IN APP QUIZ AS A TEACHING TOOL FOR COELIAC DISEASE

Technical Document

Student: James Nolan – C00226267

Supervisor: Joseph Kehoe

Date: 20/04/2021

Table of Contents

Abstract

The purpose of this project is that it is part of a mobile application to assist people with Coeliac Disease in adhering to a gluten free lifestyle.

This part of the application will consist of an engaging quiz that incorporates gamification techniques such as a leader board and statistics to help patients increase their diet compliance.

Introduction

This is a technical manual for the fourth-year student project ‘In app quiz as a teaching tool for Coeliac Disease’, or CeliQuiz for short. It is a mobile application designed for both iOS and Android with the aim of helping its users learn more about Coeliac Disease through gamification techniques.

The project was built using React Native, an open-source framework used for creating native cross-platform applications for iOS and Android, developed by Facebook. Also used is Expo, an open-source platform for making universal native apps for Android, iOS, and the web with JavaScript and React. The backend is primarily Firebase, but there is also a Flask application for storing the questions.

Running the Application

Dependencies

- React Native – 0.63.2
- Expo – 41.0.1

Back-end

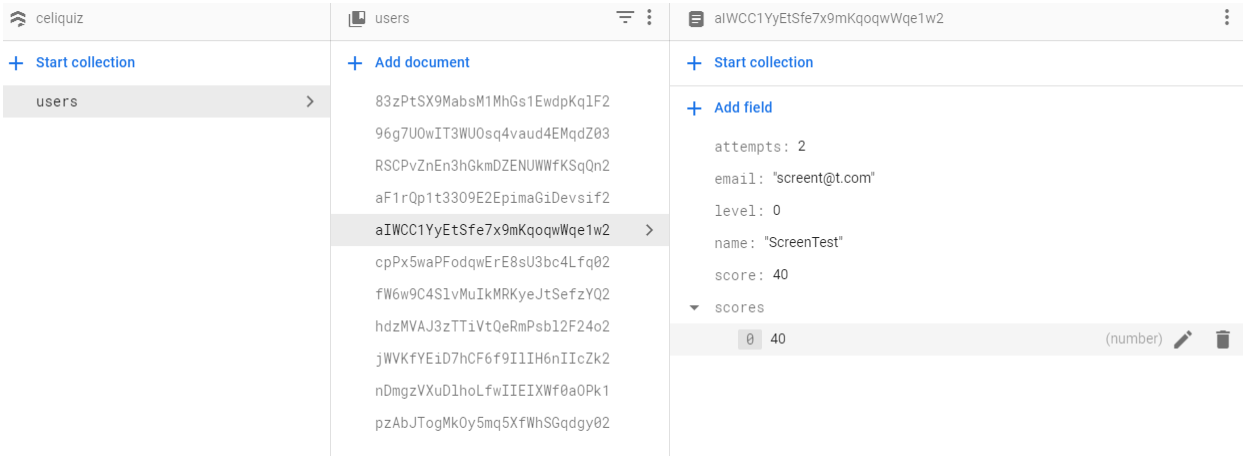
There is no need to install anything as it is all running from Firebase and pythonanywhere.com.

Authentication

Firebase authentication is used in this project to both obtain and create authenticated credentials for a user. As below shows, a user registers through an email and password combination. The user's credentials are passed to the Firebase Authentication SDK. Credentials are generated if the user is signing up for the first time. When they log in, the backend service will verify those credentials and return a response to the application.

Firestore

Cloud Firestore is a NoSQL database that stores the data used in this project. It is a scalable database and offers the ability to set up real-time listeners within the code to keep app data in sync. 6 The database stores information in collections, these collections contain documents which contain data. As below shows, when a user registers, they automatically have a document created that stores more information than Authentication can.



Flask Application

In the Flask application, the questions are stored in json format so that they can be easily pulled to the app.

```
import json
from flask import Flask, jsonify

app = Flask(__name__)
app.config["DEBUG"] = True

@app.route('/')
def index():
    return jsonify({"response_code":0,"results":[{"type":"multiple","question":" People with Coeliac disease cannot eat which kind of food?","correct_answer":"Grains","incorrect_answers":["Nuts","Dairy","Fruits"
    {"type":"multiple","question":"How is Coeliac disease treated?","correct_answer":"A change in diet","incorrect_answers":["Antibiotics","Surgery","Sleep"]},
    {"type":"multiple","question":"How common is Coeliac disease in the EU?","correct_answer":"1 in 100","incorrect_answers":["1 in 100,000","1 in 2","1 in 1000"]},
    {"type":"boolean","question":"GIP tests detect dietary problems in treated celiac patients.","correct_answer":"True","incorrect_answers":["False"]},
    {"type":"multiple","question":"Gluten is:","correct_answer":"A protein found naturally in certain foods","incorrect_answers":["An additive to foods","A preservative","A flavoring"]},
    {"type":"multiple","question":"Eating gluten-free is critical if you:","correct_answer":"Have Coeliac Disease","incorrect_answers":["Want to lose weight","Want to sleep more","Want More Energy"]},
    {"type":"boolean","question":"Can you tell which prepared foods have gluten just by looking at them?","correct_answer":"False","incorrect_answers":["False"]},
    {"type":"multiple","question":"Which is a good gluten-free breakfast choice?","correct_answer":"Scrambled Eggs","incorrect_answers":["Toast","Bagel","Pancakes"]},
    {"type":"boolean","question":"A person with celiac disease can't eat oats.","correct_answer":"True","incorrect_answers":["False"]},
    {"type":"boolean","question":"Food manufacturers are required to list gluten on their labels","correct_answer":"False","incorrect_answers":["True"]},
    {"type":"boolean","question":"Some fruit and vegetables contain gluten.","correct_answer":"False","incorrect_answers":["True"]},
    {"type":"multiple","question":"Which of these is a sympton of gluten ingestion?","correct_answer":"Indigestion","incorrect_answers":["Fever","Jaundice","Headache"]},
    {"type":"boolean","question":"Women are 3 times more likely to develop Coeliac disease.","correct_answer":"True","incorrect_answers":["False"]},
    {"type":"multiple","question":"Coeliac Disease is","correct_answer":"An autoimmune disease","incorrect_answers":["A virus","A bug","A lifestyle"]}]})

if __name__ == '__main__':
    app.run()
```

