Day 12 : Technical Competency: Recruiters

Technical Competency: Recruiters' POV – Apni IT Skills Badhao!

TATA CONSULTANCY SERVICES (TCS) ka yeh presentation technical competency aur IT skills develop karne par hai, especially recruiters ke perspective se.

Recruiters Kya Expect Karte Hain? – Quality vs. Quantity!

Recruiters aksar dekhte hain ki interviewees ko kai technologies jaise Active Directory, Natural Language Processing (NLP), aur Internet of Things (IoT) ki knowledge hoti hai. Lekin, yeh knowledge aksar **superficial** hoti hai, matlab bas upar-upar se pata hota hai, real working knowledge nahi hoti.

Simple Bhasha Mein: Interview dene waale students ko bahut saari technologies ka naam toh pata hota hai, par jab unko deep mein pucho ya practically use karne ko bolo, toh unko zyada nahi aata. Sirf "I know everything!!!" bolne se kaam nahi chalega, working knowledge honi chahiye.

Ek Example:

- **Superficial Knowledge:** "Java ek programming language hai... BigData ka matlab bahut saara data store karna hai... NLP matlab computer actual language padhta hai... Database data store karta hai fetching ke liye... Data warehousing data ko warehouse mein dalna hai... AI matlab computer intelligent ban raha hai."
- **Working Knowledge:** "Java ek Object-Oriented Language hai. Isme saari classes Object class ko extend karti hain. Interfaces standardization ke liye use hote hain taaki common calling code ban sake. Final methods class ke behavior ko fix karte hain jise users override na kar saken. Threads processes ko asynchronously chala sakte hain."

Moral: Sirf kitabi gyaan nahi, asli kaam ka gyaan hona chahiye!

Basic IT Skills: Kya Zaroori Hai?

Ek skill hona helpful ho sakta hai, but it's not enough to be ahead of the line.

Must-Have Skills:

- **Koi bhi ek Programming Language:** Jaise Python, C, Ruby, C++, Java.
    - **Hinglish Mein:** Koi bhi ek programming language mein strong pakad banao, jo tumhe pasand ho.
- **OOPS Concepts:** Object-Oriented Programming ke concepts ko real-world coding examples se relate kar pao.
    - **Hinglish Mein:** OOPS concepts (Abstraction, Encapsulation, Polymorphism, Inheritance, Class, Object) ko sirf theory mein nahi, balki practical examples se samjho. Jaise real life mein object kaise behave karte hain, wahi concepts code mein kaise apply hote hain.
    - **Image Description (OOPS Concepts Wheel):** Ek wheel diagram dikh raha hai jismein "Object" se shuru hokar "Class", "Inheritance", "Polymorphism", "Abstraction", aur "Encapsulation" tak concepts dikhaye gaye hain, jo OOPS ke pillars hain.

- **Database:**
    - RDBMS (Relational Database Management System) ki knowledge.
    - Schema / Tables / Columns / Indexes ki samajh.
    - SQL (Structured Query Language) mein DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), TCL (Transaction Control Language) ki commands aani chahiye.
    - **Hinglish Mein:** Database mein data kaise organize hota hai (tables, columns), aur usko retrieve, modify, ya manage kaise karte hain SQL commands se, yeh sab aana chahiye.
- **HTML/JS/CSS:** Simple HTML page banana aana chahiye, validations ke saath, aur simple styling ke saath.
    - **Hinglish Mein:** Web pages banane ke basics (HTML for structure, CSS for styling, JavaScript for interactivity) clear hone chahiye. Simple forms banana aur validate karna pata ho.

Recruiters Ko Kya Pasand Aata Hai? – The Delight Factor!

Recruiters aise candidates pasand karte hain jo "Awesome" hote hain aur jinko woh apni team mein lena chahte hain.

Aise log kya karte hain?

