Team info

Team name	Member 1	Member 2	Member 3	
seg.bio	Hansen Pan	Boyu Shen	Tian-hao Zhang	
Front-row Vanguard	Andrew Ren	Tae Ahn		
ActionCopilot	Yunhan Liu	Mengyan Cao	Tianyi Li	
ClearFeed	Nathan Shea			
Mark's Readers	Tianli Qu	Alexandra Sklyarova	Kai Gowers	
LaTech	Franco Z.	Andrew Boessen		
Sole-utions Lab	Alex Lehman	Xander W.	Julie H.	
PetDash	Theodore Utomo			
StyleX	Jasroop Dhingra			
Clutch	Sanjeet Brar			
Sport Scope	Zhiyuan Li	Tim Hays	Peixin Yang	

Final Project Schedule

Date	Task	Final Project Schedule
9/2/25	Week 1 Brainstorm startup ideas	Complete Survey
9/4/25		
9/9/25	Week 2 Form teams	
9/11/25		
9/16/25		
9/18/25		Team info due
9/23/25	Week 4 Initiate startup ideas and confirm with the team	
9/25/25		Problem statement due (250-350 words)
9/30/25	Week 5 Customer Discovery	
10/2/25		
10/7/25	Week 6-7 MVP Scoping	
10/9/25		*Mid-term Exam*
10/14/25		*Fall Break*
10/16/25		Customer profile summary due (one-page)
10/21/25	Week 8-10 Prototyping & Tech build	
10/23/25		Elevator Pitch
10/28/25		Product requirements document due (one-page)
10/30/25		
11/4/25		
11/6/25		
11/11/25	Week 11-12 Refinement based on feedback	
11/13/25		
11/18/25		
11/20/25		

11/25/25		
11/27/25		*Thanksgiving*
12/2/25		Project Presentation I
12/4/25		Project Presentation II

Assignment details:

Problem statement (Due 9/25)
A problem statement is a concise, evidence-based description of a specific user's pain in its real context. It contrasts the current state with a clearly desired state, names the gap (what blocks progress), explains why now, and defines a minimal, testable path (MVP) and success criteria. It is not a pitch or feature list; it's a falsifiable claim you can validate with users.

Write a crisp problem statement that frames your startup idea and a focused MVP (250-350 words).

Format:

- User & context
- Problem (current vs. desired state)
- Why this problem
- What the MVP will look like
- How will the team work together to succeed
- Desired outcome (end goal)

Customer profile summary

- Conduct customer interviews (at least 3); analyze pain points and market potential.
    - To understand whether their initial product ideas match the market needs
- If the customer interview does not apply, read at least 3 industry-related research papers/articles to understand the market; summarize and analyze the articles for the profile summary

Product requirements document
A Product Requirements Document (PRD) is a blueprint for building a product. It defines *what* needs to be built, *why* it needs to be built, and *how* it should function from the user's perspective.

- Overview / Purpose
    - What the product or feature is
    - Why it is being built (the problem statement)
    - Link to business goals, strategy, or customer needs
- Objectives & Success Metrics

    - What success looks like (KPIs, OKRs, measurable outcomes)
    - Example: “Reduce onboarding drop-off rate from 30% → 15% within 3 months.”
- Target Users & Use Cases
    - Who the users are (personas or customer segments)
    - Main use cases or user stories (e.g., *As a student, I want to track study hours so I can measure my progress*).
- Scope
    - In-Scope**:** Features and functionalities to be included
    - Out-of-Scope: Explicitly state what is excluded to avoid scope creep
- Detailed Requirements
    - Functional requirements (what the product should do)
    - Non-functional requirements (performance, security, accessibility, scalability, etc.)
    - UX/UI requirements (wireframes, mockups, user flow diagrams)
- Dependencies & Constraints
    - Technical dependencies (APIs, databases, third-party services)
    - Business/operational constraints (budget, deadlines, compliance)
- Timeline / Milestones
    - Key deadlines or deliverables (this is for your team’s own reference)
- Risks & Assumptions
    - Potential risks (e.g., limited engineering resources, uncertain user adoption)
    - Assumptions being made (e.g., users have stable internet access)

Product Requirements Document (PRD) + Prototype Progress

Overview

Each team will submit a **one-page PRD** and show **tangible prototype progress**. The goal for this checkpoint is to clearly define your product vision and demonstrate the first version of a core user interaction.

1. Product Requirements Document (PRD)

A **Product Requirements Document (PRD)** is a blueprint for building your product. It defines what needs to be built, why it needs to be built, and how it should function from the user’s perspective.

