# Javascript

|  |  |
| --- | --- |
| FE Development | Back end Development |
| * HTML, CSS, JS, Bootstrap * UI is based on the pages. Each and everything will be different page, each page need to take the response from * Server load is high * [www.google.com/login/](http://www.google.com/login/) register * Multi Page Application * Single Page Application   JavaScript Framework   * Routing * Data transfer | * Node JS   **https://www.moesif.com/blog/api-product-management/api-analytics/Top-5-NodeJs-REST-API-Frameworks/** |
| |  |  | | --- | --- | | Angualr | React | | * SPA * Data Binding (Two way binding) | * Custom Components * Own library/ own function | | * Express JS Framework * **Nest Js** |

SPA:

* Routing
* Data binding

<https://www.lambdatest.com/blog/best-javascript-frameworks/>

|  |  |
| --- | --- |
| Angular - google | React |
| Framework -> contains all required components in your application   * All library * Typescript | React is a library   * What need we choose * Single way Data Binding |
| DOM Manipulation | VDOM Manipulation |
| Component Based Architeyure  .html, css, ts  Code separation  Home | Componnet   * May contains everything * File (JS, design, css)   Home -> nav, side, |
| Design a page using HTML  Angular Directives  @Module  @ngif  @ngfor  Html  <div \*ngFor> | Design a Page  Javascript for all   * Html * Css * Logic   Javascript  JSX -> javascript extenasible  bable |

<https://www.geeksforgeeks.org/angular-vs-reactjs/>

<https://legacy.reactjs.org/docs/getting-started.html>

<https://react.dev/>

Google.com -> angular -> process -> send to client -> DOM render

React -> server send response to VDOM -> send the response to DOM

React:

* It a library
  + React library
    - Used to create a react componnet
  + React Dom library
    - Used to render the Componnets

Design a page using the elements

* In react , we just create a custom element (componnets)
  + React.createElemenst(
  + )
* Those elements must render in a single page

ASP.NET core:’

Login Components

* Tet, text pass,

React Application:

* Js
* JSX (babel) -> define a js function , must return the html output

JS Function -> values

JS Functions-> html elements -> babel library

Create react application:

* npx - node package runner , runs the library once without instating in local sysystem / application
* npm -> install the package in application or s/m

Nest Js:

Npm I @nest

React application

Package Create-react-app

React application we are developing using the framework

* next Js

React 18 Version:

Create a react Project using npx

1. npx create-react-app app-name

Way 2 create react project based vite framework

npm install –save-dev web-vitals

Create vite Project

React

* Component Based architecture (Page)
  + Define the components
  + Core building Block

Different way is available to create components

* Class based Approach
  + Create a class in js
  + Extends the React.componets
  + Then create render methods, retuns the value
* Functional Approach
  + Define a function , function retuns UI

Design a Components:

* React Bootstrap ()
* Material UI

Custom componnets,

* Navbar

Single Project-> reat bootstap / material UI

How Components can maintain data

* Passing data between components
* Manage the data in a components

1. React Components
2. States and props
   1. States is property used to manage the data in a componnts
   2. Props are properyy to transfer data from one components to another