- **Basic Algorithms:** Searching (Linear/Binary), Sorting (Bubble, Selection, Insertion, Merge), Tree Traversal, Dijkstra, Linked Lists, Hashing.
    - **Hinglish Mein:** Data structures aur common algorithms ki strong understanding honi chahiye. Sirf naam nahi, unko implement karna aur unki complexities (time and space) ko samajhna aana chahiye.
- **Design Patterns:** Factory, Singleton, Adaptor, Object Pool.
    - **Hinglish Mein:** Software design patterns ki basic knowledge ho, yeh re-usable solutions hote hain common problems ke liye.
- **Web Application Flow:** HTTP, Request, Response, HTML.
    - **Hinglish Mein:** Jab tum browser mein kuch type karte ho aur website open hoti hai, toh background mein kya hota hai? Request kaise jaati hai, server kaise process karta hai, aur response kaise aata hai, yeh process samajhna zaroori hai.
    - **Image Description (Web Application Flow):** Ek flowchart dikh raha hai jismein CLIENT (browser, widgets) WEB SERVER se communicate kar raha hai HTTP, AJAX, JSON, HTML ke through. WEB SERVER mein WEB SERVICES aur BUSINESS LOGIC hoti hai.
- **One Digital Skill:** NLP, AI Algorithms, Data Warehousing, Big Data.
    - **Hinglish Mein:** Kisi ek trending digital skill mein deep knowledge rakho. Yeh tumhe baaki candidates se aage rakhega.

Important Pointers for Interviews:

- **Final Year Project par Time Spend Karo:**
    - Project wisely choose karo – current business issues ya digital trends se related ho.
    - **Hinglish Mein:** Apne final year project ko seriously lo. Aisa project chuno jo industry ki real problems solve karta ho ya latest technologies use karta ho. Isse tumhari practical knowledge dikhegi.
- **Apni Technical Strengths Upfront Communicate Karo:**
    - Interviewers good people ko chodna nahi chahte.
    - **Hinglish Mein:** Interview shuru hote hi, jo topics tumhe bahut acche se aate hain, unko clearly bata do. Interviewer ko tumhari strengths pata chal jayengi.
- **Jo Nahi Pata, Use Accept Karo:**
    - Interviewers ke paas zyada time nahi hota. Conversation ko known topics ki taraf direct karo.
    - **Hinglish Mein:** Agar kisi cheez ka answer nahi pata, toh saaf-saaf bol do. Guess mat karo. Phir politely conversation ko un topics par le aao jinke baare mein tumhe achha gyaan hai.
- **Solution tak Pahunchne ke Logical Paths Do:**
    - Agar answer sure nahi ho, toh Methodology describe karo ya Draw karke samjhao.
    - **Hinglish Mein:** Agar exact answer nahi pata, toh ye batao ki tum us problem ko kaise approach karoge, kya steps loge, ya agar koi diagram banakar samjha sakte ho toh banao. Tumhari problem-solving approach important hai.
- **Effectively Communicate Karo:**
    - Long sentences use mat karo; concise, clear aur loud raho.
    - **Hinglish Mein:** Jo bhi bolo, seedha, saaf, aur confident hokar bolo. Lambi-lambi kahaniyan mat banao.

Basic IT Competencies: Deep Dive

Basics of HTTP / Web Application

Request-Response Cycle:

1. **Browser User Types an address:** Jaise careers.tcs.com/a/b/c.html.
2. **Browser asks DNS:** careers.tcs.com kya hai?
3. **Gets the IP of a machine:** DNS us address ka IP address deta hai.
4. **Sends request text to that machine on a "Port" (80/443):** Browser us IP address par request bhejta hai.
    - **Request Message Header:**
        - **Request Line:** GET /doc/test.html HTTP/1.1

        - **Request Headers:** Host, Accept, Accept-Language, Accept-Encoding, User-Agent, Content-Length.
        - **Request Message Body:** bookId=12345&author=Tan Ah+Teck (Agar kuch data send kar rahe ho). Headers aur body ke beech ek blank line hoti hai.

5. **Machine / Server:**
    - OS required "Port" par running process (Server) ko dhundhta hai.
    - Request us process ko deta hai.
    - Process

/a/b/c.html file (Resource) ko dhundhta hai.

    - HTML ko machine par wapas bhejta hai.
    - **Response Message Header:**
        - **Status Line:** HTTP/1.1 200 OK
        - **Response Headers:** Date, Server, Last-Modified, ETag, Accept-Ranges, Content-Length, Connection, Content-Type.
        - **Response Message Body:** <h1>My Home page</h1> (Actual HTML content). Headers aur body ke beech ek blank line hoti hai.

Hinglish Mein: Jab hum koi website kholte hain, toh hamara browser (client) ek request banata hai (jismein hum kya dekhna chahte hain aur hamari details hoti hain). Yeh request internet par server tak jaati hai. Server us request ko process karta hai aur desired webpage ko response mein wapas bhejta hai, jise hamara browser show karta hai.

Digital Competencies: The World Is Changing!