Your PRD should include the following sections (one paragraph or bullet list per section):

Overview / Purpose

- What the product or feature is
- Why it is being built (problem statement)
- How it connects to business goals, strategy, or user needs

Objectives & Success Metrics

- Define **1–2 measurable success outcomes** (KPIs or OKRs)
    - Example: *“Reduce onboarding drop-off rate from 30% → 15% within 3 months.”*
    - Example: *“Achieve 80% task completion accuracy in user testing.”*

Target Users & Use Cases

- Identify **1–2 primary user personas**
- Provide **2–3 key user stories** (e.g., *“As a student, I want to track study hours so I can measure my progress.”*)

Scope

- **In-Scope:** List **at least 3 key features** planned for the MVP
- **Out-of-Scope:** List **1–2 items** your MVP will *not* include

Detailed Requirements

- **Functional Requirements:** What the product should do (e.g., login, recommendation system, etc.)
- **Non-Functional:** Performance, security, accessibility
- **UX/UI Requirements:** Include **at least one wireframe, mockup, or user flow diagram** for a *core screen or function*

Dependencies & Constraints

- Note any technical or business dependencies (APIs, data sources, deadlines)

Risks & Assumptions

- List **1–2 key risks** and **assumptions** your team is making

2. Prototype Checkpoint

- Implement **at least one core screen or function** that demonstrates the main user flow or interaction.
    - Example: “Login and dashboard flow working with placeholder data.”
    - Example: “Input form connects to mock database.”

Week1_question

Brainstorming for Final Project

In one paragraph, answer the following questions:

- Name a real problem and the specific group of people who experience it most.
- Why is this problem important to solve (what's at stake if it's ignored)?
- How might generative AI help address the problem (supporting, simplifying, or enhancing, not just automating)?
- What's the smallest MVP you could imagine building in 2–3 weeks to test this idea?
- What outcome or success metric would you track (e.g., number of users trying it, time saved, improved satisfaction), and how would you measure it?

Week2_survey

Sangtae Ahn

A common challenge for early-stage entrepreneurs and students exploring startups is distinguishing between a “real” problem and one that only feels urgent to them. This uncertainty often leads to wasted time, money, and energy chasing ideas without true product-market fit, leaving founders discouraged and customers underserved. Generative AI could help by synthesizing user interviews, market data, and competitor insights into clear summaries that highlight whether a problem is widely felt or too niche. As a small MVP in 2–3 weeks, I could build a tool that lets founders input raw interview transcripts and receive an AI-generated report classifying the problem’s scope, urgency, and affected audience. I would track success by measuring the number of users testing it and whether they report improved clarity in identifying a “real” problem, using short surveys before and after their experience.

Andrew D Boessen

One problem that I want to work on solving is an enhanced PDF / document reader that can aid student and researchers by giving context, examples, or additional explanation to any concept selected. Generate AI could be used to parse the documents and do semantic and topic similarity to provide sources and references. It could also be used to summarize and add additional information to anything unclear in the doc. For example if a paper references the previous work of another paper, and the reader is unfamiliar with the original paper, the AI tool could be a brief recap and highlight relevant details to the current paper the user is reading. The interface would ideally would not have to make user's prompt the AI. The user could simply highlight and select the text they want more explanation on. A simple MVP would include a document reader that in connected with an LLM agent that can search document itself and the web and other resources for context. This could be a simple RAG system or more complicated. It would then summarize the sources in a short format. Some metrics to track include how often users search specific terms and the frequency that context search tool is used.

Sanjeet Brar

class lecture note taker. Listens to class lecture in real-time or can process a recorded lecture and generate notes. Students would be able to take advantage of this, giving them the freedom to focus in digesting class material as the lecture is given, instead of writing notes. Some students benefit from writing notes down, so this may not be for them, but for me this is something I would 100% use. I always found myself doing better when trying to fully understand the material instead of splitting attention between writing notes and listening. Also student athletes or people that are very busy can benefit from submitting recorded lectures and just having notes, as for some lighter classes it saves time to just have notes as opposed to watching a lecture at 2x speed. Smallest mvp, probably some type of site that takes an audio file and generates concise and well-organized notes. Success metric would probably be amount of users, user feedback on things like effectiveness, time saved, etc, and honestly my personal opinion on its performance. Since I am the target audience, I should be able to have some sense on how effective it is.

