Camera

expo-camera provides a React component that renders a preview for the device's front or back camera. The camera's parameters like zoom, auto focus, white balance and flash mode are adjustable. With the use of Camera, one can also take photos and record videos that are then saved to the app's cache. Morever, the component is also capable of detecting faces and bar codes appearing in the preview. Run the example on your device to see all these features working together!

Installation

Terminal
Copy
```
→ npx expo install expo-camera
```

If you're installing this in a bare React Native app, you should also follow these additional installation instructions.

Usage

Only one Camera preview can be active at any given time. If you have multiple screens in your app, you should unmount Camera components whenever a screen is unfocused.

Basic Camera usage

Open in Snack

```
import { Camera, CameraType } from 'expo-camera';
import { useState } from 'react';
import { Button, StyleSheet, Text, TouchableOpacity, View } from 'react-native';

export default function App() {
  const [type, setType] = useState(CameraType.back);
  const [permission, requestPermission] = Camera.useCameraPermissions();

  if (!permission) ...

  if (!permission.granted) ...

  function toggleCameraType() {
    setType(current => (current === CameraType.back ? CameraType.front :
CameraType.back));
  }

  return (
    <View style={styles.container}>
      <Camera style={styles.camera} type={type}>
        <View style={styles.buttonContainer}>
          <TouchableOpacity style={styles.button} onPress={toggleCameraType}>
            <Text style={styles.text}>Flip Camera</Text>
          </TouchableOpacity>
```

```
        </View>
      </Camera>
    </View>
  );
}

const styles = StyleSheet.create({ ... });
```

Web Support

Luckily most browsers support at least some form of web camera functionality, you can check out the web camera browser support here. Image URIs are always returned as base64 strings because local file system paths are not available in the browser.

Chrome iframe usage

When using Chrome versions 64+, if you try to use a web camera in a cross-origin iframe nothing will render. To add support for cameras in your iframe simply add the attribute `allow="microphone; camera;"` to the iframe element:

```
<iframe src="..." allow="microphone; camera;">
  <!-- <Camera /> -->
</iframe>
```

API

```
import { Camera } from 'expo-camera';
```

```
import { Camera, CameraType } from 'expo-camera';

import { useState } from 'react';

import { Button, StyleSheet, Text, TouchableOpacity, View, } from 'react-native';

export default function App() {

  const [type, setType] = useState(CameraType.back);

  const [permission, requestPermission] = Camera.useCameraPermissions();

  const [offcamera,setoncamera] = useState(Camera.Constants.FlashMode.off);

  if (!permission) {

    return <View />;

  }

  if (!permission.granted) {

    return (

      <View style={styles.container}>

        <Text style={{ textAlign: 'center' }}>We need your permission to show the camera</Text>

        <Button onPress={requestPermission} title="grant permission" />

      </View>

    );

  }

  function toggleCameraType() {

    setType(current => (current === CameraType.back ? CameraType.front : CameraType.back));

  }

  function Flashing()

  {

      setoncamera(currently=> Camera.Constants.FlashMode.torch)

  }

  function Flashing1()

  {

      setoncamera(currently=> Camera.Constants.FlashMode.off)

  }
```

```jsx
  return (
    <View style={styles.container}>
      <Camera style={styles.camera} type={type} flashMode = {offcamera}>
        <View style={styles.buttonContainer}>
          <TouchableOpacity style={styles.button} onPress={toggleCameraType}>
            <Text style={styles.text}>Flip Camera</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.button} onPress = {Flashing} >
            <Text style={styles.text}>Flash Open</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.button} onPress = {Flashing1} >
            <Text style={styles.text}>Flash Close</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.button} onPress = {current=>current = permission.granted ?
permission.denied:alert("bye")} >
            <Text style={styles.text}>Close Camera</Text>
          </TouchableOpacity>
        </View>
      </Camera>
    </View>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center'
  },
  camera: {
```

```
   width: 500,

   height: 600

 },

 buttonContainer: {

  flex: 1,

  flexDirection: 'row',

  backgroundColor: 'transparent',

  margin: 64,

 },

 button: {

  flex: 1,

  alignSelf: 'flex-end',

  alignItems: 'center',

 },

 text: {

  fontSize: 24,

  fontWeight: 'bold',

  color: 'white',

 },

});
```

Home Task

Make following app which consist of camera, flash, torch, focus, video, video start, resume and stop.

Use different icons for each properties. Example shown below