* Introduction
  + Why did you leave your last job
* Project Portfolio
  + Have you worked every directly with Client? Have been in Customer facing role
* OOP
  + Write down the basic concepts of OOP
    - Abstraction
    - Encapsulation
    - Inheritance
    - Polymorphism
    - Manipulator
  + Inline functions, virtual function, friend class, friend function,
  + Overloading and overriding
  + Finalize method
  + Interface and traits
  + What is early and late binding
  + SOLID Principles
  + How would you model and represent a Reservation being made by a Passenger for a Flight in object-oriented code?
  + Define Modularity
    - Persistence
    - Collaboration
* Data Structure and Algorithm
* Software Development Life Cycle
  + What do you think about pair programming? Agile Process
  + What is DRY?
  + Name the list of techniques for specifying the requirements
  + What is requirement tracing? / Backward Tracing and Forward Tracing
  + Which tools do you like to use for keeping track of requirements?
  + How do you manage conflicts in a web application when different people are editing the same data
  + What kinds of diagrams have you used in designing parts of an architecture, or a technical design?
* CSS (LESS, SASS, SCSS)
  + What is CSS Combinators?
* JAVASCRIPT and (Frameworks)
  + How do you Organize you JavaScript Code
  + 2 programing paradigm important for JS app Development
    - Prototypal
    - Functional
  + JSON, XML and YML
  + Synchronous and Asynchronous Programming
  + Node Package Manager
  + ANGULARJS and (Related Questions)
  + VUEJS and (Related Questions)
  + REACTJS and (Related Questions)
* PHP and (Frameworks)
  + PSR Standard
  + Composer Package Manager
  + LARAVEL & LUMEN and (Related Questions)
  + CODEIGNITER
* SEO if (required)
  + How to choose topic and keywords
  + What is sitemap
  + robot.txt
  + SEO and SMO
  + What is crawler and crawler algorithm
  + Back linking
  + outbound link
  + Differentiate between on-page and off-page
  + What is Page Rank
* RESTful Service
  + Stands for
  + What is Request method, how many types are in request method
  + What is Resource in REST API
  + How many types of Resources are there in REST API
  + What is Statelessness in RESTful web services
  + What are HTTP Status Codes
    - 200
    - 201 It means, CREATED, when a resource is successful created using POST or PUT request. Return link to newly created resource using location header.
    - 204
    - 304
    - 400
    - 401
    - 404
    - 409
    - 500
* MySQL
  + Selection and Projection
  + Types of SQL Languages
  + Select Case statement
  + JOINs (Right Join, Outer Join, Full Join, Cross Join, Self-Join), UNIONs, UNIONs All, MINUS, Intersect, Cartesian Products
  + DROP and TRUNCATE
  + Normalizations
  + Splitting
  + Insert Ignore
  + Insert Update
  + triggers
  + Procedures
  + Routines
  + Views
  + What is SQL Injection?
  + What are aggregative functions?
  + Scalar Functions
  + What is NonSQL Databases
  + Which port MySQL uses in apache Server
* Code Versioning
  + What is PR?
* Server Management
* Web Development
  + What is CORS, How does it works?
  + What are the difference between stateless and statefull protocols
  + Which type of protocols HTTP is in Stateless and statefull
  + What is Request method, how many types are in request method
  + Social Media API usage, payment Integration, Real Time Integration
* Communication Skills

**React JS interview questions**

* What tools can you use in the build steps to optimize the compiled output of React.js code?
* How do you create Higher Order Components in React.js? What are the limitations of HOCs?
* What are some ways you may improve your website's scrolling performance?
* What are the most common security threats on the frontend and how to prevent them?
* What do you need to consider if you had to support IE6?
* What is base64? What is the common use-case? What is good and what is bad about base64? (performance, security, ease of use, …)
* What are the differences between Long-Polling, Websockets and Server-Sent Events?

**Difference between micro services and monolithic architecture**

* What are microservices authentication and authorization technical solutions
  + Distributed session management
    - Sticky session
    - Session replication
    - Centralized session storage
  + Client tokens
    - Header (JWT)
    - Payload based (necessary info)
    - Signature (private/public key)
    - SAML (Security Assertion Markup Language)
  + Single Sign on (SSO)
  + Client token with API gateways
  + OAuth and OAuth2