**React** **.JS**

In 2013

React.js was developed by Facebook to address its need for a dynamic and high performing User Interface(UI). React JS is a JavaScript **library** which brought together the **speed of JavaScript** and a **new way of rendering pages**, leading to a **responsive and dynamic user input**. In 2015, two years after the team open sourced React.js and its popularity grew, they released React Native.

React.js often referred to as React or ReactJS is a JavaScript library responsible for building a hierarchy of UI components or in other words, responsible for the rendering of UI components. It provides support for both frontend and server-side.

*Remember, React.js is just the V part of the* ***MVC framework****, a library meant for rendering your views.*

.**React’s virtual DOM is faster** than the conventional full refresh model, **since the virtual DOM refreshes only parts of the page**.

.You can reuse code components in React JS, saving you a lot of time.

* It improves the debugging speed making your developer’s life easier.
* Even to those unfamiliar with React, it is easily readable.Many frameworks require you to learn an extensive list of concepts which are only useful within the framework. React strives to do the opposite.

**Difference between library and framework**

The key difference between a library and a framework is **"Inversion of Control"**. When you call a method from a library, you are in control. But with a framework, the control is inverted: the framework calls you.

A library is a module that you call from your code, and **a framework is a module which calls your code**. A framework can be made out of different libraries.

A software framework is a re-usable design for a software system (or subsystem). A software framework may include support programs, code libraries, a scripting language, or other software to help develop and glue together the different components of a software project. Various parts of the framework may be exposed through an API..

So I'd say a library is just that, "a library". It is a collection of objects/functions/methods (depending on your language) and your application "links" against it and thus can use the objects/functions/methods. It is basically a file containing re-usable code that can usually be shared among multiple applications (you don't have to write the same code over and over again).

A framework can be everything you use in application development. It can be a library, a collection of many libraries, a collection of scripts, or any piece of software you need to create your application. Framework is just a very vague term.

Strictly speaking, jQueryis a library, but, to an extent, it does meet the definition of a software framework. Although many would argue that jQuery doesn't meet the definition of a software framework strictly enough, the fact is that no other JavaScript framework fully meets the definition of a framework either.

Introduced in 2006, jQuery is **one of the earliest frontend frameworks**. ... What's more, the recent development in this framework enables developers to build native mobile applications with its HTML5-based UI system–jQuery Mobile. Besides, jQuery is browser-friendly and supports any browser you intend to use.

Bootstrap is **a free and open-source CSS framework directed** at responsive, mobile-first front-end web development

**React Native**

React Native is a framework for building native applications using JavaScript. React Native compiles to native app components, which makes it possible for you to build native mobile applications. In React JS, React is the base abstraction of React DOM for the web platform, while with React Native, React is still the base abstraction but of React Native. So the syntax and workflow remain similar, but the components are different.

**React hooks**

-Allow you to create every component as a functional component

-Sharing logic between components becomes easies

Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class.

Hooks are functions that let you “hook into” React state and lifecycle features from function components. Hooks don’t work inside classes — they let you use React without classes.

**IT Kamasutra**

**3.lesson**-Node.js install nodejs.org

(Node.js eto platforma, kotoraya kak I brauzer ponimaet js. Eto tak nazyvaemaya chernaya consolka js, kotoraya imeet programmy vspomogatelnie/moduli, kotorye pomogayut nam pisat code. Npm eto paketnyi menedjer etix modulei pomogaet ustanavlivat, npx pomogaet ustanavlivat i zapuskat’ ix.

**4.lesson**

Command prompt-um gnam ayn texy vortex stexcel enq papka or studyReact

**npx create-react-app my app(stex grel anuny mer cankacac)**

**cd my app**

**npm start**

**5.lesson spa(single page application)**

Obnavlyaetsya ne obshaya stranica html, a chast, I s server prixodit ne html a js s pomoshyo ayax zaprosa, v json formate prixodit otvet I js otrisovivaet. Server ne perezagrujaetsya,skorost’ uvelichicaetsya, pri clicke odrugogo url-a. prixodit vsegda odna stranica html pustaya pochti, v kotorom menyaetsya content js otrisovivaet content. Server zdes’ uje rabotaet kak api. V mob prilojeniyx voobshe html ne nujen, a tol’ko dannye I tam drugie instrumenty dlya otobrajeniya dannyx. V pervom sluchae u nas bil brauzernyi sait a pri react.js e` brauzernoe prilojenie.server stanovitsya universalnim.

