Final project

Rubric

The class includes a final project. This project should build off the course content. The project is graded as follows:

- Technical accomplishment: 30%
- Correctness and code quality: 20%
- Utility: 10%
- Documentation: 20%
- Presentation/report quality: 20%

Group Size

Students may elect to complete the project individually or in teams of two, three, or four. If you have more than four in the group, you should seek instructor permission. Projects with more group members will be graded with greater expectations.

Technical Accomplishment: The following acts or features will count towards technical accomplishment in cumulative fashion. The more of these things you incorporate, the higher your technical accomplishment score will be, but you are of course not expected to have all or even most of these items. Incorporating 4-5 of these items will likely achieve a maximum technical score.

- Hosting a final project online so that it is accessible over the web in a way that is not a Jupyter notebook or similar. This will usually be a web application.
- Using search augmented retrieval
- Using conversational chat
- Using multiple APIs or data sources
- Using multiple models
- Using multiple media (e.g. images or audio with LLMS)
- Using multiple calls to the same API
- Resistance to prompt injection attacks
- Building sophisticated agent workflows (hierarchical or asynchronous agents)
- Using non-standard tools (agents)

Utility is roughly speaking the usefulness of your project to humanity: the degree to which there exist living breathing humans who have a problem solved by your project and the degree to which those people might have the ability and willingness to pay for your solution. Of course, “pay” in this sense might be “publicly fund” if you’re working on a project with social benefit.

Correctness is roughly speaking the degree to which your code is both logically and mathematically correct and without errors. Does it do the thing you set out to do in the best way possible? Have you handled edge cases? Code quality includes standard indicia such as unit tests, docstrings, type hints, small function sizes, sensible abstractions, and the clear use of version control.

Documentation includes both code and end user documentation.

Deliverables:

- **Main deliverable:** A written document describing the purpose and design of your project or app. *This must include appropriate citations to both academic work and software used in the project. It should include details of the code*.
    - Please take care to communicate extremely clearly regarding what you view as the technical accomplishment and utility.
- A brief in-class presentation on the last day of class **OR** a brief video accessible to instructors in which you demonstrate use of the application or project.
    - Groups that both volunteer and are selected to present in class will receive 5 bonus points.
    - The volunteer sign up sheet will be distributed at a later date.
- Students will fill out a form evaluating their own contributions and the contributions of their teammates in terms of effort and quality. The form will be provided.

All materials are due at 1:00pm on December 9th.