**1) MERN** ( Backend )

Node js backend  
React js frontend

-> JS - Asynchronous

          - Loosely Typed

          - Single Threaded

Learn - Solve in Leetcode - understand the above 3 more in JS

-> Backend - what is backend ?  
                   - How is it built ?

                   - HTTP servers Express

                   - Databases ( NO SQL / SQL )

-> Frontend - what is Frontend?

                   - How is it built ?

                   - REACT

                   - state management

                   -    - Recoil, Redux  
**Till here Checkpoint 1 :**The above are to understand and get actual scenario of creating and hosting web Development.  
Follow further points, to get paid......  
i) Type script  
Not learned but in one or other projects it will be used.  
google or chatgpt

Technologies:  
GRPC, TRPC, openAPI, SPEC,  
Dockerization, AWS (Deployment), ORM,  
Git/ GitHub (Version Control ), GraphQL  
Vercel :   
     - Next js

     - Turbo Repo  
**Till here Checkpoint 2 :  
1)**Create a lot of projects on your own (Hard part)

**2)**Checkpoint My Knowledgewith open source Codebases ( right way to build industrial skills )

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DevOps \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Build code plan test                              release deploy operate monitor

**4 Technologies  
1) docker / Containerization**

-> Dockerize your Own Full stack APP

-> Setup Open source codebases Locally using Docker

**2) IAC**

-> Terraform

-> Creating your own terraform provider

**3) K8s**

put containers in various machines (DO, GCP)  
**4) HELM**

-> Package manager

      - templates

-> Commands - helm update

**5)**Writing low-latency code ( For high pay ) (Build trading system)

 GoLang

RUST

Arbitrary

market making

**6) AI**

**Buzz words that you need to learn that lead to a high paying job**

                                                          \*

**Build Technical Experience as a student throug -**

**Personal Projects  
Hackathons**

**Internships**

**Open source**

**Freelance**

**ESSENTIALS:**

0. DSA

1. Cloud Computing (Be Cloud Certified...AWS/Google/Azure)

2. Git/Version Control (learn basics on YouTube)

3. Basic Database Knowledge DBMS, SQL

4. Basic Knowledge of Linux

5. Basic Docker Knowledge (learn docker in 1hour)

6. Basic API knowledge (API for each language) JAVA API, PY API

7. Pipeline Familiarity (GitHub actions),

8. Issue management - tracking and communicating progress (to yourself and others)

9. Self-control - you won't perfect things 1st or 2nd try. So, accept your crappy first try and iterate on it later

10. Sketching - diagrams, flows, analysing — planning ahead or understanding a problem BEFORE "programming" will give you so many benefits

11. Keep moving forward.

We tend to sometimes get stuck on problems. Sometimes it's better to go to take a break and work on something different, and coming back later with a different perspective. Don't expect to find the "perfect approach" on the first try. You won't. Save yourself the trouble. A scuffed solution is better than never finishing. I promise. And that comes from somebody who tends to get too into the small details.

1. Identify discrimination

2. Ignore & move On

3. Learn from people who struggle & at your position earlier

4. Believe & work on yourself (open source & remote & skills)

**Your Life Your Responsibility To be 100% true to your skills & Work**

ATS friendly Resume: Keywords in JD, in Resume Sections

Blogging

Website

Portfolio

Leet Code

GitHub

LinkedIn

Twitter

Self-Show Case

**SKILLS:**

**Basics:**1: JAVA, OOP, SQL, OS, CN, DSA, SD, SDLC

**Phase 1 study:** from documentations, Opensource road maps, Blogs, Notes

**Phase 2 practical:** Hackerank, Linkedin tests pass out

**Technology 1:**FSD HTML, CSS, JS

**Phase 1 study:** from documentations, Opensource road maps, Blogs

**Phase 2 practical:** freecodecamp (as of now), Open source

**Technology 2:**DevOps

**Phase 1 study:** from documentations, Opensource road maps, Blogs

**Phase 2 practical:** Certified, Opensource prj

**Technology 3:**AI

**Phase 1 study:** from documentations, Opensource road maps, Blogs

**Phase 2 practical:** Open source

**General:**

**Java Development:** Open source Prj

**1**.JAVA, OOP, SQL, OS, CN, DSA, SD, SDLC

**2**.FSD

F: HTML, CSS JS

B: JS NPM

**3**.DEVOPS

Kubernetes, Containerization Docker, Terraform, Ansible, GCP, Git, Linux

**4.**Java Development - Angular, SpringBoot, API

Nov 14 - Nov 26

Dev1: Leetcode - DSA

Dev2: Coursera - GCP

After sem exams: Hackerrank linkedin Tests - JAVA, SQL, OOP, DSA, OS, CN

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | STUDY |  |  | IMPLEMENT |  |  |
| **Programming Languages:** |  |  |  |  |  |  |  |
| Java |  |  |  |  |  |  |  |
| SQL |  |  |  |  |  |  |  |
| HTML |  |  |  |  |  |  |  |
| CSS |  |  |  |  |  |  |  |
| JS |  |  |  |  |  |  |  |
| PY |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Core Subjects** |  |  |  |  |  |  |  |
| OOP |  |  |  |  |  |  |  |
| DSA |  |  |  |  |  |  |  |
| OS |  |  |  |  |  |  |  |
| CN |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **DevOps** |  |  |  |  |  |  |  |
| Kubernates |  |  |  |  |  |  |  |
| Docker |  |  |  |  |  |  |  |
| Terraform |  |  |  |  |  |  |  |
| Ansible |  |  |  |  |  |  |  |
| Git |  |  |  |  |  |  |  |
| LINUX |  |  |  |  |  |  |  |
| GCP |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Java Development** |  |  |  |  |  |  |  |
| Angular |  |  |  |  |  |  |  |
| SpringBoot |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **AI** |  |  |  |  |  |  |  |