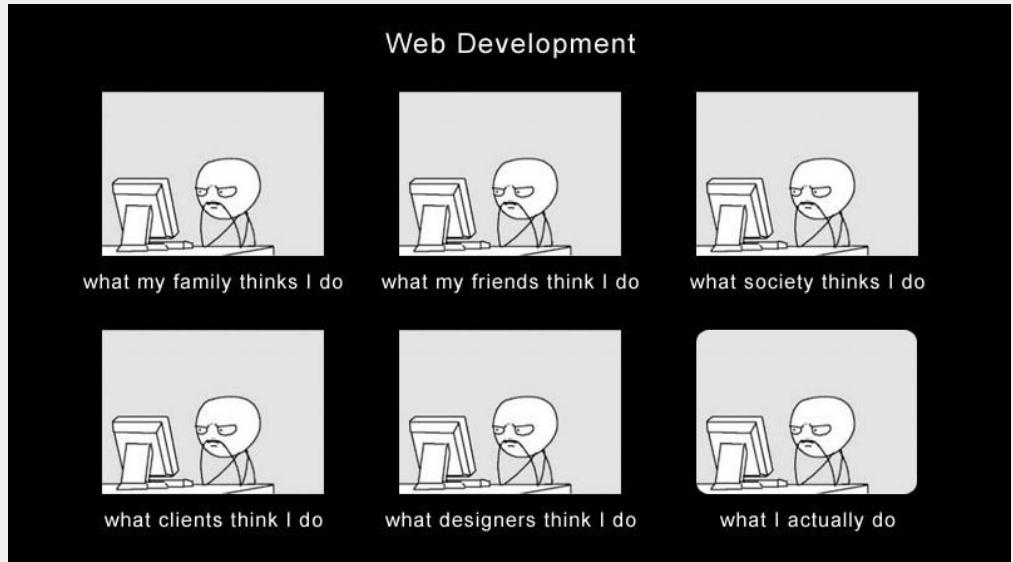


# **LAB #4**

**Adv. CiC  
Spring 2026**

# Agenda

1. Housekeeping
2. Internet Architecture
3. Web Development
4. APIs
5. Lab 4 Exercise



# Housekeeping

## Due this past week

- Reading 4
- Lab 3

*Let me know if any  
questions about  
proposal feedback.*

## Upcoming

1. Project pt. 2 - **2/19 @ 3pm**
2. Reading 5 - **2/19 @ 3pm**
3. Lab 4 - **2/20 @ 1pm**

## Additional Resources

- Online tools
  - Google, ChatGPT, etc.
  - *Make sure to review AI policy!*
- ED!!!
- Office hours
  - 1 hr/week set time
  - Appts by email

rogrammers looking  
: programming memes



Kyle 🌿 @KylePlantEmoji · 23 godz.

Me: I'm so sorry, my dog ate my  
homework

Comp Sci Professor: your dog ate yo  
coding assignment?

Me:

Prof:

Me: it took him a couple bytes

255

18,7K

146K

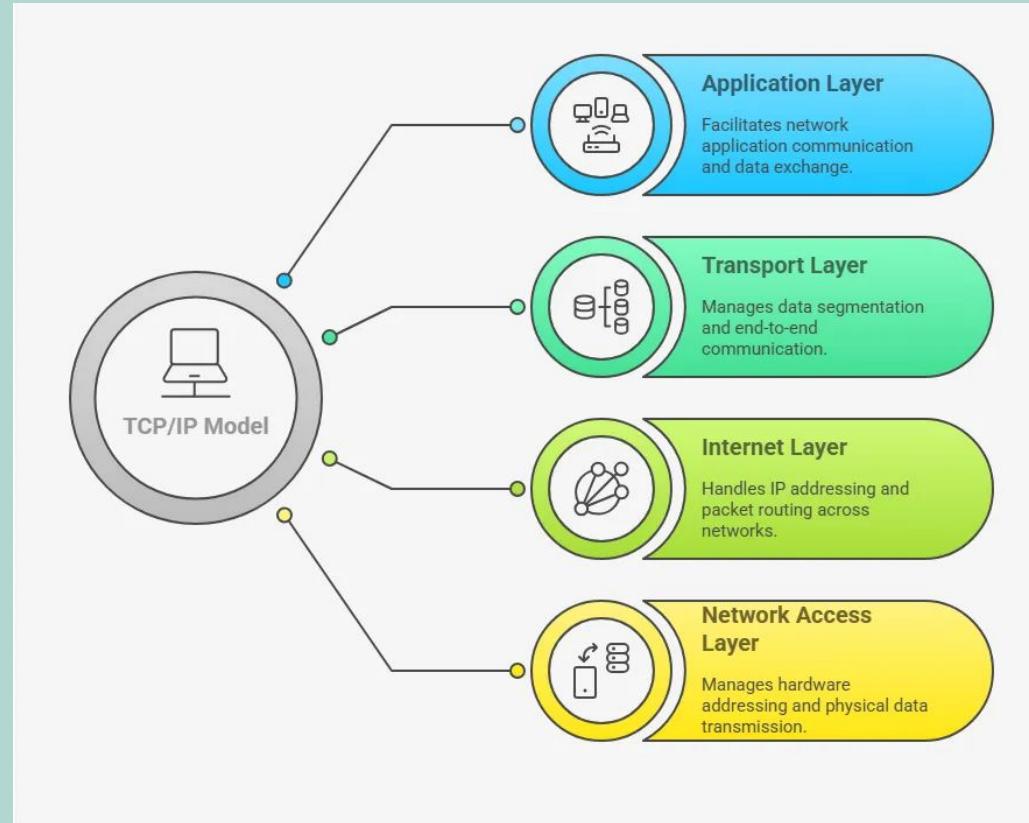


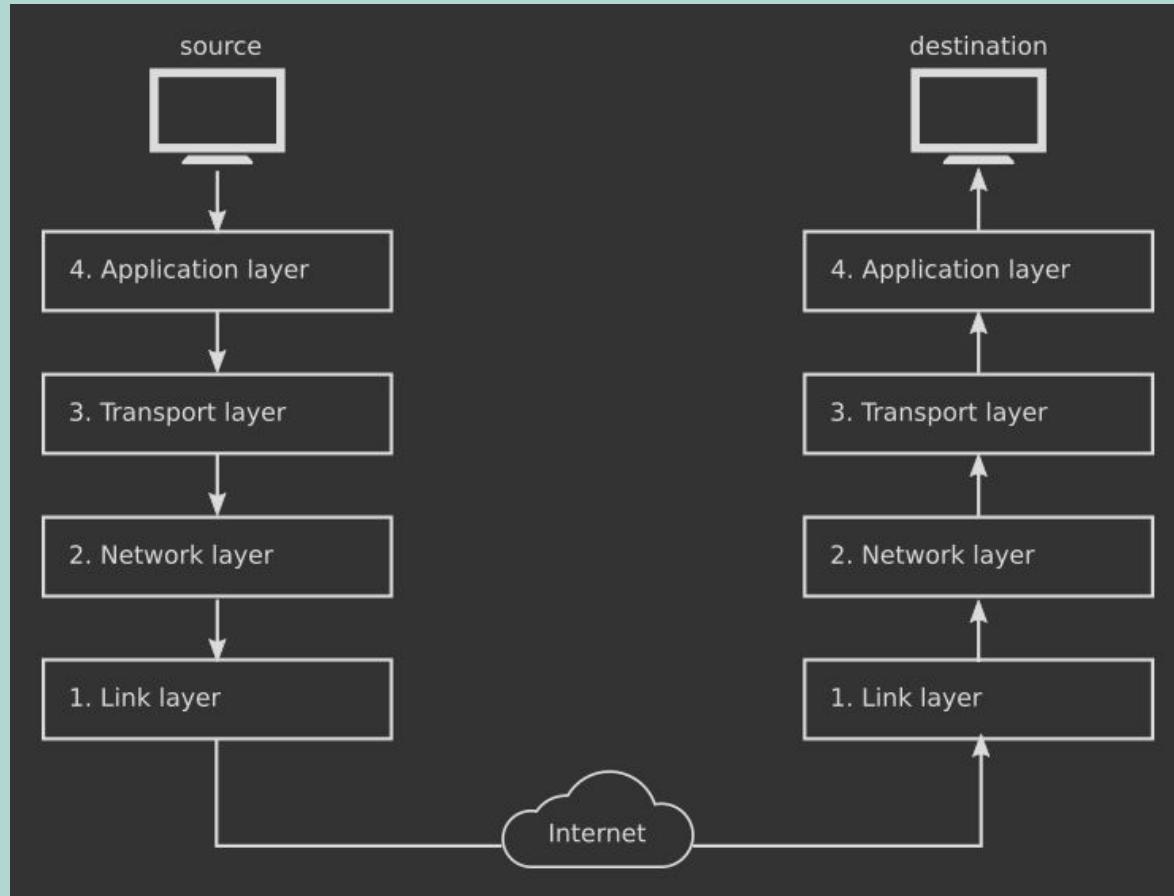
Questions  
(on Lectures, Labs, or Readings)?

