

Introduction

INFO 153B/253B: Backend Web Architecture

Kay Ashaolu - Instructor

Aishwarya Sriram - TA

Introductions

Course Goals

- Demystify how systems are built
- Learn what is powering the applications you use every day
- Be able to have intelligent conversations about the services you use
- Know the challenges of building "magic"
- Understand the trade-offs made when building "magic"
- Learn the basics of backend web architecture

Course Layout

Fridays: Lecture 9 am - 11 am (in person)

Mondays: Lab 9 am - 10 am (in person)

- INFO 153B - Undergraduates
- INFO 253B - Graduates
- Both courses have the same lecture and lab time as well as learn the same material
- Course work differs for each class

Course Materials

- Course Website: Main Information Hub
- Bcourses - Work submission, Grade notifications
- Github - Where your work will be stored

Course Schedule

- **Part I of class (~8 weeks)**
 - Labs and Assignments
 - Learn how to build backend systems
- **Part II of class (~7 weeks)**
 - 153B: Labs and Take Home Project
 - 253B: Group Project
 - Learn the theory behind backend systems

Course Work

- Short multiple choice quiz during lecture
- Two major homework assignments building a backend web application
- 4 Labs (253B), 8 Labs (153B)
- Group project (253B), Take home exam (153B)

Grade Breakdown

INFO 253A	INFO 153A
Quizzes: 10%	Quizzes: 10%
Labs: 20%	Labs: 40%
Assignment: 30%	Assignment: 30%
Group Project 40%	Take Home Exam: 20%

Readings?

- After some thought and research, we are moving away from textbook readings in the class
- In this day, it's a bit rare for software engineers to learn new technologies by reading a book from beginning to end
- We will use a couple of courses in this class - links to the lessons will be posted on the website (**and made available to enrolled students free of charge**)
 - Fuentes, Jose Salvatierra. REST APIs with Flask and Python in 2024
 - Ashaolu, Kay. System Design Course for Junior Engineers

More on Quizzes

- There will be a short three question multiple choice quiz at every lecture
- The quiz is designed to test basic knowledge of the reading
- This will be conducted in person via BCourses
- Note: attendance will be taken via a separate mechanism - quiz submissions from students that are absent that week will get a zero for the quiz
 - If you need to be excused, please reach out via email

More on Labs

- All labs will be assigned on Monday and due the following Monday before lab (9 am PST)
- The lab section will be dedicated in introducing these labs and giving you the chance to work on them during class and have support
- Labs will be led by the TA

More on Labs

- Labs provide opportunity to try out what you learned
- Labs will be graded on a simplified basis out of 10:
 - Correct: 10/10
 - Not correct, but shows significant effort: 8.5/10
 - Not correct, low to no effort: 7/10
 - No submission: 0/10
- You will be given a github classroom link to generate your own git repository where you will save your work
- You will submit your work submitting the link to your repo in BCourses

More on Assignments

- Assignments must be done individually, but you can ask others for help
- Dates when assignments are assigned and due are posted on the course website
- Lab sections nearing when assignments are due will be dedicated to assignment help
- Every day it is late after that we will deduct 2 points from the final score
- You will be given a github classroom link to generate your own git repository where you will save your work
- You will submit your work submitting the link to your repo in BCourses
- **Please contact us if you run into any issues**

More on the Group Project

- **INFO 253B only**
- The group project is an opportunity to build a project you come with other students
- Must be in groups of 4-5. There are **no exceptions** to this rule
- We will form teams and ask you to draft up a proposal of what you would like to build, and we will give you feedback. Then you will build your proposal
- There will be presentations on the last week of class before RRR week

More on the Take Home Exam

- **INFO 153B only**
- Take home will be assigned the Friday of Group Presentations and be due the next Friday of RRR week
- Will have seven days to complete, but is designed to be completed within 2 hours
- Will be similar to an assignment but not as extensive

Technologies you will use

Languages / Frameworks

- Python
- Flask
- SQL/SQLAlchemy

Tools

- Git and GitHub
- Terminal
- Docker
- Visual Studio Code (IDE)

But Don't Worry

- This is not primarily a programming class
- Grades are based on the demonstration of understanding the base concepts
- Grades are not based on extra bells and whistles, but are reserved for extra credit
- We are here to help

This is still a Graduate Class

- Perform well without supervision
- Understand classroom prep materials from a variety of different areas on the web
- Ask for help proactively
- Get frameworks working on your own

