

## PREPERATION NOTE FOR TOI & Job Interview (MERN)

*Prepared by Vahab*

*Please follow me on LinkedIn  
and Github...*

[Github](#)      [LinkedIn](#)

### Review Manifest key points

=====

- synchronous and asynchronous function
- callback and promise details
- temporal dead zone
- object Destructuring
- cors
- currying
- core modules in Node.js
- events module
- session Timeout
- view engine purpose
- express.js
- types of Middleware
- closure
- promise methods
- higher order function
- Default parameters
- Ternary operator
- Null()
- JIT
- Error first callback
- BodyPreser
- Fs module
- Capped Collection
- Distinct function
- Add to set
- All in operator
- Mongo Utility commands
- Aggregation
- First Class function
- Normalization
- Denormalization
- Trigger
- Usage of sql
- DDM.DDL
- Constraints
- Acid Properties
- Entities & Relationships
- Trunket,Delete
- Higher order Components
- Stateless function
- State Management
- Lifecycle of React
- Difference between context and Redux
- Object concept
- Object reference
- Node basic features
- ES6
- Middleware, next()
- Advantages of express
- React functional Components
- Class Components
- pure components
- Data types in JavaScript
- Event emitters
- Event delegation
- Call apply bind
- Reduce
- Streams
- Helmet
- Rest & Restful Api
- URI
- Options in Http
- http status Code
- Features of Mogo Db
- Bson
- Backup and Restore
- Transactions
- Child to parent data transfer
- Why we use pure component
- Git fetch vs git pull
- Git revert
- Git Stash
- Nginx
- Types of Load Balancing
- Proxy vs Revese Proxy
- DNS
- CDN
- Callback hell
- Promise, Promise. All(), Promise.Raise(), Promise.any()
- Promise.finally
- Hoisting
- Declaration and initialization Difference in js
- Async await
- Array methods
- Memoizaiton
- Spread operator
- Difference Let var const
- Prototype
- Reg expression
- Json
- Evl evm
- How to use settime out
- Set Intervel
- indexOf()
- deep cloning
- Thus
- Slice Splice

- Browser object method
- Shift Unshift
- Immediate invoke function
- 3 different state in promise
- Template literals
- Setdate type
- Rest parameter
- Unary function
- Local scope
- Recursion
- IIF syntax
- Session Handling
- isNaN()
- REPL
- Node.js Working
- Process
- Process.nextTick()
- setImmediate
- JWT (full working)
- DOM & BOM
- Session Cookie
- HTTP,HTTPS
- Req,res
- Res.download
- Version control
- Types of argument passed in app.use
- Register in app.use
- Node.js architecture
- NPM,API,NVM,NPX
- Sort
- TCP & UDP
- Thread Pool
- UUID
- Adhoc Queries
- TextIndex
- Socket
- Cluster
- JRE
- DOM events
- Webstorage
- First order function
- Unary function
- Cache Control no store
- Package.json
- Global objects
- Type conversion vs Type coercion
- Dependency vs DevDependency
- ReactRouter
- React HOOKs
- Array methods
-

## Asked Questions In TOI

### Communication Round

- Self intro
- Leadership and coordination
- What type of company do you want to work
- Explain first and Second project
- Team lead bad situation how to survive
- Salary expectation
- Struggle you faced
- Toughest decision
- How you schedule time
- How you handle the pressure

1. How do you handle conflicts with coworkers ?

2. What are your weaknesses?

3. Tell me about a time when you had to make a difficult decision.

4. How do you handle feedback - positive or negative ?

5. How will you negotiate about the salary during the interview even if you don't have any experience?

6. Why do you want to work for our company ?

7. What do you know about our company's product/services?

8. What, according to you, is the ideal workplace or company ?

9. How quickly do you adapt to new technology?

10. What motivates you to do your job ?

## Technical Round

### Riyas

- TLS
- reverse proxy
- forward proxy
- WAF
- Ufw
- UDP and TCP
- Load Balancing
- unary opt
- bitwise operator
- Literals
- maxAge vs Expires
- implements and extends
- set collection
- type casting
- Promise.all
- Transaction
- \$unset
- horizontal scaling
- Geospatial index
- GridFS
- CAP Theorem
- React.Fragment
- React Motion
- Thunk
- Saga
- Webpack code splitting
- Quadratic Probing
- load factor
- resize a hash table
- stack under flow
- stack in recursive function
- balanced tree
- Hierarchical Linked List
- heap in manage priority queues
- DFS Complexity
- BFS Complexity
- DFS and BFS
- graph indexing
- Spanning tree
- represent a graph in memory
- shortest distance b/w two nodes
- find the minimum spanning tree in a graph"
- implements and extends
- Docker
- CDN
- DDos
- Micro services
- set collection
- factory function
- type casting and type coition
- Proxy object
- \$unset
- chunks
- Replication
- Render Props
- Thunk
- Generator function
- Saga
- React.PureComponent
- load factor
- resize a hash table
- AVL tree
- suffix tree
- represent a graph in memory
- detect cycles in a graph
- find the minimum spanning tree in a graph
- shortest distance b/w two nodes
- Quick sort
- Why you chose AWS
- Use of nginx
- How nginx masking ip address
- Ip hiding
- Certbot
- Firewall
- Microservice how to create
- API gateway
- Socket.io working of event
- How to pass response from client to server
- Conditional operator true false and ternorey operator
- Bind operator
- Escape seq
- Hoisting
- Optional chaining
- Nullish
- Heigher order function pure function
- Implimets
- Extends
- Implicit Typecasting and Explicit typecasting
- Tocontinue ToStringToFixed
- parseFloat method
- EventEmitters
- Res.render()

- Statically typed vs dynamically typed language
- Throw
- Transaction
- Replication
- How to create Replication
- Promise.all()
- \$undwind,\$project
- Capped true while creating collection
- Sharding horizontal Scaling
- Chunkz
- View createView
- Injection
- State and props in class
- How to change state in classname
- Babel
- Webpack
- Diffing Algorithm
- Profiler
- React.fragment
- useNavigate()
- useHistory()
- reduxStore
- reducer
- reduxToolkit
- useSelector
- dynamic Rendering
- client side rendering
- lazy load
- datastructure double hashing
- recursion in stack
- garbage collector js
- stack Underflow
- perfect balanced
- Trie complexity

- Insertion Sort

### **Akshaya**

- Rest API
- http vs https
- protected routes
- cors
- SLDC models
- Example for higher order function
- Prams vs Query
- Streams in node.js
- Cap theorem
- CSR vs SSR
- Shadow Dom
- Applications of doubly circular linked list
- Applications of heap
- 

### **Shahbas**

- currying
- How to optimize the performance of a react js application
- How we can implement memorization
- Unit Testing
- Working of server side
- All operator in operator
- Sharding
- Replication and replication set
- Single page application
- Fragmentation
- State and props
- Pure component
- use Memo

- Data structure Explanation and classification
- Memory leakage
- Complexity analysis
- Tools of complexity analysis
- Linked list types and explanations Differences
- Pre order in order post Order traversal
- Difference of a node
- Hash table application of hash table
- Traversal of graph (BFS and DFS)
- Binary Search Tree (practical)
- Binary search (array inside many object)
- Status codes
- Post put Difference
- Pagination explanation
- Skip and limit (mongo aggregation)
- Dotenv explanation
- MVC
- Module, import module
- Middle ware
- Exception handling
- Finally usage
- Queue
- Hosting AWS explanation
- Define javascript and its overall working
- Hoisting
- Micro task and callback
- Promise
- Template literals

- App.use
- App.set
- Difference b\w use and set
- Cors
- Lookup
- Transaction
- Difference b\w Authorization and authentication
- JWT works Explanation
- How JWT works and Why JWT
- Difference b\w JWT and session
- http and https explain
- Rest API
- What are Http methods
- 
- Streams,
- Event looping,
- Why react is popular,
- Props, context,
- Complexity of stack and queue,
- Array operations,
- -Duplicate elements from an array

“ Duaa mein yaad Rakhna”- Vahab

### **Vishnu Sivan**

- Mern vs mean,
- How to test a project,
- Npm run audit,
- Disadvantages of JavaScript,
- Learn more about git,
- Event delegation,
- Indexing,
- Transactions,
- CAP Theorem,
- Disadvantages of an array,
- Applications of heap and BST,
- BST validation
- Steps for hosting the site,
- Pm2,
- Replica set,
- Sharding,