Session 4 ⇒ 07/01/2023

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 ?

1.)

- No, JSX is not at all mandatory for react.
- It could be used when you are planning to avoid compilation.
- It acts as a syntactic sugar for `React.createElement()`.

2.)

- No, ES6 is not at all mandatory for react.
- Same functionalities could be build without using the features of ES6.

3.)

- `{TitleComponent}` ⇒ the component will not get rendered. Why ?🤔. Simple. {Just a function is written But Not Called}. Instead `{TitleComponent()}` would have worked perfectly.
- `{<TitleComponent/>}` ⇒ the component will be rendered.
- `{<TitleComponent></TitleComponent>}` ⇒ the component will be rendered.

4.)

```
{ // Single Line Comment }

{ /* Multi
     Line
```

```
     Comment */
}
```

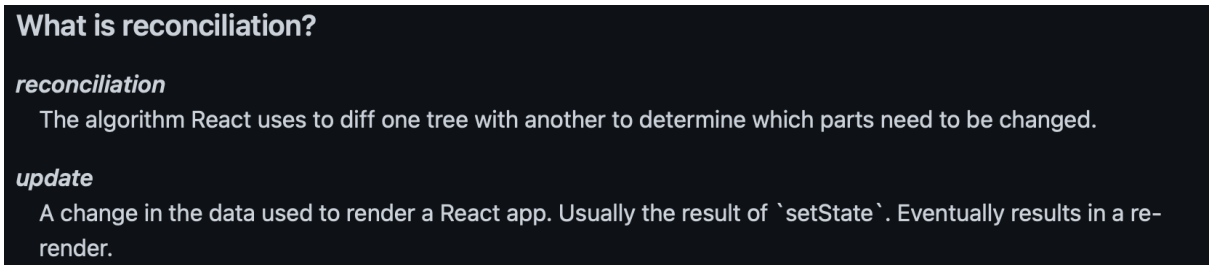
5.)

```
// React.Fragment => a component exported by React
// JSX can only have One Parent!
// Suppose if i want two parents, then either i would have to add them inside a div
// or we can make use of a component provided by the React
// React.Fragment is like a empty tag
// But it looks ugly to use this big term again and again
// So, just use <> </> these empty tags . But, what are these ? A shorthand for React.Fragment
// We can't pass any attribute to the empty tag

// Most Important => React.Fragment can only have key props.
// But this is not possible while using the shorthand property as it acts like
// empty tags which can't have any attributes.
```

6.)

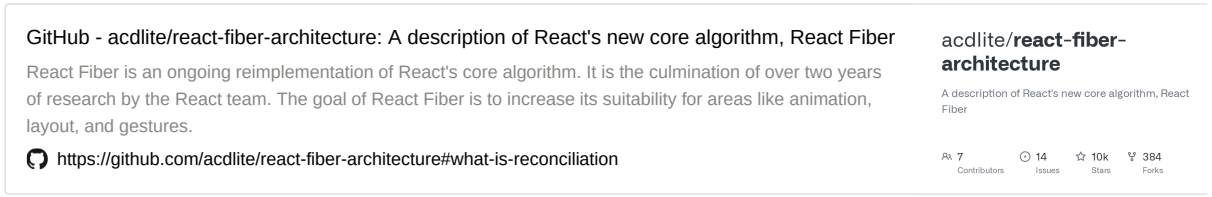
- Virtual Dom is a representation of the actual DOM.

7.)

- Reconciliation is the process of finding the differences between the two working trees (the actual dom and the virtual dom) and rerendering only those elements which got changed.
- 

8.)

- React Reconciliation Engine For Incremental Rendering.
- Best Suited For Animation, Layout, Gestures.
- Currently Being Used By Invact Metaversity.
- Build using the powers of React Fiber.



9.)

- To uniquely identify the elements present at the same level in a tree

- for example, if inside a div i have two tags one h1 and one p, then if either of them changes then react(diff algo) will be able to find out which tag changed.
- But think of a scenario where in i have 3 h1 inside a div and now if i insert one more h1 inside the div, then how will it detect which h1 got inserted as all are same tags, so in such a case it would end up rerendering all of the 4.
- In order to avoid such happenings, it is advisable to pass keys to siblings so that only the changed tags get rerendered and boost performance

10.)

- Yes we can definitely use index as keys.
- But Not Recommended
- Why So?
- Because it might happen that two or more elements have the same keys at some instance, and we know that keys must be unique.
- Therefore, we try to use only unique values as keys.
-

```
// Never use index as a key.
// valid but not recommended
// no key <<<<<< index key (use ONLY when you don't have anything else) <<<<<<<<<< unique key (best practice)
```

11.)

- Functional Components are basically a function. And we have heard that we pass props into the component. What does this signify? Simple. Props are simply arguments when passed and simply parameters while being received inside the func component.
- Props ⇒ shorthand for properties
- passing some properties(data) into my components
- when i need to pass some data from the parent component to the child component then i need to pass properties to the child component
- Ways to Pass : -
  - Simply pass the required object

```
const FnComp = (props) => {
  console.log(props.restaurant.name);
}

<FnComp restaurant={Restaurant[0]}/>
```

  - Use Destructuring

```
const FnComp = ({ restaurant }) => {
  console.log(restaurant.name);
}

<FnComp restaurant={Restaurant[0]}/>
```

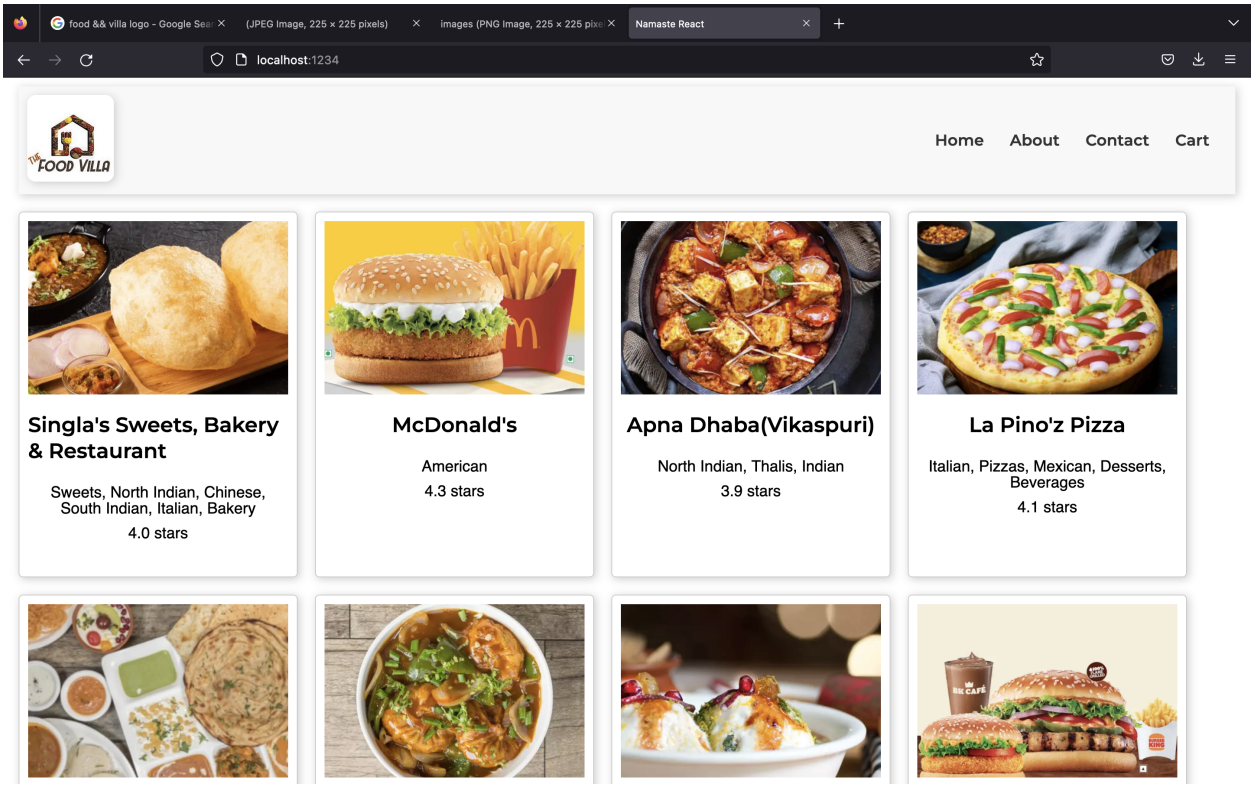
- Destructuring Along With Spread Operator

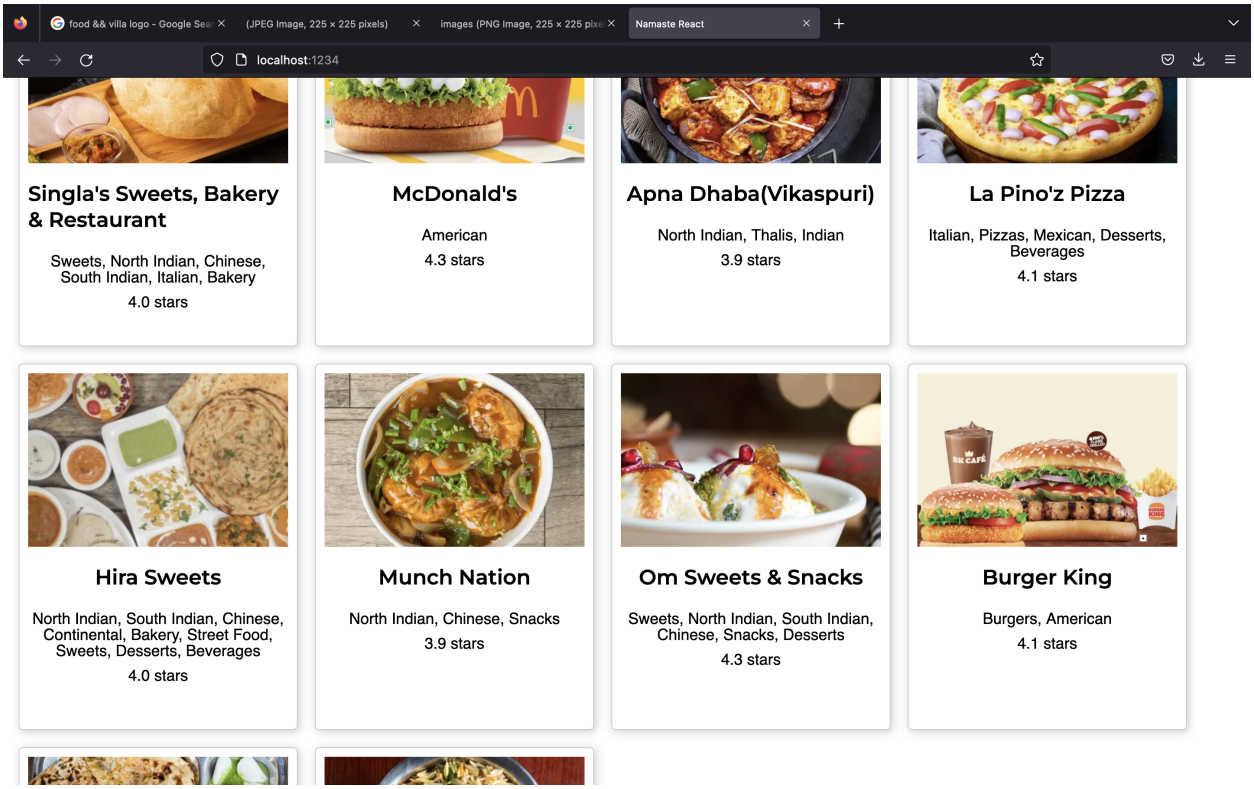
```
const FnComp = ({ cloudinaryImageId, name, cuisines, avgRating }) => {
  console.log(cloudinaryImageId, name, cuisines, avgRating);
}

<FnComp restaurant={...Restaurant[0].data}/>
```

12.)

- Config Driven UI
- UI handled with the help of Backend
- The website needs to be seen as different in different geographical areas as different places have different deals, offers, discounts, products, climate effect, etc.
- But for satisfying this thing, should we make different website for different locations ??
- No ! We must provide dynamic data from the backend ⇒ according to which the UI will be rendered.
- If there are no offers running in Delhi, then there won't be any data coming inside the offers carousel for the location: 'Delhi'.
- So since this is empty ⇒ won't be rendered ⇒ our website looks different as different configs.

Coding Assignment Output ➖




food && villa logo - Google Sear
(JPEG Image, 225 × 225 pixels)
images (PNG Image, 225 × 225 pixe
Namaste React
localhost:1234
Singla's Sweets, Bakery & Restaurant
Sweets, North Indian, Chinese, South Indian, Italian, Bakery
4.0 stars
McDonald's
American
4.3 stars
Apna Dhaba(Vikaspuri)
North Indian, Thalis, Indian
3.9 stars
La Pino'z Pizza
Italian, Pizzas, Mexican, Desserts, Beverages
4.1 stars
Hira Sweets
North Indian, South Indian, Chinese, Continental, Bakery, Street Food, Sweets, Desserts, Beverages
4.0 stars
Munch Nation
North Indian, Chinese, Snacks
3.9 stars
Om Sweets & Snacks
Sweets, North Indian, South Indian, Chinese, Snacks, Desserts
4.3 stars
Burger King
Burgers, American
4.1 stars