## **Available Features**:

In a single page application a whole page is split into smaller components, each component may contain other components. Each component’s presentation logic and required data remains within that component class.

## Angular:

Angular is a complete MVC framework for front-end development with lots of features. In angular we can create components. Angular has dependency injection and service class concept so all business logic and backend server communication happens in various service classes as a result components class mainly focus on presentation and dom manipulation logic . So separation of concern is well defined. Angular has built in routing mechanism which is very important for web application. Besides so to support clean code and separation of concern it has “Directive” and “Pipe” which are basically to change the appearance or behavior of a DOM element. Followings are some key features of angular:

* Complete MVC or MVP framework
* Dependency injection
* Templates, based on an extended version of HTML
* Routing, provided by @angular/router
* Ajax requests using @angular/common/http
* @angular/forms for building forms
* Component CSS encapsulation
* XSS protection
* Utilities for unit-testing components.

## React:

React is on the other hand is not a complete framework, it only a component library. As a component library it does it job very well. But for a single page web application we need a few more things: Routing mechanism, Data flow and beck-end server communication etc. Since it is not a complete framework to support it lackings various community driven tools are created which form a react based ecosystem. For example, to support routing there is a well maintained library “React-router”. We can use flex architecture pattern proposed by facebook or redux or mobx for data and event flow. Following are some key features of react:

* Instead of classic templates, it has JSX, an XML-like language built on top of JavaScript
* State management using *setState* and the Context API.
* No Routing mechanism(need 3rd party library for that)
* XSS protection
* Utilities for unit-testing components.

## Conclusion:

In feature wise comparison Angular is clear winner without any doubt. But with react all the missing feature is available through well supported 3rd party libraries and there are multiple options to choose from. So feature will not be a big concern.

# Learning & Adoption

## Angular:

Since angular is most feature rich so the learning curve is very steep will introduce more new concepts than React. To work with angular need to learn the following things:

* Typescript
* Ng Module
* Dependency Injection
* Decorator
* Components
* Directives
* Pipes
* Templates
* Rx.js(rxjs itself is a big library to learn but extremely powerful tools.)
* AoT
* Zone and angular change detection
* Jasmin and Karma for unit testing

## React:

React is pretty simple to learn and onboard. Any developer will be able to catch up quickly. There are only few new concepts:

* JSX
* Component and its lifecycle
* State Management
* React hook
* React context API
* Jest and Enzyme for unit testing

Some more things need to learn to build complete single page application with React:

* React Router v4
* Redux/MobX(not mandatory but knowledge will help a lot)
* Webpack(not mandatory but knowledge will help a lot)

## Conclusion

Clearly react is winner. One of the best things about react is a developer needless to learn whole react based ecosystem to onboard. Only basic knowledge of react can help him/her to contribute to our team. But things are not easy with angular. Only basic knowledge of angular is big and needs time to learn and actually basic knowledge will not be enough.

# Migration to single page application

## Angular:

Angular is a complete framework, to migrate from our existing multi-page server-side rendered application will not easy due to this reason. We have to create and maintain a angular application along with our current application and implement a reasonable amount our old features to angular first then we can remove existing one.

## React:

React is component library. It can actually be used as js templating library to which data will be served in JSON format. React can be used in our existing application to render a part of the page, for example we can use react to render our sidebar. As result of this migration is easy. We can introduce react in our existing multi-page application, create private api to serve JSON. Then also we have to create a single page app using react but there we can re-use those react components as well as those private apis which we created inside our existing application.

## Conclusion:

Clearly react is winner.