1. Welcome to CSC 4351

Syllabus and Tools, Content Overview

Agenda 📅

1. Intro class/self
2. Meet Some Peers!
3. Course Design Discussion
4. Syllabus/Policy Review
5. How do you build an app?

Why do we care? 🤔

Every single lecture we have in this class will have a slide called "Why do we care?" that explains why today's topic is important to either:

- Success in this class
- Success as an engineer in industry

So… Why do we care in general? 🤔

- What are we doing here?
  - Learning tools for building stuff
  - …with other people in the mix
- How will that help you?
  - Working through this class will give you problem-solving skills and exposure to concepts that will help you hit the ground running at a technical job!

So… Why do we care today? 🤔

- Why can't we just start writing code right now?

So… Why do we care today? 🤔

- The syllabus is basically a step-by-step guide of how to succeed in this class!
- It’s important to get the tools set up now so you won’t be slowed down getting them set up during class later.
- Getting the high-level picture of where we’re going will make the individual parts make more sense!

Structure of the Class

- Phase 1: Basic Tools (Git, Python, Flask)
    - Most of your work will be on **individual assignments**
    - But we'll also choose groups in these first couple weeks
- Phase 2: Add-ons (Databases, User Input)
    - Assignments will build on each other into a **project**
- Phase 3: **Group Project**
    - Laying the groundwork for second semester work
    - This is really the focus of the **overall** class.
- Misc: Career Days
    - Every month, we'll devote one lecture to a career-oriented topic like resume-writing.

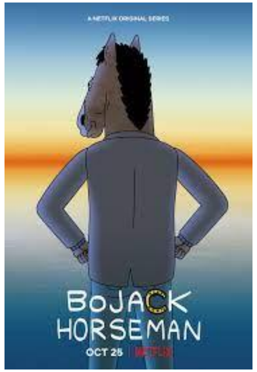
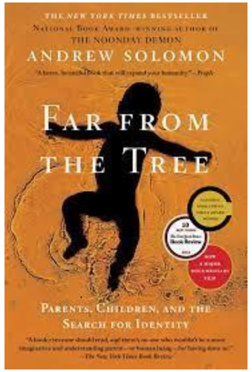
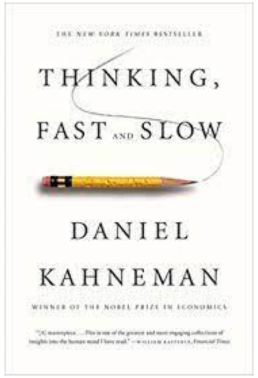
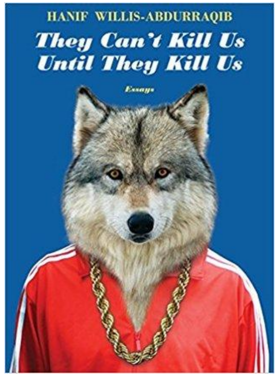
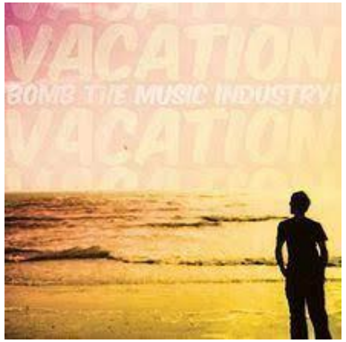
About Me

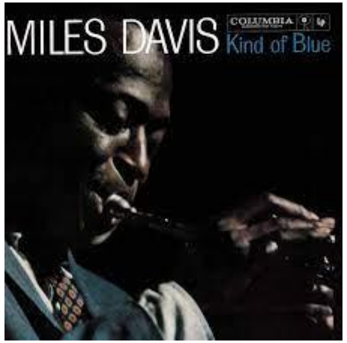
facebook connectivity

THUUZ sports

Georgia State University

About Me


A NETFLIX ORIGINAL SERIES
BOJACK HORSEMAN
OCT 25

ANDREW SOLOMON
FAR FROM THE TREE
PARENTS, CHILDREN, AND THE SEARCH FOR IDENTITY

THINKING, FAST AND SLOW
DANIEL KAHNEMAN
WINNER OF THE NOBEL PRIZE IN ECONOMICS

HANIF WILLIS-ABDURRAQIB
They Can't Kill Us Until They Kill Us
Essays

VACATION
BOMB THE MUSIC INDUSTRY!

