

# CS 175 (W26): Project in Artificial Intelligence

## Project Proposal

Due date: Friday, January 23, 2026 (Pacific Time)

Roy Fox

<https://royf.org/crs/CS175/W26>

For the proposal, you have to finalize your team roster, propose an idea for your course project, describe the evaluation plan, and set up a project website with a page for this submission.

### Part 1 Setting a Repository and a Website (10 points)

For the project that you will be working on this quarter, we expect you to maintain a public GitHub repository for the code, and an associated website. You can find a detailed step-by-step guide below, and feel free to reach out with any questions.

#### Part 1.1 Setting up the Repository

Here are steps to create a new repository on GitHub:

1. Pick a short, catchy name for your team/project. This should be the name of the group on Canvas, the name of the repository on GitHub, and we will refer to it below by PROJECT. You can change it later, if needed.
2. If you don't have a GitHub account, sign up for one for free, and ask all of your teammates to sign up for one as well. If you're unfamiliar with `git`, read some tutorials on how it works<sup>1</sup>, and get comfortable with it, as you will be using it to maintain the website and the code for the project.
3. To create a repository that will store your project code and website, pick one of the group members under whose account the repo will reside. We will refer to this user below by the GitHub username USER. (Advanced users are welcome to create the repo under a shared group account or an organization, if you like, but do connect each of your individual accounts.)
4. That person should log in, go to the + on the top right, and click *New repository*. Name your repository by the group/project name (the PROJECT above). Make sure the owner is USER, make it public, click *Add a README file*, and then click *Create repository*. Congrats, your repository should now be created with a URL like `https://github.com/USER/PROJECT!`
5. To add your group members to the repository, go to *Settings* tab, *Collaborators* page (under *Access*), click *Add people* under *Manage access*, enter their GitHub IDs, and *Add (them) to this repository*. They should be able to commit/edit files in the repo once they accept the invitation.

---

<sup>1</sup>Here are two: <https://try.github.io> and <https://guides.github.com/activities/hello-world>.

## Part 1.2 Setting up the Website

Note: advanced users can ignore the following instructions and create their own *GitHub Pages* page. The only constraints are: (1) your webpage should present the docs/ folder in the master branch of your repo, (2) it should have one dedicated page or tab for the team and another for the proposal (and later, for the progress and project reports), and (3) the links to these pages should be obvious on the project's homepage.

1. Download the zip file at <https://royf.org/crs/CS175/W26/gh-skeleton.zip>.
2. Unzip the file. Inside the docs/ folder, you will find 2 folders and a total of 5 files.
3. Create a docs/ folder in your project repo, and commit the files from the above docs/ folder into it. It is easiest to do it on your machine after checking out your project locally, or you can do it through GitHub's web interface.
4. On GitHub, in *Settings* again, go to the *Pages* page (under *Code and automation*), and select the master branch and the /docs folder. Click *Save*.
5. Now open a new tab, and go to <https://USER.github.io/PROJECT>. There should be a website there (may take a few minutes to come up).

Now that your website is up, personalize it. Through any teammate's account, do the following:

1. Edit `_config.yml` and change the title from PROJECT to the name of the repository (do not make other changes to this file unless you know what you're doing). Again, you can make these edits on the GitHub website, or preferably after checking out the repo locally. A few minutes after the commit (and the push, if committed locally), your website should change to reflect the new name on the front page and navigation bar.
2. (Optional) Edit `index.md` to list more information. Do not remove or edit the first four lines (the stuff in between and including the ---, often called *front matter* or *preamble*) unless you know what you are doing. You should remove the default text on this page that has some basic information about Markdown, and a link to a complete guide on how to format using Markdown.
3. Edit `team.md` and replace the placeholders with your teammates and their UCINetIDs. You can optionally make more edits to add information (photos, homepage/GitHub links, whatever you'd like) or make it look prettier.

Again, if you encounter a problem you are unable to solve, please reach out on the course forum, making sure you detail what you tried and what indication of error you're seeing.