Front-end

To run the frontend open Command Prompt and change directory to the project.

```
C:\Users\user>cd quizceli

C:\Users\user\quizceli>
```

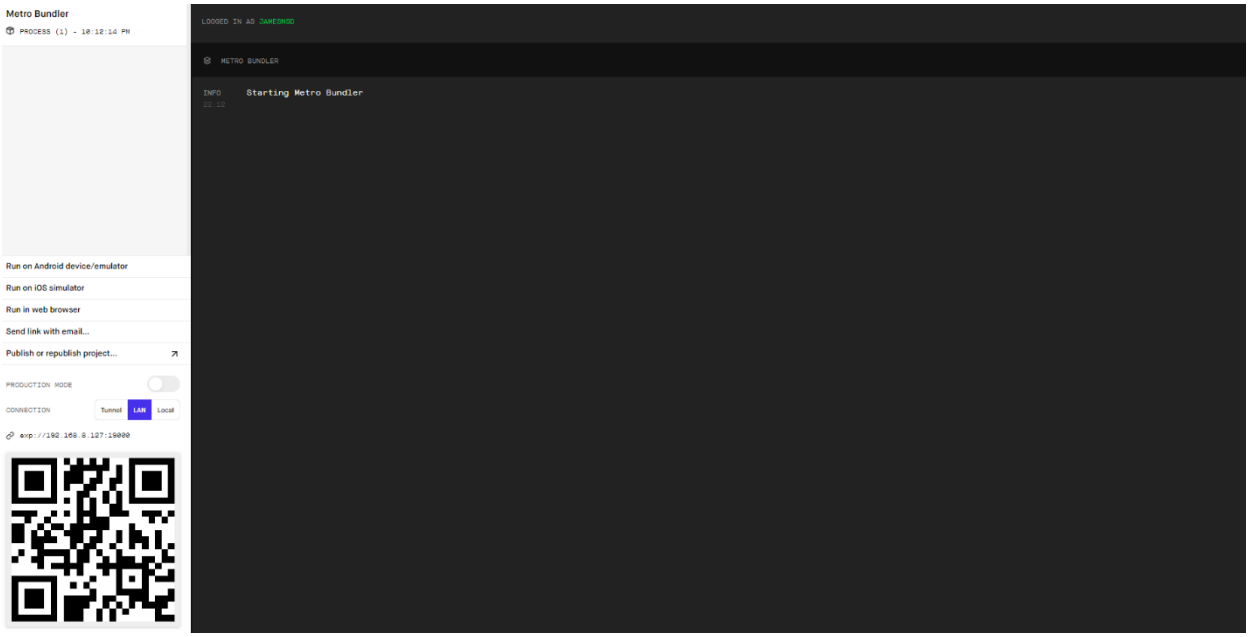
Next type yarn add expo to install Expo packages.

```
C:\Users\user\quizceli>yarn add expo
```

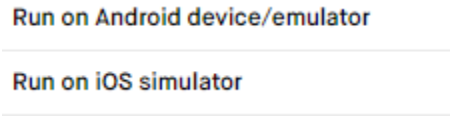
Next type expo to start to launch the application.

```
C:\Users\user\quizceli>expo start
```

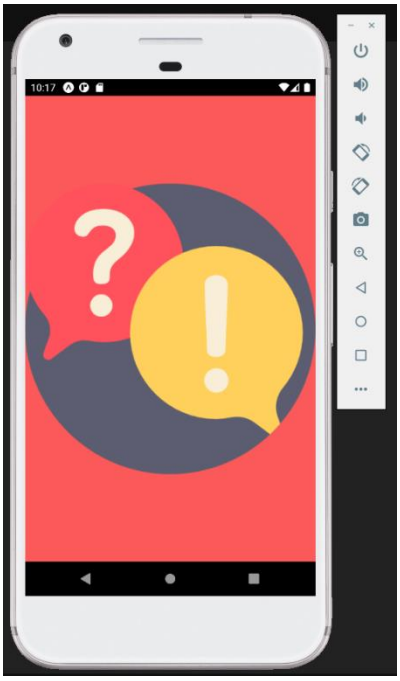
A local version of the application will automatically run in your browser.



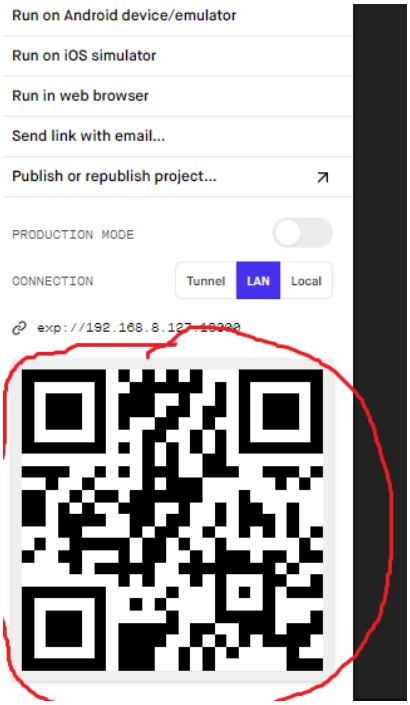
To run on an Android or iOS simulator select ‘Run on Android device/emulator, or ‘Run on iOS simulator’.



This will launch the application on your opened simulator.



To run on a physical Android, download the Expo app from the Play Store and scan the QR code.



