

**Ownership Chain**

**Summer Internship Project Report**

Submitted To

**DataRitz Technologies**

**Duration : 6 weeks**

**By**

**ANSHI AGGARWAL**

**1803213022**

**Under the guidance of**

**DR. Bhuwneshwar Prasad Sharma**

**CERTIFICATE**

This is to certify that ProjectReportentitled “**Ownership Chain**” which is submitted by team-

Innovators in partial fulfillment of the requirement for the summer internship of degree

B.Tech in Department of Computer Science and Engineering of ABES Engineering

College, is a record of the candidate own work carried out by him under my/our supervision.

The matter embodied in thesis is original and has not been submitted for the award of any

other degree.

**Supervisor: Dr. Bhuvenshwar Prasad Sharma**

**Date:03/july/2020**



**ACKNOWLEDGEMENT**

*It gives us a great sense of pleasure to present the report of the B. Tech summer internship Project undertaken during B. Tech. Second Year. We owe special debt of gratitude to DR. Bhuvneshawar Prasad Sharma,* ***DataRitz Technologies*** *for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of theday.*

*We also take the opportunity to acknowledge the contribution of Head, Department of* ***DataRitz Technologies*** *for his full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the DataRitz Technologies for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.*

*Signature: Team INNOVATORS*

*Name &Roll no: Anshi Aggarwal (1803213022)*

***Date- 03/july/2020***

**TABLE OF CONTENTS**

Page No.

Declaration ii

Certificate iii

Acknowledgements iv

Project Title 5

Project Description 6

User Story 7

Team Members 8

Entities 9

Use Case Diagram 10

ER Diagram 11

Data Flow Diagram 12-13

Class Diagram 14

Data Dictionary 15

Prototype 16-27

Actual Screens 28-32

Screen wise code 33-58

References 59



**PROJECT TITLE**

**OWNERSHIP CHAIN**

**TEAM MEMBERS**

**TEAM INNOVATORS**

1. MAYANK SHARMA- 1803210093
2. KAMINI JHA- 1803210071
3. KOMAL- 1803210082
4. ANSHI AGARWAL- 1803213022



**PROJECT DESCRIPTION**

In our project, “OWNERSHIP CHAIN”, the basic idea is that the user can upload all the images of the things that he/she has bought or he/she can upload the picture of the bill of any particular thing and image of the things that has been purchased. By this , the user will have a proof of that purchased entity that will help in future. Because many a times we do not keep the bills in a proper place or things get stolen and thus this creates a problem whenever the user is asked to show that the thing belong to him only. So , with the help of this app user can keep the record of all the products he/she had bought.



**USER STORY**

User can login in the app if he/she already have an account, otherwise he/she have to sign up for creating new account. User can upload a scanned image of the things or the bills or user can also select any image from the gallery. User can also view all the uploaded images. User can also add the description about the image that he/she has uploaded. User can change his password and user can see his/her details.