Mengyan Cao

Many professionals or simply busy people struggle not just with fitting tasks into their schedules, but with knowing the right order to approach them—especially when certain tasks can't start until others are completed. Without a system to organize these dependencies, people often lose track, work inefficiently, and feel overwhelmed. Generative AI could ease this by transforming a long, unordered task list into a clear sequence, automatically accounting for dependencies and suggesting when each step should be tackled. A minimal MVP over 2–3 weeks could be a simple web app where users enter multiple tasks at once, mark any that must come first, and receive back a streamlined, AI-generated queue. Success could be evaluated by tracking how many users complete all tasks in a suggested queue and by measuring perceived gains in clarity and focus through quick in-app check-ins.

Jasroop K Dhingra

The Problem: City residents are constantly blindsided by local developments—zoning hearings, street closures, block cleanups, transit changes—because notices are buried in PDFs on city websites or taped to lampposts. This disconnect leaves residents feeling disempowered and fuels mistrust in local government.

Why It Matters: Lack of awareness means fewer people show up to public meetings, fewer volunteers for cleanups, and more frustration. Cities waste time dealing with backlash instead of building engagement.

A generative AI agent could scrape city feeds, notices, and community calendars, then auto-summarize them into short, engaging, neighborhood-specific digests. Add visuals (maps, icons) and translations to boost accessibility.

A simple web newsletter or WhatsApp broadcast for one neighborhood. Backend scrapes local government calendars; AI summarizes in plain language; frontend delivers a weekly "neighborhood bulletin." Start small: one ZIP code, one newsletter, one pilot audience.

Success Metrics:

- Primary: Open/engagement rates (residents clicking for more info, RSVPing to events).
- Secondary: Meeting attendance, volunteer signups, or resident feedback.
- How to Measure: Track click-throughs and compare city meeting attendance before vs. after bulletin launch.

Kai Gowers

A real problem many college students and researchers face is struggling to understand dense or old historical texts, especially those written in archaic English with unusual grammar and vocabulary, such as Shakespeare, 18th-century legal texts, or early scientific works. This often leads to confusion, slower reading, and missing important context or arguments. If ignored, readers risk poor comprehension, wasted study time, and weaker academic or professional outcomes. A current Generative AI solution to this is to give an LLM the entire document and ask it to summarize it. However, due to the blackbox nature of LLMs, we can't be sure that the LLM accurately summarized the document without reading the whole

document ourselves. It is very possible that the LLM missed a very important part of the document, and this makes this current solution flawed. Generative AI could help by serving as a reading companion that summarizes each page in plain, modern language as the user progresses, ensuring nothing important is overlooked while also clarifying difficult phrasing. An MVP could be a lightweight browser or PDF extension that highlights each page and shows a short, accessible summary beside it. The main success metric would be improved comprehension and time saved: you could measure this by quick user surveys ("Was the summary accurate and easier to understand?"), comprehension quizzes comparing AI-assisted vs. non-assisted reading, and engagement metrics like pages summarized per session.

Timothy A Hays

College football coaches spend countless hours analyzing plays of upcoming opponents and then selecting the best plays to write down on cards for that week's practices. These cards are used by the scout teams to help the offense or defense get a realistic feel for the upcoming opponent. This work can not only be automated by AI to create cards for the coaches without requiring them to draw them up manually, but AI could also analyze which plays the team should be most concerned about based on the opponent's recent performance and their own team's historical tendencies.

I can envision creating a system that automatically compiles the best plays an offense has run in the past 2–3 weeks, supported by a limited AI analysis tool to highlight which plays need to be prioritized. Success could be measured in terms of time saved and by comparing the predicted plays against the ones the opponent actually runs in the upcoming game, essentially measuring how accurately the AI forecasted the game plan.

Julia L Hohenberg

Problem: Many elderly patients struggle to remember or correctly interpret the details their doctors share during appointments, often due to medical jargon, information overload, or memory difficulties. This confusion can result in missed medications, skipped follow-up appointments, or improper self-care, all of which increase health risks and reduce quality of life.

NOTE: this could also be expanded to other populations. To be honest, I feel like I could use this too.

AI Help: Generative AI can take detailed medical visit notes and translate them into clear, plain-language summaries that highlight only the most important instructions. Instead of long paragraphs filled with technical terms, the AI could output short, actionable reminders such as "Take this pill before breakfast" or "Schedule your next visit in two months." It could also format instructions in larger text, with bullet points or color-coded sections, making them easier for elderly patients to follow. It could also be a summary PDF if we don't have the UI for reminders set up yet.

MVP: A minimal viable product could be a simple tool where a caregiver, patient, or clinic staff member copies the doctor's notes into the system, and the AI instantly generates a one-page, easy-to-read summary. This summary could include key "to-dos," medication reminders, and next steps, designed to be printable for the patient to take home or stored digitally for family members to reference.

We could generate 'fake' doctor's notes and for our metric we could have participants try and understand the original doctor's note vs. the AI summary.

Metric: Success could be measured by tracking patient understanding and adherence. For example, doctors could check at follow-up visits whether patients remembered instructions correctly or completed tasks such as taking medication on time. Improved recall and higher adherence rates would indicate that the tool is making a measurable impact on patient health and independence.

In our specific context as a school project, we could have different test cases (i.e., my grandma lives in MA and is accessible). We could also test with non-elderly patients for demo purposes. In saying that, it might make sense to expand this to non-elderly populations so it could be better tested at BC for our project.

Alexander A Lehman

Climate change is a massive issue, and on top of that, the environment is in danger from unsustainable agricultural/hunting/deforestation/resource extraction practices, and lack of restorative initiative. Many people are impacted, such as small farmers, local communities, indigenous groups, and conservation groups. With this being ignored, climate change will undoubtedly worsen, water/soil quality could be at risk (and therefore food), as well as biodiversity, which will further spiral the environment downwards. Generative AI could help in multiple ways, including consolidating research into location-specific guides based on data-driven recommendations, creating simulations of scenarios where a specific action is taken / what could happen if a given practice continues or what a specific area should look like, or taking satellite imagery data and translating it into policy suggestions based on the status/health of different regions, or just simply translating such data into a clear-cut deliverable for policymakers and the public to comprehend easily. The smallest MVP could be an interface taking satellite data, region metrics, current practices, and current regulations and generate a report based on observed changes in the data and actionable recommendations, as well as what-if analysis on undesirable practices continuing. Some possible metrics could be: number of NGO/conservation groups/just general users who download a conservation plan/report, number of citations of the generated reports, user satisfaction/return rate , average time saved compared to doing your own research.

Tianyi Li

Self-directed learners—from career changers to students and hobbyists—all face the same core problem: finding a clear learning path through a chaotic landscape of online resources. This overwhelming process, which often ignores a person's existing transferable skills, can lead to wasted time, frustration, and a significant loss of personal potential when they ultimately give up. To solve this, a generative AI tool could act as a personalized "learning concierge," enhancing a person's journey by translating their current skills and interests into a structured learning roadmap. The smallest version of this tool could simply ask users for their goals and background, then instantly output a tailored plan that highlights transferable skills, identifies knowledge gaps, and provides a clear sequence for learning. The success of this MVP would be measured not by the number of users, but by a simple feedback mechanism: asking users to rate how much clearer their path feels after using the tool. This metric directly addresses the human-centric goal of the project, focusing on perceived value and clarity rather than raw usage numbers.

Yunhan Liu

A real problem faced by busy professionals and students is managing their schedules effectively—many struggle with remembering tasks, finding time slots for important activities, and avoiding overlaps. If this problem isn't solved, they risk missed deadlines, reduced productivity, and increased stress. Generative AI could help by intelligently suggesting time slots based on existing commitments, rephrasing vague tasks into actionable items, and sending timely reminders in natural, personalized language. A smallest MVP in 2–3 weeks could be a lightweight web or mobile app where users input tasks, and the AI suggests optimal time slots and sends push notifications for reminders. Success could be measured by the number of active users who schedule at least one task per day and the percentage of reminders that lead to completed tasks, tracked via simple in-app analytics.

Zhiyuan Li

At the New York coach camp, I met many people but had trouble remembering names and conversations, which made it hard to follow up. If this problem isn't solved, I could miss chances to build friendships, find mentors, or grow my career in basketball. Generative AI could help by keeping track of who I met, what we talked about, and suggesting quick follow-up messages. A small MVP could be an app where I scan or log a contact and get a short recap with a draft message. Success would be measured by how many contacts are saved and how many suggested follow-ups are actually sent.

He Pan

A real problem event attendees face is forgetting or losing track of the people they meet, which makes it hard to follow up and weakens the value of networking. If ignored, attendees miss potential collaborations, friendships, or career opportunities. Generative AI can help by enhancing networking recall—summarizing who you met, what was discussed, and suggesting tailored follow-up messages to keep conversations alive. A 2–3 week MVP for MissedShots could be a mobile/web prototype where users quickly log or scan a new connection, and the AI generates a short memory recap plus a suggested message. Success would be measured by number of connections logged and percentage of users who send AI-suggested follow-ups, tracked directly in the app.

