Chapter 01 - Inception

Link to my Code

Theory -

- What is Emmet?
- Difference between a Library and Framework?
- What is CDN? Why do we use it?
- Why is React known as React?
- What is crossorigin in script tag?
- What is diference between React and ReactDOM
- What is difference between react.development.js and react.production.js files via CDN?
- What is async and defer? - see my Youtube video ;)

Coding -

- Set up all the tools in your laptop
    - VS Code
    - Chrome
    - Extensions of Chrome
- Create a new Git repo
- Build your first Hello World program using,
    - Using just HTML
    - Using JS to manipulate the DOM
    - Using React
        - use CDN Links
        - Create an Element
        - Create nested React Elements
        - Use root.render

- Push code to Github (Theory as well as code)
- Learn about Arrow Functions before the next class

References:

- https://beta.reactjs.org/apis/react/createElement
- https://www.youtube.com/watch?v=IrHmpdORLu8

Chapter 02 - Assignment - Igniting our App

Please Note: *Write the answers and code on your own while finishing your assignments. Try to put down your thoughts into words by yourself in your own words. (This will help you develop muscle memory and you will remember all the concepts properly)* ✌️

 # Theory Assignment:

- - What is `NPM`?
- - What is `Parcel/Webpack`? Why do we need it?
- - What is `.parcel-cache`
- - What is `npx` ?
- - What is difference between `dependencies` vs `devDependencies`
- - What is Tree Shaking?
- - What is Hot Module Replacement?
- - List down your favourite 5 superpowers of Parcel and describe any 3 of them in your own words.
- - What is `.gitignore`? What should we add and not add into it?
- - What is the difference between `package.json` and `package-lock.json`
- - Why should I not modify `package-lock.json`?
- - What is `node_modules` ? Is it a good idea to push that on git?
- - What is the `dist` folder?
- - What is `browserlists`
  Read about dif bundlers: vite, webpack, parcel
- Read about: ^ - caret and ~ - tilda
- Read about Script types in html (MDN Docs)

 # Project Assignment:

- In your existing project

- - intialize `npm` into your repo
- - install `react` and `react-dom`
- - remove CDN links of react
- - install parcel
- - ignite your app with parcel
- - add scripts for “start” and “build” with parcel commands
- - add `.gitignore` file
- - add `browserlists`
- - build a production version of your code using `parcel build`

 # References

- Creating your own create-react-app
- Parcel Documentation
- Parcel on Production
- BrowsersList: https://browserslist.dev/

Chapter 03 - Laying the Foundation

Topics

- JSX
- React.createElement vs JSX
- Benefits of JSX
- Behind the Scenes of JSX
- Babel & parcel role in JSX
- Components
- Functional Components
    - Composing Components

Assignment

- What is JSX?
- Superpowers of JSX
- Role of `type` attribute in script tag? What options can I use there?
- `{TitleComponent}` vs `{<TitleComponent/>}` vs `{<TitleComponent></TitleComponent>}` in JSX

Coding Assignment:

- Create a Nested header Element using React.createElement(h1,h2,h3 inside a div with class “title”)
    - Create the same element using JSX
    - Create a functional component of the same with JSX
    - Pass attributes into the tag in JSX
    - Composition of Component(Add a component inside another)
    - `{TitleComponent}` vs `{<TitleComponent/>}` vs `{<TitleComponent></TitleComponent>}` in JSX
- Create a Header Component from scratch using Functional Components with JSX
    - Add a Logo on left
    - Add a search bar in middle
    - Add User icon on right
    - Add CSS to make it look nice

References

- Babel: https://babeljs.io/
- Attribute Type: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script#attr-type

- JS Modules: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules
- Babel Playground: https://babeljs.io/repl#
- React without JSX: https://reactjs.org/docs/react-without-jsx.html

Chapter 04 - Talk is cheap, show me the code!

Assignment

- Is JSX mandatory for React?
- Is ES6 mandatory for React?
- `{TitleComponent}` vs `{<TitleComponent/>}` vs `{<TitleComponent></TitleComponent>}` in JSX
- How can I write comments in JSX?
- What is <React.Fragment></React.Fragment> and <></> ?
- What is Virtual DOM?
- What is Reconciliation in React?
- What is React Fiber?
- Why we need keys in React? When do we need keys in React?
- Can we use index as keys in React?
- What is props in React? Ways to
- What is a Config Driven UI ?

Coding Assignment:

- Build a Food Ordering App
    - Think of a cool name for your app
    - Build a AppLayout
    - Build a Header Component with Logo & Nav Items & Cart
    - Build a Body Component
    - Build RestaurantList Component
    - Build RestaurantCard Component
    - Use static data initially
    - Make your card dynamic(pass in props)
    - Props - passing arguments to a function - Use Destructuring & Spread operator
    - Render your cards with dynamic data of restaurants
    - Use Array.map to render all the restaurants

PS. Basically do everything that I did in the class, in the same sequence. Don't skip small things.

References

- Code Link: https://bitbucket.org/namastedev/namaste-react-live/src/master/
- React without JSX: https://reactjs.org/docs/react-without-jsx.html
- Virtual DOM: https://reactjs.org/docs/faq-internals.html
- Reconciliation: https://reactjs.org/docs/reconciliation.html

- React Fiber Architecture: https://github.com/acdlite/react-fiber-architecture
- React Without ES6: https://reactjs.org/docs/react-without-es6.html
- Index Keys as Anti-Pattern: https://robinpokorny.com/blog/index-as-a-key-is-an-anti-pattern/

Chapter 05 - Let's get Hooked!

Assignment

- What is the difference between **Named** Export, **Default** export and *** as** export?
- What is the importance of config.js file
- What are React Hooks?
- Why do we need a **useState** Hook?

Coding Assignment:

- Clean up your code
- Create a Folder Structure for your app
- Make different files for each Components
- Create a config file
- Use all types of import and export
- Create a Search Box in your App
- Use useState to create a variable and bind it to the input box
- Try to make your search bar work

References

- Code Link - https://bitbucket.org/namastedev/namaste-react-live/src/master/

Chapter 06 - Exploring the world

- What is a Microservice?
- What is Monolith architecture?
- What is the difference between Monolith and Microservice?
- Why do we need a useEffect Hook?
- What is Optional Chaining?
- What is Shimmer UI?
- What is the difference between JS expression and JS statement
- What is Conditional Rendering, explain with a code example
- What is CORS?
- What is async and await?
- What is the use of `const json = await data.json();` in getRestaurants()

Coding Assignment :

- Play with the useEffect Hook to see when it is called?(before or after render)
- Play with dependency array in useEffect Hook
- Play with the developer console by putting a debugger in render and useEffect
- Call an actual API to get data
- Handle Error in your API call
- Build Shimmer UI when data in not loaded
- Render your UI with actual API data
- Make Search functionality work
- Make a Login Logout button which toggles with a state

Chapter 07 - Finding the Path

Assignment

- What are various ways to add images into our App? Explain with code examples
- What would happen if we do console.log(useState())?
- How will useEffect behave if we don't add a dependency array ?
- What is SPA?
- What is difference between Client Side Routing and Server Side Routing?

Coding Assignment:

- Add Shimmer Effect without installing a library
- Install react-router-dom
- Create a appRouter and Provide it to the app
- Create a Home, About, Contact Page with Link (use child routes)
- Make a Error page for routing errors
- Create a Restaurant Page with dynamic restaurant ID
- (Extra) - Create a login Page using Formik Library

Resources:

- React Router DOM - https://reactrouter.com/en/main
- Client Side Routing - https://reactrouter.com/en/main/start/overview
- Formik - https://formik.org/

Chapter 08 - Let's get Classy

Theory Assignment:

- How do you create Nested Routes react-router-dom cofiguration
- Read abt createHashRouter, createMemoryRouter from React Router docs.
- What is the order of life cycle method calls in Class Based Components
- Why do we use componentDidMount?
- Why do we use componentWillUnmount? Show with example
- (Research) Why do we use super(props) in constructor?
- (Research) Why can't we have the callback function of useEffect async?

Coding Assignment:

- Create a Class Based Component
    - Create 2 class based child components
    - Pass props from Parent to child
    - Create a constructor
    - Create a state variable inside child
    - Use this.setState to update it
    - What if there are multiple state variables?
    - Write a console.log for each lifecycle method
    - Play with the console logs to find out the correct order of their execution
- Create interval inside componentDidMount?
    - Use clearInterval to fix the issue caused by that interval

React Life Cycle Method Diagram - https://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/

Code Link - https://bitbucket.org/namastedev/namaste-react-live/src/master/

Chapter 09 - Optimizing our App

Theory -

- When and why do we need lazy()?
- What is suspense?
- Why we got this error : A component suspended while responding to synchronous input. This will cause the UI to be replaced with a loading indicator. To fix, updates that suspend should be wrapped with startTransition? How does suspense fix this error?
- Advantages and disadvantages of using this code splitting pattern?
- When do we and why do we need suspense?

Coding -

- Create your custom hooks
- Try out lazy and suspense
- Make your code clean.

References:

- https://reactjs.org/docs/hooks-custom.html
- https://beta.reactjs.org/apis/react/lazy#suspense-for-code-splitting

Chapter 10 - Jo dikhta hai vo bikta hai

Theory:

- Explore all the ways of writing css.
- How do we configure tailwind?
- In tailwind.config.js, what does all the keys mean (content, theme, extend, plugins)?
- Why do we have .postcssrc file?

Coding:

- Configure Tailwind and try to build your whole app using tailwind.

References:

- https://styled-components.com/
- https://tailwindcss.com/
- https://getbootstrap.com/
- Material UI - https://mui.com/

Chapter 11 - Data is the new Oil

Theory:

- What is prop drilling?
- What is lifting the state up?
- What is Context Provider and Context Consumer?
- If you don’t pass a value to the provider does it take the default value?

Coding:

- Practice React Context with code examples
- Try out Nested Contexts

References:

- Lifting State Up
- React Context

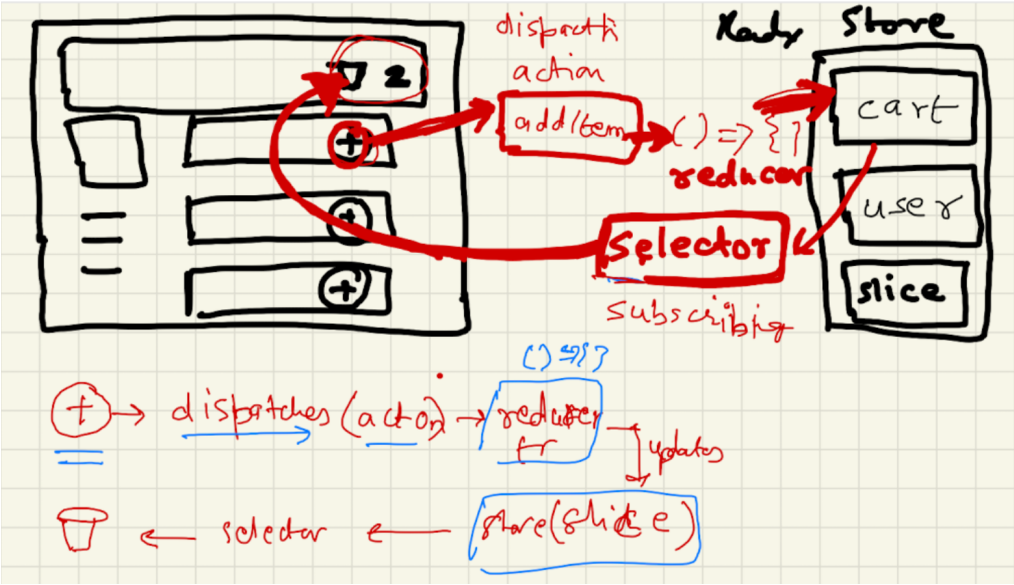
Chapter 12 - Let's Build our Store

Theory:

- useContext vs Redux.
- Advantage of using Redux Toolkit over Redux.
- Explain Dispatcher.
- Explain Reducer.
- Explain slice.
- Explain selector.
- Explain createSlice and the configuration it takes.

Coding:

- Practice making a store, slices and do read and write operations using Redux Store
- Build Cart Flow using Redux Store



Chapter 13 - Time for the test

Theory:

- What are different types for testing?
- What is Enzyme?
- Enzyme vs React Testing Library
- What is Jest and why do we use it?

Coding:

- Setup React Testing Library
- Write Unit Tests for Header Component to test for Logo, Cart - 0 items and Online Status
- Write Integration Test case for search feature on the Homepage
- Write Integration Test case for Add to Cart flow