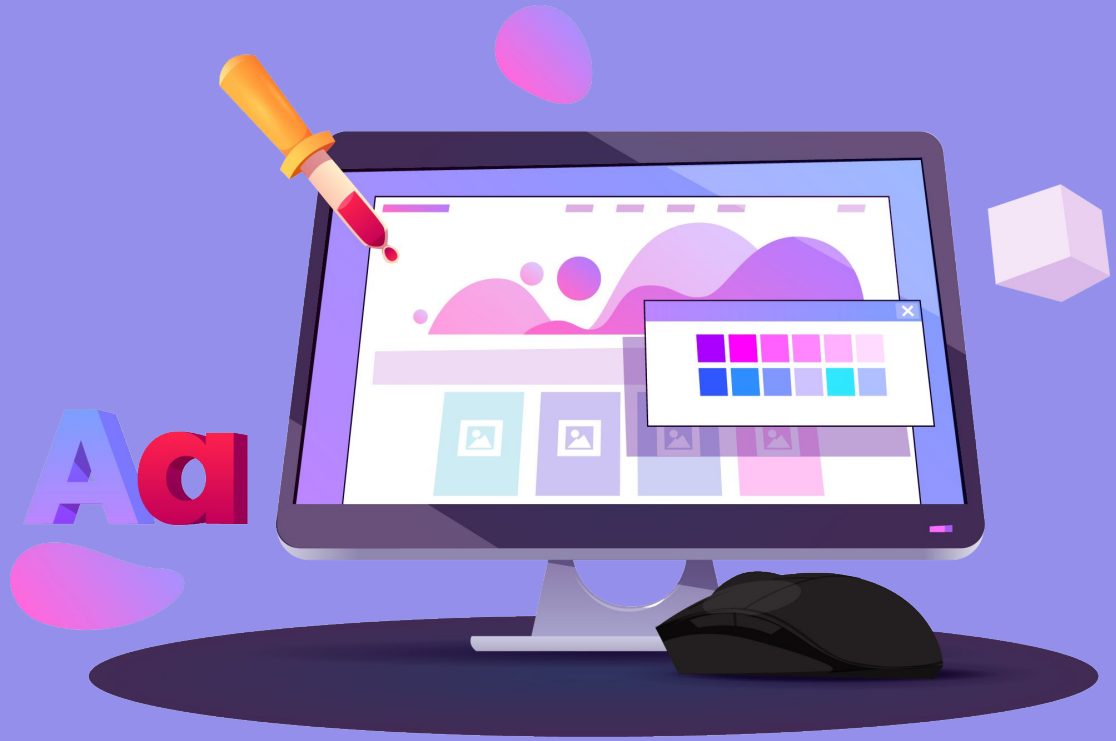
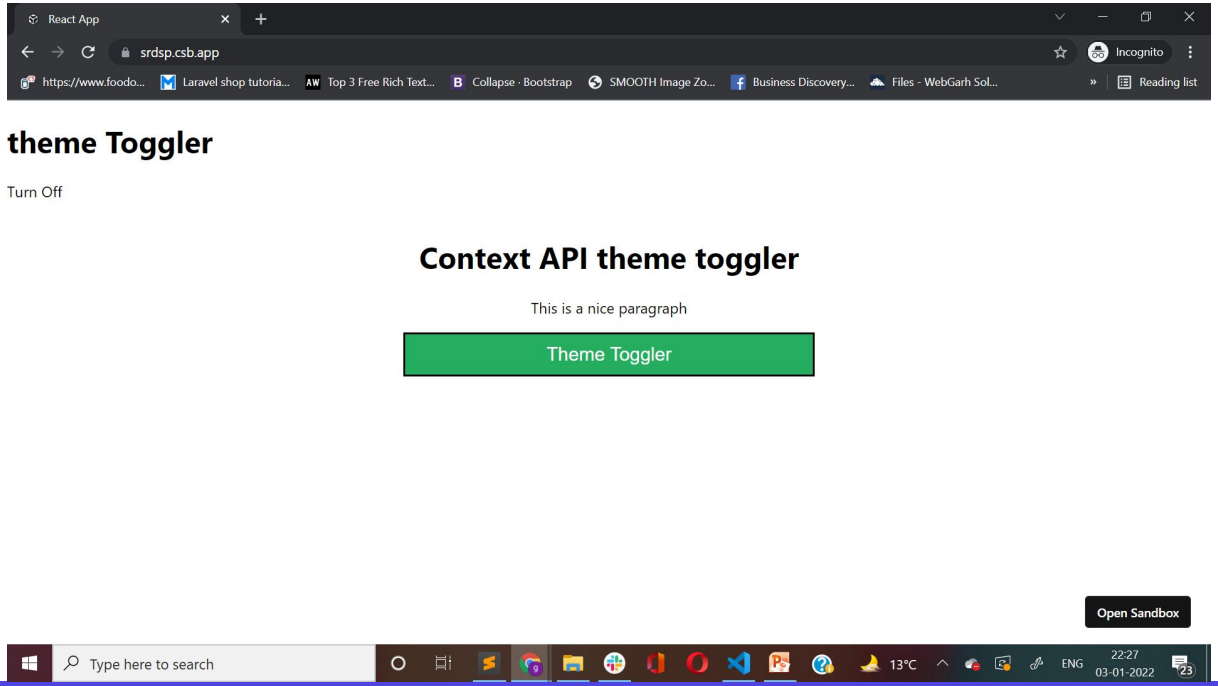
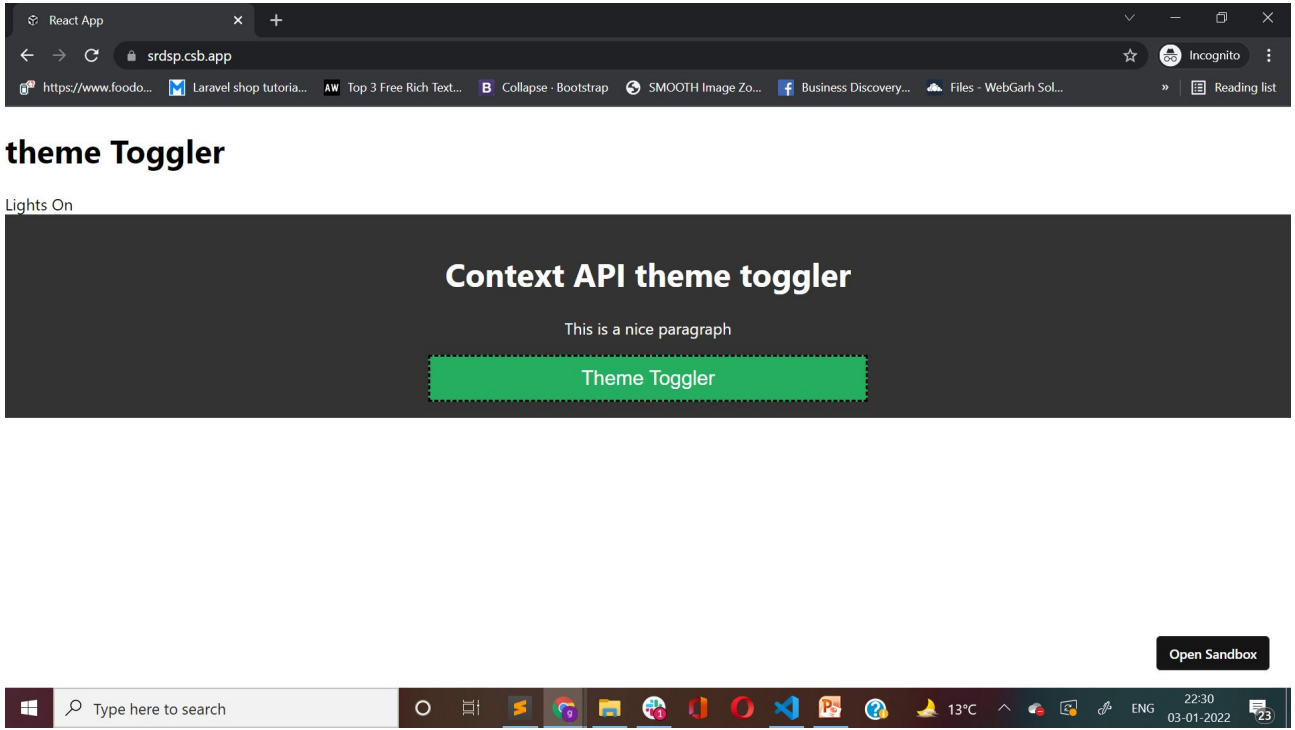
Context API