**6.lesson index.js,**

Package.json file delaet papku proektom, analiziruya node moduli, katorye zapuskayutsya.otrisovka proisxodit s pomoshyu jsx-a.

(rules common

1.class zareservirovano, I po etomu vmesto classa` className

2. import ./ -. Lejit v toi je papke chto I ya

3.funkciya doljna returnit odin element` div` roditelskii

)

Index.html stranica pustaya, ona otrisovivaet app.js iz src-a. esli izmenit’ content`

function App() {

  return (

    <div className="App">

      bla

    </div>

  );

}

Ili sdelat iz toi je samoi funkcii strelochnuyu funkciyu, kot.ya budet vozvrashat napr bla` na stranice ischeznet vsyo I budet tol;ko bla`

const App=()=> {

  return (

    <div className="App">

      bla

    </div>

  );

}

**7.lesson component**

Component` eto funkciya vozvrashayushaya razmetku jsx

Classic old school function`

function something(){}

modern`

const Something=function(){}

strelochnaya to je samoe, tolko`

const Something=()=>{}

v nashem slichae mi ne vizivaet etu funkciyu tak` Something(), a poyavlyaetsya novii the` <Something/>, kotoriy doljny vstavit kuda nujno

const App=()=> {

  return (

    <div className="App">

     <div>oinak1</div>

<Something/>

    </div>

  );

}

const Something =()=> {

  return (

    <div className="App">

     <div>oinak2</div>

    </div>

  );

}

**8.lesson Export teoriya**

Ispolzuetsya modulnost.

Dlya importirovaniya componenty`

import App from './App';

;(./ nachni s toi papke v kotorom ya naxojus-> v odnoi papke vmeste so mnoi)

import Something from './../Something.js; (./ ne iz toi je papki, a na 1 level up)

export default App; (iren export piti anenq, vor karoxananq inport anel)

**9.lesson –nichego**

**10.lesson –default opasnost’**

export default App; (iren export piti anenq, vor karoxananq inport anel)

erb export enq anum default inch vor ban u anun chenq dnum, importijamanak

u ete grenq

import Somethingelse from './../Something.js; (es depqum eli kashxati, qani vor konkret anun chenq drel(im mot chashxatec))

# **12.lesson React JS (Разбивка на компоненты)**

Src-I mej components papka enq bacum u meju prinak stexcum enq header, sidebar, profile componentnery…

**13.lesson (git repositoriya)**

Kogda delaem clone I xotim startanut proekt` npm start` daet error, package json exists, but cant find node modules, qani vor gitum chenq gcum node modules-y. dra hamar` npm install

Vorpeszi tesnenq tarber jamanaknerum commit arac versianery`

**gitk --all&**

heto mer uzac commiti vra enq sexmum yev cntrl insertov copy anum

terminalum`

git checkout shift insert (id-n insert enq anum)

package lock fayly kareli e jnjelvor toxi gnal mer uzac texy

ete menq mi voreve version baceuc heto popoxum enq u heto noric uzum enq verjin versiayin gnanq, chi toxum. Git statusov stugum enq vor fylum enq katarel fopoxutyuny yev grum`

git checkout src/App.js (orinak)

**14.lesson (css modules)**

Kogda daem imya className-u riskuem povtorit’, I vse tag-I s etim className-om izmenyatsya. Daje esli dlya kajdogo componenta butem sozdavat’ novii css. React vse css-y pereobrazovivaet v js, I ono vkluchaetsya v obshemu dokumentu (single page app)