Back-End Code (Flask Application)

```
import json

from flask import Flask, jsonify


app = Flask(__name__)

app.config["DEBUG"] = True


@app.route('/')

def index():

    return jsonify({"response_code":0,"results":[{"type":"multiple","question":" People with Coeliac disease cannot eat which kind of food?","correct_answer":"Grains","incorrect_answers":["Nuts","Dairy","Fruits"]},

    {"type":"multiple","question":"How is Coeliac disease treated?","correct_answer":"A change in diet","incorrect_answers":["Antibiotics","Surgery","Sleep"]},

    {"type":"multiple","question":"How common is Coeliac disease in the EU?","correct_answer":"1 in 100","incorrect_answers":["1 in 100,000","1 in 2","1 in 1000"]},

    {"type":"boolean","question":"GIP tests detect dietary problems in treated celiac patients.","correct_answer":"True","incorrect_answers":["False"]},

    {"type":"multiple","question":"Gluten is:","correct_answer":"A protein found naturally in certain foods","incorrect_answers":["An additive to foods","A preservative","A flavoring"]},

    {"type":"multiple","question":"Eating gluten-free is critical if you:","correct_answer":"Have Coeliac Disease","incorrect_answers":["Want to lose weight","Want to sleep more","Want More Energy"]},

    {"type":"boolean","question":"Can you tell which prepared foods have gluten just by looking at them?","correct_answer":"False","incorrect_answers":["False"]},
```

{"type":"multiple","question":"Which is a good gluten-free breakfast choice?","correct_answer":"Scrambled Eggs","incorrect_answers":["Toast","Bagel","Pancakes"]},

{"type":"boolean","question":"A person with celiac disease can’t eat oats.","correct_answer":"True","incorrect_answers":["False"]},

{"type":"boolean","question":"Food manufacturers are required to list gluten on their labels","correct_answer":"False","incorrect_answers":["True"]},

{"type":"boolean","question":"Some fruit and vegetables contain gluten.","correct_answer":"False","incorrect_answers":["True"]},

{"type":"multiple","question":"Which of these is a sympton of gluten ingestion?","correct_answer":"Indigestion","incorrect_answers":["Fever","Jaundice","Headache"]},

{"type":"boolean","question":"Women are 3 times more likely to develop Coeliac disease.","correct_answer":"True","incorrect_answers":["False"]},

{"type":"multiple","question":"Coeliac Disease is","correct_answer":"An autoimmune disease","incorrect_answers":["A virus","A bug","A lifestyle"]}]})

```
if __name__ == '__main__':
    app.run()
```

Front-End Code

App.js

```
import React from 'react'
import  AppLoading from 'expo-app-loading';
import MyStack from './routes/routes.js';

  export default class App extends React.Component{

      render(){

        return (<MyStack/>);
        }
    }
```

app.json

```
{
  "expo": {
    "name": "quizceli",
    "slug": "quizceli",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "splash": {
      "image": "./assets/question.png",
      "resizeMode": "contain",
      "backgroundColor": "#fb5b5a"
    },
    "updates": {
      "fallbackToCacheTimeout": 0
    },
    "assetBundlePatterns": [
      "**/*"
    ],
    "ios": {
      "supportsTablet": true
    },
    "android": {
      "adaptiveIcon": {
        "foregroundImage": "./assets/adaptive-icon.png",
        "backgroundColor": "#FFFFFF"
      }
    },
    "web": {
      "favicon": "./assets/favicon.png"
    }
  }
}
```

Firebase.js

```
import { firebase } from '@firebase/app';
require('firebase/auth')

import firestore from 'firebase/firestore'

const settings = {timestampsInSnapshots: true};
//Firebase configuration details for app to communicate with Firebase
const config = {
    apiKey: "AIzaSyDXTJMAoN4tYV8456hnG611t1cgjndKuSw",
    authDomain: "celiquiz.firebaseapp.com",
    projectId: "celiquiz",
    storageBucket: "celiquiz.appspot.com",
    messagingSenderId: "226340250977",
    appId: "1:226340250977:web:8f0e3100f4c047517f9120",
    measurementId: "G-BP8NPDEYSG"
};
firebase.initializeApp(config);

firebase.firestore().settings(settings);

export default firebase;
```

components/Questions.js

```
import React from "react";
import { View, Text, StyleSheet, Button, TouchableOpacity, StatusBar } from "react-
native";
import { RadioGroup, RadioButton } from "react-native-flexi-radio-button";

export default class Question extends React.Component {
  constructor() {
    super();

    this.state = {
      answer: null
    };
  }
  //Radio Buttons for question type
  renderOptions = question => {
    if (question.type === "boolean") {
      return [
        <RadioButton value={"True"} key={1}>
          <Text style={styles.radioText}>True</Text>
        </RadioButton>,

        <RadioButton value={"False"} key={2}>
          <Text style={styles.radioText}>False</Text>
        </RadioButton>
      ];
    } else {
      const result = [];

      question.incorrect_answers.forEach((item, index) => {
        let key = `${question.id}-${index}`;

        if (index === this.props.correctPosition) {
          let key2 = `${question.id}-10`;
          result.push(
            <RadioButton value={question.correct_answer} key={key2}>
              <Text style={styles.radioText}>{question.correct_answer}</Text>
            </RadioButton>
          );
        }

        result.push(
          <RadioButton value={item} key={key}>
            <Text style={styles.radioText}>{item}</Text>
          </RadioButton>
        );
      });

      return result;
    }
  };

  render() {
    return (
```

```
      <View style={styles.container}>
        <Text style={{ fontStyle: 'italic', color:"#5c5e70", fontSize: 30,
fontWeight: "bold" }}>
          {this.props.question.question}
        </Text>
        <RadioGroup
          onSelect={(index, answer) => this.setState({ answer })}
          selectedIndex={null}
        >
          {this.renderOptions(this.props.question)}
        </RadioGroup>
        <TouchableOpacity
            style={styles.button}
            onPress={() => this.props.onSelect(this.state.answer)}
         >
          <View style={{ display: "flex", flexDirection: "row" }}>
            <Text
              style={{
                color: "white",
                fontWeight: "bold",
                fontSize: 16
              }}
            >
              Submit
            </Text>
          </View>
        </TouchableOpacity>

      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
    padding: 40,
    height: '100%',
    width: '100%'
  },
  radioText: {
    fontSize: 20,
    color: "#5c5e70"
  },
  button: {
    backgroundColor:"#fb5b5a",
    alignItems: "center",
    padding: 10,
    borderRadius: 25,
    marginLeft: "auto",
    marginRight: "auto",
    width:"80%",
    height:50,
    justifyContent:"center"}});
```