**TEAM MEMBERS**

# Our team - Innovators, consist of 4 members-

# Mayank Sharma- (Team Leader)- 1803210093

# Kamini Jha- 1803210071

# Komal- 1803210082

# Anshi Agarwal- 1803213022

# 

# ENTITIES

# The main entity of our program-

# User

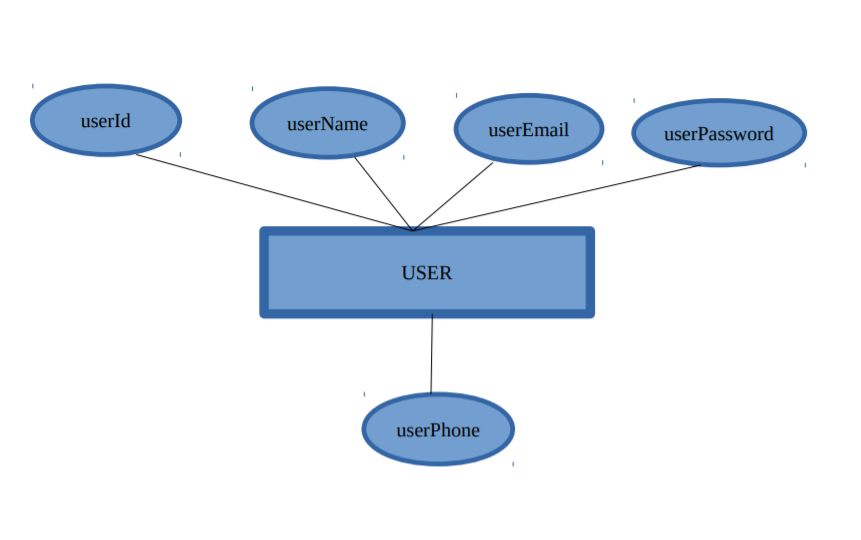
# User’s role is to upload scanned image or image from gallery , to add description for the image, user can view all the images . User can login for an existing account or can also sign up for new account and also can change the password.