## Part 2 Contents of the Proposal

### Part 2.1 Creating a Proposal Page (10 points)

You will notice that there is a greyed out link to the proposal in the navigation bar, that gives an error if you click on it. This will change once you create a docs/proposal.md file with the following front matter as the first four lines:

```
1 ---
2 layout: default
3 title: Proposal
4 ---
```

The rest of the file should contain the rest of the proposal, using the Markdown format (headings start with ##, one heading for each section below, and regular text without any formatting). Commit and push your new file, and the link should become active once GitHub updates your website in a few minutes.

## Part 2.2 Summary of the Project (20 points)

In a paragraph or so, describe the main idea behind your project. Focus on the problem setup, not the solution, i.e. what is your overall goal? If your project is application-driven, what is the application, i.e. the task or behavior that you want the learner / agent to have? If your project is method-driven, what is the method aiming to achieve and what is an example baseline method? At the very least, you should have a sentence that clearly explains the input/output semantics of your project, i.e. what information will it take as input, and what will it produce.

## Part 2.3 Project goals (20 points)

In a sentence or two each (optionally in a bulleted list), outline 3 levels of goals for the project. Each level can consist of a single task or behavior that you aim for the agent to do, or a single question you aim to answer; or if not single then at most a couple related ones. The levels are:

- **Minimum goal:** your most basic effort to get something done and working. If you only do this, you'll have positive learning experience that's worth your time.
- **Realistic goal:** your good and reasonable effort to do something cool and interesting. If you do all of this, you'll have a solid project you can be really happy about.
- **Moonshot goal:** your best effort and dream result, if everything happens to go really well. This may not be realistic, although you could be pleasantly surprised, and anyway the point of this is a "compass" of where the really amazing stuff could be.

## Part 2.4 AI/ML Algorithms (10 points)

In a single sentence, name the AI/ML algorithm(s) and method(s) you anticipate using for your project. It does not have to be a detailed description of any algorithm, even just the name of a class of methods is sufficient. Examples of this include "planning with dynamic programming", "reinforcement learning with neural function approximator", "learning from demonstrations", "min-max tree search with pruning", and so on. You can take a guess at properties of the method that you'll prefer using, such as model-free vs. model-based, on-policy vs. off-policy, etc. You won't lose points for suggesting an incorrect algorithm or method, and you'll be able to switch to different ones at any point that it makes sense later in the project.

## **Part 2.5 Evaluation Plan (25 points)**

As described in class, mention how you will evaluate the success of your project. In a paragraph, focus on the quantitative evaluation: what experiments you run in order to evaluate, what the metrics are, what the baselines are (naïve approaches that you'll try first), by how much you estimate that your approach could improve the metric (just a ballpark), etc. In another paragraph, describe what qualitative analysis you will show to verify the project works, such as the debugging, sanity, and toy cases for the approach, how you will visualize the results (internal and/or external) of the algorithm to verify it works, what qualitative properties you expect of a successful result, etc.

## **Part 2.6 AI Tool Usage (5 points)**

AI tools are becoming good at some things, not yet at others. Using them for what they're good for and watching out for where they fall short is probably something you're already doing and may choose to do during your project. Our policy in this course is to allow any use of such tools for your projects. The only requirement is that you record and report all aspects of the project in which you used AI tools. Please indicate this clearly in a separate section of your proposal, or otherwise indicate that no AI tools were used.

You will not lose points in any part of this course for using AI tools smartly, but you may lose points if you choose to blindly rely on subpar output of such tools — and be warned, they often fail in ways ranging from obvious to subtle, and their failures may be a time sink to identify. In addition to “hallucinations”, two high-level usage patterns to be careful about are: The Takeover — you contribute nothing substantial to the project, at which point ask yourself, if AI can really do your entire job for you, what you are gaining by pretending otherwise; and The Reversal — you ask the AI for detailed instructions what to do and your only contribution is to then go and do it, in which case it's no longer your AI tool, but rather you're its human tool. My take: at this point in time, the concern is not that the above becomes true but rather that you perform your project as if it is.