Tianli Qu

One real problem is math lecture retention in college. Many students often feel lost in fast-paced, lecture heavy classes, especially if there are limited outside materials to review. This provides a significant barrier for students to truly understand advanced topics and leads to students simply "rolling" with the class or coping with rote memorization. Generative AI can help provide concise summaries for students to study as well as take organized notes as well as find appropriate external sources to reference when learning hard topics.The smallest MVP I can imagine is a voice to latex generator or OCR to latex generator, along with deepseek api calls to help explain topics. A success metric to use would be how accurate the information provided is and how well a student has comprehended a subject compared to a control sample student

Andrew L Ren

College students (particularly first-generation students) often struggle to translate their coursework and extracurriculars into strong resumes and cover letters. While they may have the skills, many lack the confidence or guidance to present them effectively. Networking can be helpful, but access is uneven, and interviews often feel performative rather than reflective of true competency (e.g., memorizing LeetCode problems or navigating subjective behavioral questions). These challenges are intensified by the rise of AI, which has made the job selection process more competitive and unforgiving. Without support, many capable students risk being overlooked for internships and jobs, limiting career opportunities and upward mobility simply because they don't know how to market themselves. This restricts the general talent pool and widens equity gaps in employment outcomes. Generative AI could serve as a supportive assistant, helping students reframe their experiences into professional, industry-relevant language. Rather than automating the process, it would guide students to learn how to present themselves more effectively. AI could also simulate mock interviews, offering common behavioral and technical questions with tailored feedback (and, with visual input, potentially even feedback on body language). An MVP might take the form of a web app where students paste bullet points or experiences and receive AI-suggested resume or cover letter phrasing, paired with a companion chatbot that provides 3–5 mock interview questions with instant feedback on clarity and content. Success metrics could include repeat student usage, self-rated improvement in confidence, and the number of job or internship offers secured, tracked through follow-up surveys.

Nathan M Shea

The problem I'm aiming to focus on is the abundance and overwhelmingness of information and media that we all have access to at all times, and the group I'm choosing to focus on is Gen-Z, as it is the group I am most familiar with and identify with. This problem is vital to solve because everyday people are age fall victim to the ultra-effective, ultra-addictive dominators of the attention economy. The tech giants that are the magnificent 7, and our dependence on technology create a day to day environment where young people are consuming immense amounts media selected algorithmically through the filter of prioritization of viewer retention and ad views. This filtration process takes advantage of our dopamine reward system and gives us the illusion of personalization. We're losing the ability to choose what we're seeing, consuming, and learning about. This is especially impactful to the way we hear about and understand current events. If this problem is ignored, our personalities and world view will continue to become less nuanced and more mundane, and we'll fall victim to billionaires who prioritize personal gain over societal advancement. AI can help address problem by intuitively filtering out junk content, and cleaning up what we see and consume, prioritizing a more academic lens of truth, and genuine societal and global relevance over brain rot and quick dopamine hits. What I envision for a small MVP is a daily news, AI-generated podcast in which this filtration process is also personalized to the user's interests, improving daily based on feed back, and aiming to broaden topics as the user expands their knowledge and explores new topics. To measure success, I'd track user retention, number of users, frequency of use, and user feedback used both to improve their personal podcast, and the overall product itself.

Boyu Shen

A real problem faced by biology and neuroscience researchers is the difficulty of producing accurate image annotations, since manual labeling is time-consuming and inconsistent while existing models often generate unreliable results. This matters because inaccurate annotations can stall research progress, reduce reproducibility, and waste expert labor. Generative AI could help not only by automating labeling but by supporting and enhancing the process: for example, suggesting plausible structures based on learned biological priors, simulating edge cases to strengthen training, or generating uncertainty maps to guide experts' attention to the hardest regions. A foundational verifier AI agent would build on this by verifying outputs against metrics and biological constraints, then deciding whether to accept, fine-tune, or swap models, creating a feedback loop that continually improves quality. The smallest MVP in 2–3 weeks could be a prototype that integrates one or two baseline models, applies Dice/IoU and uncertainty checks, and flags outputs for acceptance or retrial. The key success metric would be time saved and annotation quality improved, measured by reduction in manual correction time and higher agreement with ground truth across benchmark datasets.

