

```

import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View,
  TextInput,
  Button,
  TouchableOpacity,
  Image,
  Alert,
  Switch,
  AsyncStorage,
  YellowBox
} from 'react-native';
import { Icon } from 'react-native-elements';
import firebase from 'firebase';
import Toast from 'react-native-simple-toast';
import Dialog from "react-native-dialog";
import _ from 'lodash';

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

class LoginScreen extends Component {
  constructor(props) {
    super(props);
    this.state = {
      email: '',
      password: '',
      rememberMe: false,
      error: '',
      resetPasswordDialog: false
    }
  }

  async componentDidMount(){

    //on toggle
    //remember user
    rememberUser = async () => {
      try {
        //save email and password ot async storage
        await AsyncStorage.setItem('EMAIL', this.state.email);
        await AsyncStorage.setItem('PASSWORD', this.state.password);
      } catch (error) {
        console.log("async rememberMe error: " + error)
      }
    }

    //reset password
    //call firebase reset password by email
    resetPassword(){
      firebase.auth().sendPasswordResetEmail(this.passwordChangeEmail)
        .then(() => alert("Password reset email sent"))
        .catch((error) => this.setState({error:error.message}))
    }
  }

```

```

//sign in user
signinUser(){
  //compare email and password against firebase authentication
  firebase.auth().signInWithEmailAndPassword(this.state.email,
this.state.password).then(() => {
    if(this.state.email == 'admin@payasyoudrive.com'){
      this.props.navigation.navigate('AdminOptions')
    } else {
      this.props.navigation.navigate('BottomTab')
    }

  }).catch((error) => {
    this.setState({error:error.message})
  });
}

//toggle remember me
toggleRememberMe = value => {
  this.setState({ rememberMe: value })
  if (value === true) {
    this.rememberUser();
  } else {
    this.forgetUser();
  }
}

//clear async storage
async forgetUser(){
  try {
    await AsyncStorage.removeItem('EMAIL');
    await AsyncStorage.removeItem('PASSWORD')
  } catch (error) {
    console.log("async forgetUser error: " + error)
  }
};

render() {
  return (
    <View style={styles.container}>
      <View style={styles.content}>
        <View style={{flex:2}}>

          </View>

        <View style={{flex: 3}}>
          <Image
            source={require('../assets/logo-yellow-blue.png')}
            alignSelf='center'
            style={styles.logo}
          />
          <Text style={styles.header}>
            <Text style={{color: '#EFC066'}}>P</Text>
            ay
            <Text style={{color: '#EFC066'}}> A</Text>
            s
            <Text style={{color: '#EFC066'}}> Y</Text>
            ou
            <Text style={{color: '#EFC066'}}> D</Text>rive</Text>
          </View>

```

```

<View style={{flex:6}}>
  <FormInput
    icon='mail'
    type='antdesign'
    onChangeText={email => this.setState({email})}
    placeholder='Email Address'
  />

  <FormInput
    icon='key'
    type='antdesign'
    onChangeText={password => this.setState({password})}
    placeholder='Password'
    secureTextEntry={true}
  />

  <View style={styles.rememberMecontainer}>
    <View>
      <Text style={styles.rememberme}>Remember Me</Text>
    </View>

    <View>
      <Switch
        value={this.state.rememberMe}
        onValueChange={(value) => this.toggleRememberMe(value)}
        trackColor='#2E6CB5'
        thumbColor='#2E6CB5'
      />
    </View>
  </View>

  <View>
    <Dialog.Container visible={this.state.resetPasswordDialog}>
      <Dialog.Title>Reset password</Dialog.Title>
      <Dialog.Description>
        Enter account email address. Then check your emails for reset
link.
      </Dialog.Description>
      <Dialog.Input
        onChangeText={(passwordChangeEmail) => this.passwordChangeEmail
= passwordChangeEmail}
        label="Email Address"
        placeholder="Enter email"
      />
      <Dialog.Button label="Cancel" onPress={() => this.setState({
resetPasswordDialog: false})}/>
      <Dialog.Button label="Confirm" onPress={() => {
        this.setState({resetPasswordDialog: false})
        this.resetPassword()
      }}/>
    </Dialog.Container>
  </View>

  <ButtonComponent
    icon='arrowright'
    type='antdesign'
    onPress={() => this.signInUser()}
    text='Login'
  />

```

```

    />

    <TouchableOpacity onPress={() =>
this.props.navigation.navigate('WelcomeScreen')}>
      <Text style={styles.signup}>Sign Up</Text>
    </TouchableOpacity>

    <TouchableOpacity style={styles.buttonContainer} onPress={() =>
this.setState({resetPasswordDialog: true})}>
      <Text style={styles.forgotpassword}>Forgot Password?</Text>
    </TouchableOpacity>

    <View style={styles.errorcontainer}>
      <Text style={styles.error}>{this.state.error}</Text>
    </View>

    <View>
      <Dialog.Container visible={this.state.resetPasswordDialog}>
        <Dialog.Title>Reset password</Dialog.Title>
        <Dialog.Description>
          Enter account email address. Then check your emails for reset
link.
        </Dialog.Description>
        <Dialog.Input
          onChangeText={(passwordChangeEmail) => this.passwordChangeEmail
= passwordChangeEmail}
          label="Email Address"
          placeholder="Enter email"
        />
        <Dialog.Button label="Cancel" onPress={() => this.setState({
resetPasswordDialog: false})}/>
        <Dialog.Button label="Confirm" onPress={() => {
          this.setState({resetPasswordDialog: false})
          this.resetPassword()
        }}/>
      </Dialog.Container>
    </View>
  </View>
</View>
);
}
}

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#EFF1F3'
  },
  content: {
    flex: 1,
    alignSelf: 'center',
    width: '80%'
  },
  header: {
    flex: 2,
    fontWeight: "bold",
    fontSize: 32,
    color: "#373E45",

```

```

      textAlign: 'center'
    },
    logo:{
      flex: 2,
      width: 150,
      height: 150,
      resizeMode: 'contain'
    },
    rememberMecontainer: {
      flexDirection: 'row',
      alignSelf: 'center',
      paddingVertical: 10
    },
    rememberme: {
      color: '#373E45',
      fontSize: 15,
      marginRight: 20,
      padding: 3,

    },
    signup: {
      textAlign: 'center',
      fontSize: 20
    },
    forgotpassword: {
      textAlign: 'center',
      paddingTop: 20,
    }
  });

export default LoginScreen;

```