

SECTION B: WEB DEVELOPMENT

(For: Full Stack, MERN, Java Full Stack, Web Designing)

1. A user complains that a form on your website isn't submitting properly. Walk through the systematic troubleshooting steps you would take, starting from the user's browser.

- First, I would ask the user which browser and device they are using and try to reproduce the issue on my side.
- I would check if all required fields are filled correctly and validate inputs on the client side.
- I would open the browser Developer Tools and look for JavaScript errors in the Console.
- I would inspect the Network tab to see if the form request is being sent and whether it gets a response from the server.
- Finally, I would check the backend logs to ensure the server is receiving the request and processing it without errors.

2. Explain the difference between front-end and back-end development using the analogy of a restaurant. What "roles" do HTML, CSS, JavaScript, and a server-side language play in this analogy?

- The front-end is like the dining area of a restaurant where customers interact. The back-end is like the kitchen where food is prepared.
- HTML is the structure of the restaurant, like tables, chairs, and menu layout.
- CSS is the decoration and styling, such as colors, lighting, and presentation.
- JavaScript is the waiter who takes orders and responds to customer actions.
- A server-side language (like Java, Node.js, or Python) is the kitchen staff that processes orders and prepares the food (data).

3. You're building a social media app where users can post messages. Describe how data flows from when a user types a message to when it appears on another user screen. Mention at least 4 different components or technologies involved.

- First, the user types a message on the website or app screen.
- The front-end (HTML, CSS, JavaScript) takes the message input.
- JavaScript sends the message to the server through a request.

- The backend receives the message and processes it.
- The message is stored in the database.
- When another user opens the app, the server sends the stored message.
- The message is shown on the other user's screen.

Components/Technologies involved:

- Front-end (HTML, CSS, JavaScript)
- JavaScript & API/HTTP request
- Backend server
- Database

4.What are the three core technologies that make up the “front-end” of any website, and what is the specific responsibility of each?

- HTML (HyperText Markup Language):

It is used to create the structure and content of the website, like headings, paragraphs, images, and links.

- CSS (Cascading Style Sheets):

It is used to style and design the website, like colors, fonts, layout, and appearance.

- JavaScript:

It is used to make the website interactive and dynamic, like buttons, forms, sliders, and animations.

5.Imagine you need to store user profiles (name, email, profile picture) and their posts (text, timestamp). How would you structure this data in a database? Describe what tables you'd create and how they'd relate to each other.

- Firstly, Create a Users table for user details.
- Fields include user_id, name, email, and profile picture.
- Create a Posts table for post details.
- Fields include post_id, text, timestamp, and user_id.
- User_id links posts to users.
- One user can have many posts.
- **Use 2 tables:** Users and Posts.

- Users table stores: user_id, name, email, profile_picture.
- Posts table stores: post_id, user_id, text, timestamp.

CREATE TABLE Users (

User_id INT PRIMARY KEY,

Name VARCHAR(50),

Email VARCHAR(50),

Profile_picture VARCHAR(100)

);

CREATE TABLE Posts (

Post_id INT PRIMARY KEY,

User_id INT,

Text VARCHAR(255),

Timestamp DATETIME,

FOREIGN KEY (user_id) REFERENCES Users(user_id)

);

SECTION G: TECHNICAL MINDSET & PROBLEM SOLVING

1. Describe your process for learning a completely new technical skill or technology. How do you approach it, what resources do you use, and how do you know when you've understood it well enough?

- First, I read articles or watch tutorials to understand the basics.
- Then, I practice small examples or mini-projects to get hands-on experience.
- I use YouTube, online courses, forums, and documentation to learn more.
- I know I've understood it when I can solve problems on my own and explain it to someone else.

2. You're stuck on a technical problem for several hours. What do you do? Outline at least three specific strategies you would use to make progress.

- Take a short break to refresh my mind.
- Break the problem into smaller parts and try solving each part.
- Search online for solutions, tutorials, or similar examples.
- Ask friends, teachers, or forums for help.
- Try different approaches or experiments until it works.

3. How would you explain a technical concept from your domain (like a database, API, or machine learning model) to a non-technical family member?

- I use simple examples from daily life.
- Database → like a filing cabinet storing all files.
- API → like a waiter: you give the order, waiter tells the kitchen, and brings results.
- I avoid technical terms and focus on what it does and why it's useful.
- I may draw small diagrams to make it easier to understand.

4. What does “debugging” mean beyond just fixing code errors? Describe the mindset and systematic approach you would take to debug any complex system problem.

- Debugging means finding why something is not working and fixing it.
- Mindset: patient, logical, and curious, not frustrated.

Steps I follow:

- Observe the problem and identify the exact cause.
- Break the system into small parts and test each part.
- Check logs, outputs, or error messages.
- Try possible fixes and verify if it works.
- Document the solution for future reference.

5. Why is documentation important in technical work, even if you're the only one who will ever see the code or project?

- Documentation helps me remember what I did in the project.
- It makes it easier to update or fix the code later.
- Saves time when I revisit the project after weeks or months.
- Helps others understand the project if they ever work on it in the future.