Alexandra Sklyarova

I took a data visualization class over the summer, and one thing that we talked about was the limitations of charts/ graphs for visually impaired people (in particular the issues that can arise from screenreaders). This problem is important to solve because in a digital world, it is imperative that all people have equal access to information and technology, especially since many things in the world are driven by data. I would like to create a generative AI that is adaptive to individual peoples' needs when using a screenreader. So instead of replacing the reader, it would add an intelligence layer that learns from the patterns of usage and reads information in order of preference. Perhaps it can even have a chat bot/ agent feature that allows the user to interact with the information presented on the screen in a better way. The smallest MVP I could build would probably be a simple Streamlit app (or small browser extension) that takes in a chart and outputs an adaptive, screenreader-friendly description. To measure the success, we could put out an anonymous survey for blind/VI users or test participants several factual questions about a chart ("Which year had the highest revenue?") and measure % of correct answers using your adaptive descriptions. Then we could compare against baseline: raw chart with no description or just a standard screenreader. We could also measure how much they enjoy the feature against how much they like just a regular screenreader.

Theodore J Utomo

One big problem amongst pet owners is the inability to decipher pet body language. Being able to understand a pet's health through their body language can often help prevent health issues before anything becomes serious. If a kid has a fever, they can tell their parents and get medicine or visit the doctor, if a cat feels sick or has a stomach ache, its owner often won't know it's feeling ill until its too late. This is where GenAI can be applied. There has been thorough research on deep learning models built for analyzing pet body language, and if this technology can be integrated into existing home pet cameras (similar to ones like Furbo) while using GenAI to deliver user friendly analysis, it can deliver insights into the mental and physical health of pets that pet owners would often need to seek professional advice for. At

the very least, this can save hundreds of dollars in trips to the vet, but it can also save tens of thousands of dollars for pet owners by catching lapses in health that lead to chronic conditions if left untreated.

The smallest MVP I can imagine building in 2-3 weeks to test this idea is a mobile app that allows you take a picture/video of a pet, and the app will analyze it using deep learning models, and then use GenAI to deliver an analysis of the body language to the user.

A success metric I would track is number of users trying it out and see if the app's analysis aligns with real life professional opinion.

Michael A Widener

Diabetic foot ulcers (DFUs) pose a serious risk to patients with diabetes, preceding 80% of non-traumatic amputations, and clinicians currently struggle to prognosticate their dynamic progression from scalar data, leading to inconsistent care and delayed interventions. This problem primarily affects clinicians treating DFU patients, as well as the patients themselves who may receive late or suboptimal treatment. Generative AI can enhance care by producing personalized, visual forecasts of wound progression that integrate temporal imaging and patient data, offering interpretable spatiotemporal projections rather than static risk scores. The smallest viable MVP could be a system that ingests a short temporal sequence of wound images plus key patient metadata and generates a simple conditional image forecast for the next few days, allowing clinicians to visualize likely progression patterns. Success could be measured by clinician engagement with the visual forecasts, improvements in confidence or decision-making speed in simulated cases, or user studies comparing intervention choices with and without the AI-generated visualizations.

Peixin Yang

International students recruiting in the U.S. struggle with interviews because feedback is very limited. If we ignore this fact, it would hurt the confidence and income of these students. Generative AI can be a good support for simulating interviewers, simplify by forming STAR bullets with follow-ups from raw information. Also, Gen AI can provide drills to practice certain ways of speaking that might raise the probability of passing the interview. A 2-3 week MVP might be a browser mock-interview that takes a résumé and job post, runs a 10-question session, returns an revised sample answer, and provide a practice plan. The seccess metrics are: (1) ≥50% reduction in filler words per minute, measured from automatic speech transcripts with a fixed filler-word list; (2) median “time to first acceptable draft” under 5 minutes, based on event timestamps from session start to the first answer that clears a quality threshold ; (3) repeat practice, defined as the share of new users completing ≥2 sessions within 7 days; and (4) one-click CSAT ≥4.5/5, reported as a rolling average over the last 200 ratings.

Franco Zabaleta

A real problem that everyone in an academic setting has experienced is none other than having to use latex to type everything from math to data sets to even code in some cases formally, it takes so long to do even after you have succeeded in learning the language, it's important to solve because with all the technology in the world there should be a better way to turn in reports both in a school and work setting,

like why aren't people just able to write on a computer and the math shows up based on the symbols they used, if ignored this problem of having to use latex will waste many hours of an undergrads and beyonds life typing stuff which could just be automatically converted in due time itself. Gen Ai might help solve this problem, because there can just be an image classification algorithm to recognize all the different symbols one writes whether it be sigma and or dot product, or just simple variables and equations, as well as text(this problem has been solved before hence I will focus on latter problem I just mentioned). The smallest MVP I could have in 2-3 weeks is maybe just a program that uses Alex.net database of math symbol images and then generates a way to recognize the different symbols based on different edges and or organization of edges in due time itself. Basically, a program that can scan a simple square image by analyzing each separate square of pixels and then generating a computer version of said image, by either putting a contrast filled filter on it and or turning the pixels not the color scale of pencil marks to white to make it look more computer generated or something one can manually input into a computer. The outcome or success metric I could use is basically if a whole equation can be written and or more difficult if a whole simple intro to abstract mathematics number theory proof could be made from one a person writes, and or simply just track how many symbols are generated with the filter and or how many errors there are per each batch of writing in real life one sends to the computer, I am not sure if this is possible, but it is something that would be a cool project as I have never enjoyed writing stuff on latex, and would want to see how much time people save using my program vs just typing it on latex, obviously for shorter equations or proofs, difference in time isn't huge but as the proof or amount of math grows, as well as mathematical complexity of the image contents increase then so does the amount of time saved relative to taking hours to type every single symbol and text individually.

Tian-hao Zhang

A real problem is the burdensome process of machine learning process. This can be really time consuming and burdensome, as some steps like requesting data or verifying ground truth data are especially long. This problem is important to solve because it reduces progress and makes the process a lot more expensive and time consuming. Gen AI can solve this issue by introducing an agent equipped with tools that can automate these processes. Typical LLMs can only search the web or output text while agents have a lot more functionality. The smallest MVP I can imagine making in 2-3 weeks is a simple agent capable of proofreading, requesting, coding, and running programs. A success metric may be how many customers the product gains or how much time on average is saved.

Week3_topic

Team 1 — STEM Reading & Math Copilot

- **What it does:** Helps students read technical PDFs by breaking problems into steps, explaining where formulas come from, and converting equations to clean LaTeX.
    - **Example use case:** While reading a proof/derivation, they can highlight two lines and ask for the missing algebra/calculus step.
- **Who it's for:** Students in math/engineering/science courses.

Pairs to connect: Andrew D. Boessen, Kai Gower, Franco Zabaleta

Team 2 — Lecture & Accessibility Note-Taker

- **What it does:** Turns live lectures into clear bullet notes and adds accessible descriptions for charts/figures so screen-readers can read them.
    - **Example use case:** After a 50-minute lecture, a student uploads slides + audio and instantly gets clean bullet notes with alt-text for every chart so a screen reader can narrate the figures.
- **Who it's for:** Any student, especially those who benefit from accessible materials.

Pairs to connect: Sanjeet Brar, Tianli Qu, Alexandra Sklyarova

Team 3 — Meeting → Actions Productivity Copilot

- **What it does:** Converts meeting or event notes into specific action items with owners, due dates, and calendar reminders—so nothing gets lost.
    - **Example use case:** A club officer pastes Zoom notes and gets a checklist ("Email sponsors—Maya, Fri 5pm; Book room—Alex, Mon 10am") plus draft follow-up emails and calendar invites.
- **Who it's for:** Students and teams juggling classes, clubs, and projects.

Pairs to connect: Mengyan Cao, Yunhan Liu, Sangtae Ahn, Tianyi Li

Team 4 — Career / Interview / Networking Coach

- **What it does:** Preps students for interviews with practice prompts and rubric-based feedback, and organizes outreach/follow-ups to build relationships.
    - **Example use case:** Preparing for a data-analyst interview, a student runs a 10-minute mock session, receives rubric scores with targeted feedback, and generates a personalized thank-you email.
- **Who it's for:** First-gen/international students and anyone recruiting.

Pairs to connect: Andrew L. Ren, Peixin Yang, Zhiyuan Li, He Pan

Team 5 — Health & Care Insights

- **What it does:** Makes health info easier to understand with plain-language guidance and visual timelines (e.g., how a wound changes over time or when to seek care for a pet).
    - **Example use case:** A caregiver uploads weekly photos of a wound and receives a simple timeline (“healing slowing this week”), plain-language guidance, and “watch-for” alerts to discuss with a clinician.
- **Who it’s for:** Patients, caregivers, and pet owners who need clarity.

