* **Design Patterns**
* **Workflow models and types**
* **Virtual DOM**
* **MySQL Injection**
* **Data Sanitization**
* **Scrum Methodology**
* **GitHub commands**
* **JS Hoisting (which data types are hoisted or which are not)**
* **Temporal dead zone (TDZ)**
* **Callback Function & Callback Hell**
* **Closure Function**
* **Lexical scope**
* **Higher Order Functions**
* **Higher Order Components**
* **Anonymous Functions**
* **Debouncing**
* **Thread Pooling**
* **JS Array Methods (Array.from(), Array.of(), push, pop, shift, unshift, splice, foreach, map, filter, find, findIndex, reduce, reduceRight, every, some, includes, indexOf, lastIndexOf, concat, slice, join, flat, flatMap, reverse, sort, length, isArray)**
* **Difference between map & foreach and what they return**
* **Difference between reduce & reduceRight**
* **How to solve Callback Hell,?**
* **Difference between Promise & async/await**
* **Is JS single thread or multi thread,?**
* **Is JS's behavior sync or async,?**
* **Promises (resolve, reject)**
* **Async/await**
* **.then, .catch, .finally  
    
    
    
  React JS**
* **How virtual DOM works,?**
* **State vs Props (proper definition prhi)**
* **Data Binding**
* **Unidirectional vs Bidirectional Data Binding (with examples)**
* **How to do bidirectional data binding,?**
* **React Hooks (useState, useEffect, useMemo, useCallback, useLayoutEffect, useContext)**
* **React workflows (Flux, Redux)**
* **How redux works,?**
* **Difference between localStorage & sessionStorage**
* **Web API's (setTimeout, XMLHttpRequest etc)**
* **Call Stack**
* **Callback Queue**
* **Event Loop**
* **Core concepts of OOP**
* **Object**
* **Class**
* **Interface (must)**
* **Encapsulation**
* **Abstraction**
* **Polymorphism**
* **Inheritance**
* **Some of SQL questions:**
* **Data Normalisation (1nf, 2nf etc)**
* **Joins**
* **Types of Joins**
* **Alias**
* **React Lifecycle (must)**
* **Class based components Lifecycle**
* **Function based components Lifecycle (it manages through useEffect)**