# **Internet Architecture**

# What is the Internet?

system of interconnected computers and networks that communicate using shared protocols to exchange information around the world





# How TCP/IP Works

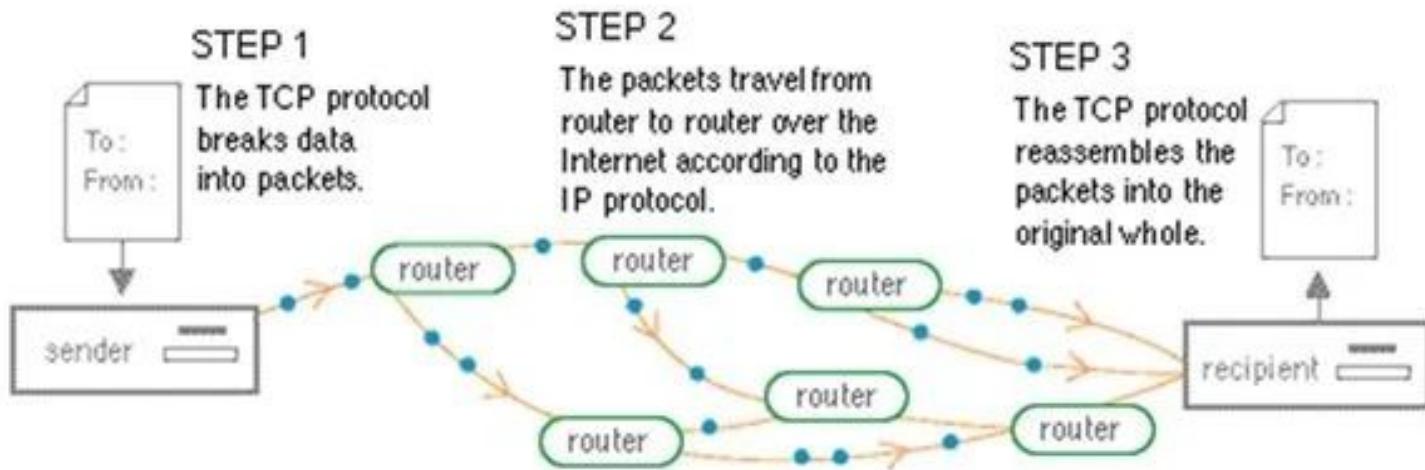
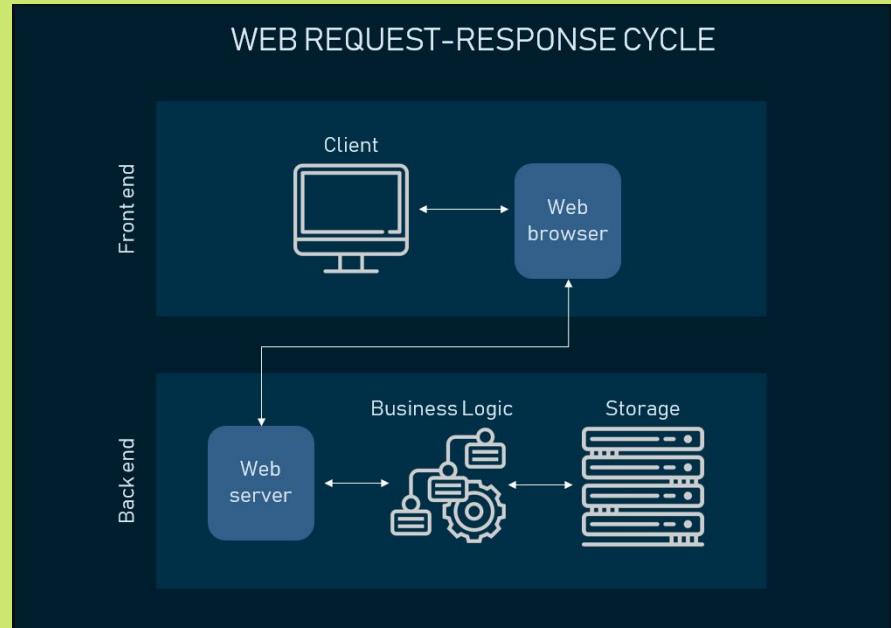


