

```

import React, { Component } from 'react';
import {
  StyleSheet,
  View,
  Text,
  TouchableOpacity,
  Image
} from 'react-native';
import firebase from 'firebase';
import Select2 from 'react-native-select-two';
import { Icon } from 'react-native-elements';
import { Dropdown } from 'react-native-material-dropdown';
import Toast from 'react-native-simple-toast';
import Dialog from "react-native-dialog";

import ButtonComponent from '../components/ButtonComponent';
import FormInput from '../components/FormInput';
import DropdownInput from '../components/DropdownInput';

class EditUserDetails extends Component {
  constructor(props) {
    super(props);
    this.state = {
      email: '',
      firstname: '',
      lastname: '',
      licence: '',
      currentPassword: '',
      newPassword: '',
      showchangesdialog: false,
      changepassworddialog: false,
      error: '',
      coverage: ''
    };
  }
  async componentDidMount(){
    const { currentUser } = firebase.auth();

    //EVENT: firebase call
    //populate field placeholders from existing information on db
    firebase.database().ref(`/users/${currentUser.uid}/`).on('value', snapshot => {
      this.email = snapshot.val().email
      this.firstname = snapshot.val().firstname
      this.lastname = snapshot.val().lastname
      this.licence = snapshot.val().licence
      this.coverage = snapshot.val().coverage
      this.setState({
        email: this.email,
        firstname: this.firstname,
        lastname: this.lastname,
        licence: this.licence,
        coverage: this.coverage
      })
    })
  }

  //event listener submit changes
  submitChanges(){
    const { currentUser } = firebase.auth();

```

```

const {
  firstname,
  lastname,
  licence,
  currentPassword,
  coverage
} = this.state

//check password entered is correct
if(this.reauthenticate(currentPassword)){
  var Data = {
    firstname: firstname,
    lastname: lastname,
    licence: licence,
    coverage: coverage
  }

  return firebase.database().ref(`/users/${currentUser.uid}/`).update(Data)
  .then(result => {
    alert('Changes submitted succesfully...')
  })
  .catch(error => { this.setState({error}) })
} else {
  Toast.show('Invalid password entry')
}
}

//change password modal
changePassword = (currentPassword, newPassword) => {
  //ceck entered password
  this.reauthenticate(currentPassword).then(() => {
    var user = firebase.auth().currentUser;
    //firebase updatePassword function
    user.updatePassword(newPassword).then(() => {
      alert("Password changed succesfully")
    }).catch((error) => { this.setState({error}) });
  }).catch((error) => { this.setState({error}) });
}

//compare $currentPassword to firebase auth password
async reauthenticate(currentPassword){
  var user = firebase.auth().currentUser;
  var cred = firebase.auth.EmailAuthProvider.credential(
    user.email, currentPassword);

  return await user.reauthenticateWithCredential(cred)
    .then(() => {
      console.log('Success')
    })
    .catch(e => {
      console.log(e)
    })
}

render(){
  return(
    <View style={styles.container}>
      <View style={styles.content}>

```

```

<View style={{ flex: 3, justifyContent: 'center' }}>
  <Image
    source={require('.././../assets/face-scan.png')}
    style={styles.image}
  />
  <Text style={styles.headingtext}>{this.state.email}</Text>
  <FormInput
    icon="user"
    type="antdesign"
    placeholder={this.firstname}
    keyboardType="default"
    onChangeText={(firstname) => this.setState({firstname})}
    // ref={(input) => { this.firstnameInput = input }}
    secureTextEntry={false}
  />
  <FormInput
    icon="user"
    type="antdesign"
    placeholder={this.lastname}
    keyboardType="default"
    onChangeText={(lastname) => this.setState({lastname})}
    // ref={(input) => { this.lastnameInput = input }}
    secureTextEntry={false}
  />
  <DropdownInput
    icon="drivers-license-o"
    type="font-awesome"
    label={this.licence}
    data={[{
      value: 'Full Licence'
    }, {
      value: 'Provisional Licence'
    }]}
    onChangeText={(value) => this.setState({ licence: value})}
  />
  <DropdownInput
    icon='attach-money'
    type='material'
    data={[{
      value: 'Third Party Insurance'
    }, {
      value: 'Third Party Fire & Theft'
    }, {
      value: 'Comprehensive'
    }]}
    onChangeText={(value) => this.setState({ coverage: value})}
    label={this.coverage}
  />
</View>
<View style={{width: 250, alignSelf: 'center'}}>
  <Text style={{textAlign: 'center', color: 'red'}}>
    {this.state.error}</Text>
  </View>
<View>
  <Dialog.Container visible={this.state.showchangesdialog}>
    <Dialog.Title>Enter password</Dialog.Title>
    <Dialog.Description>
      Enter password to make changes
    </Dialog.Description>
  </Dialog.Container>
</View>