MY CHEMICAL ROMANCE
THE BLACK PARADE
PARENTAL ADVISORY EXPLICIT CONTENT

MILES DAVIS
COLUMBIA
Kind of Blue

1000 gecs

EIR’s Mission

- Build applicable, real-life knowledge of software engineering and CS fundamentals
- Build communities of challenge and support
- Engage faculty to scale innovations to CS curricula

ENGINEER

IN RESIDENCE

Let’s socialize! 👯

Big Ideas of this Class

Big Idea #1 - Personalization

Big Idea #2 - Make Mistakes!

Big Idea #3 - Feedback

- One of the most valuable, and hardest to practice, skills in software (or any career) is coming up with ways to make your work environment better.
- In this class, that means thoughtfully considering how you could be learning more, or better.
- If something isn't working for you, or if you see an opportunity for some part of this class to be better, **please tell me!** There is zero chance you will offend me.

TED, I WANT YOUR HONEST FEEDBACK ON HOW I'M DOING AS YOUR BOSS.
DILBERT.COM @SCOTTADAMSSAYS
THIS FEELS LIKE A TRAP.
NOT AT ALL! I *LOVE* CON–STRUCTIVE CRITICISM!
2-27-21 2021 Scott Adams, Inc./Dist. by Andrews McMeel
I THINK YOU'RE ALMOST PERFECT.
ALMOST?

Independent Work Time: Syllabus (15 min)

1. In iCollege, use the link to open the syllabus.
2. Carefully read the full syllabus.
3. Comment on the Google Doc (at least) one thing you either like, dislike, or are confused about in the syllabus. Make sure I can somehow see your first name and last name.
4. If you have extra time, feel free to start in on HW1 (other components: joining the Discord, filling out a survey)

Administrivia

- Not going to go over the whole syllabus right here
- Just want to touch on some things for emphasis

Class 1 of 2

CSC 4351 and 4352 are a *sequence.*

- If you fail CSC 4351, you can’t enroll in 4352. You’ll have to take 4350 in the spring to graduate on time.
- I've designed this class to be easy to pass with consistent effort, so I wouldn’t stress too much about it.

Class Tools

- Coding: VSCode (strongly encouraged)
  - Option to provision an Azure VM (Virtual Machine) if you don’t want to develop locally. Talk to me if you want to take that option.
- **Communication: Discord**
- Assignments: iCollege, Gradescope
- **I’ll send instructions for setting up all of these as part of the first couple homework assignments.**

Grading Breakdown

- Assignments: 50%
- Group Project: 30%
- Attendance/Participation: 10%
- **Career Prep: 10%**

Late Policy

Trying something new this year!

- Everyone gets **10 late days** for the semester, no questions asked.
- Groups get **5 late days** on group assignments.
- If you turn in HW1 three days late, there's no penalty, but you'll only have 7 more late days remaining.
- Once your late days run out, it's 10% off per day on each assignment.
- *Recommendation: save your late days for the harder programming assignments!*
- Extenuating circumstances exist! Let me know if something truly out of your control is preventing you from doing your work. Your health comes first.

Academic Dishonesty

- Most software engineering relies on external resources. It is not always easy to tell what constitutes dishonesty. My bar is this: **you need to be able to explain and reproduce any code you write in this class**.
- Questions about what constitutes dishonesty? **Ask me.**
- Know the policy and tempted to disregard it? **Reach out to me - there are always options.**
- My grades tend to be pretty inflated. You will be fine in this class!

Discussion

This is the first time this course has ever been taught. How can we make it as useful for you and future generations as possible? Some big questions on my mind:

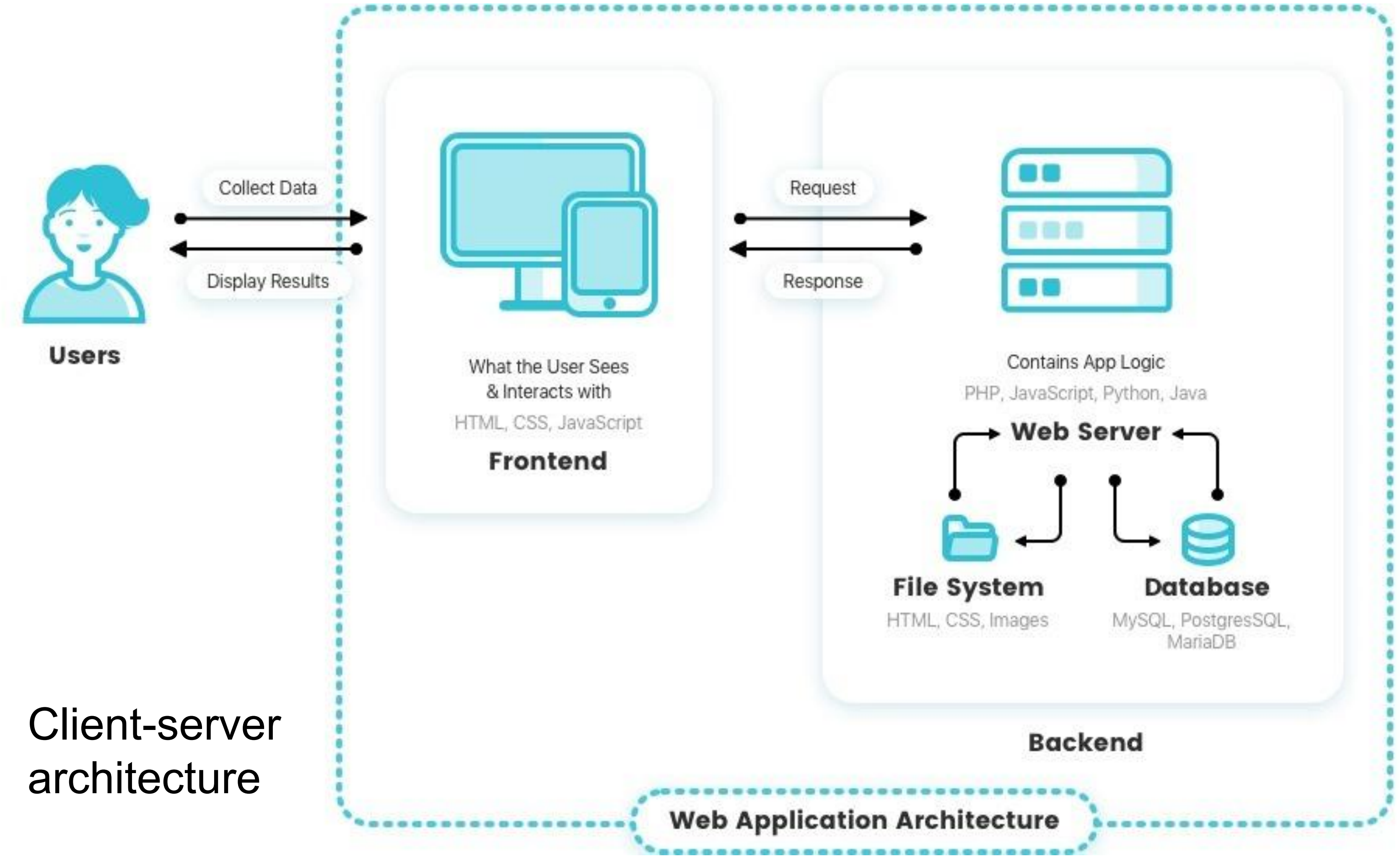
- Introducing a tech stack vs. independent discovery
- Pulse check on career stuff
- Group selection
- How much non-project work should there be?
- Anything else?

Break

Let’s plan an app...