Responsibilities

Me

- Explain core concepts
- Understand design trade-offs
- Introduce new technologies
- Tie everything together

You

- Understand lectures and do class prep if assigned
- Explore new technologies
- Work independently and in groups

Office Hours and Wait List

- Kay's Office Hours: Monday's 4 - 5 pm: please see the course website to book directly
- Aishwarya's Office Hours: TBD
- Wait list will start being resolved next week
- 153B and 253B share the same 49 spots in the classroom, so even if you see that there is room in CalCentral, there may not be room
- Depending on 253B enrollment, we could expand 153B
- Concurrent Students if I can ask to hold until end of next week to determine space in the class

Wait List - Auditing

- Apologies, but there will be no auditing this class this year
- Course materials will be publicly available on the course website. Just note that some content may be behind a paywall that would be provided to enrolled students

Questions?

**Before we move on, I want
to address the elephant in
the room...**

**AI IS GOING TO REPLACE
MID-LEVEL ENGINEERS!**

**MARK ZUCKERBERG SAID
SO!**

**WHY LEARN ANYTHING IN
THIS CLASS?**

Let's listen to the actual clip:

https://www.youtube.com/embed/uDL_6A6zB0w?enablejsapi=1

So here's my take on this

1. It is a mistake to jump on the AI bandwagon and think that software engineers will not be necessary and AI will just build our applications perfectly
2. It is also a mistake to think that software development, as we know it right now, is not about to change significantly in the next few years (think about the creativity that Mark mentioned in this clip)

The truth is...

We just don't really know.

We didn't know how the internet was going to take off in the 90s

We didn't know how web 2.0 was going to completely shift how we interacted with computers in the 2000s

We're at another inflection point now

So what do you do?

What doesn't change too often is...

The fundamentals. The core ways that systems work. Ideas and thoughts that have persisted and stayed the test of time.

Examples? Guess what hasn't changed...

The data structures and algorithms we use

Fundamentals

We can be pretty sure that no matter what happens:

- We'll be using hash tables/dictionaries
- We'll be using arrays
- We'll be using objects and functions
- We'll need to sort and order things efficiently
- We'll need to use some advanced math to achieve amazing things (like how the weights in flat file LLMs are calculated)

and so on

So my ultimate take is

If you learn the true fundamentals of how to actually build things, **AI becomes a tool for that purpose**, and not a mystical force that build things for you

The question though is, are there really fundamentals for back end system design?

I will argue in this class that they are, it's just the information is either in people's heads, or scattered in a bunch of books

Part I: Learning by building

This class will be split into two parts to facilitate learning the fundamentals:

Part I: Learning by building - everyone is going to learn how to build backend systems using Python/Docker/SQLAlchemy etc...

There's power in rolling up your sleeves and you yourself building a simple backend system that you could probably ask ChatGPT to build for you now

Part I: Learning by building

When you build something on your own, you gain understanding that you can't get from having something build it for you.








You gain that "intuition" that comes only from doing the work yourself.

Part I: Learning through theory

Part II: Learning through theory - As our 253B students work on their projects, and 153B students do a series of labs, the entire class is going to learn about what I call **System Design Theory**

This concept is an attempt to introduce a way of approaching back end systems **in a technology agnostic way** that utilizes seven fundamental building blocks that are used to build a variety of applications

The seven building blocks

Type	Building Block	Icon
Task	Worker	
Task	Service	
Storage	Key-Value Store	
Storage	File Store	
Storage	Queue	
Storage	Relational Database	
Storage	Vector Database	

Why Part II?

Part II purpose is to show you how you've used at least five-six of those seven building blocks in what you have built yourself

And show you how those same building blocks you used can be used to build so many other applications

We will go through 15 case studies explain how these seven building blocks can be used to build these systems

The caveat

This is not to say that this "System Design Theory" is made to build the "most efficient" system design ever

These system designs are not the most efficient.

The goal is to inspire your intuition for systems.

If you build your intuition, your mind will be able to make connections and utilize new tools (like AI) better for your purposes.

Isn't this how LLMs work in the first place?

Building intuition future proofs yourself

I truly believe that growing intuition is the key to not just staying up to date with the latest, but being able to utilize the latest to your advantage

Having an open mind helps as well

This class aims to help you do both

Questions?