**REACT CONTENTS**

1. **Introduction To React**

* **Features of React**

1. **DOM, Shadow DOM & Virtual DOM**
2. **Cons of React**
3. **Installation of React**
4. **React using npm**
5. **Introduction to React Component**

* **How to use multi components in React Application**

1. **Class Component**
2. **Difference between Function and Class component**
3. **Data Bindings**

* **Simple, Nested Bindings**
* **Fetching Data from API and Bind**

1. **How to Use State functionality and Event Binding in Functional**

**Component**

* **How to use Two Way Data Binding**
* **State, FakeStoreAPI**

1. **How to Use State functionality in and Event Binding Class**

**Component**

* **Dynamic Style Change, Theme Change.**

1. **Forms and Validations**

* **Using React (Lengthy Process)**
* **Using 3rd Party API/Library i.e Formik (Easy & Industrial Standard)**
* **Formik Validations. (React Forms)**
* **Data Binding & Validations using YUP Library.**
* **Formik Components.**
* **Validation State.**

1. **Component Life-Cycle Hooks.**

* **For Class Component.**
* **For Functional Component.**
* **Predefined React Hook**

**\*useState()**

**\*useEffects()**

**\*useContext()**

**\*useCookies()**

**\*useReducer()**

* **Custom Hook**

1. **Working with APIs (Get/Post/Patch/Delete)**
2. **React Routing**

* **Intro to Routing**
* **Basic Routing Components**
* **Complete Login & Register Example using Routing**
* **Route Parameters**
* **Nested Routes**

1. **Typescript Concepts (33-38 ½)**
2. **Using typescript in React Application**
3. **Material UI for React Project**
4. **Route Hooks**
5. **Redux**

**THNINGS NEEDS TO BE PRACTISE**

1. **How to work with APIS, GET/POST/PATCH/PUT**

**Using Forms (Using Formik) (Refer Video No 27)**

1. **Use Login and Register**
2. **Perform data binding and validations. (Done)**

**FAQs in REACT**

* + - 1. **What is the pros of developing screen using typescript with React applications?**

**Using typescript in react application helps to develop robust, scalable & maintainable applications. It helps to reduce runtime errors because of we can handle the data type in in frontend as per the expected structure. In case of typescript errors can be noticed at the time of compilation.**

* + - 1. **How to solve the error while downloading react app with typescript?**
* **npx create-react-app react-typescript-app --template typescript**

**(got error)**

* **npx create-react-app react-typescript-app --template typescript --legacy-peer-deps  
  (tsconfig.json did not get created)**
* **Manually created tsconfig.json file  
  inside this file added  
  *{***

***"compilerOptions": {***

***"target": "es5",***

***"lib": ["dom", "dom.iterable", "esnext"],***

***"allowJs": true,***

***"skipLibCheck": true,***

***"esModuleInterop": true,***

***"allowSyntheticDefaultImports": true,***

***"strict": true,***

***"forceConsistentCasingInFileNames": true,***

***"noFallthroughCasesInSwitch": true,***

***"module": "esnext",***

***"moduleResolution": "node",***

***"resolveJsonModule": true,***

***"isolatedModules": true,***

***"noEmit": true,***

***"jsx": "react-jsx"***

***},***

***"include": ["src"]***

***}***

* **Then   
  rm -rf node\_modules package-lock.json**
* **Then  
  npm install**
* **Then   
  npm install --save-dev @types/react @types/react-dom  
    
  Then at last worked fine.**