

FULL STACK LAB – COMPLETE VIVA QUESTIONS WITH ANSWERS

WEEK 3 – REACT.JS

Q1. What is React?

A: A JavaScript library for building fast, component-based UIs.

Q2. What is a Component?

A: A reusable UI block; can be functional or class-based.

Q3. Functional vs Class Components?

A: Functional uses hooks; class uses lifecycle methods.

Q4. What is JSX?

A: Syntax that lets you write HTML-like code inside JavaScript.

Q5. useState Hook?

A: Adds reactive state to functional components.

Q6. Counter Program Explanation?

A: count stored in state; buttons update state using setCount().

Q7. What is useEffect?

A: Hook to run side effects like API calls.

Q8. Why dependency array []?

A: Runs the effect only once on mount.

Q9. API Fetch Program?

A: useEffect calls fetch, fetch returns JSON, stored in state using setUsers().

Q10. What are Props?

A: Read-only data passed from parent to child.

Q11. Props vs State?

A: Props = external input; State = internal data.

Q12. Iterative Rendering?

A: Using map() to generate lists dynamically.

Q13. Why key in map()?

A: Helps React efficiently update lists.

Q14. React Form Handling?

A: Inputs store values in state (controlled components).

Q15. Virtual DOM?

A: Lightweight copy of DOM enabling faster updates.

Q16. How React updates UI?

A: Uses diffing algorithm + virtual DOM.

Q17. What is Babel?

A: Converts JSX/ES6 → browser-compatible JS.

Q18. One-way data binding?

A: Data flows parent → child only.

WEEK 6 – NODE & EXPRESS

Q1. What is Node.js?

A: JavaScript runtime built on Chrome V8 engine.

Q2. Why use Node.js?

A: Asynchronous, fast, scalable server applications.

Q3. What is Express.js?

A: Minimal web framework for Node providing routing and middleware.

Q4. How to install Express?

A: npm install express

Q5. Hello World program explanation?

A: app.get("/") handles route, app.listen() starts server.

Q6. What is Middleware?

A: Functions executed before route handlers. Example: `express.json()`.

Q7. What is REST API?

A: API based on HTTP methods (GET, POST, PUT, DELETE).

Q8. CRUD Operations?

A: Create = POST, Read = GET, Update = PUT, Delete = DELETE.

Q9. Params vs Body?

A: Params in URL; body contains JSON data.

Q10. `res.send` vs `res.json`?

A: `send()` → text/HTML; `json()` → JSON response.

Q11. Routing?

A: Defining URL endpoints.

Q12. What is Nodemon?

A: Auto-restarts server on file changes.

Q13. Event-driven architecture?

A: Node runs callback functions based on events.

WEEK 7 – MYSQL

Q1. What is MySQL?

A: Relational database management system.

Q2. What is RDBMS?

A: Stores data in rows/columns with relations.

Q3. SQL vs MySQL?

A: SQL = language; MySQL = database software.

Q4. Table, Row, Column?

A: Table = collection, Row = record, Column = field.

Q5. CREATE TABLE?

A: Used to define table structure.

Q6. AUTO_INCREMENT?

A: Automatically increases unique ID.

Q7. INSERT Query?

A: Adds data to table.

Q8. SELECT Query?

A: Retrieves data.

Q9. DELETE vs DROP?

A: DELETE removes rows; DROP removes table.

Q10. Primary Key?

A: Unique, not null identifier.

Q11. VARCHAR?

A: Variable-length string.

Q12. UPDATE Query?

A: Modifies existing data.

Q13. WHERE Clause?

A: Filters records.

Q14. DISTINCT?

A: Removes duplicate values.

Q15. Normalization?

A: Organizing data to reduce redundancy.

Q16. JOINS?

A: Combine rows from multiple tables. Types: INNER, LEFT, RIGHT.

Q17. WHERE vs HAVING?

A: WHERE filters rows; HAVING filters groups.

Q18. ORDER BY?

A: Sorts results.

Q19. Constraints?

A: Rules like NOT NULL, UNIQUE, PRIMARY KEY.

COMMON VIVA

Q1. What is Full Stack?

A: Working on frontend + backend + database.

Q2. Frontend vs Backend?

A: Frontend = UI; Backend = server logic.

Q3. What is API?

A: Interface that allows communication between systems.

Q4. What is JSON?

A: Lightweight data format used in APIs.

Q5. Async Programming?

A: Executes without blocking main thread.

Q6. Callback vs Promise vs Async-Await?

A: Callback → function inside function; Promise → resolves later; Async■await → cleaner syntax.

Q7. What is NPM?

A: Node package manager.

Q8. GET vs POST?

A: GET retrieves; POST submits data.

Q9. What is CORS?

A: Prevents unauthorized cross-site requests.

Q10. MERN Stack?

A: MongoDB, Express, React, Node.