Digital skills aaj ke time mein bahut zaroori hain. Kuch key areas:

- **Artificial Intelligence (AI):**
    - **Hinglish Mein:** AI matlab machines ko insaano jaisa sochna sikhana. Jaise computer ko decision lena sikhana, data se patterns dhundhna.
    - **Image Description (AI and Statistics):** Ek circuit board ke design mein "Artificial Intelligence" likha hai, jo "equals" sign ke baad ek pie chart aur magnifying glass ke saath "STATISTICS" se juda hai. Iska matlab AI kaafi हद तक statistics par depend karti hai.
    - **Bayes Theorem:** $P(A|B)=P(B)P(B|A)P(A)$
        - $P(A|B)$: The probability of A being true given that B is true.
        - $P(B|A)$: The probability of B being true given that A is true.
        - $P(A)$: The probability of A being true.
        - $P(B)$: The probability of B being true.

        - **Hinglish Mein:** Bayes Theorem ek mathematical formula hai jo probability calculate karta hai ek event ki, given ki koi doosra event already ho chuka hai. Yeh AI mein prediction aur classification ke liye use hota hai.
    - **Linear vs. Non-Linear Fitment:** Yeh dikhata hai ki data points par kaise models fit kiye jaate hain. AI mein data ko analyze karne ke liye alag-alag mathematical models use hote hain.
    - **Supervised Learning:** Agar naya data aata hai (jaise koi naya person apni age aur vocabulary batata hai) aur tumhara system usko "Bad Vocabulary" batata hai, toh tum is naye point ko apne graph mein add kar sakte ho future use ke liye. Isse "Supervised Learning" kehte hain.
        - **Hinglish Mein:** Supervised learning AI ka ek type hai jismein machine ko labeled data (input aur uska correct output) dekar train kiya jaata hai. Jaise agar hum machine ko bahut saare birds ki pictures (labeled "bird") dikhayein, toh woh naye bird ko pehchan lega.
- **Data Warehousing:**
    - **Hinglish Mein:** Data warehousing ek process hai jismein alag-alag sources se data collect karke ek centralized repository (data warehouse) mein store kiya jaata hai. Yeh data analysis aur reporting ke liye use hota hai.
    - **Student Management System Example:**
        - Ek Student Management System mein "Normalized" tables hote hain jo transactions (save/read) ke liye design kiye jaate hain.
        - Management ko weekly stats chahiye (e.g., Daily Fee Collection Amount, QQQ Salary Spending, Fee Collection vs Store Spending for a College for a Quarter).
        - Aisi queries long aur complex hoti hain Student Management System par.
        - Har raat/weekly/monthly, data ko Extract, Transform (summarizing, joining by college, etc.), aur Load karke ek naye Database mein daala jaata hai.
        - Yeh naya database, jo primarily reporting aur analysis ke liye hai, usko "Data Warehouse" kehte hain.
    - **Image Description (Data Warehousing Flow):** Left side mein Library Data, Student Data, Payroll Data, aur Store Data dikh raha hai. Yeh sab data ek bade cylinder jaisi structure mein jaa raha hai, jise "Data Warehouse" kaha gaya hai. Right side mein OLAP Engine, Reports, aur Data Mining ka zikr hai, jo data warehouse se derive kiye jaate hain.
    - **Hinglish Mein:** Socho college ka sara data (students, library, fees, salary) alag-alag jagah hai. Data warehousing mein, yeh sara data ek jagah ikattha kiya jaata hai, usko clean aur organize kiya jaata hai, taaki future mein usko analyze karke important decisions liye ja saken (jaise kitni fees aa rahi hai, kahan kharcha ho raha hai).
- **Big Data:**
    - **Hinglish Mein:** Bahut bade aur complex datasets ko Big Data kehte hain jo traditional data processing software handle nahi kar sakte. Ismein high volume, velocity, aur variety ka data hota hai.

- **Natural Language Processing (NLP):**
    - Information Retrieval, Sentiment Analysis, Information Extraction, Machine Translation, Question Answering.
    - **Hinglish Mein:** NLP machine ko insaani bhasha samjhna sikhata hai. Jaise Google Translate, ya jab tum phone par "Hey Google" bolte ho, toh woh NLP ka use karta hai. Isse machine text ko padh sakti hai, uska matlab nikal sakti hai, aur jawab de sakti hai.

Overall take-away: Sirf theory nahi, practical knowledge aur problem-solving approach develop karo. Apni strengths ko highlight karo aur jo nahi aata use honestly accept karo, but solution tak pahunchne ka logical path provide karo. Aur sabse important, current digital trends aur technologies par dhyan do!