routes/routes.js

```
import React from 'react'

import { createStackNavigator } from '@react-navigation/stack'
import { NavigationContainer } from '@react-navigation/native'
//importing class names from screens
import SignUp from '../screens/SignUp.js'
import Login from '../screens/Login.js'
import Main from '../screens/Main.js'
import QuizMain from '../screens/QuizMain.js'
import Settings from '../screens/Settings.js'
import TakeQuiz from '../screens/QuestionScreen.js'
import HighScore from '../screens/HighScore.js'
import WelcomeScreen from '../screens/WelcomeScreen.js'
import Stats from '../screens/Stats.js'
import ForgotPassword from '../screens/forgotPassword.js'

const Stack = createStackNavigator();
//creating navigation stack
function MyStack() {
    return (
      <NavigationContainer>
        <Stack.Navigator>
          <Stack.Screen name="Login" component={Login} options={{ headerShown: false
}}/>
          <Stack.Screen name="SignUp" component={SignUp} options={{ headerShown:
false }}/>
          <Stack.Screen name="QuizMain" component={QuizMain} options={{ headerShown:
false }}/>
          <Stack.Screen name="Main" component={Main} options={{ headerShown: false
}}/>
          <Stack.Screen name="Settings" component={Settings} options={{ headerShown:
false }}/>
          <Stack.Screen name="TakeQuiz" component={TakeQuiz} options={{ headerShown:
false }}/>
          <Stack.Screen name="HighScore" component={HighScore} options={{
headerShown: false }}/>
          <Stack.Screen name="WelcomeScreen" component={WelcomeScreen} options={{
headerShown: false }}/>
          <Stack.Screen name="Stats" component={Stats} options={{ headerShown: false
}}/>
          <Stack.Screen name="ForgotPassword" component={ForgotPassword} options={{
headerShown: false }}/>
        </Stack.Navigator>
      </NavigationContainer>
    );
  }

  export default MyStack;
```

screens/forgotPassword.js

```
import React from 'react';
import { StyleSheet, Text, View, TextInput, TouchableOpacity, StatusBar } from
'react-native';
import firebase from '../Firebase';
import iid from '@react-native-firebase/iid';

export default class ForgotPassword extends React.Component {

  state = { email: ''}
//function to send email to entered by user
  Reset = async() => {
      try {
          await auth().sendPasswordResetEmail(email);
      } catch (e) {
          console.log(e);
      }
  };

  render(){
    return (
      <View style={styles.container}>
        <StatusBar backgroundColor="black"/>
        <Text style={styles.logo}>CeliQuiz</Text>
        <View style={styles.inputView} >
          <TextInput
            style={styles.inputText}
            placeholder="Enter Your Email..."
            placeholderTextColor="white"
            onChangeText={text => this.setState({email:text})}/>
        </View>
        //onPress the email entered is ran to the function and sent
        <TouchableOpacity onPress={() => this.Reset(this.state.email)}
style={styles.loginBtn}>
          <Text style={styles.loginText}>Reset Password</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.SignUpBtn}  onPress={() =>
this.props.navigation.navigate('Login')}>
          <Text style={styles.loginText}>
            Return To Log In
          </Text>
        </TouchableOpacity>

      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
    alignItems: 'center',
    justifyContent: 'center',
```

```
},
logo:{
  fontStyle: 'italic',
  fontWeight:"bold",
  fontSize:45,
  color:"#fb5b5a",
  marginBottom:110
},
inputView:{
  width:"80%",
  backgroundColor:"#5c5e70",
  borderRadius:25,
  height:50,
  marginBottom:20,
  justifyContent:"center",
  padding:20
},
inputText:{
  height:50,
  color:"white"
},
loginBtn:{
  width:"80%",
  backgroundColor:"#fb5b5a",
  borderRadius:25,
  height:50,
  alignItems:"center",
  justifyContent:"center",
  marginTop:70,
  marginBottom:10,

},
loginText:{
  color:"white",
},

SignUpBtn:{
  width:"80%",
  backgroundColor:"#669999",
  borderRadius:25,
  height:50,
  alignItems:"center",
  justifyContent:"center",
  marginTop:3
},
PwordBtn:{
    width:"80%",
    backgroundColor:"black",
    borderRadius:25,
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:5
  }});
```

screens/Login.js

```
import React from 'react';
import { StyleSheet, Text, View, TextInput, TouchableOpacity, StatusBar } from
'react-native';
import firebase from '../Firebase';
import iid from '@react-native-firebase/iid';

export default class Login extends React.Component {

  state = { email: '', password: '' }
  //function for logging in
  Login  = (email, password) => {
      firebase
         .auth()
         .signInWithEmailAndPassword(email, password)
         .then((s)=> {
         //onPress user is logged in and brought to main menu
            this.props.navigation.navigate('Main');
         })
         .catch(function(error) {
           alert(error.message);
         });
  };

  render(){
    return (
      <View style={styles.container}>
        <StatusBar backgroundColor="black"/>
        <Text style={styles.logo}>CeliQuiz</Text>
        <View style={styles.inputView} >
          <TextInput
            style={styles.inputText}
            placeholder="Enter Your Email..."
            placeholderTextColor="white"
            onChangeText={text => this.setState({email:text})}/>
        </View>
        <View style={styles.inputView} >
          <TextInput
            secureTextEntry
            style={styles.inputText}
            placeholder="Enter Your Password..."
            placeholderTextColor="white" //#003f5c
            onChangeText={text => this.setState({password:text})}/>
        </View>

        <TouchableOpacity onPress={() => this.Login(this.state.email,
this.state.password)} style={styles.loginBtn}>
          <Text style={styles.loginText}>Login</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.SignUpBtn}  onPress={() =>
this.props.navigation.navigate('SignUp')}>
          <Text style={styles.loginText}>
            Sign Up
          </Text>
```

```
        </TouchableOpacity>

        <View style={{flexDirection:'row',alignItems:'space-around',marginTop:10}}>
            <Text style={{color:'black'}}>Forgot Password?   </Text>
            <TouchableOpacity onPress={() =>
this.props.navigation.navigate('ForgotPassword')}>
                <Text style={{color:'black',fontWeight:'bold',fontSize:15}}>Click
here</Text>
            </TouchableOpacity>
        </View>

      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
    alignItems: 'center',
    justifyContent: 'center',
  },
  logo:{
    fontStyle: 'italic',
    fontWeight:"bold",
    fontSize:45,
    color:"#fb5b5a",
    marginBottom:110
  },
  inputView:{
    width:"80%",
    backgroundColor:"#5c5e70",
    borderRadius:25,
    height:50,
    marginBottom:20,
    justifyContent:"center",
    padding:20
  },
  inputText:{
    height:50,
    color:"white"
  },
  loginBtn:{
    width:"80%",
    backgroundColor:"#fb5b5a",
    borderRadius:25,
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:70,
    marginBottom:10,

  },
  loginText:{
```

```
    color:"white",
  },

  SignUpBtn:{
    width:"80%",
    backgroundColor:"#669999",
    borderRadius:25,
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:3
  },
  PwordBtn:{
      width:"80%",
      backgroundColor:"black",
      borderRadius:25,
      height:50,
      alignItems:"center",
      justifyContent:"center",
      marginTop:5
    }

});
```

screens/SignUp.js

```
import React from 'react'
import { StyleSheet, Text, TextInput, View, Button, TouchableOpacity, StatusBar }
from 'react-native'
import firebase from '../Firebase';
import firestore from '../Firebase';

export default class SignUp extends React.Component {

  constructor(){
    super();
  }
  //seet variables for app state
  state = { name: '', email: '', password: '', score: 0, level: 0, attempts: 0}
  //function to register user and bring to main screen
  Register = (email, password) => {
      firebase.auth().createUserWithEmailAndPassword(this.state.email,
this.state.password)
          .then((userCredentials)=>{
              if(userCredentials.user){
                userCredentials.user.updateProfile({
                //set display name
                  displayName: this.state.name
                })
                .then(() => {
                      //Once the user creation has happened successfully, the useris
added into firestore
                      //with the appropriate details.

firebase.firestore().collection('users').doc(firebase.auth().currentUser.uid)
                      .set({
                          name: this.state.name,
                          email: this.state.email,
                          score: this.state.score,
                          level: this.state.level,
                          attempts: this.state.attempts,
                      })
                      //catch any errors
                      .catch(error => {
                          console.log('Something went wrong with added user to
firestore: ', error);
                      })
                })
                .then((s)=> {
                  this.props.navigation.navigate('WelcomeScreen');
                })
              }
          })
          .catch(function(error) {
            alert(error.message);
          });
  };

  render(){
```

```
    return (
      <View style={styles.container}>
        <StatusBar backgroundColor="black"/>
        <Text style={styles.logo}>CeliQuiz</Text>

        <View style={styles.inputView} >
          <TextInput
            style={styles.inputText}
            placeholder="Enter Name"
            placeholderTextColor="white"
            onChangeText={text => this.setState({name:text})}/>
        </View>

        <View style={styles.inputView} >
          <TextInput
            style={styles.inputText}
            placeholder="Enter Email"
            placeholderTextColor="white"
            onChangeText={text => this.setState({email:text})}/>
        </View>

        <View style={styles.inputView} >
          <TextInput
            secureTextEntry
            style={styles.inputText}
            placeholder="Enter Password"
            placeholderTextColor="white"
            onChangeText={text => this.setState({password:text})}/>
        </View>

        <TouchableOpacity onPress={() => this.Register(this.state.email,
this.state.password)} style={styles.registerBtn}>
          <Text style={styles.registerText}>Register</Text>
        </TouchableOpacity>


        <View style={{flexDirection:'row',alignItems:'space-around',marginTop:10}}>
          <Text style={{color:'black'}}>Already have an account?   </Text>
          <TouchableOpacity onPress={() => this.props.navigation.navigate('Login')}>
            <Text style={{color:'#fb5b5a',fontWeight:'bold',fontSize:15}}>Login
here</Text>
          </TouchableOpacity>
        </View>

      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
    alignItems: 'center',
    justifyContent: 'center',
```

```
  },
  logo:{
    fontStyle: 'italic',
    fontWeight:"bold",
    fontSize:45,
    color:"#fb5b5a",
    marginBottom:110
  },
  inputView:{
    width:"80%",
    backgroundColor:"#5c5e70",
    borderRadius:25,
    height:50,
    marginBottom:20,
    justifyContent:"center",
    padding:20
  },
  inputText:{
    height:50,
    color:"white"
  },
  registerBtn:{
    width:"80%",
    backgroundColor:"#fb5b5a",
    borderRadius:25,
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:50,
    marginBottom:10
  },
  registerText:{
    color:"white"
  },

});
```

screens/WelcomeScreen.js

```
import React, {Component} from 'react';
import {
  View,
  Text,
  Image,
  StyleSheet,
  StatusBar,
  ImageBackground,
} from 'react-native';
import AppIntroSlider from 'react-native-app-intro-slider';

const slides = [
  {
    key: 1,
    title: 'Hi!',
    text: 'Welcome to CeliQuiz.',
  },
  {
    key: 2,
    title: 'Compete!',
    text: 'Can you make it to the top of the Leader Board?',
  },
  {
    key: 3,
    title: 'Learn!',
    text: "Take the Quiz and find out more about Coeliac Disease",
  },
];

class WelcomeScreen extends Component {
  constructor(props) {
    super(props);
    this.state = {};
  }

  _renderItem = ({item}) => {
    return (
      <View style={styles.slide}>
        <ImageBackground source={item.image} style={styles.image}>
          <Text style={styles.title}>{item.title}</Text>
          <Text style={styles.text}>{item.text}</Text>
        </ImageBackground>
      </View>
    );
  };
  _onDone = () => {
    this.props.navigation.navigate('Main');
  };

  _renderNextButton = () => {
    return (
      <View>
        <Text style={styles.next}>Next</Text>
```

```
      </View>
    );
  };
  _renderDoneButton = () => {
    return (
      <View>
        <Text style={styles.done}>Done</Text>
      </View>
    );
  };
  _renderSkipButton = () => {
    return (
      <View>
        <Text style={styles.skip}>Skip</Text>
      </View>
    );
  };
  render() {
    return (
      <View style={{flex: 1}}>
        <StatusBar
          translucent
          backgroundColor="#f5f5f5"
          barStyle="light-content"
        />
        <AppIntroSlider
          renderItem={this._renderItem}
          data={slides}
          onDone={this._onDone}
          dotStyle={styles.dots}
          activeDotStyle={styles.activeDots}
          renderDoneButton={this._renderDoneButton}
          renderNextButton={this._renderNextButton}
          renderSkipButton={this._renderSkipButton}
        />
      </View>
    );
  }
}

const styles = StyleSheet.create({
  slide: {
    flex: 1,
    flexDirection: 'column',
  },
  image: {
    flex: 1,
    resizeMode: 'cover',
    justifyContent: 'flex-end',
    paddingBottom: 100,
    backgroundColor: '#ffffff',
  },
  text: {
    color: '#5c5e70',
    textAlign: 'center',
```

```
  },
  title: {
    fontStyle: 'italic',
    fontWeight:"bold",
    fontSize: 45,
    color: '#fb5b5a',
    textAlign: 'center',
  },
  dots: {
    backgroundColor: '#5A5858',
  },
  activeDots: {
    backgroundColor: '#fb5b5a',
  },
  next: {
    fontSize: 14,
    fontWeight: '700',
    color: '#5A5858',
  },
  done: {
    fontSize: 14,
    fontWeight: '700',
    color: '#3c240c',
  },
  skip: {
    fontSize: 14,
    fontWeight: '700',
    color: '#fb5b5a',
  },
});

export default WelcomeScreen;
```

screens/Main.js

```
import React from 'react'
import { StyleSheet, Platform, Image, Text, View, TextInput, Alert, TouchableOpacity,
Dimensions, StatusBar } from 'react-native'
import * as firebase from "firebase";
import { Ionicons } from "@expo/vector-icons";

export default class Main extends React.Component {
  state = { currentUser: null }
//get current user from auth
  componentDidMount() {
    const { currentUser } = firebase.auth()
    this.setState({ currentUser })
}

render() {
    const { currentUser } = this.state
return (
        <View style={styles.container}>
                <StatusBar backgroundColor="black"/>
                //display users name when logged in
                <Text style={styles.logo}>Hi {currentUser && currentUser.displayName
}!</Text>

                <TouchableOpacity style={styles.button} onPress={() =>
this.props.navigation.navigate("QuizMain")}>
                    <Text style={{color: "white", fontWeight: "bold", fontSize: 18}}
>
                        Quiz
                    </Text>
                </TouchableOpacity>

                <TouchableOpacity style={styles.button} onPress={() =>
this.props.navigation.navigate("HighScore")}>
                    <Text style={{color: "white", fontWeight: "bold", fontSize: 18}}
>
                        LeaderBoard
                    </Text>
                </TouchableOpacity>

                <TouchableOpacity style={styles.button} onPress={() =>
this.props.navigation.navigate("Stats")}>
                    <Text style={{color: "white", fontWeight: "bold", fontSize: 18}}
>
                        Stats
                    </Text>
                </TouchableOpacity>

                <TouchableOpacity style={styles.button} onPress={() =>
this.props.navigation.navigate("Settings")}>
                    <Text style={{color: "white", fontWeight: "bold", fontSize: 18}}
>
                        Settings
                    </Text>
```

```
                </TouchableOpacity>

        </View>
    )
  }
}
const styles = StyleSheet.create({

  logo:{
    fontStyle: 'italic',
    fontWeight:"bold",
    fontSize:45,
    color:"#fb5b5a",
    marginBottom:110
  },

  container: {
    flex: 1,
    backgroundColor: "white",
    alignItems: 'center',
    justifyContent: 'center',
  },

  button: {
    width:"80%",
    backgroundColor:"#fb5b5a",
    borderRadius:25,
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:10,
    marginBottom:10,
  },

  paragraph: {
    fontSize: 16,
    color: "#777",
    textAlign: "center",
    padding: 10,
    marginTop: 20,
    lineHeight: 25
  }

})
```

screens/Settings.js

```
import React from 'react';
import { StyleSheet, Text, View, TextInput, TouchableOpacity, StatusBar } from
'react-native';
import firebase from '../Firebase';

export default class Settings extends React.Component {
//log out function
  Logout = async () => {
      try {
          await firebase.auth().signOut();
          this.props.navigation.navigate('Login');
      } catch (e) {
          console.log(e);
      }
  }

  render() {
    return (
      <View style={styles.container}>
        <StatusBar backgroundColor="black"/>
        <Text style={styles.logo}>Settings</Text>
        <TouchableOpacity onPress={() => this.Logout()} style={styles.button}>
            <Text style={styles.loginText}>Logout</Text>
        </TouchableOpacity>

        <TouchableOpacity onPress={() =>
this.props.navigation.navigate('changePasswordScreen')}>
            <Text style={styles.loginText}>Change Password</Text>
        </TouchableOpacity>

      </View>
      );
    }
  }

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
    alignItems: 'center',
    justifyContent: 'center',
  },
  logo:{
    fontStyle: 'italic',
    fontWeight:"bold",
    fontSize:45,
    color:"#fb5b5a",
    marginBottom:110
  },
  button: {
    width:"80%",
    backgroundColor:"#fb5b5a",
    borderRadius:25,
```

```
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:10,
    marginBottom:10,
  },
  loginText:{
    color:"white",
    fontWeight: "bold",
    fontSize: 18
  },
});
```

screens/QuizMain.js

```
import React from 'react';
import { StyleSheet, Text, View, TouchableOpacity, StatusBar } from 'react-native'
import { Ionicons } from "@expo/vector-icons"
import TakeQuiz from "../screens/QuestionScreen"
import * as firebase from "firebase";

export default class QuizMain extends React.Component {
  state = { currentUser: null }

  componentDidMount() {
    const { currentUser } = firebase.auth()
    this.setState({ currentUser })
}

  render() {

    return (
      <View style={styles.container}>
        <StatusBar backgroundColor="black"/>
        <Text style={styles.logo}>Test Your Celiac Knowledge!!</Text>
        <Text style={styles.paragraph}>
          This quiz will ask all the questions to see how well you know life with
Celiac Disease.
        </Text>
        <TouchableOpacity
            style={styles.button}
            onPress={() => this.props.navigation.navigate("TakeQuiz")}
         >
          <View style={{ display: "flex", flexDirection: "row" }}>
            <Ionicons name="md-play" size={32} color="white" />
            <Text
              style={{color: "white", fontWeight: "bold", marginLeft: 10, marginTop:
5}}
            >
              Start Questions
            </Text>
          </View>
        </TouchableOpacity>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
    alignItems: 'center',
    justifyContent: 'center',
  },

  logo:{
    fontStyle: 'italic',
```

```
    fontWeight:"bold",
    fontSize:45,
    color:"#fb5b5a",
    textAlign: "center",
    marginBottom:110
  },

  button: {
    width:"80%",
    backgroundColor:"#fb5b5a",
    borderRadius:25,
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:50,
    marginBottom:10
  },

  paragraph: {
    fontSize: 20,
    color: "#5c5e70",
    textAlign: "center",
    padding: 10,
    marginTop: 15,
    lineHeight: 25
  }
});
```

