Agenda.
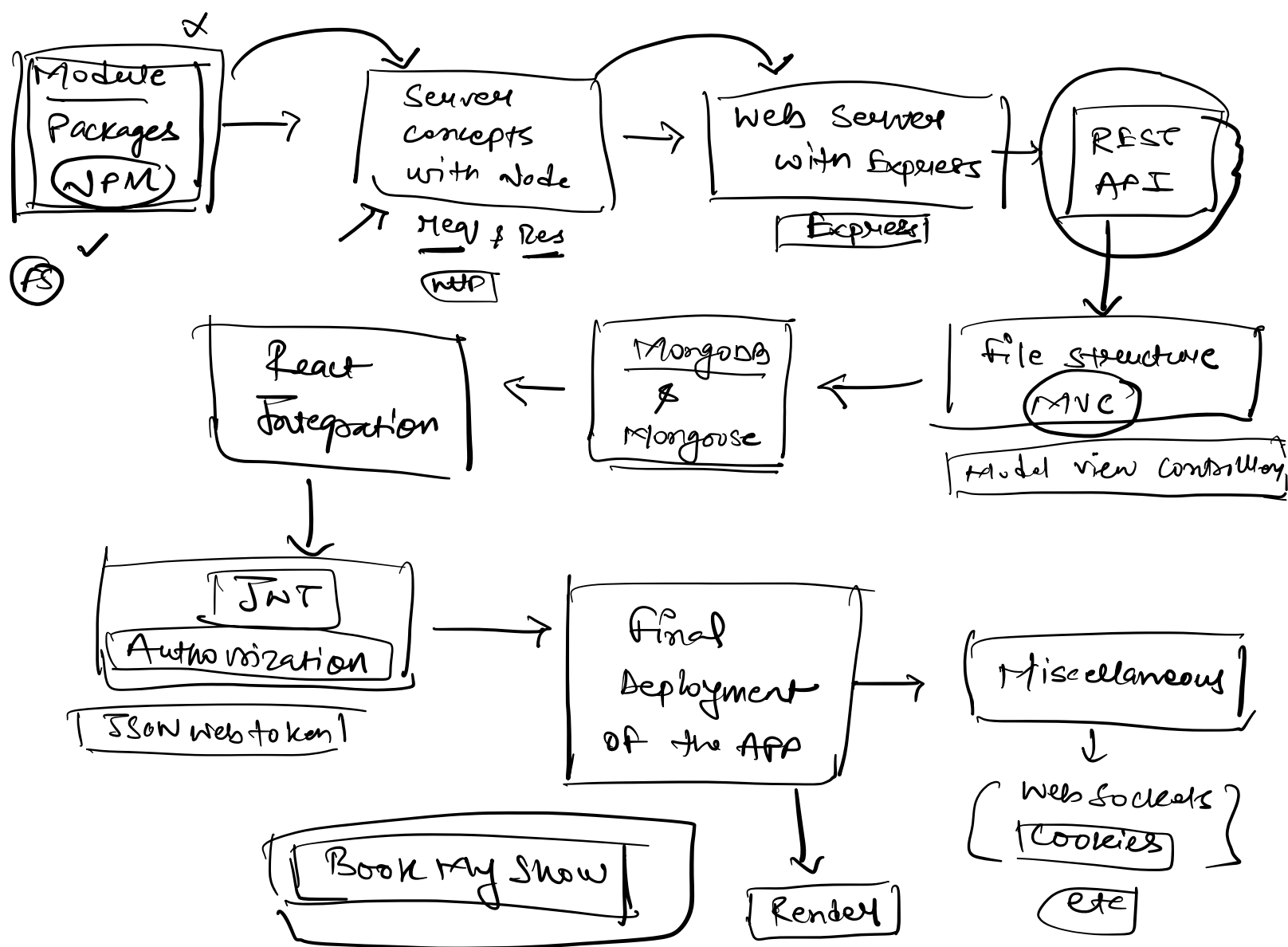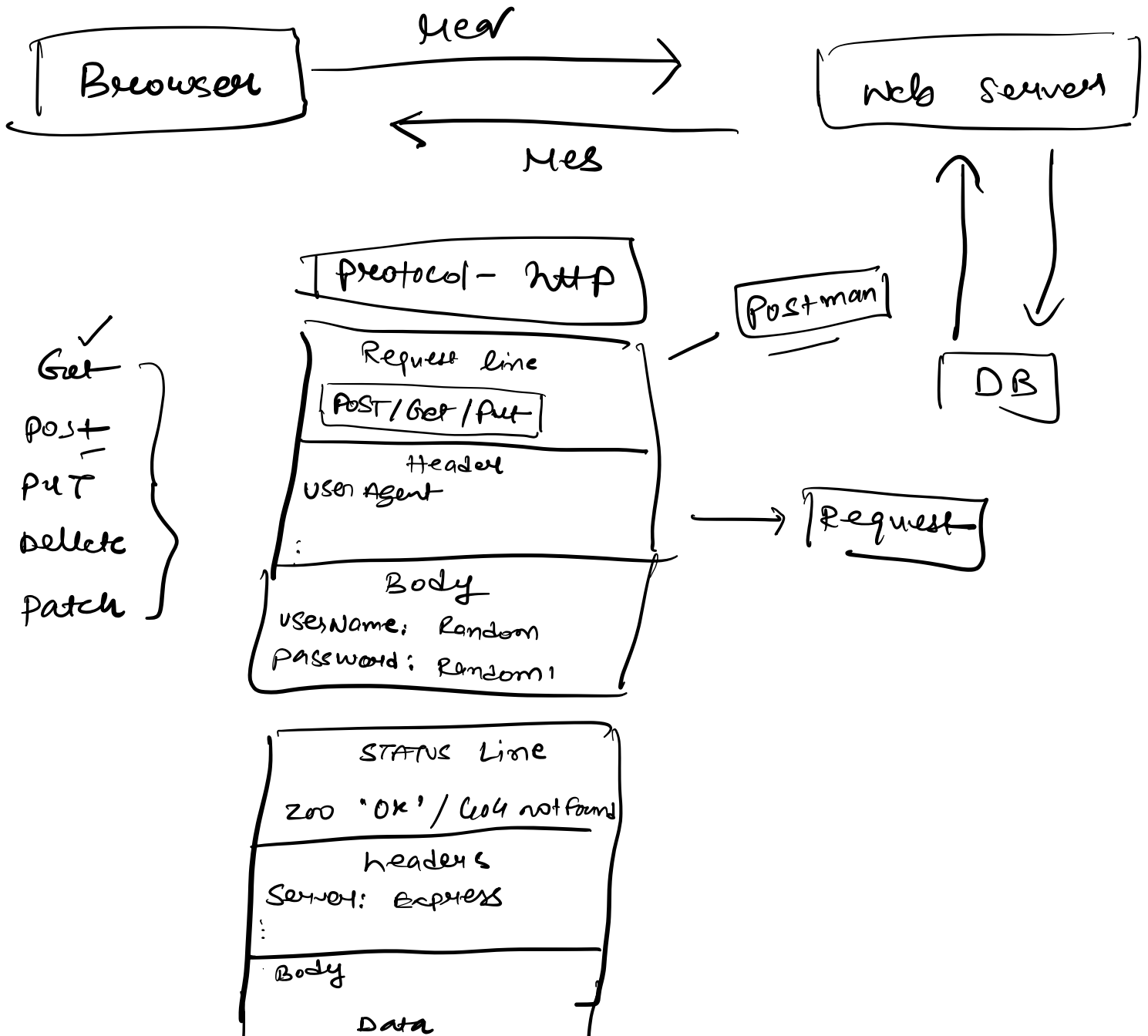
1) Understanding In-built modules of Node.js (FS)

