React:

Single Page Application , Responsive SPA Application

1. Creating React Appliction
   1. npx create-react-app (19.0)
2. React Architecture
   1. Component based Architecture
      1. Define everything as Component
3. Components:
   1. Class Based Approach (traditional Model)
   2. Functional Approach (recommended for all)
      1. Define a function using function
      2. Define functions using arrow
4. Create a components
   1. Use the HTML, CSS for designing the components
   2. Use predefined templates or predefined components
      1. Bootstrap,
      2. React bootstrap (user defined components, navbar, buttons) , bootstrap (css, js)
         1. Responsive Web Page
            1. Compatible with design

Write more code of CSS media Query

1. Communication Between Components
   1. **Props** -> transfer data between one components to another
   2. **State** -> Manage the data in the component
   3. Manage the data in application
      1. Local storage ()
      2. **Context API ()**
2. React
   1. Props and state
      1. Class based approach (this.props, this.state)
      2. Function Approach(props, useState)
         1. Manage Sttae is
         2. Receive data from component Easy
   2. Routing:
      1. Browser Router
         1. Defined path
      2. Navigate using path, Link, Navlin (react-router)
3. Formik -> design a form and validate Form
   1. Developed based on React Hooks library
      1. React hooks
         1. useState,
4. Yup for schama defnation with validation,
5. use as a state or use as a element
6. How to store data in json File.

Life Cycle:

* Mounting
* Updating
* Unmounting

ContextAPI:

* Provide the data / data sharing
* Clicent
* Client Server
  + Providers, consumers

Hooks Methods:

* useState -> manage data in a component
* useEffect -> mange the lifecycle of hooks, functional component
* createContext() -> used to manage the data in application
  + provider -> application
  + consumer in componnets
  + in Application , some data I need to use in multiple place without calling that
    - styles
      * button, card, bg ,

1. Create Context

Data we manage throw context API

Services we have in services,

React Redux:

State Manage Tools:

* Context API -> mainly focus on data
* Redux -> manage and functionality
* App -> added a data -> it reflect all other components
* React Redux – react
* Redux toolkit -> any JS library framework

Work a store

Ecommerce application

Product count 9

1 product

Redux:

* Store
* Reducer

npm install @reduxjs/toolkit

npm install react-redux

in Redux:

Dispacher -> to invoke events

Selectors -> to invoke values

Core Concepts:

Redux: (redux toolkit)

* Is used to provide data , it provides updated data everytime
  + Based on the state on application, redux maintains the current state/ precious Statte, based on that provide proper info to outside
* Store -> provides the data
  + Collecting the slicers
    - Slicers basically a object contains the data and its functions
    - Data are present in initialstate
      * Changes of data need to track based on the reducers
    - Reducers are simply collecting the action

Product Management System:

* BE (Nest Js) endpoints (Users/ Product)
* FE (react Application) Access Nest JS API endpoints in application