Relevel
by Unacademy



APP FEATURE WE WILL BE BUILDING TODAY IS Theme Toggler




React App
srdsp.csb.app
Incognito
https://www.foodo...
Laravel shop tutoria...
Top 3 Free Rich Text...
Collapse · Bootstrap
SMOOTH Image Zo...
Business Discovery...
Files - WebGarh Sol...
Reading list
theme Toggler
Lights On
Context API theme toggler
This is a nice paragraph
Theme Toggler
Open Sandbox
Type here to search
13°C
ENG
22:30
03-01-2022

What are Props

- Arguments passed to the React component.
- Props are passed to the component via HTML attributes.
- Props stand for a property.
- Special React keyword for property used to pass data from one component to another.
- Props are variables or objects.Arguments passed to the React component.
- Props are passed to the component via HTML attributes.
- Props stand for a property.
- Special React keyword for property used to pass data from one component to another.
- Props are variables or objects.

```
import React, { Component }
from "react";
import ReactDOM from
"react-dom";

class App extends
React.Component {
render() {
return (
<div>
{" "}
<h2>welcome to
{this.props.name} </h2>
<p>Unacademy is the largest
edtech company </p>
</div>
);
}
}

export default App;
```

What is Prop Drilling

- Prop drilling is mainly where the same data is sent to most levels due to end level requests.
- Pass data from the parent component to the correct child component

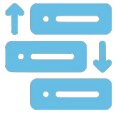
```
import "./styles.css";
import React, { useState } from "react";
const GrandChild = (props) => {
return (
<div>
<h3>Grandchild: </h3>
<Child brand={props.brand} />
</div>
);
};
const Child = (props) => {
return (
```

```
<div>
<h2>Child: {props.brand}</h2>
</div>
);
};
const App = () => {
const [brandname] = useState("Amazon");
return (
<div>
<h1>Hello</h1>
<GrandChild brand={brandname} />
</div>
);
};
export default App;
```

How to avoid Prop Drilling

There are numerous ways to avoid Props Drilling, such as

React Context API



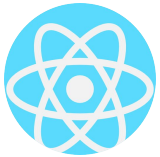
Composition

Redux or MobX

HOC

Render Props

React Context API



- A React application can efficiently create global variables that can be passed.
- It is an alternative to "drill prop" or move props from grandparent to child to parent and so on.
- Context is mainly used when certain data needs to be accessed by multiple components at various nesting levels.
- If you just want to avoid passing certain accessories on some level, aligning components is often an easier solution than context.

How to use Context

1. setup a context provider & define the data you want to store

2. use a context consumer where ever you need the data from the store in 2 types of components:

Example

```
import React, { Fragment } from "react";
import Provider from "./provider";
import Context from "./context";

const Agents = () => {
return <AgentOne />;
};

const AgentOne = () => {
return <AgentTwo />;
};

const AgentTwo = () => {
return <AgentBond />;
};

const AgentBond = () => {
return (
```

```
const AgentBond = () => {
return (
<Context.Consumer>
{(context) => (
<Fragment>
<h3>Agent Information</h3>


<p>Mission Name: {context.data.mname}</p>
<h2>Mission Status:
{context.data.accept}</h2>
<button
onClick={context.isMissionAccepted}>Choos
e to accept</button>
</Fragment>
)}
```

```
</Context.Consumer>
);
};

const App = () => {
return (
<div>
<h1>Context API</h1>
<Provider>
<Agents />
</Provider>
</div>
);
};

export default App;
```

Practice Problem

- Create a ToDo App using Context API.

Thank you