# 

# USE CASE DIAGRAM

# 

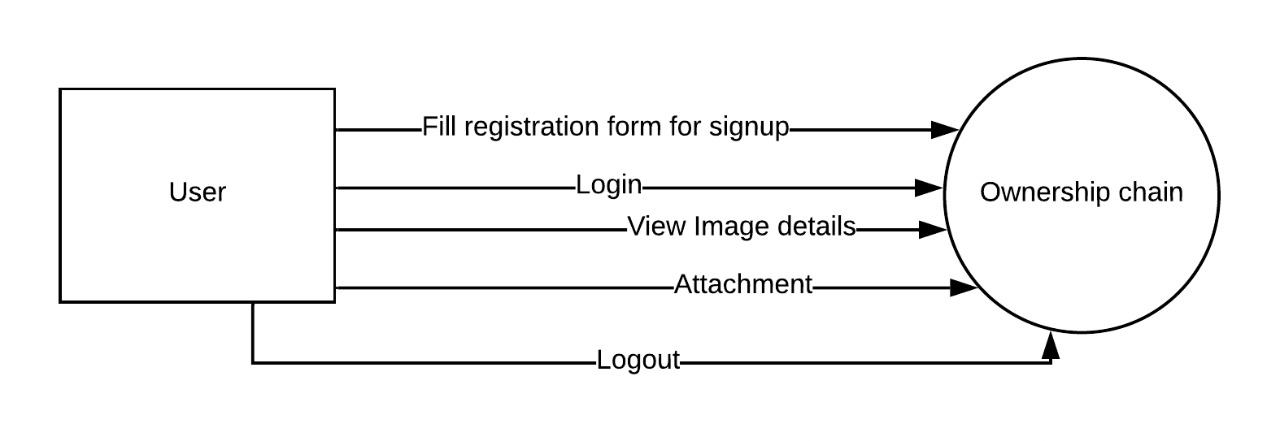
**ER DIAGRAM**





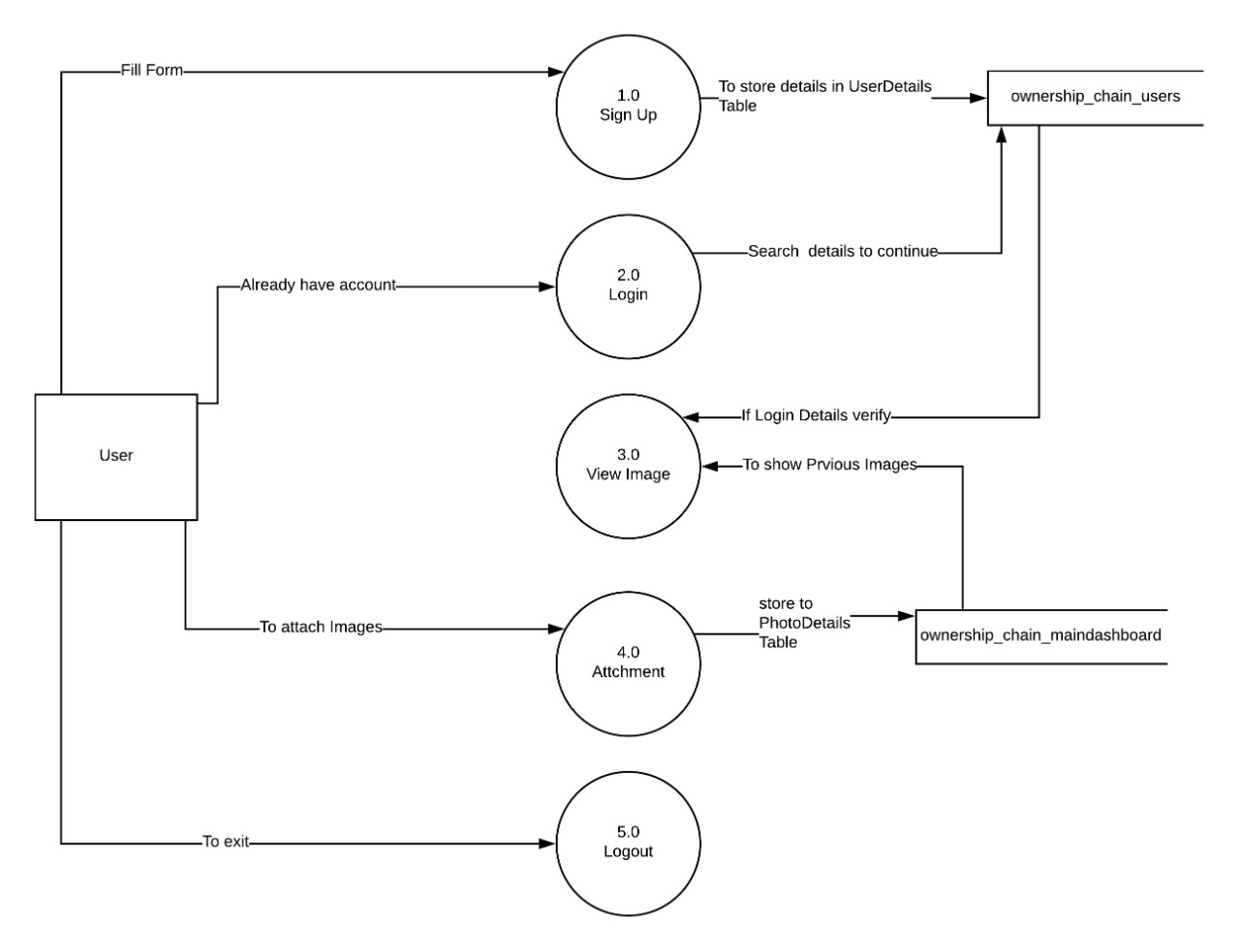
**DATA FLOW DIAGRAM**

**0th level DFD-**

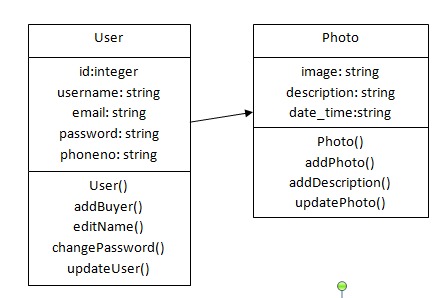




**1st level DFD-**

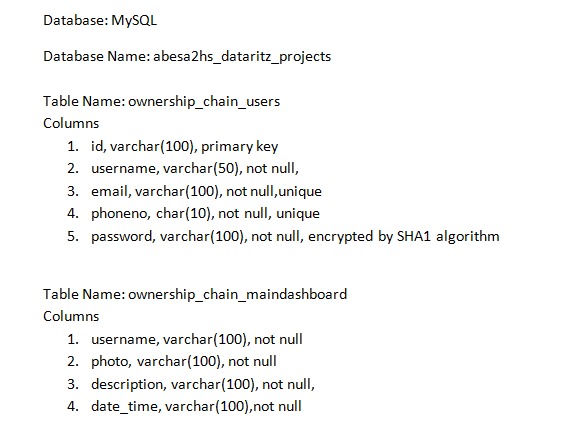


**CLASS DIAGRAM**





**DATA DICTIONARY**





**PROTOTYPE**

1. **Welcome page-**

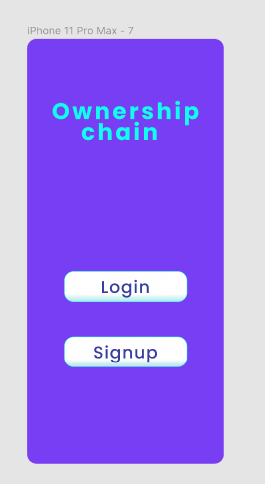


**Functionality-**

It shows the app name with the logo of the app.



1. **Login-Signup Screen-**



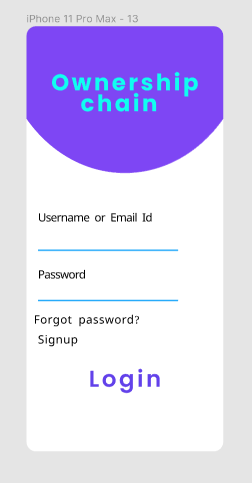
**Functionality-**

This screen shows two options to the user of sign in and login. If a user already has an account then he/she can login else sign in to the app.

**Controls and Rules-**

* Button- Login
* Button- Sign Up

1. **Login-**



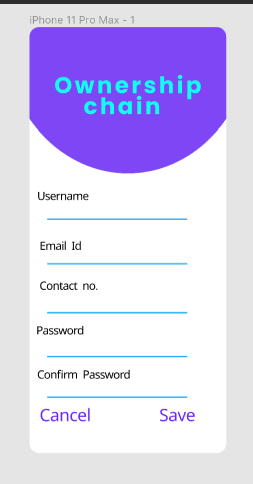