Pairs to connect: Julia L. Hohenberg, Theodore J. Utomo, Michael A. Widener

Team 6 — Sports Analytics + ML Ops

- **What it does:** Automatically scouts opponents’ plays and highlights tendencies, with built-in checks to verify the model’s output.
    - **Example use case:** Before game day, the team uploads two opponent games and gets a one-page scout report (most frequent formations/plays, down-and-distance tendencies) with confidence/verification flags.
- **Who it’s for:** Student teams/coaches needing fast, reliable scouting.

Pairs to connect: Timothy A. Hays, Boyu Shen, Tian-hao Zhang

Team 7 — Civic/Community Info

- **What it does:** Turns city agendas and notices into short, multilingual bulletins with clear “what/where/when” and any actions residents should take.
    - **Example use case:** The app pulls next week’s city agenda and outputs a 60-second brief in English/Spanish/Chinese—what’s changing, when/where to show up, and how to submit a comment.
- **Who it’s for:** Residents who want to stay informed without reading long documents.

Pairs to connect: Jasroop K. Dhingra, Alexander A. Lehman, Nathan M. Shea

Week9_pitch

Wed (10/22)
~~12:00-12:15 - Franco Z., Andrew~~
~~15:00-15:15 - Sangtae Ahn, Haozhen Gong , Andrew Ren~~
~~15:15-15:30 - Zhiyuan, Tim, Percy~~
15:30-15:45
15:45-16:00

Thu (10/23)
~~11:30-11:45 - Julie Hohenberg, Xander Widener, Alex Lehman~~
~~11:45-12:00 - Kai Gowers, Tianli Qu, Alexandra Skylarova~~
12:00-12:15
12:15-12:30
12:30-12:45
12:45-13:00
~~15:00-15:15 - Mengyan Cao, Yunhan Liu, Tianyi Li~~
~~15:15-15:30 - Theodore Utomo~~
15:30-15:45
15:45-16:00

Fri (10/24)
15:00-15:15 - Nate Shea
~~15:15-15:30 - Jasroop Dhingra~~
15:30-15:45 - seg.bio: Ethan Shen, Hanson Pan, Tian-hao Zheng
15:45-16:00 Sanjeet Brar

Plan

12-Week Venture Studio with Occasional Guest Speakers

Goal: Walk CS students through building an MVP and pitching a startup, enriched by external expert insights.

Weeks 1–2: Foundations & Team Formation

- **Week 1**: Introduce startup mindset, Lean Startup principles, and objectives.
    - **TODOs:** Team formation around problem areas.
    - Pair teams with Shea Center mentors (if applicable)
- **Week 2**: Deep dive into designing high-impact AI ventures.
    - **Guest Speaker:** Bring in an AI researcher or entrepreneur to discuss "identifying high-impact AI opportunities".
    - **Deliverable:** Problem statement & team charter.

Weeks 3–4: Customer Discovery & Validation

- **TODOs:** Conduct 5 customer interviews (depends on how many members are in each group); analyze pain points and market potential.
    - To understand whether their initial product ideas match the market needs
- **Deliverable:** Interview summary + opportunity hypothesis

Weeks 5–6: Ideation & MVP Scoping

- Brainstorm, prioritize solution ideas, and define MVP scope.
- **Deliverable:** One-page simple Product Requirements Document (PRD). (provide some templates for students to start with)
- **Guest Speaker (optional):** Bring in a founder or technologist to share how they scoped MVPs effectively.

Weeks 7–8: Prototyping & Tech Build

- Build a lightweight prototype (wireframe, mockup, or coded version) and run 3–5 user tests.
- **Deliverable:** Prototype demo.
- **Guest:** Invite Shea Center mentors to provide feedback to prototypes through a start-up lens.

Weeks 9–10: Iteration & Business Basics

- Refine prototype based on feedback; introduce business modeling (e.g. business model canvas).

- **Deliverable:** Updated prototype + draft business model canvas.

Weeks 11–12: Pitching & Showcase (Demo Day)

- Run workshops on storytelling, designing pitch decks, and handling Q&A.
- **Deliverable:** Final MVP demo and 5–7 minute investor-style pitch.
- **Guest Finale:** Organize a Demo Day-style event with external reviewers, such as entrepreneurs, investors, or Shea Center partners.

Resource

```
from langchain_ollama import ChatOllama

llm = ChatOllama(
    model="mistral:latest",
    base_url="http://cscigpu08.bc.edu:11434",
    temperature=0
)

messages = [
    ("system", "You are a helpful assistant."),
    ("human", "Explain machine learning in simple terms.")
]

response = llm.invoke(messages)
print(response.content)
```

Here is a list of available models for the model parameters: