React.js

Lecture 1

Agenda

- Introduction about React
- Single page application
- Virtual Dom
- Environment setup.
- React app structure.
- What is JSX ?
- Building Reusable components.
- Component lifecycle methods
- State and Props.
- Handling events

REACT

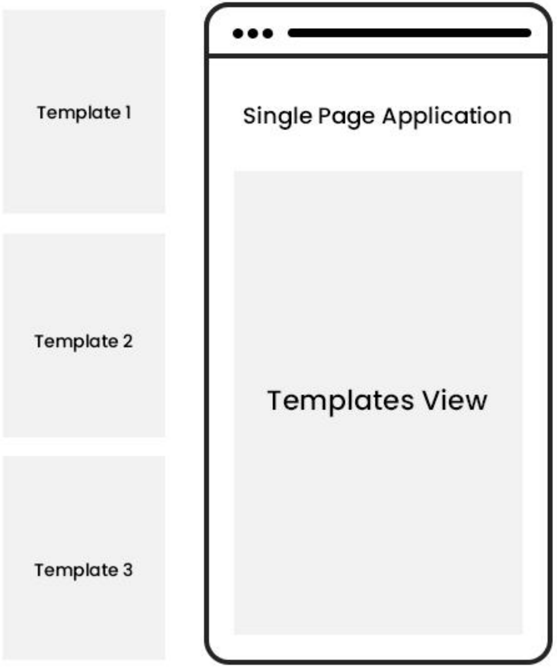
React is a JavaScript library for building user interfaces , developed at Facebook and released to the world in 2013.

Current stable version : 17.0.2

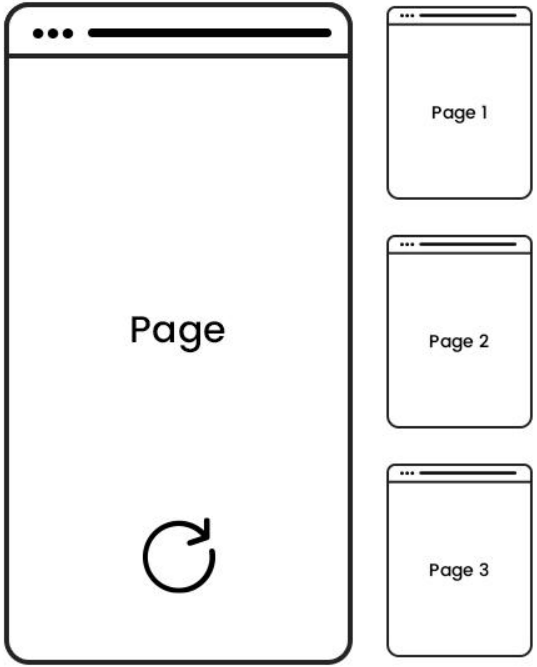
Why React ?

- Easy creation of dynamic applications.
- Improved performance using Virtual DOM.
- Reusable components
- Easy to learn.
- Dedicated tools for easy debugging.

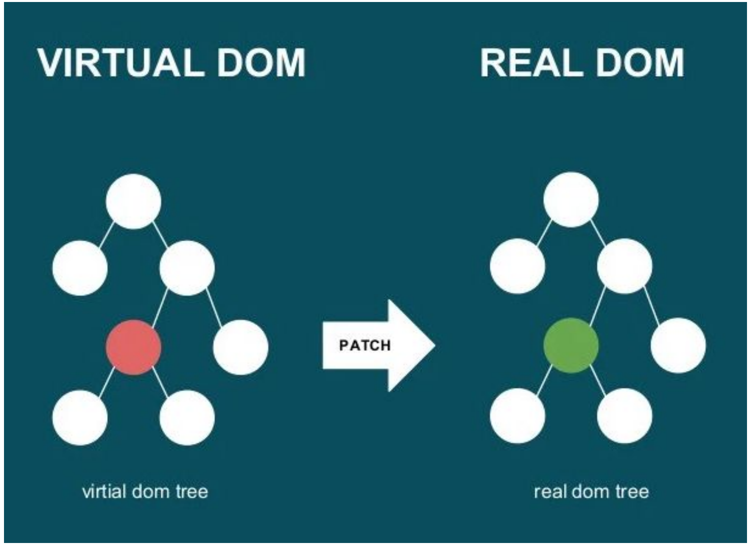
Single page application



No page refresh on request



Whole page refresh on request



Virtual DOM

The virtual DOM is only a virtual representation of the DOM

- When new elements are added to the UI, a virtual DOM, which is represented as a tree is created.
- Each element is a node on this tree. If the state of any of these elements changes, a new virtual DOM tree is created.
- This tree is then compared with the previous virtual DOM tree, the virtual DOM calculates the best possible method to make these changes to the real DOM. This ensures that there are minimal operations on the real DOM.