**Functionality-**

Here, the user can login into the account ( if already exists) using two credentials: Username/email and password. Users can also navigate to the signup page if does not have any account. There is also the option of forgot password if the user didn’t remember the password, user get their password on Gmail.

**Controls and Rules-**

* Text Box- Username or Email id
* Cannot be null
* Text Box- Password
* Cannot be null
* More than 8 letters
* First letter should be Capital
* Text- Forget password
* Text- Sign up
* Button- Login

1. **Sign up-**



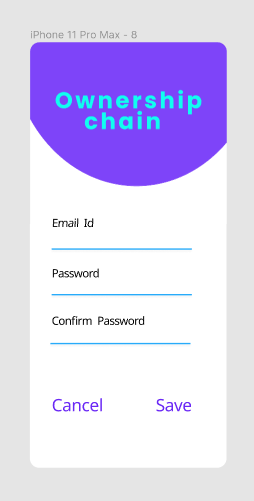
**Functionality-**

User can create his/her account on the app by signup screen. Here, the user need to provide three credentials: username, email id and contact number.

**Controls and Rules-**

* Text Box- Username
* Cannot be null
* Text Box- Email Id
* Cannot be null

1. **Change Password-**



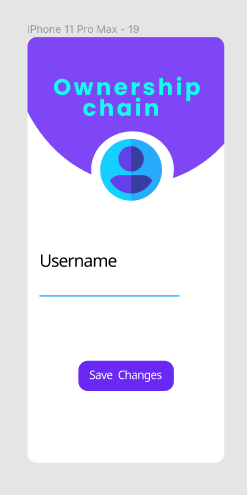
**Functionality-**

On this screen the user can change the password of account.