Figure 2. How data travels over the Net.

# Web Development

# Basics of Web Development

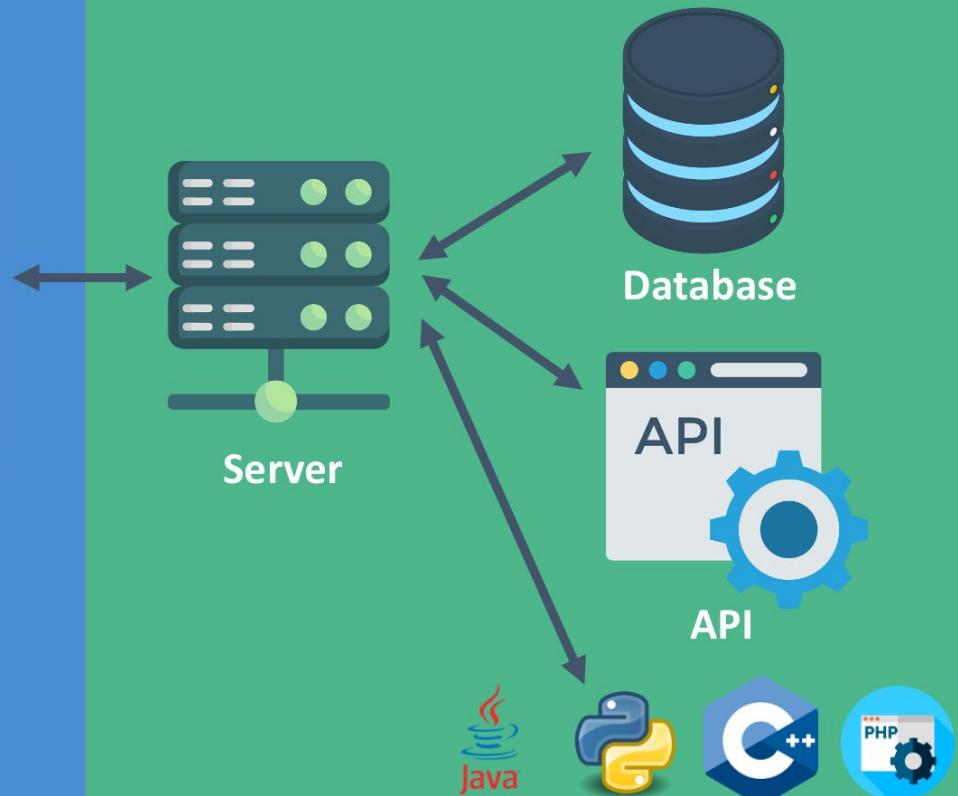
- **How the web works:** At its core, the web is just computers sharing data - your browser downloads code and content from a server, then runs and displays it on your device.
- **Frontend development:** The frontend is everything users see and interact with in their browser (layout, design, buttons, animations)
- **Backend development:** The backend runs on servers and handles data, logic, and storage: processing requests, managing databases, and sending information back to the frontend.