Case Study (12) :

Case Study Ideas :

1. **"Superficial vs. Working Knowledge" Case Study (Recruitment Focus)**

- **Problem Statement:** Ek company (jaise TCS) high volume mein fresher engineers hire kar rahi hai. Recruiters ko lagta hai ki bahut saare candidates ke paas trending technologies (AI, IoT, NLP, Big Data) ki **superficial knowledge** hai, **working knowledge** nahi hai. Isse recruitment process mein time waste hota hai aur right talent milna mushkil ho jaata hai.
- **Case Study Question:**
    - Aisi situation ko improve karne ke liye recruitment process mein kya changes laaye ja sakte hain?
    - Candidates ko apni working knowledge demonstrate karne ke liye kya platforms ya methods provide kiye ja sakte hain?
    - Interviewers ko kaise train kiya jaaye ki woh superficial knowledge ko jaldi identify kar saken?
    - Candidates ko apni skills ko

"Delight Factor" tak pahunchane ke liye kya advice di jaani chahiye?

- **Possible Solutions (Case Study ke Hisaab Se):** Skill-based assessments, live coding challenges, project-based interviews, mock interview sessions with feedback.

2. **"Final Year Project Implementation" Case Study (Practical Skill Development)**

- **Problem Statement:** College students aksar final year projects ko bas degree complete karne ke liye karte hain, na ki real-world problem-solving ke liye. Isse unki practical skills develop nahi ho paati aur recruiters unke projects mein depth nahi dekhte.
- **Case Study Question:**
    - Ek college (ya student group) ko current business issues ya digital trends (jaise AI for healthcare, IoT for smart city, Big Data for e-commerce) par based ek impactful final year project choose aur implement karne mein kaise guide kiya jaaye?

    - Project selection se leke communication tak, kya steps hone chahiye taaki project recruiters ko impress kar sake?
    - Kon-kon si technical competencies (OOPS, Database, Algorithms, Web Application Flow) is project mein use ho sakti hain aur unhe kaise integrate kiya jaaye?
- **Possible Solutions (Case Study ke Hisaab Se):** Mentorship programs, industry collaboration for project topics, focus on design patterns and algorithms in project implementation, effective communication of project outcomes.

3. **"Digital Transformation for a Small Business" Case Study (Digital Competencies Application)**
    - **Problem Statement:** Ek small manufacturing company hai jo abhi bhi apne operations manually manage karti hai. Unhe market mein competitive rehne ke liye digital transformation ki zaroorat hai. Unke paas raw data bahut hai, par use analyze nahi kar paate.
    - **Case Study Question:**
        - Is company ko AI, Data Warehousing, aur Big Data jaise digital competencies ko apnane mein kaise help ki ja sakti hai?
        - Ek basic Data Warehouse kaise design kiya ja sakta hai jisse unka manufacturing data (production, inventory, sales) analyze ho sake?
        - AI/ML algorithms ka use karke inventory prediction ya quality control mein kaise improvement laayi ja sakti hai?
        - Unke existing customer feedback (textual data) se sentiment analysis (NLP ka use karke) kaise kiya ja sakta hai?
    - **Possible Solutions (Case Study ke Hisaab Se):** Step-by-step implementation plan for data warehousing, choosing appropriate AI models (e.g., linear regression for prediction), integrating NLP tools for text analysis.

4. **"Optimizing a Web Application for Performance" Case Study (Basic IT Competencies)**
    - **Problem Statement:** Ek e-commerce website hai jiske users ko slow loading times aur frequent errors ki shikayat hai. Website ka current architecture bahut basic hai aur traffic badhne par perform nahi kar pa raha.
    - **Case Study Question:**
        - Website ke performance issues ko diagnose karne ke liye HTTP Request/Response flow ko kaise analyze kiya jaayega?
        - OOPS concepts , Database optimization (SQL queries, indexing) , aur basic Algorithms (e.g., efficient searching in products) ka use karke performance kaise improve ki ja sakti hai?
        - HTML/JS/CSS mein kya changes kiye ja sakte hain client-side performance badhane ke liye?
        - Web Services aur AJAX jaise advanced web application flow concepts ko kaise integrate kiya ja sakta hai smoother user experience ke liye?
    - **Possible Solutions (Case Study ke Hisaab Se):** Code refactoring using OOPS principles, database indexing, optimizing SQL queries, implementing client-side validations, using asynchronous calls for data fetching.