React

* React is javascript library for building user interface.
* It client side javascript library.
* Declarative, component focused approch.
* Declarative approch:

Define desired target states and let react figure out the actual javascript DOM instructions.

* Building Single Page Applications (SPAs) :-

React can also used to control the entire frontend of a web application.

React can be used to control parts of HTML pages or entire pages

* Note: React is lean and focosed component based UI library. Certain features like routing, etc. a re added via community packages.
* Course content:
  + Basic and foundation (Key features)
    - Components and Building UI
    - Working with events and data
    - “props” and “State”
    - Style react app and components
    - React hooks
  + Advance concepts(Building for production)
    - Side effects
    - “Refs”
    - More react hooks
    - React context API and redux(third party library)
    - Forms
    - http requests
    - custome hooks
    - routing
    - deployment
    - Nextjs

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# **Component driven user interfaces**

* Module Content:
  + React core syntax
  + JSX
  + Working with component
  + Working with data
* Components:
  + React is all about components.
  + What is component?

components are like JavaScript functions. They accept arbitrary inputs (called “props”) and return React elements describing what should appear on the screen.

* + Why Components?

1) Reusability

2)Seperation of concerns

3)Split big bunck of code into multiple smaller functions.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# **Create new react project**

* create-react-app
* <https://create-react-app.dev/>
* <https://github.com/facebook/create-react-app>
* Node js required.
* Step 1: open the folder on terminal where project is created

cd desktop

* Step 2: run command

npx create-react-app app-name

ex: npx create-react-app my-app

Note :

To update the version of create-react-app

Use following command

npm install react-scripts@latest

* Step 3: Now open created app folder

Ex: cd my-app

* Step 4: Start project

npm start

* Step 5: Open project folder in VSCODE

Steps to run project on vscode terminal:

1. npm install
2. npm start

Note : Folder Structure of project

my-app/

README.md

node\_modules/

package.json

public/

index.html

favicon.ico

src/

App.css

App.js

index.css //this css content common styling

entire page

index.js //execution starting point

# ***Analyzing a standerd react project***

//index.js file

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));

Selecting root using DOM

Using <App> component render App component to main page

Index.html file (public folder)

<body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

</body>

App.js file

function App() {

  return (

    <div>

      <h2>Let's get started!</h2>

    </div>

  );

}

export default App;

App is root component New components are nested in app component

# **Create Custom Components**

* for component create separate folder “components” to src folder
* This components are like tree structure and App component is root component.

Render to single Html page

Note:

1) Component Name Should be starting with capital letter(Naming Convension). Because lower case are built in html elements.

1. In component we must have only one root component.

<div>

<div>First line</div>

<h1>Title</h1>

<p>Paragraph</p>

</div>

<div>First line</div>

<h1>Title</h1>

<p>Paragraph</p>

Not allowed Allowed

* Functional Component:

function ExpenseItem(){

    return(

        <h1>Expense Component</h1>

    );

}

export default ExpenseItem;

* Styling component:
  + Create .css file
  + Import this file into component
  + Use className instead of class to style element.

import './ExpenseItem.css';

function ExpenseItem(){

    return(

        <div className ="expense-item">

            <div>11 JAN, 2021</div>

            <div className="expense-item\_\_description">

                <h2>Car Expences</h2>

                <div className="expense-item\_\_price">$4.5</div>

            </div>

        </div>

    );

}

export default ExpenseItem;

* Using dynamic data to component:

import './ExpenseItem.css';

function ExpenseItem(){

    const expenseDate = new Date(2021, 2, 11);

    const expenseTitle = "Car Insurance";

    const expensePrice = 4.5;

    return(

        <div className ="expense-item">

            <div>{expenseDate.toLocaleDateString()}</div>

            <div className="expense-item\_\_description">

                <h2>{expenseTitle}</h2>

                <div className="expense-item\_\_price">${expensePrice}</div>

            </div>

        </div>

    );

}

export default ExpenseItem;

* Passing data using “props”:-

Component.js

import './ExpenseItem.css';

function ExpenseItem(props){

    return(

        <div className ="expense-item">

            <div>{props.date.toLocaleDateString()}</div>

            <div className="expense-item\_\_description">

                <h2>{props.title}</h2>

                <div className="expense-item\_\_price">${props.amount}</div>

            </div>

        </div>

    );

}

export default ExpenseItem;

app.js

import ExpenseItem from "./components/ExpenseItem";

function App() {

  const expenses = [

    {

      id: "e1",

      title: "Toilet Paper",

      amount: 94.12,

      date: new Date(2020, 7, 14),

    },

    { id: "e2", title: "New TV", amount: 799.49, date: new Date(2021, 2, 12) },

    {

      id: "e3",

      title: "Car Insurance",

      amount: 294.67,

      date: new Date(2021, 2, 28),

    },

    {

      id: "e4",

      title: "New Desk (Wooden)",

      amount: 450,

      date: new Date(2021, 5, 12),

    },

  ];

  return (

    <div>

      <h2>Let's get started!</h2>

      {expenses.map(function (value) {

        return (

          <ExpenseItem

            title={value.title}

            amount={value.amount}

            date={value.date}

          />

        );

      })}

    </div>

  );

}

export default App;

* Wrapper Components:

import "./Card.css"

function Card(props){

    const classes = "card " +props.className;

    return(

        <div className={classes}>

            {props.children}

        </div>

    )

}

export default Card;