Dlya togo, chtobi reshit’ etot vopros , my pereimenovivaem napr navbar.css v **navbar.module.css** (ono pozvolyaet pereobrazovivat’ className-y v unikal’nie napr` item\_11d533xdfd)

Import-y`

import './Css/Profile.css';

poxaren`

import classes from './Css/App.module.css';

isk className=’item’ poxaren`

className={classes.item}

kam classes-I poxaren cank ban`

import s from './Css/App.module.css';

isk className=’item’ poxaren`

className={s.item}

**v nazvaniyax stilya ne ispolzuem defis`- (!!!)**

mojno nijnee podcherkivanie `\_, ili camel case

esli xotim k element dobavit napr 2 classa`

<div className={css.dialog\_item + " " + css.dialog\_active}>Dima</div>

**15.lesson**

V papke components sozdaet papki dlya kajdogo komponenta` profile, sidebar, header…

I peretaskivaem ego jsx file I css file(v papke profile1 Profile.js I Profile.module.css)

Dlya importirovaniya image`

import avatar from './../../images/avatar.jpg'

<img src={avatar} alt="avatar" width='200px' height='auto' />

Lracum`

Senc dzev ka background-I picture-i`

 <div className={classes.background\_img} style={{

        backgroundImage: "url(" + "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ9R6WVd7lIJePzBBD2MiZRy511oh9ONh0H\_w&usqp=CAU" + ")",

      }}>

Inline background` **(backgroundImage !!!!)**

<div className={classes.avatar\_img} style={{

          backgroundImage: "url(" + avatar + ")"

        }}>

Kam`

<div className={classes.avatar\_img} style={{

          backgroundImage: `url(${avatar})`

        }}>

Multi inline style`

style={{

backgroundImage: "url(" + "https://images.pexels.com/photos/34153/pexels-photo.jpg?auto=compress&cs=tinysrgb&h=350" + ")",

backgroundPosition: 'center',

backgroundSize: 'cover',

backgroundRepeat: 'no-repeat'

}}

**Bootstrap miacnelu hamar`**

npm install --save bootstrap

or

npm install --save bootstrap@latest

import '../node\_modules/bootstrap/dist/css/bootstrap.min.css';

or

import 'bootstrap/dist/css/bootstrap.min.css';

(bayc or pillsy chi ashxatum, menak static stylerna berum)

**16.lesson**

Komponenta eta funkciya no mi ego vizivaem kak teg. Funkciya vizivaetsya za kadrom react. Kajdaya komponenta vizivaetsya vmeste s parametromi. Samaya vajnaya eta props I razmetku otobrazit po raznomu. Props react peredaet vsegda kak pustoi obyekt.

T k my ne pojem vizvat funkciyu napramuyu` napr App(props) I peredat vse chto xotim, vmesto propsa, a vizivaem funkciyu kak teg`(u tega est’ attributy, napr calue, classname, s kotorymi mojem nastroit teg), takje mojno peredat Dannie s komponentoi, kak attribute.`

<Header name=”dima”/> avromaticheski peredaetsya object {name: ‘dima’}, mojem takje peredat’ bolshe znachenii` <Header name=”dima” age= ‘’26’’ /> object props{name: ‘dima’,

age:’26’}

dlya etrisovki` <div>{props.name}</div> <div>{props.age}</div>

**19.lesson (ssilki routing)**

Dlya ssilok ispolzuetsya routing. Dlya etogo nado ustanovit paket.

Terminal

**npm install react-router-dom --save** (kam npm I react-router-dom)

node modulnery git repository chen ynknum, dra hamr petq u chi ashxati, dra hamar noric nmp i react-router-dom piti anel

package.json um avelanum e ` "react-router-dom": "^5.2.0",

app.js

    <Route path='/profile' component={Profile} /></div>

<div><Route path='/dialogs' component={Dialogs} /></div>

Sidebar-um el ete uzum enc contenty poxvi`

 <a href="/profile"><h5>Profile</h5></a>

      <a href="/dialogs"><h5>Messages</h5></a>

      <a href="/news"><h5>news</h5></a>

Bayc ays depqum, page-y amboxchutyamb upload e linum, isk menq uzum enq vor ,asy update lini