Controls and Rules-

* Text Box- Email Id
* Cannot be null
* Text Box- Password
* Cannot be null
* More than 8 letters
* First letter should be Capital
* Text Box- Confirm Password
* Cannot be null
* More than 8 letters
* First letter should be Capital

1. **Change Username-**



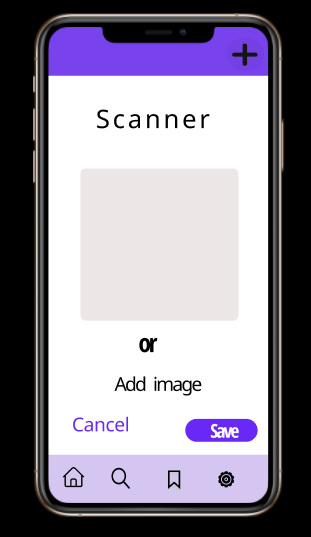
**Functionality-**

On this screen, user can change his/her username that will be shown to his/her profile.

**Controls and Rules-**

* Text Box- Username
* Cannot be null
* Button- Save Changes

1. **Scanning and attaching ownerships**



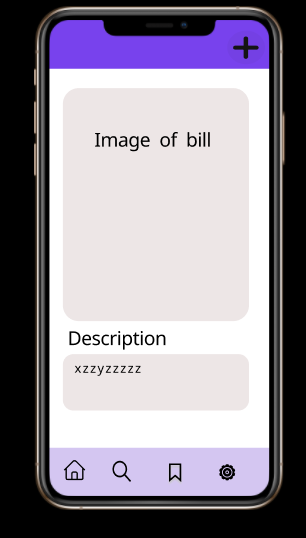
**Functionality-**

This screen shows the image and Description by tap the image to display.

**Controls and Rules-**

* Image- Images
* Text Box- Description
* Cannot be null
* Button- Home
* Button- Search
* Button- Bookmark
* Button- Setting

1. **Image Description-**



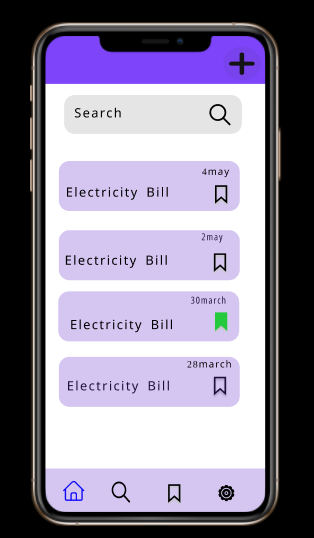
**Functionality-**

User can provide the description and Subject of the provided image and then save that.

**Controls and Rules-**

* Image- Scanned Image
* Text Box- Description
* Cannot be null
* Text Box- Subject
* Cannot be null
* Button- Save
* Button- Home
* Button- Search
* Button- Bookmark
* Button- Setting

1. **Main Dashboard-**



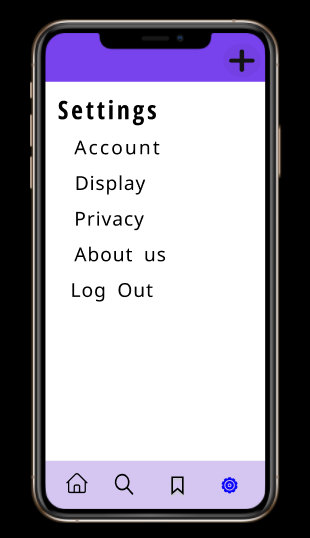
**Functionality-**

On this screen, user can view all his/her previously existing images with the uploaded dates.