2) How to create a small web server with Node.js

3) Understanding Req & Res with the http module

4) Serving static html & JSON files via the server

Roadmap of the Module

```
┌─────────────┐        ┌──────────┐        ┌──────────────┐       ╭─────────╮
│ Module      │  x     │ Server   │        │ Web Server   │       │ REST    │
│ Packages    │───────▶│ concepts │───────▶│ with Express │──────▶│ API     │
│  (NPM)      │        │ with Node│        │              │       ╰─────────╯
└─────────────┘        └──────────┘        └──────────────┘            │
   (FS) ✓                Req & Res            [Express]                 │
                          [http]                                        ▼
                                                               ┌──────────────┐
┌──────────────┐       ┌──────────┐        ┌──────────────┐    │ file structure│
│  React       │       │ MongoDB  │        │              │◀───│    (MVC)     │
│ Integration  │◀──────│    &     │◀───────│              │    └──────────────┘
│              │       │ Mongoose │                            Model view Controller
└──────────────┘       └──────────┘
      │
      ▼
┌──────────────┐                    ┌──────────────┐
│   JWT        │                    │  Final       │        ┌──────────────┐
│ Authorization│───────────────────▶│ Deployment   │───────▶│ Miscellaneous│
└──────────────┘                    │ of the APP   │        └──────────────┘
 [JSON web token]                   └──────────────┘               │
                                           │                       ▼
       ┌──────────────┐                    ▼                 ┌──────────────┐
       │ Book My Show │                 ┌────────┐           │ web sockets  │
       └──────────────┘                 │ Render │           │  [Cookies]   │
                                        └────────┘           └──────────────┘
                                                                   (etc)
```

When to use Node.js? & why to use Node.

1) Heavy I/O operations - Media streaming ✓
2) Small web servers with less computation ✓

Starting at 10:55 PM

Browser ——— Req ———→ Web Server
        ←——— Res ———

Protocol - HTTP

Postman

Request line
[POST / Get / Put]

Header
User Agent
:

Body
UserName: Random
Password: Random1

——→ Request

Get ✓
Post
PuT
Delete
Patch

DB

STATUS Line

200 "OK" / 404 notfound

headers
Server: express
:

Body
Data

html / JSON

Response Status. (status codes]

(2)xx → 200   201   202

Success   [OK] | Created | | Accepted |

4xx → 400   (404)   (400)

| Errors | | Bad Request | | Not Found | | Bad Request |

5xx →   500   502

| Server | Internal | Bad
Errors | server   Gateway |
      Error |

3xx
| Redirections)

Application.

