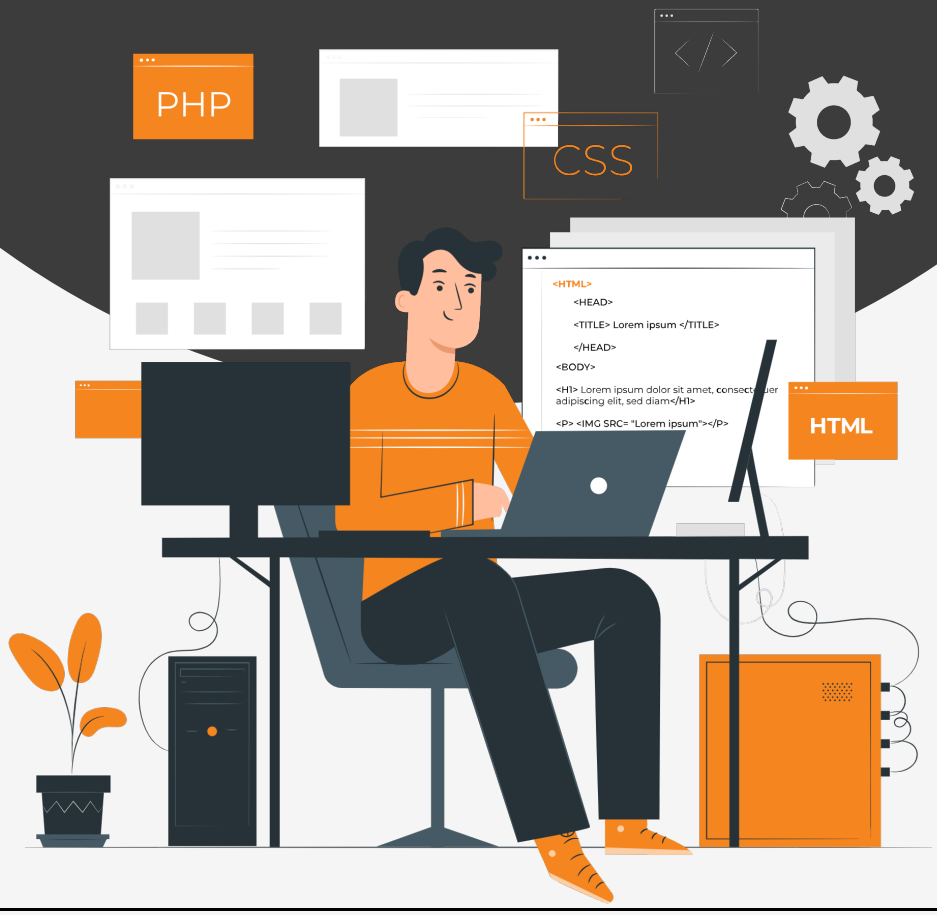
Lesson:

Image Component



PW SKILLS

The Image component in React Native is used to display images on the screen. It allows you to load and render local or remote images in your application.

Here are a few examples of using the Image component:

Example 1: Local Image

The Switch component in React Native is a simple UI control that allows users to toggle between two states. It's visually analogous to a physical switch or a light toggle and serves as an alternative to checkboxes or buttons for binary choices. With React Native, the Switch is rendered natively on both Android and iOS, providing a familiar look and feel for users across platforms.

```
import React from 'react';
import { View, Image, StyleSheet } from 'react-native';

const App = () => {
  return (
    <View style={styles.container}>
      {/*Add any image from your local */}
      <Image source={require('./assets/image.jpg')} style={styles.image} />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  image: {
    width: 200,
    height: 200,
  },
});

export default App;
```

In this example, we have a parent View component called container which is styled to center its content. Inside the container, there's an Image component displaying a local image. The source prop of the Image component takes the path to the local image file using the require function. The style prop is used to define the dimensions of the image.

Example 2: Remote Image

```
import React from 'react';
import { View, Image, StyleSheet } from 'react-native';

const App = () => {
  return (
    <View style={styles.container}>
      <Image
        source={{ uri: 'https://picsum.photos/200' }}
        style={styles.image}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  image: {
    width: 400,
    height: 400,
  },
});

export default App;
```

In this example, we have a parent View component called container which is styled to center its content. Inside the container, there's an Image component displaying a remote image. The source prop of the Image component takes an object with the uri property pointing to the remote image URL. The style prop is used to define the dimensions of the image.

Example 3: Image with Placeholder

```
import React from 'react';
import { View, Image, StyleSheet } from 'react-native';

const App = () => {
  return (
    <View style={styles.container}>
      <Image
        source={require('./images/avatar.jpg')}
        style={styles.image}
        resizeMode="cover"
        defaultSource={require('./images/placeholder.png')}
        loadingIndicatorSource={require('./images/loading.gif')}
      />
    </View>
  );
};

```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  image: {
    width: 200,
    height: 200,
  },
});

export default App;
```

In this example, we have a parent View component called container which is styled to center its content. Inside the container, there's an Image component displaying a local image. The source prop of the Image component takes the path to the local image file using the require function. The style prop is used to define the dimensions of the image. Additionally, we use the resizeMode prop to control how the image is resized within its container. The defaultSource prop provides a placeholder image that is displayed while the actual image is loading. The loadingIndicatorSource prop specifies a loading indicator