screens/QuestionScreen.js

```
import React from 'react';
import { View, ActivityIndicator, Picker, StyleSheet, Text, Button, TouchableOpacity,
Image, StatusBar } from 'react-native';
import Question from "../components/Question";
import firebase from '../Firebase';
import firestore from '../Firebase';

export default class TakeQuiz extends React.Component {

      constructor(props) {
        super(props);
        //set values for quiz
        this.state = {

          loading: false,
          questions: [],

          scores: 0,
          current: 0,
          attempts: 0,
          level: 0,

          correctScore: 5,
          totalScore: 50,

          results: {
            score: 0,
            correctAnswers: 0
          },
          completed: false
        };
      }
      //import questions from pythonanywhere
      fetchQuestions = async () => {
        await this.setState({ loading: true });
        const response = await fetch(
          `http://jamesnsd.pythonanywhere.com/`
        );
        const questions = await response.json();

        const { results } = questions;

        //trying to shuffle questions by id
        results.forEach(item => {
          item.id = Math.floor(Math.random() * 10000);
        });

        await this.setState({ questions: results, loading: false });
      };
      //reset function for when reset button is pressed
      reset = () => {
        this.setState(
          {
```

```
            questions: [],
            current: 0,
            scores: 0,
            attempts: 0,

            results: {
              score: 0,
              correctAnswers: 0,
            },
            completed: false
          },
          () => {
            this.fetchQuestions();
          }
        );
      };
      //checks if the answer given was correct and assigns score to the user, then
sets state to next question,
      //or if the questions have run out
      submitAnswer = (index, answer) => {
        const question = this.state.questions[index];
        const isCorrect = question.correct_answer === answer;
        const results = { ...this.state.results };

        results.score = isCorrect ? results.score + 5 : results.score;
        results.correctAnswers = isCorrect
          ? results.correctAnswers + 1
          : results.correctAnswers;

        this.setState({
          current: index + 1,
          results,
          completed: index === 9 ? true : false
        });
      };

      componentDidMount() {
        this.fetchQuestions();
      }
      //sets the users score, adds the score to their array, and increments their
attempts.
      //Happens when the user selects leaderboard.
      setUserValues= (score) => {


firebase.firestore().collection('users').doc(firebase.auth().currentUser.uid)
            .update({
              score: this.state.results.score,
              scores:
firebase.firestore.FieldValue.arrayUnion(this.state.results.score),
              attempts: firebase.firestore.FieldValue.increment(1),
              level: firebase.firestore.FieldValue.increment(0.25),
            })
            //ensure we catch any errors at this stage to advise us if something does
go wrong
```

```
            .catch(error => {
                console.log('Something went wrong: ', error);
            })
            .then((s)=> {
                alert("Score Submitted!!")
            })
          .catch(function(error) {
            alert(error.message);
          });
      };

      render() {
        return (
          //Submits answer and calls next question
          <View style={styles.container}>
            {!!this.state.questions.length > 0 &&
              this.state.completed === false && (
                <Question
                  onSelect={answer => {
                    this.submitAnswer(this.state.current, answer);
                  }}
                  question={this.state.questions[this.state.current]}
                  /*Randomises the answers on screen*/
                  correctPosition={Math.floor(Math.random() * 3)}
                  current={this.state.current}
                />
              )}

            <View
              style={{ flex: 1, alignItems: "center", justifyContent: "center" }}
            >
              {this.state.completed === true && (
                <View style={styles.container}>

                  <Text style={styles.logo}>Quiz Completed</Text>
                  <Text style={styles.scores}>Correct Answers:
{this.state.results.correctAnswers}</Text>
                  <Text style={styles.scores}>
                    Incorrect Answers: {10 - this.state.results.correctAnswers}
                  </Text>
                  <Text style={styles.scores}>Total Score: {50}</Text>
                  <Text style={styles.scores}>Obtained Score:
{this.state.results.score}</Text>

                  <TouchableOpacity
                          style={styles.buttonsub}
                          onPress={() => this.setUserValues(this.state.score)}
                        >
                          <Text
                             style={{color: "white", fontWeight: "bold"}}
                          >
                             Submit Score!
                          </Text>
                  </TouchableOpacity>
```

```
                <TouchableOpacity
                        style={styles.button}
                        onPress={() => this.props.navigation.navigate("HighScore")}
                      >
                        <Text
                           style={{color: "white", fontWeight: "bold"}}
                        >
                           Leaderboard
                        </Text>
                </TouchableOpacity>

                <TouchableOpacity
                        style={styles.button}
                        onPress={this.reset}
                      >
                        <Text
                           style={{color: "white", fontWeight: "bold"}}
                        >
                           Restart
                        </Text>
                </TouchableOpacity>

                <TouchableOpacity
                        style={styles.button}
                        onPress={() => this.props.navigation.navigate("Main")}
                      >
                        <Text
                           style={{color: "white", fontWeight: "bold"}}
                        >
                           Home
                        </Text>
                </TouchableOpacity>

              </View>
            )}
          </View>
        </View>
      );
    }
  }

  const styles = StyleSheet.create({
    container: {
  flex: 1,
  backgroundColor: 'white',
  alignItems: 'center',
  justifyContent: 'center',
    },

    logo: {
      fontStyle: 'italic',
      fontWeight:"bold",
      fontSize:30,
      color:"#5c5e70",
      marginBottom:110
```

```
  },

  loadingQuestions: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center"
  },

  scores: {
    fontSize: 16,
    color: "#777",
    textAlign: "center",
    padding: 10,
    marginTop: 10,
    lineHeight: 25
  },

  button: {
    backgroundColor: "#fb5b5a",
    padding: 10,
    marginTop: 10,
    borderRadius: 25,
    width: 150,
    height:50,
    alignItems:"center",
    justifyContent:"center"

  },

  buttonsub: {
    backgroundColor: "#669999",
    padding: 10,
    marginTop: 10,
    borderRadius: 25,
    width: 150,
    height:50,
    alignItems:"center",
    justifyContent:"center"

  }
});
```