Getting started

- Install node : https://nodejs.org/en/
- Open new terminal in your directory.
- Create new react app : `npx create-react-app app-name`
- Enter your app folder.
- Run your react app : `npm start`

Let's explore our

React app structure

JSX

- JSX stands for JavaScript XML and allows us to write HTML in React.
- With JSX :

```
const myelement = <h1>I Love JSX!</h1>;

ReactDOM.render(myelement, document.getElementById('root'));
```

- Without JSX :

```
const myelement = React.createElement('h1', {}, 'I do not use JSX!');

ReactDOM.render(myelement, document.getElementById('root'));
```

Try : https://babeljs.io/

Class vs Functional component

- Component represent the part of user interface , before React Hooks (will be discussed later) when we want to create a dynamic component, we have to create a class component and use lifecycle methods to change states to make it reusable and encapsulate.

- It is regular ES6 classes that extends component class form react library, also known as “stateful” components because they implement logic and state.It must have render() method returning html.

Class vs Functional component

- **Functional component :**

```
const App = ( ) => {return (<h1> Hello </h1>)};
```

- **Class component :**

```
class App extends Component{
render( ){
return (<h1> Hello </h1>)}
};
```

Component lifecycle

There are 3 phases in a React component lifecycle:

- Mounting
- Updating
- Unmounting

Component lifecycle

Mounting

When mounting you have 4 lifecycle methods :

- **Constructor**

  The constructor is the first method that is called when mounting a component.

Component lifecycle

- **render()**

  From the render() method you return the JSX that builds the component interface.

- **componentDidMount()**

  This method is the one that you will use to perform API calls, or process operations on the DOM.

Component lifecycle

Updating

- **componentDidUpdate()**

  This method is called when the component has been updated in the DOM. Use this to call APIs that must be updated when the DOM changes.

Component lifecycle

Unmounting

- **componentWillUnmount()**

  The method is called when the component is removed from the DOM. Use this to do any sort of cleanup you need to perform.

Props and State

Components need data to work with. There are two different ways that you can combine components and data: either as props or state.

props

Props is short for properties and they are used to pass data between React components. React’s data flow between components is uni-directional (from parent to child only).

In a class component, props are passed by default. There is no need to add anything special, and they are accessible as this.props in a Component instance.

Props and State

State

- special built-in object called state, which allows components to create and manage their own data.
- In the Component constructor, initialize this.state
- Can Update state using this.setState() method
- Don’t mutate (change) state directly , always use setState( ) method to change state value that cause this component to rerender and update view

Handling Events

Handling events with React elements is very similar to handling events on DOM elements.

Example :

```
<button onClick={addUsers}>
    Add Users
</button>
```

Thank you

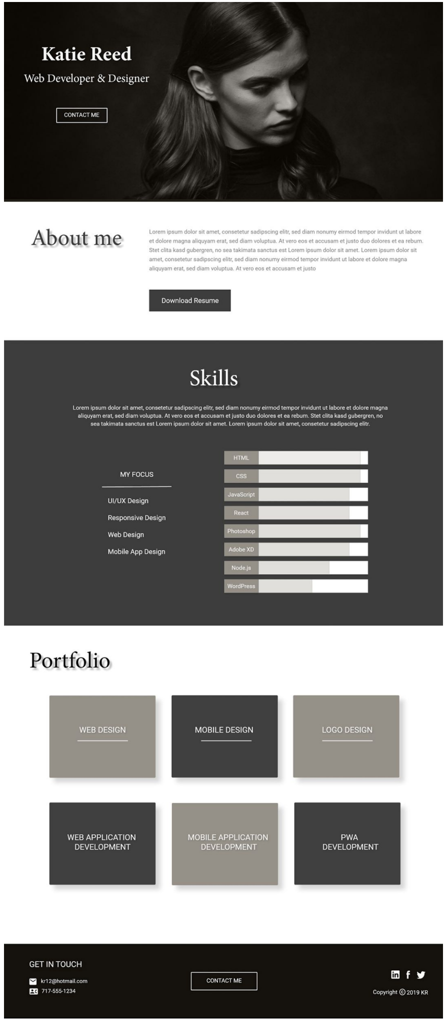
Lecture 1
Lap

Task 1 : Portfolio

Create a web page using React that contains the following sections :

- Hero section
- Bio and about me (education and experiences) section with a link to download resume
- Skills section
- Portolio and projects section
- Footer contains contact us section with social media icon links such as facebook , twitter , linkedin , github (fontawesom - [Bonus])

Using Bootstrap as UI library (search for it) and create reusable components for your page.



Katie Reed

Web Developer & Designer

About me

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo

Skills

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr.

Portfolio

GET IN TOUCH

kr12@hotmail.com

717-555-1234

Copyright © 2019 KR