1. mp4 ,   2. mp4 , 3. Pdf , 4. xl , 5. doc

6. mp4 ,   7. mp3

media	documents	
↓	↓	
mpu , MP3	xl , pdf	

SRC file — [f1.txt] — [SRC file Path]

Destination Directory — [my directory] ———→ [Path]

↓

f1.txt

[paste]
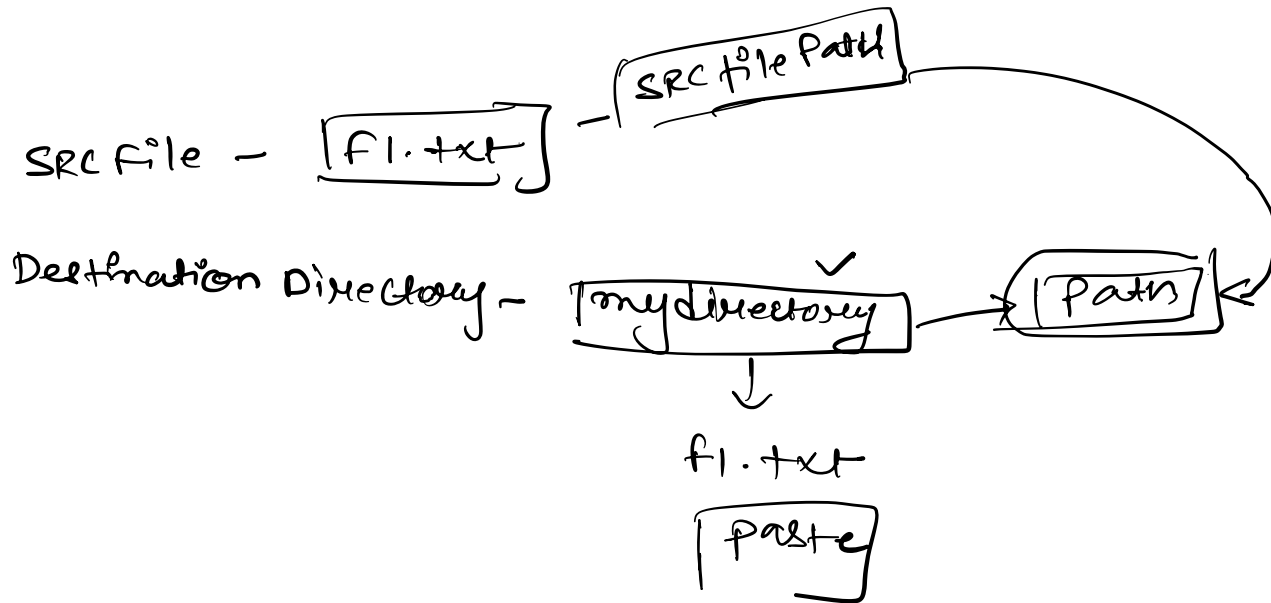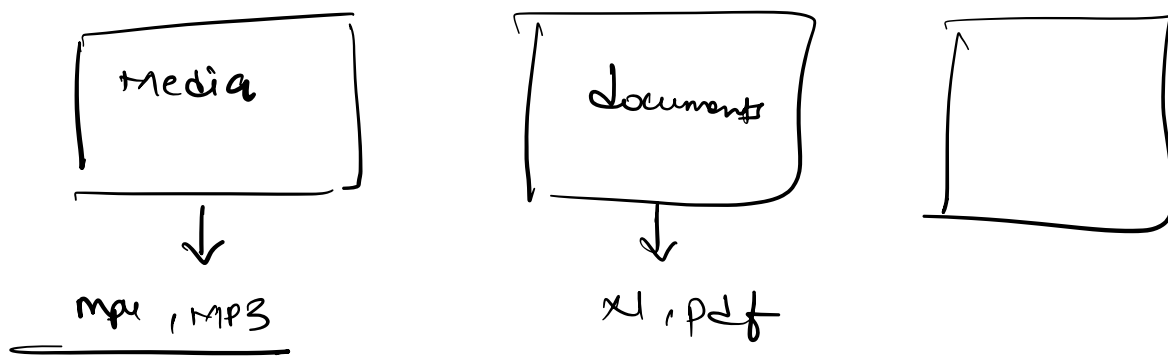
(MERN)

Companies which have their own stack.

(Google Amazon)

[Software Engineer]

↓

[DSA skills]

[Problem solving skills.]

Arrays
strings
DP
Recursion

Tree
Graph
Tries
Stacks
Queues
L L

Companies —   Role- specific   Take-home Assignment
                                                        ↓
                                                       code
                    [MERN developer]   ↓
                                                       github
                    Array }   ↓
                    String } ✓   (deploy)
                    (JS)
                    React }   (6 — 15 LPA)
                    Node }