Getting Started with Web Development

Pratik Luitel

About me

About old me

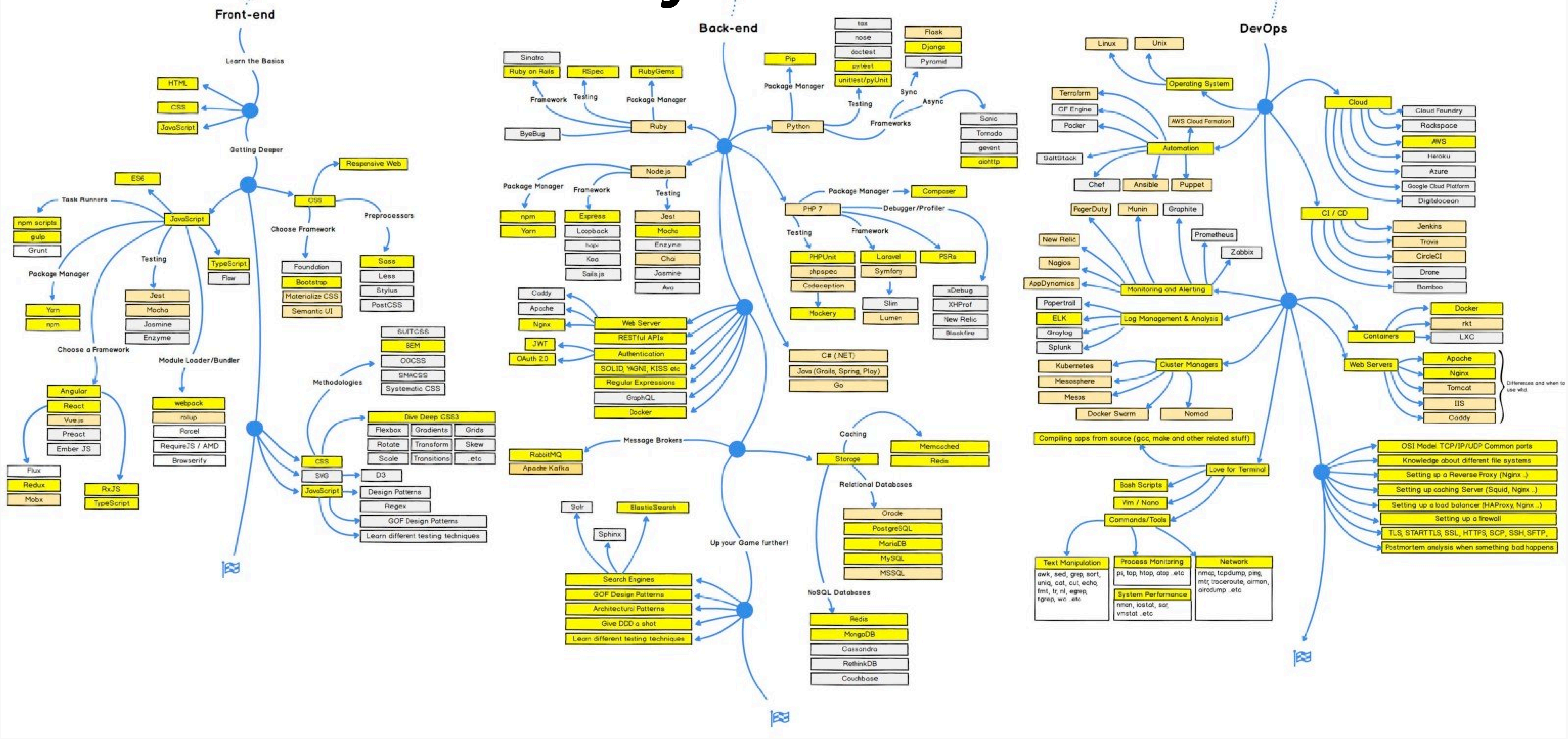
Fun(?) Fact: I have been a professional web developer for <2 years

Learnt C (without consent) in +2
Got into Data Science, ML/DL early into college.
Spent college thinking I’d do deep learning research.
Got into Data Engineering right out of college.
Had to put a Deep Learning application on the web, so I HAD to be a Web Dev.

Lesson: Life (and the internet) is a box of chocolates

Lesson 2: Almost all development is web development

Case Study: Horrors like this



Lesson 3:
Just writing code is not web development

Web developement happens mostly in your head

The InterWeb

A realm filled with communication channels between computers

A combination of processing, communicating through these channels, storing data, and (optional) rendering/displaying the result

Developing something for the internet then, needs:

- Somewhere to Store data
- Some logic to Process data
- Some logic to communicate with other computers on the internet
- Somewhere to put the servers that do this processing and storing
- Some logic to format the result and show it

The Store

"Respect for the truth is the basis for all morality.”
- Duke Leto Atreides, Dune

Practical advice for starting out:

- Learn SQL (Start with SQLite)
- Become comfortable with the syntax and how things work there.
- Learn about noSQL
- Refer to the database section of Slide 5
- Try stuff out and learn about their use cases

The Backend

The layer of translation and processing
1 way and 2 way communication

Practical advice for starting out:

- Choose your language and framework.
- I recommend a minimal framework in any language you're familiar with.
- Learn the REST contract (de facto standard)
- Learn with a minimal FE first. It's more fun this way.
- Learn basics about Design Patterns and system architecture.

