Internship Process

**Eligibility**: HTML, CSS, Javascript || Any object oriented/imperative language.

**-------------------------------------------------------------------------------------------------------------------------------**

* Every phase has a entry level task/test to get into that.
* Any one can attempt any phase entry level test/task at any point of time.
* In case failure in any phase , they have to repeat it.
* Things might be vary based different conditions/situations.
* Phase- 2 or 3 or both are important to get into team.

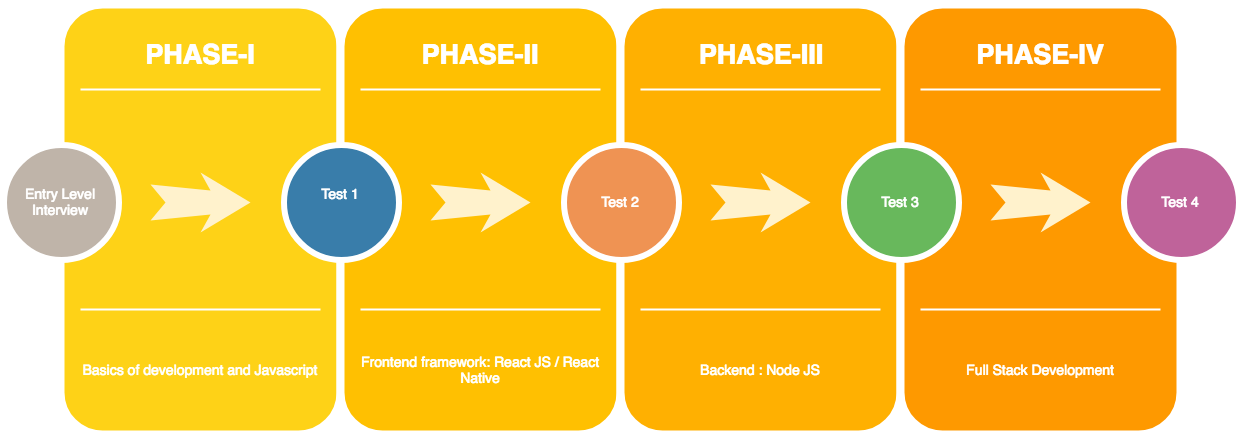
**-------------------------------------------------------------------------------------------------------------------------------**

**Tech Stack/Tools** :

* HTML/CSS
* Javascript
* React (Redux)
* Git (Github/Gitlab/Bitbucket)
* Node Js (Express JS)
* Postman & JIRA/ Trello
* Dynamodb
* Postgresql
* Docker
* Kafka (Confluent)
* Microservice
* AWS (Lambda, ECS, ES, Route 53)
* **Heroku**
* **Mongodb**
* **MySQL**
* **React-Native**

**\*** Bold ones are secondary priority.

**Phases**:



* Phase-1 (Basics of development and Javascript):
  + HTML, CSS, JavaScript, Bootstrap
  + OOP In JavaScript & Fundamental problem solving approaches using Javascript
  + Javascript Web apps frontend & Basic JS Frameworks
  + ES5,ES6,ES7
* Phase-2 (Frontend framework: React JS):
* Intro to React
* React / React Native with redux
* JSX and React Components
* State, Life Cycle, and Events
* Lists and Forms
* Ajax Requests with React
* Modeling Application State
* Managing App State with Redux
* Intermediate Redux: Middleware
* Antd / material UI
* Phase-3 (Backend : Node JS):
* NodeJS and Express JS Intro
* Node.js is used in modern web development
* Express framework to set up a web server
* Building REST API
* MongoDB , Mongoose intro
* Writing CURDs
* MVC Pattern / use cases and practice
* Events & Streams & Modules
  + Databases Intro ( SQL and Nosql)
  + REST API
* Phase-4 (Full Stack Development):
  + MicroServices Architecture vs Monolithic Style
  + Different Design Patterns
  + Application Development Phase -1
  + Application Development Phase -2
  + Application Development Phase -3
* Phase- 5 (Tech infrastructure):
  + Infrastructure
  + AWS
  + Docker ….etc

Tasks:

1. Task 1 - **Create react app manually**.
2. Task 2 - **Converting a simple Blog into react components**
3. Task 3 - **Timer App**
4. Task 4 - **Sales cart in Single Page Application**
5. Task 5 - **Usurvery with Firebase database**
6. Task 6 - **Firebase Login Setup**
7. Task 7 - **Google Sign in with Firebase**
8. Task 8 - **Reddit API with axios**
9. Task 9 - **Auth0 login System**
10. Task 10 - **Github API with Fetch and Auth0 login system**

References:

* Codecademy
* Udemy --installing react video
* Counter App (state and props)
* Udemy -- CountDown Champ (js time , state and props)
* React Docs(Handling Events to Composition vs inheritance)
* Thinking in React (covers events, json, conditional rendering)
* Timer (React practise)

**List of Technologies:-** Reactjs , Redux ,Ant Design,Git, Node JS, Dynamodb,Postman,Nodejs,Express server, React-Router-Dom.

**List out the sources with reference URL**

* Udemy: <https://www.udemy.com/course-dashboard-redirect/?course_id=1042110>
* Code academy: <https://www.codecademy.com/learn/react-101>
* React Docs: <https://reactjs.org/docs/hello-world.html>
* Ant Design: <https://ant.design/docs/react/introduce>
* <https://www.udemy.com/node-with-react-fullstack-web-development/learn/v4/overview>
* <https://www.udemy.com/ultimate-web/learn/v4/overview>
* <https://www.udemy.com/react-js-and-redux-mastering-web-apps/learn/v4/overview>
* <https://www.udemy.com/react-js-mastering-redux/learn/v4/overview>
* <https://www.techiediaries.com/react-router-dom-v4/>
* <https://reacttraining.com/react-router/web/guides/philosophy>
* <https://aws.amazon.com/dynamodb/getting-started/>
* <https://www.getpostman.com/docs/postman/api_documentation/viewing_documentation>
* <https://ant.design/>
* <https://www.djamware.com/post/58b27ce080aca72c54645983/how-to-create-nodejs-expressjs-and-mongodb-crud-web-application>
* <https://medium.freecodecamp.org/building-a-simple-node-js-api-in-under-30-minutes-a07ea9e390d2>
* <https://zellwk.com/blog/crud-express-mongodb/>
* <https://www.callicoder.com/node-js-express-mongodb-restful-crud-api-tutorial/>
* <https://codeburst.io/writing-a-crud-app-with-node-js-and-mongodb-e0827cbbdafb>
* <https://medium.freecodecamp.org/building-a-simple-node-js-api-in-under-30-minutes-a07ea9e390d2>