Po etomy vmesto qa href, ispolzuem navlink

 <NavLink to="/profile" activeClassName={css.actLink}><h5>Profile</h5></NavLink>

//chashxatec im mot activeClassName-y

      <NavLink to="/dialogs" activeClassName={css.actLink}><h5>Messages</h5></NavLink>

A na vercu importiruem

import { NavLink } from 'react-router-dom';

**22.lesson**

Dlya kajdogo otdelnogo cheloveka sozdaetm otdelnyj url k dialogu s nim`

div className={css.dialog\_item}>

                        <NavLink to="/dialogs/4">Sveta</NavLink>

                    </div>

                    <div className={css.dialog\_item}>

                        <NavLink to="/dialogs/5">Kara</NavLink>

                    </div>

Menq app js-icgalis einq route-ov depi dialogs`

<div><Route path='/dialogs' component={Dialogs} /></div>

Dra hamar karoxanum enq gnal /dialogs/1, /dialogs/2 ….

Bays ete grenq`

<div><Route exact path='/dialogs' component={Dialogs} /></div>

Chenq karox, exact-y petq e galis.

**22.lesson(props dlya dialogov)**

Dlya kajdogo

const Dialog\_item = (props) => {

    return (

        <div className={css.dialog\_item + " " + css.dialog\_active}>

            <NavLink to={'/dialogs/' + props.id}>{props.name}</NavLink>

        </div>

    )

}

A v classe Dialogs`

const Dialogs = () => {

    return (

        <div className={css.dialog\_flex}>

            <div className={css.dialog\_item1}>

                <h3>Dialogs</h3>

                <Dialog\_item name='DimichNew' id="1" />

                <Dialog\_item name='Sasha' id="2" />.. . .

**25.lesson(metod massiva map)**

V elemety massiva mojvo polajit componety.

Esli imeem massiv arr I ego vivesti, to elementy otabrajatsya ryadom let **arr=[‘mek’, ‘erku’, ‘ereq’]`**

**mekerkuereq**

takje mojno otabravit arrayjsx elementov.

Massiv odnix elementov mojno pereobrazit v massiv drugix elementov s pomoshyu

Napr. Imeeem Array imeaem let oldArr = [‘Viktor’, ‘Andrei’, ‘Sveta’, ‘Kara’],

Let newArr=oldArr.map((el)=>{

If(isMaleName(el)){

return 1;  
}else{

return 0;

}

}) (ili return isMaleName(el)? 1:0;)

(ili return “<div>”+el+”</div>”)

(ili return ‘<div>${el}</div>’ )

veradardznum e ` newArr=[1, 1, 0, 0]

ili esli ne xotim return pisat`

Let newArr=oldArr.map(el=>({

en:el,

ru:translateIntoRu(el)

})

)

Karox enq grel`

let newArr=oldArr.map(el=>(<Message enq={el.enq} ru={el.ru}/>))

orinak unenq`

let MessagesData = [

        { id: '1', message: 'Hi' },

        { id: '2', message: 'how are you' },

        { id: '3', message: 'Fine' },

        { id: '4', message: 'yo' },

        { id: '5', message: 'yoww' },

    ];

    let newMessagesData = MessagesData.map(

        el => (<Message\_item message={el.message} />)

    )

Texadrum enq mer uzac texum`

<div>

 {newMessagesData}

</div>

**26.lesson(BLL, UI)**

**UI-user interface**

**BLL-business logic layer**

UI –eto react v kotoryi prixodyat dannye s propsov

A iz propsov daniie v react js prixodyat ne iz bazy dannyx, a iz abstraktnoi. BLL dlya react eto redux

**27.lesson(component vs render)**

Dlya togo chtoby perebrosit dannye is App.js v razl komponenty, nado

<div><Route path='/dialogs' component={Dialogs} /></div>

-i poxaren`

<div><Route path='/profile' render={() => <Profile />} /></div>

Vorpeszi poxancenq propser-y.