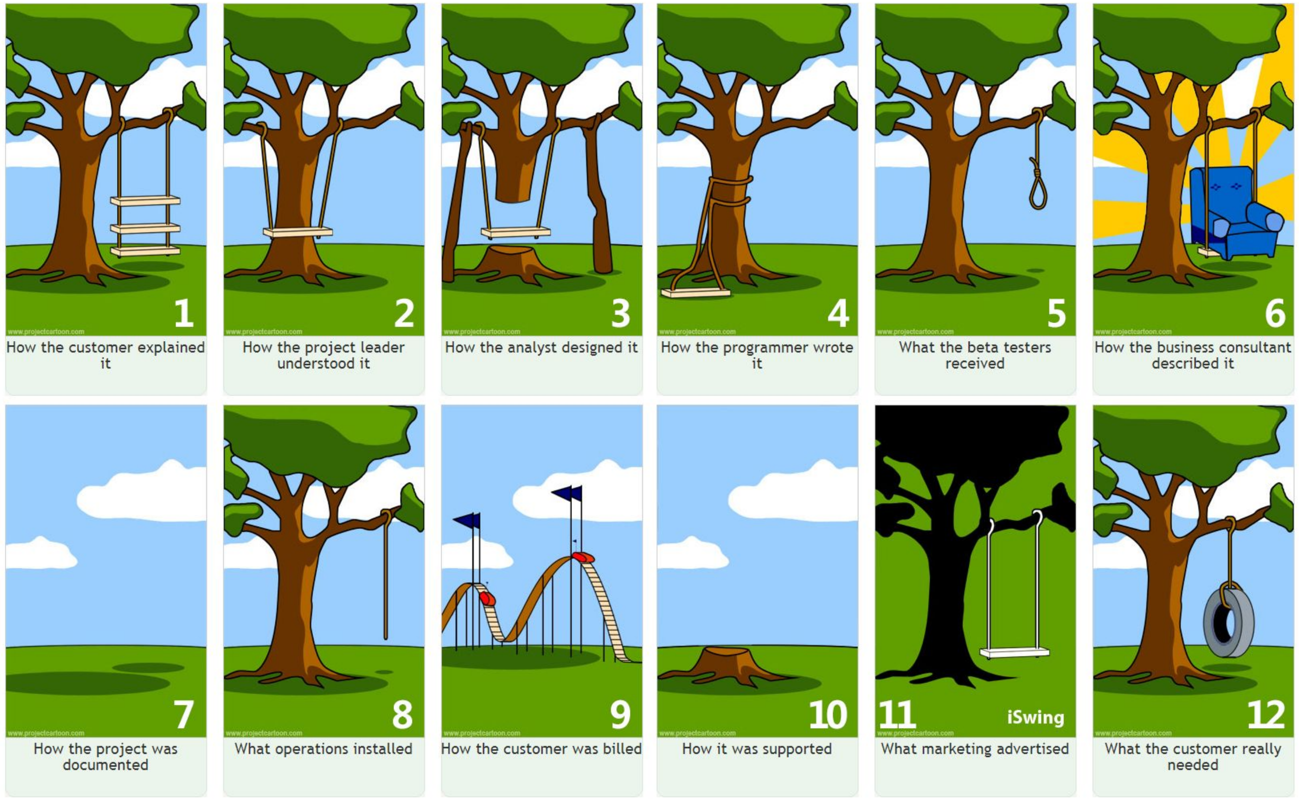
- Let’s see where all the topics in the syllabus fit into an app’s architecture!
- We'll try to cover the following components of an app's design:
  - Front-end engineering
  - Back-end engineering
  - API design
  - Database design
  - How it all fits together in full-stack engineering
- Don't worry if you get lost during this bit! It's not supposed to make 100% sense yet, but it's important to know high-level what we're headed towards.

Whiteboarding ✍️



Client-server architecture

Lesson 1: Crisp communication is needed


1
How the customer explained it
2
How the project leader understood it
3
How the analyst designed it
4
How the programmer wrote it
5
What the beta testers received
6
How the business consultant described it
7
How the project was documented
8
What operations installed
9
How the customer was billed
10
How it was supported
11
iSwing
What marketing advertised
12
What the customer really needed

Lesson 1+: Engineers are only part of the equation!

We know software engineers write, test, and architect code, but who else is involved?

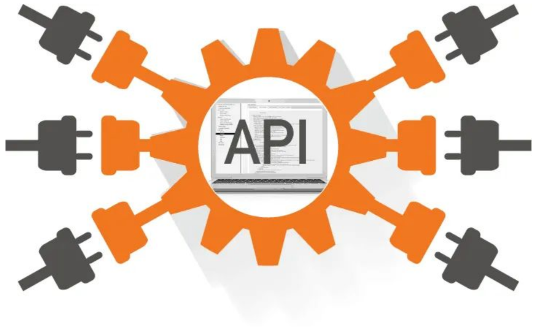
- Software Engineers
- Product Managers
- Product Designers
- Content Strategists
- Data Scientists
- Data Engineers
- Technical Program Managers
- UX Researchers
- The list goes on…

Lesson 2: Iterative planning + development are the norm

- Facebook, Instagram, Snapchat, Twitter, Google, etc. didn’t spring into existence out of nowhere
- These apps all incrementally built upon, planned in advance, and added multiple complex systems over time
- Most of these started out as a simple web app
- Your projects, and this course, will be the same
- As you get more senior, you will be more and more responsible for the planning (technical and non-technical) and not just the building!

Lesson 3: Use previous work

- Successful apps usually don't reinvent the wheel - they use third-party APIs, existing frameworks, and borrow graciously from other sources to build their product as fast as possible!
- When designing our apps, we won’t build out all minute features (i.e. a Maps product or a Login product) - we’ll use third-party tools for that!



Until next time... ✌️

- HW0 is to **meet with me** before September 9th! Pick a 10 minute slot on Calendly (link on iCollege)
- Follow instructions on iCollege for HW1 (due before next class).

Coming soon...

- HW2 (Development Environment Setup) will be posted next week.
- Office hours coming soon! In the meantime, just email me about questions or setting up a meeting.
- Next class, we'll talk about Git or resume-writing – I'll hold a poll.