# FRONT-END



# BACK-END



# HTML VS CSS

**HTML**



Creates the content & structure of web page

**Complete Web Page**

HyperText Markup Language



**CSS**



Controls the presentation & styling of web page

Cascading Style Sheets

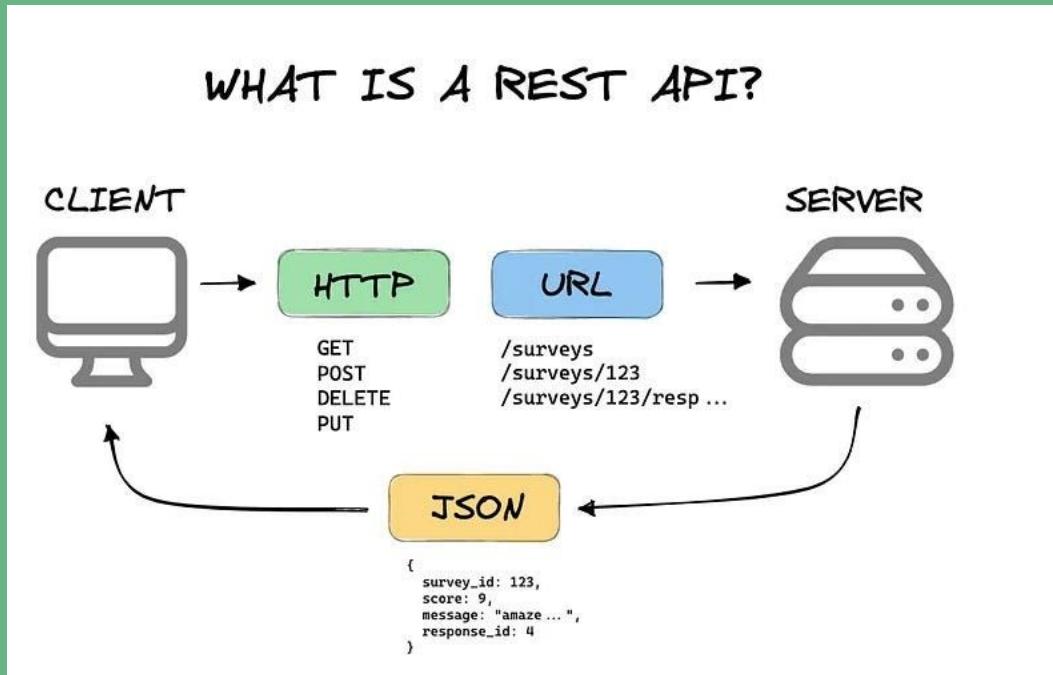
How are the two different?



# APIs

# APIs

- An API is a **way for different software systems to talk to each other** and exchange data in a structured, predictable way
- **Why they matter:** APIs let apps reuse existing services (like maps, payments, or weather data) instead of building everything from scratch
- **Big picture:** They make modern software modular - different teams and systems can work independently but still connect smoothly



Web app  
in  
Browser



Request



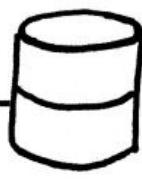
Response



Web  
Server

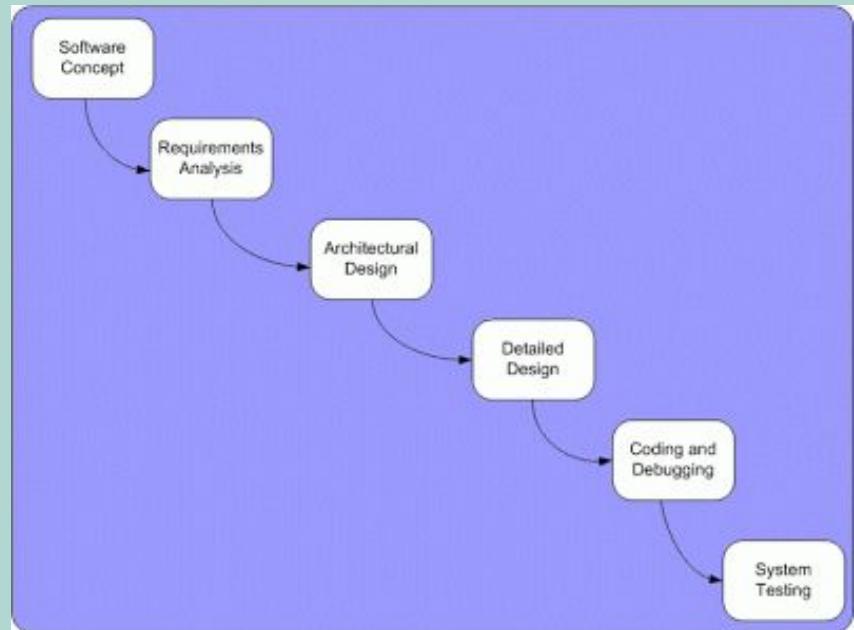


Database



# Document-Driven Development

- **Clarity:** Writing docs first forces clarity - you define how something should work before getting lost in implementation details
- **Collaboration:** everyone builds against the same shared expectations, reducing confusion and rework