```

```

        </Dialog.Description>
        <Dialog.Input
            onChangeText={(currentPassword) =>
this.setState({currentPassword})}
            secureTextEntry={true}
            label="Password"
        />
        <Dialog.Button label="Cancel" onPress={() => this.setState({
showchangesdialog: false})}/>
        <Dialog.Button label="Confirm" onPress={() => {
            this.setState({showchangesdialog: false})
            this.submitChanges()
        }}/>
    </Dialog.Container>
</View>

<View>
    <Dialog.Container visible={this.state.changepasswordddialog}>
        <Dialog.Title>Change Password</Dialog.Title>
        <Dialog.Description>
            Confirm current password & enter new password
        </Dialog.Description>
        <Dialog.Input
            onChangeText={(currentPassword) =>
this.setState({currentPassword})}
            secureTextEntry={true}
            label="Enter Current Password"
            placeholder="Current Password"
        />
        <Dialog.Input
            onChangeText={(newPassword) =>
this.setState({newPassword})}
            secureTextEntry={true}
            label="Enter New Password"
            placeholder="New Password"
        />
        <Dialog.Button label="Cancel" onPress={() => this.setState({
changepasswordddialog: false})}/>
        <Dialog.Button label="Confirm" onPress={() => {
            if(this.state.currentPassword != '' &&
this.state.newPassword != ''){
                this.setState({changepasswordddialog: false})
                this.changePassword(this.state.currentPassword,
this.state.newPassword)
            } else {
                Toast.show("All fields are required")
            }
        }}/>
    </Dialog.Container>
</View>
</View>
<View style={styles.button}>
    <ButtonComponent
        text="Save Changes"
        icon="save"
        type="antdesign"
        onPress={() => this.setState({ showchangesdialog: true})}
    />
    <ButtonComponent

```

```

        text="Change Password"
        icon="key"
        type="antdesign"
        onPress={() => this.setState({changepassworddialog: true})}
      />
      <TouchableOpacity
        style={styles.cancel}
        onPress={() => this.props.navigation.goBack()}
      >
        <Text style={styles.canceltext}>Cancel</Text>
        <Icon
          name="close"
          type="antdesign"
          style={{marginRight: 10}}
          color="#2E6CB5"
        />
      </TouchableOpacity>
    </View>
  </View>
)
}
}

```

```

const styles = StyleSheet.create({
  headingtext: {
    fontSize: 24,
    paddingTop: 20,
    color: '#2E6CB5',
    justifyContent: 'center',
    textAlign: 'center',
    paddingBottom: 30,
  },
  content: {
    flex: 4,
    width: '80%'
  },
  button: {
    flex: 2,
    width: '100%'
  },
  container: {
    flexDirection: 'column',
    flex: 6,
    alignItems: 'center',
  },
  cancel:{
    alignSelf: 'center',
    alignContent: 'center',
    width: '70%',
    backgroundColor: 'white',
    borderRadius: 400,
    borderColor: '#2E6CB5',
    borderWidth: 1,
    padding: 15,
    flexDirection: 'row',
    justifyContent: 'space-between',
    marginBottom: '5%',
  },
},

```

```
canceltext: {
  fontSize: 20,
  marginLeft: 20,
  color: '#2E6CB5',
  fontWeight: 'normal'
},
image:{
  width: 100,
  height: 100,
  alignItems: 'center',
  alignSelf: 'center'
}
})

export default EditUserDetails;
```