

```

//TODO generate bills

import React, { Component } from 'react';
import {
  StyleSheet,
  View,
  Text,
  Image,
  AsyncStorage,
  Alert
} from 'react-native';
import firebase from 'firebase';

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

class AdminOptions extends Component {
  constructor(props) {
    super(props);
    this.state = {
      error: ''
    };
  }
  componentDidMount(){
    //security check
    //make sure user is admin
    //navigate back to log in screen if false
    if(firebase.auth().currentUser.email !== 'admin@payasyoudrive.com'){
      firebase.auth().signOut()
      this.props.navigation.navigate('LoginScreen')
    }
  }

  //logout admin
  //prompt for confirmation
  logout(){
    Alert.alert(
      'Logout',
      'Are you sure you would like to logout?',
      [
        {
          text: 'Cancel',
          onPress: () => console.log('Cancel Pressed'),
          style: 'cancel',
        },
        {text: 'Logout', onPress: () => {
          firebase.auth().signOut()
          this.props.navigation.navigate('LoginScreen')}}
      ],
    );
  }

  //to be run once a month
  //TODO automatically trigger generateBills()
  generateBills(){
    const currentMonth = this.humanizedMonth();
    const currentYear = new Date().getFullYear()
    firebase.database().ref(`/users/`).once('value').then(snapshot => {
      snapshot.forEach((childSub) => {

```

```

        var id = childSub.key
        var amount_owed = 0
        firebase.database().ref(`/users/${id}
/journeys`).once('value').then(snapshot => {
            snapshot.forEach((childSub) => {
                if(childSub.val().billing_month == currentMonth){
                    amount_owed = amount_owed + childSub.val().cost
                }
            })
            const month_year_string = currentMonth + "_" + currentYear
            firebase.database().ref(`/users/${id}/billing_history
/${month_year_string}/`).set(amount_owed)
        })
    })
    }).catch(error => this.setState({error}))
}

```

```

humanizedMonth(){
    var month = new Date().getMonth()
    switch (month) {
        case 0:
            month = "Jan"
            break;
        case 1:
            month = "Feb"
            break;
        case 2:
            month = "Mar"
            break;
        case 3:
            month = "Apr"
            break;
        case 4:
            month = "May"
            break;
        case 5:
            month = "Jun"
            break;
        case 6:
            month = "Jul"
            break;
        case 7:
            month = "Aug"
            break;
        case 8:
            month = "Sep"
            break;
        case 9:
            month = "Oct"
            break;
        case 10:
            month = "Nov"
            break;
        case 11:
            month = "Dec"
            break;
        default:
            break;
    }
}

```

```

    return month
  }

  render() {
    return(
      <View style={styles.container}>
        <View style={styles.iconContainer}>
          <Image
            source={require('../assets/settings.png')}
            style={styles.icon}
          />
          <Text style={styles.error}>{this.state.error}</Text>
        </View>
        <View style={styles.buttonContainer} >
          <ButtonComponent
            text='Algorithm Conditions'
            icon='find'
            type='antdesign'
            onPress={() =>
this.props.navigation.navigate('AlgorithmScreen')}
          />

          <ButtonComponent
            text='Process Claims'
            icon='menu-fold'
            type='antdesign'
            onPress={() => this.props.navigation.navigate('ClaimsScreen')}
          />

          <ButtonComponent
            text='Generate Bills'
            icon='shoppingcart'
            type='antdesign'
            onPress={() => this.generateBills()}
          />

          <ButtonComponent
            text='Smart Contract Control'
            icon='shoppingcart'
            type='antdesign'
            onPress={() =>
this.props.navigation.navigate('SmartContractControl')}
          />

          <ButtonComponent
            text='Logout'
            icon='logout'
            type='antdesign'
            onPress={() => this.logout()}
          />
        </View>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 5,

```

```
        flexDirection: 'column'
      },
      icon:{
        width: 100,
        height: 100
      },
      iconContainer:{
        flex: 2,
        flexDirection: 'row',
        alignItems: 'center',
        alignSelf: 'center'
      },
      buttonContainer:{
        flex: 2,
        flexDirection: 'column',
      },
      error: {
        textAlign: 'center',
        color: 'red',
        padding: 10
      },
    },
  });

export default AdminOptions;
```