**Client**

Client in this project will be created with NEXTJS. It will be going to have an feature of Server Side Rendering (SSR).

Server Side Rendering is very helpfull for Search Engine Optimization (SEO).

**NextJS**

Next.js is a React framework that **gives you building blocks to create web applications**. By framework, we mean Next.js handles the tooling and configuration needed for React, and provides additional structure, features, and optimizations for your application

Difference Between **NodeJS** vs **NextJS** ?

Node. js is ideal for fast and scalable server-side and networking applications, while Next. js provides a robust framework for building high-performing React-based web applications with server-side rendering. Ultimately, the best option for your next web project will depend on your specific needs and requirements.

**Phase 1**

Creating Homepage , Signin , Signup, Signout PAGES

Phase 2

**Phase 1**

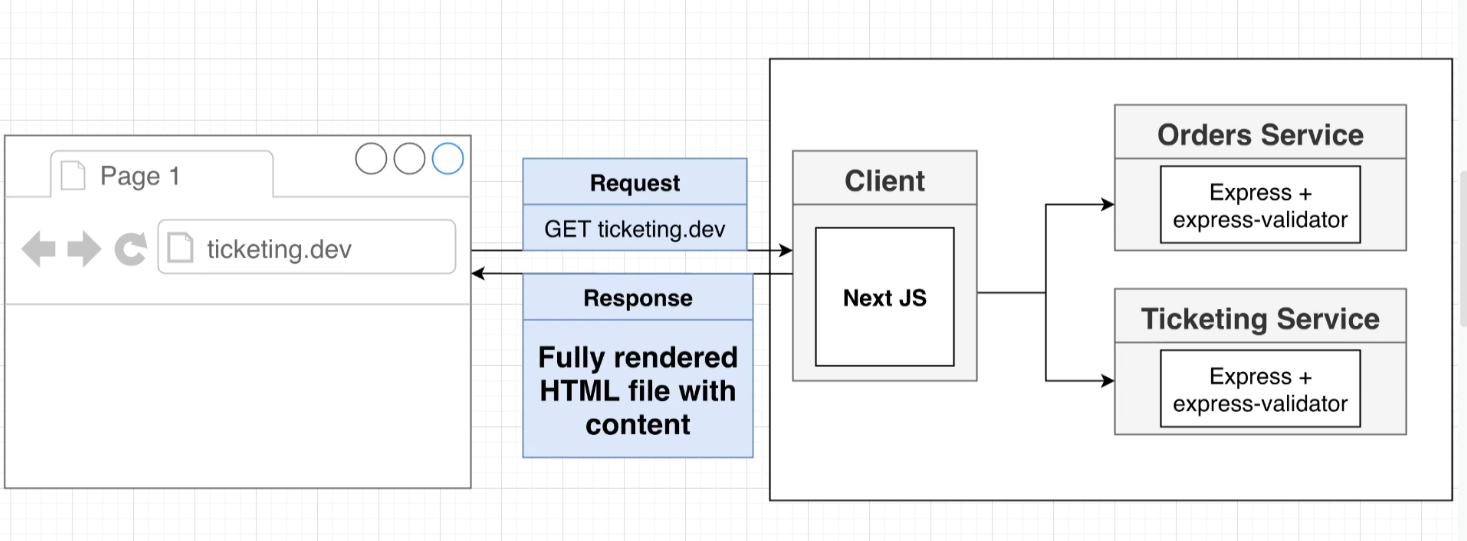
Phase 1 main focus will be on creating auth pages and redirecting.

1. Signin
2. Signout
3. Signup

**Libraries Required**

1. react
2. react-dom
3. next

**Architecture Design with respect to kubernetes**



**Basics of NextJS**

**Pages** in NextJs

To define pages routes in NextJS we create pages folder and create the same route pattern with help of folders and then create JSX file to reflect the code.

for eg --> [www.ticketing.dev/auth/signin](http://www.ticketing.dev/auth/signin)

for this path we need **folder auth** and in this we need **signin file jsx** code all under Pages

**pages**->**auth**->**signin**

Advantage of pages system

NextJS will provide extra functionality with pages.

To access the features you can capture in argument

**Problems while using client in cluster environment ?**

1. Client while calling an api service from server side which is located in same cluster but in different pod?

Two ways to solve

1. **Not Recomended** --> use service name (clusterIP) of the pod while calling from other pod

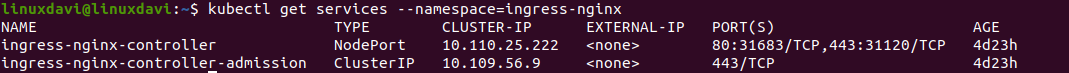
eg => **<http://auth-srv/api/users/signin>**

it is not recomended because any time we are going to change service name this api call will give error and it is not possible to change every api request name located in project.

