****Placement Preparation Guide – Frontend & Backend****

## ****Frontend Preparation Topics****

### ****1. HTML****

* **Semantic Tags** – Understand the role of tags like <article>, <section>, <nav>, etc.
* **Emmet Abbreviations** – Use shorthand for quick HTML coding.
* **Forms** – All input types (text, number, email, checkbox, radio, file, submit, etc.).
* **Multimedia Tags** – Handling audio, video, and iframe.

### ****2. CSS****

* **Selectors** – Class, ID, attribute, pseudo-selectors, etc.
* **Box Model** – Margin, padding, border, and content.
* **Positioning** – static, relative, absolute, fixed, sticky; understanding z-index.
* **Flexbox and Grid** – Layout design using modern CSS techniques.
* **Media Queries** – Responsive design.
* **Animations** – Using @keyframes for basic animations.

### ****3. JavaScript****

* **Basics** – let, var, const, data types, functions.
* **Arrays & Objects** – Operations and iteration techniques.
* **DOM Manipulations** –
  + Toggle HTML/text/class
  + Change attributes
  + Build a basic **Todo List**
* **ES6 Features** –
  + Spread/rest operator
  + Arrow functions
  + Callbacks
  + Map, filter, reduce
  + async/await, Promises
* **API Handling** – fetch() to call and handle APIs.

### ****4. React (Frontend Framework)****

1. **Core Concepts**

* JSX Syntax
* Functional Components
* Props & State
* Hooks: useState, useEffect, intro to useRef, useContext

1. **Routing**

* React Router: BrowserRouter, Route, Link, useNavigate

1. **Rendering & Events**

* Conditional Rendering
* List Rendering with map() and key prop
* Handling Events: onClick, onChange
* Controlled Forms and Validation

1. **API Integration**

* Fetch data using fetch() or axios in useEffect

1. **Mini Projects**

* Counter App
* Todo List App
* Weather App using external API
* Form with validation

## ****Backend Preparation Topics****

### ****1. Core DSA Topics****

* **Numerical Problems** – Factorials, primes, neon numbers, etc.
* **Array & String** – Traversals, pattern matching, transformations.
* **Recursion** – Fundamental recursion and variations.
* **Subarray/Subsequence** – Maximum subarray, longest subsequence problems.
* **Divide and Conquer** – Merge sort, quicksort, binary search variants.
* **Backtracking** – N-Queens, sudoku solver, combinations/permutations.
* **HashMap** – Frequency count, groupings, optimizations.
* **Sorting Algorithms** – Insertion, selection, merge, quick, etc.
* **Binary Search** – On sorted arrays and answer space.

### ****2. Data Structures****

* **Linked List** – Reversal, merge, detect cycle.
* **Stack & Queue** – Valid parentheses, min-stack, circular queue.
* **Tree & BST** – Traversals, height, diameter, insertion.
* **DFS & BFS** – Graph traversals on matrix/adjacency list.

### ****3. Advanced Concepts****

* **Sliding Window** – Fixed and variable window patterns.
* **Greedy Algorithms** – Activity selection, job sequencing.
* **Dynamic Programming** – Memoization, tabulation; problems like knapsack, LIS, LCS.
* **Java Collections** – Usage of Collections.sort(), ArrayList, HashMap, etc.

### ****4. MySQL (Database Basics)****

1. **Core Topics**

* **Database Basics**
  + What is a Database, Table, Primary Key, Foreign Key
* **Data Definition Language (DDL)**
  + CREATE, ALTER, DROP statements
* **Data Manipulation Language (DML)**
  + INSERT, UPDATE, DELETE, SELECT

1. **Important SQL Queries**

* SELECT with WHERE, AND, OR, BETWEEN, IN, LIKE
* Sorting with ORDER BY
* Filtering with LIMIT, DISTINCT
* Aggregate Functions: COUNT, SUM, AVG, MIN, MAX
* GROUP BY and HAVING
* INNER JOIN, LEFT JOIN, RIGHT JOIN
* Subqueries (Nested Queries)
* Basic UNION and CASE usage

1. **Practice Tasks**

* Write queries for:
  + Employee management (filter by salary, dept, etc.)
  + Student attendance and gradebook
  + Sales/product analytics (top products, total revenue)
* Normalize a table and explain 1NF, 2NF, 3NF
* Simple ER Diagram explanation (optional for backend roles)