screens/HighScore.js

```
import React from 'react';
import firebase from '../Firebase';
import Leaderboard from 'react-native-leaderboard';
import { Alert, View,Text, Image, Button, TouchableOpacity} from 'react-native';

//for suffix to be assigned based on number
const ordinal_suffix_of = (i) => {
    var j = i % 10,
        k = i % 100;
    if (j == 1 && k != 11) {
        return i + "st";
    }
    if (j == 2 && k != 12) {
        return i + "nd";
    }
    if (j == 3 && k != 13) {
        return i + "rd";
    }
    return i + "th";
}
export default class HighScore extends React.Component { //App
  constructor(){
    super();
  }
  //add user data to array and sets initial rank
  state = {
    data: [] ,
    userRank: 1,


}
alert = (title, body) => {
  Alert.alert(title, body, [{ text: "OK", onPress: () => {} }], {
    cancelable: false
  });
};
//sort the scores
sort = (data) => {
  //console.log("DATA",data)
  const sorted = data && data.sort((item1, item2) => {
      return item2.highScore - item1.highScore;
  })
  let userRank = sorted.findIndex((item) => {
      return item.userName === this.state.myUserName;
  })
  this.setState({ userRank: ++userRank });
  return sorted;
}
//import data from firestore
componentDidMount(){
try {
  const user = firebase.auth().currentUser;
firebase.firestore().collection("users").doc(firebase.auth().currentUser.uid)
```

```
.get()
.then(querySnapshot => {
//set your username and score
this.setState({
  myUserName:querySnapshot.data().name,
  myLastScore:querySnapshot.data().score
})
});
  firebase.firestore()
  .collection('users')
  .get()
  .then(snapshot => {
    snapshot.forEach(doc => {
      if (doc && doc.exists) {
        this.setState({

          data:[
            ...this.state.data,
            {
          userName:doc.data().name,
          highScore:doc.data().score}
          ]

        })

      }
    });
    this.sort(this.state.data)

  });
} catch (error) {

}

}

renderHeader() {
  return (
      <View colors={[ '#1da2c6', '#1695b7']}
          style={{ backgroundColor: 'white'}}>

            <View style={{
              flexDirection: 'column', justifyContent: 'center', alignItems:
'center', paddingTop:6,}}>
              <Text style={{fontSize: 45, fontStyle: 'italic', fontWeight:"bold",
color:"#fb5b5a", marginBottom:6, marginTop:6}}>
                  Leaderboard</Text>
              <Text style={{ fontSize: 25, fontWeight:"bold", color: 'black',
}}>{this.state.myUserName}</Text>


            </View>

          <View style={{
              flexDirection: 'row', justifyContent: 'center', alignItems: 'center',
```

```
              marginBottom: 15, marginTop: 15
          }}>
            <View style={{flex:3, justifyContent:'space-
around',alignItems:'center',flexDirection:'row'}}>
              <Text style={{ color: 'black', fontSize: 25,}}>
                  {ordinal_suffix_of(this.state.userRank)}
              </Text>

              <Text style={{ color: 'black', fontSize: 25,}}>
                  {this.state.myLastScore} points
              </Text>

            </View>

          </View>

      </View>
  )
}



  render(){
    return (

<View style={{ flex: 1 }}>
        {}

      {this.renderHeader()}
        <Leaderboard
        /*onRowPress={(item, index) => {
          this.alert(item.name + " clicked", item.score + " points, wow!")}}*/
        data={this.state.data}
        sortBy='highScore'
        evenRowColor= "#fb5b5a"
        labelBy='userName'/>

        </View>


        )
  }
}
```

screens/Stats.js

```
import React from 'react'
import { StyleSheet, Platform, Image, Text, View, TextInput, Alert, TouchableOpacity,
Dimensions, StatusBar } from 'react-native'
import * as firebase from "firebase";
import { Ionicons } from "@expo/vector-icons";

export default class Stats extends React.Component {

  constructor(){
    super();
  }

  state = { name: '', score: 0, level: 0, attempts: 0}

  componentDidMount(){
  try {
    const user = firebase.auth().currentUser;
  firebase.firestore().collection("users").doc(firebase.auth().currentUser.uid)
  .get()
  .then(querySnapshot => {
  this.setState({
    name:querySnapshot.data().name,
    score:querySnapshot.data().score,
    attempts:querySnapshot.data().attempts,
    level:querySnapshot.data().level
  })
  });

  } catch (error) {

  }

  }

render() {
return (
        <View style={styles.container}>
                <StatusBar backgroundColor="black"/>
                <Text style={styles.logo}>Stats</Text>

                <Text style={{ fontSize: 25, fontWeight:"bold", color: 'black',
marginBottom:20 }}>Name: {this.state.name}</Text>

                <Text style={{ fontSize: 25, fontWeight:"bold", color: 'black',
marginBottom:20}}>Level: {this.state.level}</Text>

                <Text style={{ fontSize: 25, fontWeight:"bold", color: 'black',
marginBottom:20}}>Score: {this.state.score}</Text>

                <Text style={{ fontSize: 25, fontWeight:"bold", color: 'black',
marginBottom:20}}>Attempts: {this.state.attempts}</Text>

        </View>
```

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

  logo:{
    fontStyle: 'italic',
    fontWeight:"bold",
    fontSize:45,
    color:"#fb5b5a",
    marginBottom:110
  },

  container: {
    flex: 1,
    backgroundColor: "white",
    alignItems: 'center',
    justifyContent: 'center',
  },

  button: {
    width:"80%",
    backgroundColor:"#fb5b5a",
    borderRadius:25,
    height:50,
    alignItems:"center",
    justifyContent:"center",
    marginTop:10,
    marginBottom:10,
  },

  paragraph: {
    fontSize: 16,
    color: "#777",
    textAlign: "center",
    padding: 10,
    marginTop: 20,
    lineHeight: 25
  }

})
```

INSTITUTE of TECHNOLOGY CARLOW
Institiúid Teicneolaíochta Cheatharlach

PLAGIARISM DECLARATION

*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

*I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

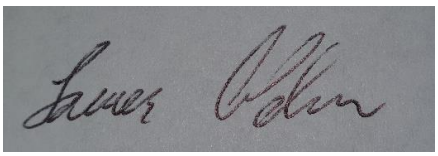
*I have provided a complete bibliography of all works and sources used in the preparation of this submission.

*I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name: (Printed) James Nolan

Student Number(s): C00226267

Signature(s):



Date: 27/04/2021