1. **Recommended** --> use ingress-nginx namespace for calling api calls within cluster

Check name of services in **namespace ingress-nginx**

=> **kubectl** get services **--namespace**=ingress-nginx



Here **ingress-nginx-controller** can be used as service name

so request will be like

[http://](http://service_name.namespace.svc.cluster.local)**[service\_name.cluster\_namespace](http://service_name.namespace.svc.cluster.local)**[.svc.cluster.local](http://service_name.namespace.svc.cluster.local)

**baseUrl =>** [http://](http://ingress-nginx-controller.ingress-nginx.svc.cluster.local)**[ingress-nginx-controller](http://ingress-nginx-controller.ingress-nginx.svc.cluster.local)**[.](http://ingress-nginx-controller.ingress-nginx.svc.cluster.local)**[ingress-nginx](http://ingress-nginx-controller.ingress-nginx.svc.cluster.local)**[.svc.cluster.local](http://ingress-nginx-controller.ingress-nginx.svc.cluster.local)

**Routing in NextJs ?**

routing is very simple in nextjs. just import Router from ‘next/router’ and then use Router to push url at the top of the stack to reflect the page in realtime.

**Example**

import **Router** from ‘**next/router**’

Router.push(‘/auth/signin’)

**Tree Structure of Herarichy**

**\_app** is the top member of herarchy we can use Component argument to render all pages from parent \_app.

**Important** **=>** **\_app** is responsible for running all pages defined **getInitialProps function** and **return the context** from parent to all other components with **pageProps** argument from common Component.

**( ctx )** contains all

**'err'**,

‘req',

'res',

'pathname',

'query',

'asPath',

'locale',

'locales',

'defaultLocale',

'AppTree',

'defaultGetInitialProps'

in this **req** feild contains all domain , host , cookie and all other stuff which is very important when we are calling from server and request need to be forward with exact same configration.

**\_app.js** => its getInitalProps function contains argument with (**AppTree**,**Component**,**router**,**ctx**)

**index.js** => its getInitialProps function contains arguments with (**ctx**)

**pages** => its getInitialProps function contains arguments with (**ctx**)

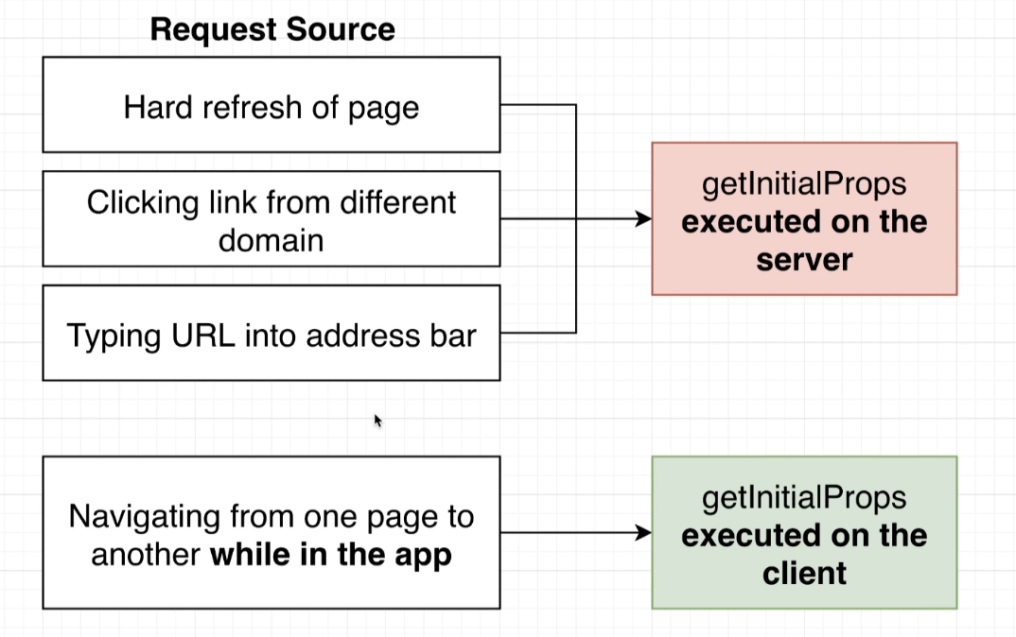
**What is getInitialProps function ?**

**getInitialProps** is special function available in NextJS .

Used for server side processing.

It is fast as compare to other client based browser languages.

Always remember api calls originated via server are handled differently than the browser side api calls

**getInitialProps** function work under three condition and two environment