20/10/24

HTML / CSS Refresher

6 classes → HTML & CSS
- → How web works
- → CSS
  - → Basics
  - → Imp. concepts of CSS
  - → Adv. CSS
  - → Responsiveness, Accessibility
  - → Performance → HTML / CSS

Project → Food Delivery App.

Full stack LLD Module

- Frontend {35 classes}
- HLD
- Backend {15 - classes}

* 11 classes on JS
  - → Basics
  - → FP, OOPS, Inheritance
  - → Async JS, concurrency
    - → Event Loop
    - → Cb, cb hell
    - → Promises, A.A
    - → Create your own promise
  - → ES6
  - → Error Handling

Poly fill → is an implementation which is not available on older browsers but present in current browsers.

Good Write

* Frontend Machine Coding → 8 classes
  - ↳ DOM
  - ↳ Browser → storage, drag & drop, intersection Observer
  - ↳ HTTP Protocol
  - ↳ Performance optimization (browser)
  - ↳ Case websites
  - Project → Kanban board

Example Questions : Star rating, Autocomplete, Timer.

* 10 classes React
  - ↳ Basic
  - ↳ Hooks
  - ↳ Routes
  - ↳ Memoization
  - ↳ Redux / context API
  - ↳ Testing, Performance

Project → E-commerce Project.

Design Patterns → standard sol.
↓
Standard prob.

Singleton Design Pattern
Factory Design Pattern

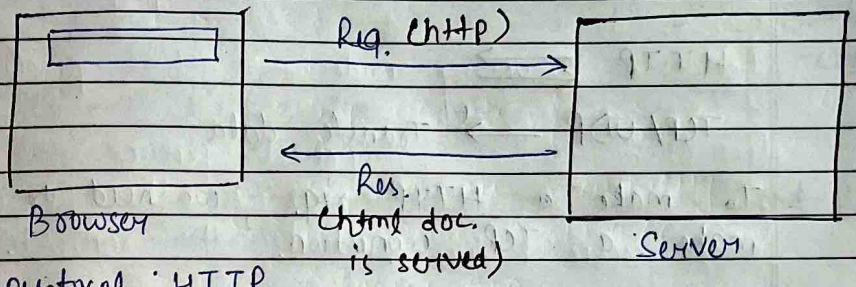
6 → HTML/CSS
11 → JS
8 → FE
10 → React

Q- How web works ?

Task → Process → steps → Bottleneck

Good Write

Request Response cycle -



- protocol : HTTP
- first file : HTML

- Page Load :
  - ↳ Getting the data from server (Navigation)
  - ↳ Parsing the data into UI (convert) (Rendering)
  - ↳ interaction (Interaction)
  - ↳ data → output

Resource Scheduling 1.1 ms

Connection Start 2.17 ms

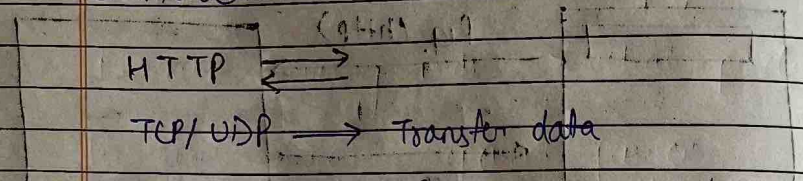
Req/Res 243.77 ms

① DNS Lookup → Domain name server
- example.com → IP Address
  ↓
  URL

② initial connection → TCP handshake establishing a TCP connection

Good Write

Protocol → A set of rules to be followed to communicate with each other.



To make a HTTP req, we need to firstly make a TCP connection.

③ SSL → Secure Socket Layer should be formed to transfer data securely.
SSL Handshake

④ Request ⟶ Response

1 ms → 1000 sec
↓
0.36 ms

Q- Reduce your wait time for a server?
Ans- . CDN - content delivery network → to geograp -hically nearest server.
. Replicate data points. (Server replication)

① Navigation -
→ DNS lookup ①
→ TCP Handshake ②
→ SSL Handshake ③
→ Req/Res cycle ④

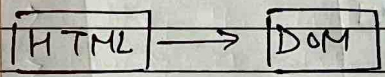
Good Write

Latency - Time b/w req. & res.
Sol - CDN
↳ placing db nearest to the client

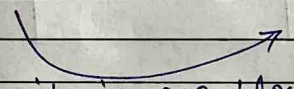
Bottlene -cks

TCP → (first packet received is 14 KB)
Slow start mechanism
↳ Hampers the initial experience of end-user.
↳ html content → light

② Parsing -
Critical Rendering Path : Steps req. to convert the HTML document → Rendered UI



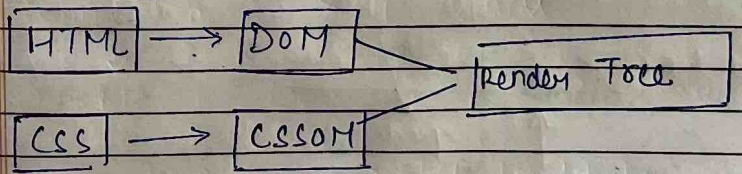
document obj. model



As soon as it encounters JS, it is loaded & executed, then only the HTML will be render -ed.
All the steps are sequential.

Parsing
↳ CSS → CSSOM

③ Render Tree -



Good Write

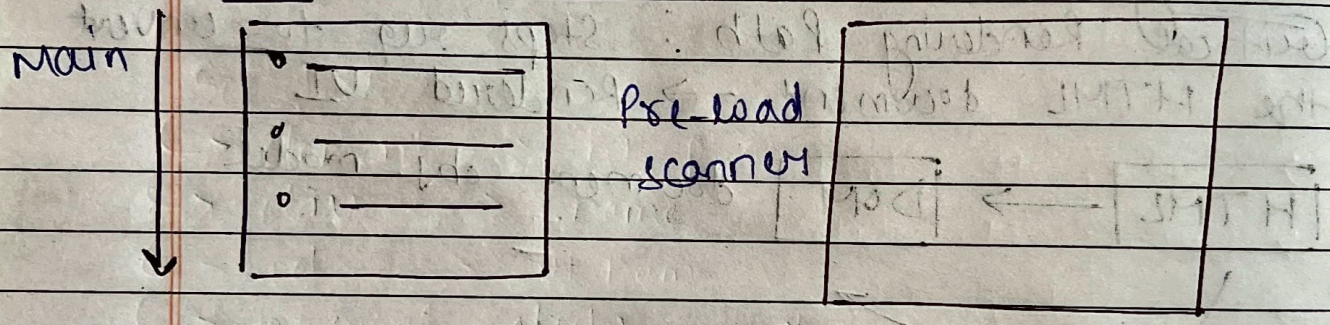
DATE: __/__/____
PAGE: _____

④ Layout - Process of laying out your rendered dom tree into the actual page. (nodes)

⑤ Render - Converting nodes → actual pixels.

Pre-load Scanner - Separate parser
↳ Scanner through content & make the request.
↳ Usecase: Reduce the wait-time

Render -



Keep the html file very small.

Interactivity -
↳ all your event listeners will be attached.
↳ defer your JS file