**COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS.**

***Lab task-mobile development***

Camera App react native

***Dated:*** *20t of November 2023*

***Prepared For:*** Sir Jawad

***Prepared By:***

Laiba binta tahir FA21-BSE-019-5A

import React, { useState, useRef } from 'react';

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

import { Camera } from 'expo-camera';

import { MediaLibrary } from 'expo-media-library';

import { MaterialIcons, FontAwesome, Ionicons } from '@expo/vector-icons';

export default function App() {

  const [cameraType, setCameraType] = useState(Camera.Constants.Type.back);

  const [flashMode, setFlashMode] = useState(Camera.Constants.FlashMode.off);

  const [isRecording, setRecording] = useState(false);

  const cameraRef = useRef(null);

  const toggleCameraType = () => {

    setCameraType(

      cameraType === Camera.Constants.Type.back

        ? Camera.Constants.Type.front

        : Camera.Constants.Type.back

    );

  };

  const toggleFlashMode = () => {

    setFlashMode(

      flashMode === Camera.Constants.FlashMode.off

        ? Camera.Constants.FlashMode.torch

        : Camera.Constants.FlashMode.off

    );

  };

  const toggleRecording = async () => {

    if (isRecording) {

      cameraRef.current.stopRecording();

      setRecording(false);

    } else {

      const { uri } = await cameraRef.current.recordAsync();

      await MediaLibrary.createAssetAsync(uri);

      setRecording(true);

    }

  };

  return (

    <View style={styles.container}>

      <Camera

        style={styles.camera}

        type={cameraType}

        flashMode={flashMode}

        ref={cameraRef}

      >

        <View style={styles.overlay}>

          <TouchableOpacity onPress={toggleCameraType} style={styles.iconButton}>

            <Ionicons name="ios-reverse-camera" size={32} color="white" />

          </TouchableOpacity>

          <TouchableOpacity onPress={toggleFlashMode} style={styles.iconButton}>

            {flashMode === Camera.Constants.FlashMode.off ? (

              <FontAwesome name="flash" size={32} color="white" />

            ) : (

              <MaterialIcons name="highlight" size={32} color="white" />

            )}

          </TouchableOpacity>

          <TouchableOpacity style={styles.iconButton}>

            {*/\* Placeholder for HDR icon \*/*}

            <Ionicons name="ios-contrast" size={32} color="white" />

          </TouchableOpacity>

          <TouchableOpacity style={styles.iconButton}>

            {*/\* Placeholder for Settings icon \*/*}

            <Ionicons name="ios-settings" size={32} color="white" />

          </TouchableOpacity>

          <TouchableOpacity onPress={toggleRecording} style={styles.iconButton}>

            {isRecording ? (

              <FontAwesome name="stop" size={32} color="red" />

            ) : (

              <MaterialIcons name="fiber-manual-record" size={32} color="white" />

            )}

          </TouchableOpacity>

        </View>

      </Camera>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    justifyContent: 'center',

  },

  camera: {

    flex: 1,

  },

  overlay: {

    flex: 1,

    flexDirection: 'row',

    justifyContent: 'space-between',

    alignItems: 'flex-end',

    padding: 16,

  },

  iconButton: {

    padding: 10,

  },

});

OUTPUT:

