple conceptual inaccuracies and explains their significance for understanding the discipline
Quality of justification	
Developing	Critiques are vague or unsupported
Proficient	Critiques are supported by specific passages or arguments from the original article
Exemplary	Critiques are well-reasoned, referencing mathematical or domain-specific arguments from the source
Depth of reflection	
Developing	Reflection is superficial or restates the activity description
Proficient	Reflection is superficial or restates the activity description
Exemplary	Reflection shows metacognitive awareness of how AI fluency can mask conceptual gaps

Appendix II

Critical thinking and Reading Code Written by GenAI

Overview: Students review an executable Python code snippet for data analysis generated by an LLM and use critical thinking skills to identify and fix logical fallacies. The lesson assesses data science literacy and code comprehension skills.

Learning Objectives: After completing this lesson, the student will be able to:

- Apply code-reading skills to understand a piece of code written by another person or entity.
- Identify logical flaws in executable code.
- Apply critical thinking in evaluating code output against intended goals.
- Perform appropriate fixes to produce an output that is logically valid and matches expectations.

Audience and Prerequisites: Tested in two graduate courses: Introduction to Machine Learning and AI Tools and Processes. Students should have experience fitting a linear or logistic regression using Python and awareness of nonlinear optimization techniques.

Time Required: 30–60 minutes, preceded by a lesson on gradient-based optimization routines. Broken into: (1) reading and understanding the LLM-generated code; (2) fixing the code.

Technology and Materials: Computer and access to Python (local or cloud). The code snippet used for this lesson is available at: https://colab.research.google.com/drive/1i4jodzzl8HqRH9YTJwvwvkTbBLMwIX4

Lesson Plan: GenAI can generate code snippets for specific tasks with a short prompt. In such a situation, it becomes critical for students to assess the validity of the output. This lesson provides an opportunity to practice code-reading skills and critically understand the output generated.

First Stage of the Lesson - Read: Students spend 10 minutes reading the shared code snippet and recording what they believe the code's intention is. The teacher then shares the intended context.

Second Stage of the Lesson - Evaluate: Students are asked to identify and fix the fallacy in the code (~15 minutes).

Third Stage of the Lesson - Share and Reflect: Class discussion on the approach students took, followed by a round-robin sharing of takeaways.

Assessment Criteria: Instructors may assess student work using the following criteria

Code reading comprehension	
Developing	Student describes the code at a surface level only (e.g., 'it compares two methods')
Proficient	Student correctly identifies the intended goal and the key steps in the implementation
Exemplary	Student identifies the intended goal, traces the logic step-by-step, and flags potential ambiguities
Logical error identification	
Developing	Student does not identify the error, or identifies the wrong part of the code
Proficient	Student correctly identifies the logical fallacy and explains why it invalidates the comparison
Exemplary	Student identifies the error, explains the underlying conceptual issue, and connects it to course theory
Correction and validation	
Developing	Student attempts a fix that does not resolve the logical error
Proficient	Student implements a correct fix and produces output that aligns with expectations
Exemplary	Student implements a correct fix, validates output against theoretical expectations, and articulates what changed and why

Appendix III

Systems Thinking using Prompt Engineering

Overview: Students learn to write prompts iteratively to practice systems thinking. The lesson makes students aware of the interrelationships and patterns of events in a data science project rather than isolated snapshots of approaches, using a large language chatbot as a collaborator.

Learning Objectives: After completing this lesson, the student will be able to:

- Differentiate between zero-shot and few-shot prompting techniques.
- Understand the different aspects of designing a data science project from a systems perspective.
- Identify ethical dilemmas and regulatory implications of an AI system design.
- Reflect independently on how ethical considerations alter technical design decisions.

Audience and Prerequisities: Students in data disciplines (data science, machine learning, AI) or students in other disciplines interested in using machine learning or deep learning models. No prerequisites required beyond basic familiarity with LLM chatbots.

Time Required: 30–60 minutes, preceded by a lesson on different prompting techniques (see Walter, 2024). Broken into three stages with discussion.

Technology and Materials: Computer and access to a large language chatbot (e.g., OpenAI's ChatGPT, Anthropic's Claude, or Google's Gemini).

Lesson Plan: The lesson uses Dugan et al. (2022) definition of systems thinking: 'a discipline for seeing wholes, a framework for seeing interrelationships and repeated patterns of events rather than just isolated incidents.' Koral Kordova, Frank, and Nissel Miller (2018) argue that by just providing an algorithm-focused education alone cripples students in executing multi-disciplinary projects.

First Stage of the Lesson - Zero Shot Prompt: Teacher shares an example zero-shot prompt. Students spend 10 minutes creating their own zero-shot prompt for a scenario or industry of their choosing. Teacher then thinks aloud to highlight gaps in the generated response.

Second Stage of the Lesson - Few Shot Prompt: Teacher demonstrates how the zero-shot prompt can be enhanced with details. Students create a few-shot prompt for their scenario.

Third Stage of the Lesson - Reflect (without AI): Students are explicitly asked not to use an LLM as they reflect on their learning. Conducted orally in a round-robin manner or as a written assignment.

Example Assignment (Data Ethics Course): **Part 1: Zero-Shot** (use as prompt or modify)

Imagine you are a data scientist asked to design an AI system for healthcare. Choose one application (examples: predicting hospital readmission, early cancer detection, personalized medication recommendations, triaging ER patients). Describe the system as if you were focused only on technical goals. Specify design decisions such as:

- Problem framing: What is the AI predicting or recommending?
- Data: What sources will you use (EHR, imaging, genomics, etc.)?
- Features: What types of variables will you include?
- Modeling: What kind of model might you choose and why?
- Evaluation: What metric would you optimize for?

Keep your answer focused on accuracy, efficiency, or performance — not ethics.

Part 2: Few-Shot (use as prompt or modify)

Now revisit your system from Part 1. Imagine your design is about to be deployed in a real hospital. Identify at least three ethical dilemmas AND their regulatory implications. Consider:

- EU AI Act: Does your system qualify as high-risk AI? How would you ensure compliance?
- CCPA/CCRA: If California patients' data are included, how would you handle their rights?
- Bias & fairness: Could data or features introduce inequities?
- Consent & explainability: How would you balance accuracy with the need for meaningful explanations?

Part 3: Reflection (without AI)

Please answer the following in your own words:

- How did your perspective on the AI system change once you considered ethical dilemmas?
- Which design decision do you now see as most ethically significant, and why?

Assessment Criteria: Instructors may assess student work using the following criteria

Prompt quality and iteration	
Developing	Zero-shot and few-shot prompts are nearly identical; little evidence of iteration
Proficient	Few-shot prompt adds meaningful constraints that produce a richer LLM response than the zero-shot
Exemplary	Few-shot prompt demonstrates deliberate systems-level thinking; student articulates what they added and why
Ethical dilemma identification	
Developing	Fewer than three dilemmas identified, or dilemmas are vague and not tied to the specific system design
Proficient	Three or more specific dilemmas identified with reference to regulatory frameworks (EU AI Act, CCPA, etc.)
Exemplary	Dilemmas are specific, well-explained, and connected to concrete design decisions; trade-offs are analyzed
Independent reflection quality	
Developing	Reflection is superficial, brief, or shows no change in perspective
Proficient	Reflection demonstrates a meaningful shift in how the student perceives the system's design or feasibility
Exemplary	Reflection shows sophisticated systems-level thinking; student articulates specific design decisions they would change and why