Hi all! This is a Google Doc where we will track the schedule, plans, and progress for the inaugural Build Your Own Proof Assistant class. It will also serve as a syllabus.

Format **1**
Specifics 1
Backing Up 2
Schedule **2**
Infrastructure **4**
Grading **4**

Format

The goal of this course is to build a proof assistant together. A general-purpose proof assistant is a lifetime project, so we will instead target some **specific domain and audience**, effectively building a **domain-specific proof assistant.** We will collectively decide which domain and audience to target in the first week.

Specifics

The schedule will consist of a **planning period**, followed by six **"sprints"** with particular subgoals in mind. Early sprints might include the whole class as one team to get the basics down (for example, we might want everyone to be familiar with the foundations of the proof assistant we build, or to participate in its initial high-level design), but later sprints will divide the class into smaller teams. The last class of each sprint will be a **"retrospective"** meeting during which we will reflect on our progress and plan more for the next sprint. This may mean adapting our goals as we go, and making sure we choose realistic enough **"minimum viable products" (MVPs)** to target to begin with.

Within each sprint, each team will have a few dedicated roles:

1. **Managers** will be responsible for ensuring that the work actually gets done for that sprint, and that the team stays on task. Managers will also report team progress to me.
2. **Technical Leads** will be responsible for tying up technical loose ends in between classes, in order to make sure the team meets its technical goals for the sprint to the degree that is possible (or otherwise communicates why it is not possible).
3. **Technical Program Managers (TPMs)** will be responsible for cross-team communication about technical details, for example to help sync up on changes that are needed to common interfaces or to goals to account for unforeseen circumstances.

Rolls will rotate. To get full credit for the course, you will need to **assume each role once**, each for the duration of **a full sprint**. There should only really be obligatory homework when you assume one of those roles, so please try to schedule those for weeks in which you know you will be able to work on this. There will usually be one person in each role, but as needed to make sure everyone can sign up, sometimes two people can fill the same role.

During the other weeks in which you do not assume any particular role, to get credit, you will still need to **join a team**. On that team, you will either need to show up and **participate** in the sprint, or you will need to let **both me and your manager** know that you will be on **"vacation"** for part or all of that sprint (for example, to attend a conference, or when you are sick; I don't want to delineate between those cases for student privacy reasons). You can take **as many vacations as you want**, but you need to **communicate** them in order to get credit. (If everyone takes vacation every week, I will change this, but for now I am just hoping for good faith. Also, if you are sick when you have a leadership role, let me know and we will work something out.)

If you would like to participate **remotely** during any given sprint, but don't want to take a vacation for that sprint, please work with your team's **manager** to find a way to do so.

The final week will be tying up **loose ends**. There are no special roles for that week, but you still must either **participate** or tell me that you are taking a **vacation** in order to get full credit

Backing Up

The structure of this class is meant to mirror how the software industry often operates, based on my time as a software engineer before academia. This is a good experience, but it is also hard! In the real world you get paid good money for this kind of thing, and even then you still often miss your goals. **So it is OK if we do not always meet our goals.** This is about learning.

With that said, if the project goes well enough, we can all **publish** the result in the end, or we can **continue** the project after the semester is over. But if not, we will **learn** a lot about how to work together on a large collaborative project, while also learning about the guts of proof assistants and how they work. **Both are successes to me!**

Schedule

This is where we will update the schedule based on our project plans. **We will start this planning process during the first week of class.** I will also assess student backgrounds during the first week of class to get an idea of whether any background lectures will be needed, and to get an idea of how to help keep teams balanced and goals and schedule realistic.

Date	Goals	Team 1	Team 2	Team 3
Jan. 21st	1. Assess backgrounds 2. Describe class goals and structure 3. Remind folks what a proof assistant is, and what its components are 4. Discuss target domain ideas	N/A	N/A	N/A

Jan. 26th	1. Discuss backgrounds and implications 2. Choose target domain 3. Sketch MVP/initial version of final goals	N/A	N/A	N/A
Jan. 28th	1. Plan 1st sprint goals 2. Plan 1st sprint teams	N/A	N/A	N/A
Feb. 2nd	**1st Sprint, Day 1**			
Feb. 4th	**1st Sprint, Day 2**			
Feb. 9th	**1st Sprint, Day 3**			
Feb. 11th	**1st Sprint Retro** 1. Reflect on progress 2. Adapt goals 3. Plan next sprint			
Feb. 16th	**2nd Sprint, Day 1**			
Feb. 18th	**2nd Sprint, Day 2**			
Feb. 23rd	**2nd Sprint, Day 3**			
Feb. 25th	**2nd Sprint Retro** 1. Reflect on progress 2. Adapt goals 3. Plan next sprint			
Mar. 2nd	**3rd Sprint, Day 1**			
Mar. 4th	**3rd Sprint, Day 2**			
Mar. 9th	**3rd Sprint, Day 3**			
Mar. 11th	**3rd Sprint Retro** 1. Reflect on progress 2. Adapt goals 3. Plan next sprint			
Mar. 16th	Spring Break	N/A	N/A	N/A
Mar. 18th	Spring Break	N/A	N/A	N/A
Mar. 23rd	**4th Sprint, Day 1**			
Mar. 25th	**4th Sprint, Day 2**			
Mar. 30th	**4th Sprint, Day 3**			

Apr. 1st	**4th Sprint Retro** 1. Reflect on progress 2. Adapt goals 3. Plan next sprint			
Apr. 6th	**5th Sprint, Day 1**			
Apr. 8th	**5th Sprint, Day 2**			
Apr. 13th	**5th Sprint, Day 3**			
Apr. 15th	**5th Sprint Retro** 1. Reflect on progress 2. Adapt goals 3. Plan next sprint			
Apr. 20th	**6th Sprint, Day 1**			
Apr. 22nd	**6th Sprint, Day 2**			
Apr. 27th	**6th Sprint, Day 3**			
Apr. 29th	**6th Sprint Retro** 1. Reflect on progress 2. Adapt goals 3. Plan next sprint			
May 4th	**Loose Ends, Day 1**			
May 6th	**Loose Ends, Day 2**			

Infrastructure

Once the class starts, we will add some common infrastructure here (e.g., Discord channels for project communication, Trello boards, planning documents, or GitHub repositories).

Grading

Grading for this class is participation-based, with the following breakdown:

- Serve as a manager for at least one sprint: 20%
- Serve as a technical lead for at least one sprint: 20%
- Serve as a TPM for at least one sprint: 20%
- Participate (or take vacations) for the other sprints: 30% total (10% each)
- Participate (or take vacation) in the loose ends week: 10%

Auditing is welcome! Both officially and unofficially. Auditors are not required to do anything in particular, though they may also assume leadership roles if everyone who is taking the course for credit is signed up for them already and we have extra slots available.