The FrontEnd

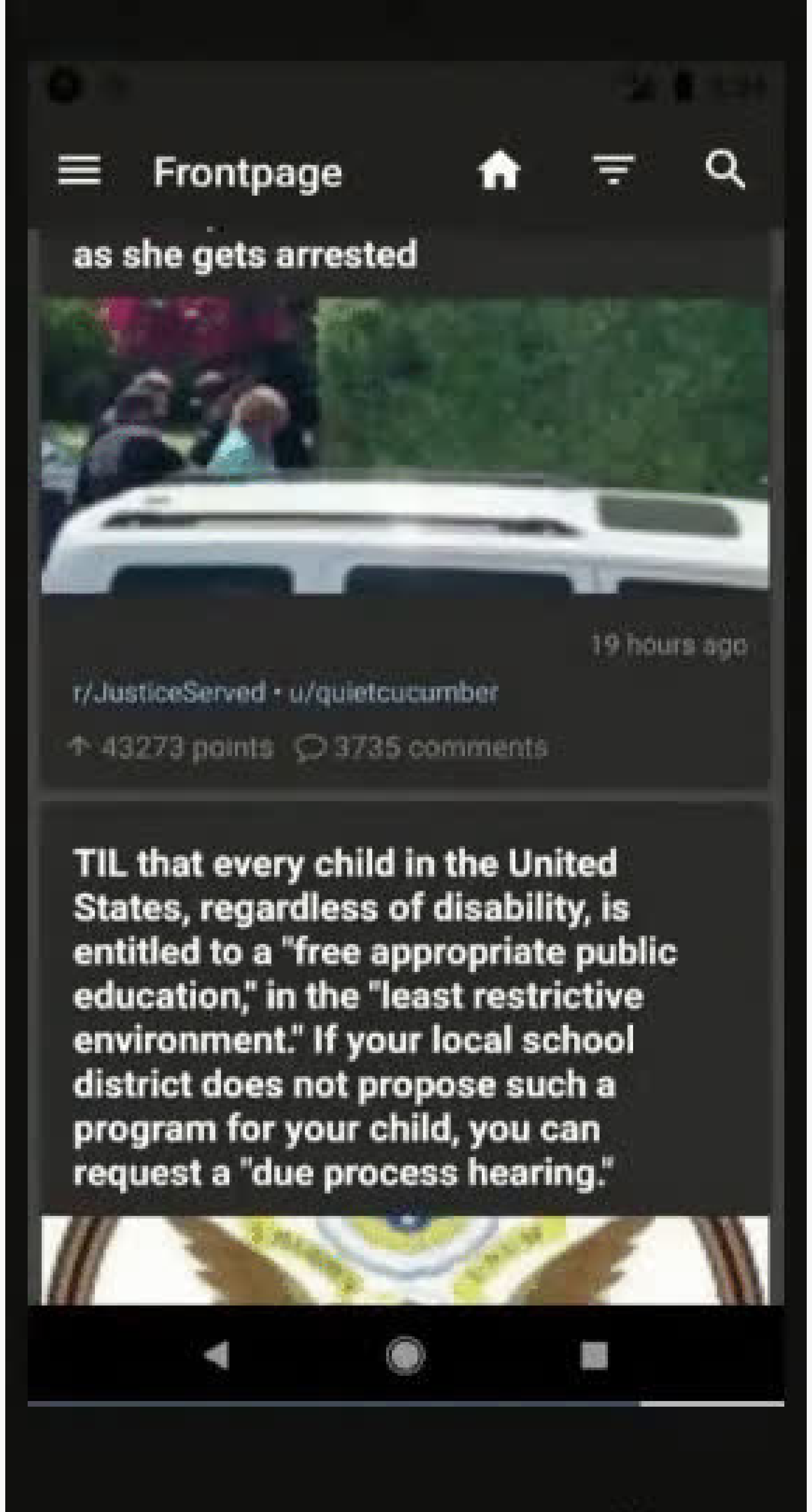
The logic that generates a beautiful page (the fruit of all our BackEnd and Database work)

Practical advice for starting out:

- Learn the basics of HTML and CSS.
- Learn a bit of JavaScript and TypeScript.
- Use some publicly available APIs to create projects (e.g. Pokemon API, Reddit API)
- Learn the basics of good User Experience and Interface Design

Example of a first web project

- Takes information from the Reddit API
- **Iteration 1:** Displays it as a Web Application using React.
- **Iteration 2:** Mock how you think the Reddit BackEnd looks like
- **Iteration 3:** Think about and try to mock how Reddit data is stored in the Database.



as she gets arrested

19 hours ago

r/JusticeServed • u/quietcucumber

43273 points 3735 comments

TIL that every child in the United States, regardless of disability, is entitled to a "free appropriate public education," in the "least restrictive environment." If your local school district does not propose such a program for your child, you can request a "due process hearing."

The Infrastructure

All this code/data/logic has to be somewhere right? In a physical computer?

Yes, and that is where DevOps comes in

DevOps

AKA, You don’t ever need to worry about it, until you do.

<u>Developer Operations</u>

Automating the boring and repeated stuff

Hosting your application on the cloud

Managing costs

Scaling and Reliability

Deployment and Networking

aws

Google Cloud

No Dev Ops is also Dev Ops. (But it’s bad DevOps)

How my journey was (AKA my book of regrets)

College days
(Portfolio Era)

Early Career
(Theory and Tutorial Hell Era)

Modern Era
i.e. Today

Excercise

Be Opinionated (Don’t be afraid to be wrong)

Know The History

Know How people do shit

Engage with the Community

Don’t jump on hype trains

Contd

Use LLMs smartly when learning

Do not ignore soft skills:
Rhetoric, Argumentation, people skills

The Web Dev Experience

When you are spending more time in meetings, Excel & JIRA and very less time doing coding..

The Hackathon Mindset
and its pitfalls

Questions?

The End

And good luck on your WebDev journey