Q: why does this look a little Waterfall-y?

Today 6:14 AM

dude your job is done for 💀 I just  
made an entire website with  
chatGPT

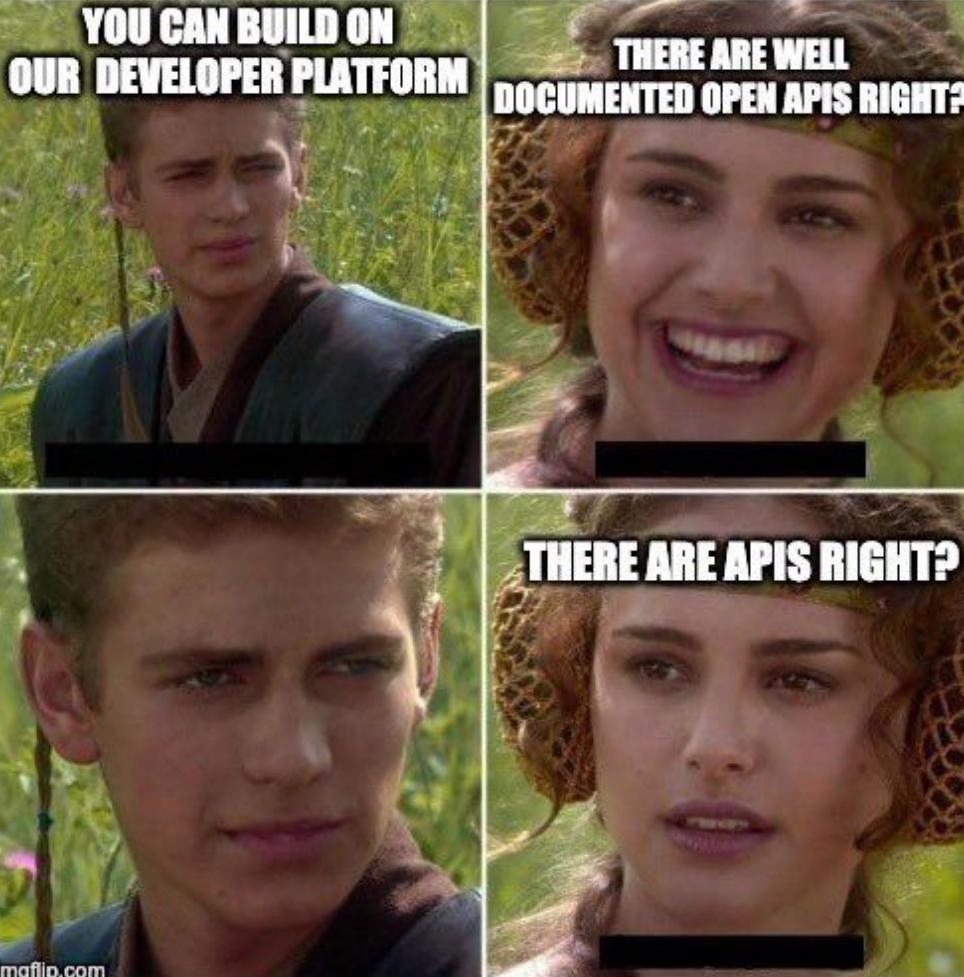
wanna see it?

sure

Read 6:16 AM

C:\Users\ben\Downloads\index.html





**YOU CAN BUILD ON  
OUR DEVELOPER PLATFORM**

**THERE ARE WELL  
DOCUMENTED OPEN APIs RIGHT?**

**THERE ARE APIs RIGHT?**



**EXERCIS**

**E**

**Lfg**