**Controls and Rules-**

* Text Box- Search
* Box- Image
* Button- Home
* Button- Search
* Button- Bookmark
* Button- Setting
* Button- Attach

1. **Settings-**



**Functionality-**

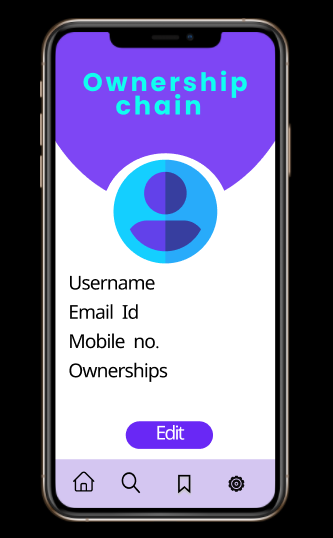
On the setting screen, user have the following options for setting:

1. Account( user can see his/her account information)
2. Display( Show Dashboard)
3. Privacy ( can change the passwords etc.)
4. About Us ( About the app)
5. Log Out (user can logout from his/her account)

Controls and Rules-

* Text- Account
* Text- Display
* Text- Privacy
* Text- About us
* Text- Logout
* Button- Home
* Button- Search
* Button- Bookmark
* Button- Setting

1. **Profile-**



**Functionality-**

On this screen user can see his/her account details such as username, email id, mobile number, Image Saved or ownerships.

**Controls and Rules-**

* Text- Username
* Text- Email Id
* Text- Mobile Number
* Text- Ownership
* Button- Edit
* Button- Home
* Button- Search
* Button- Bookmark
* Button- Setting



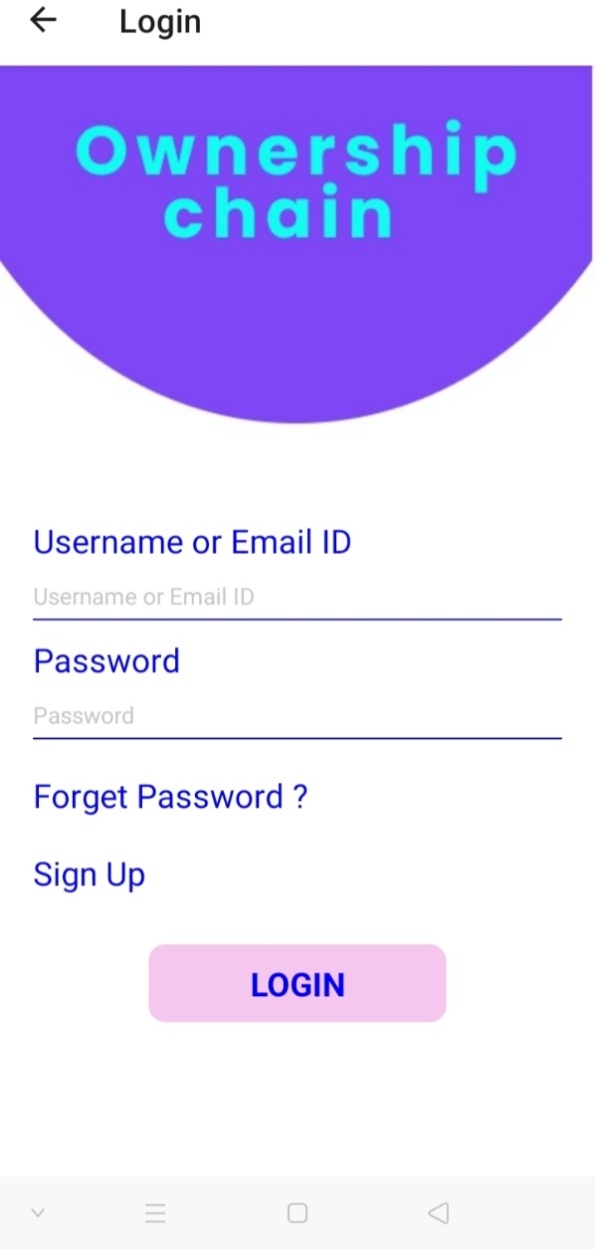
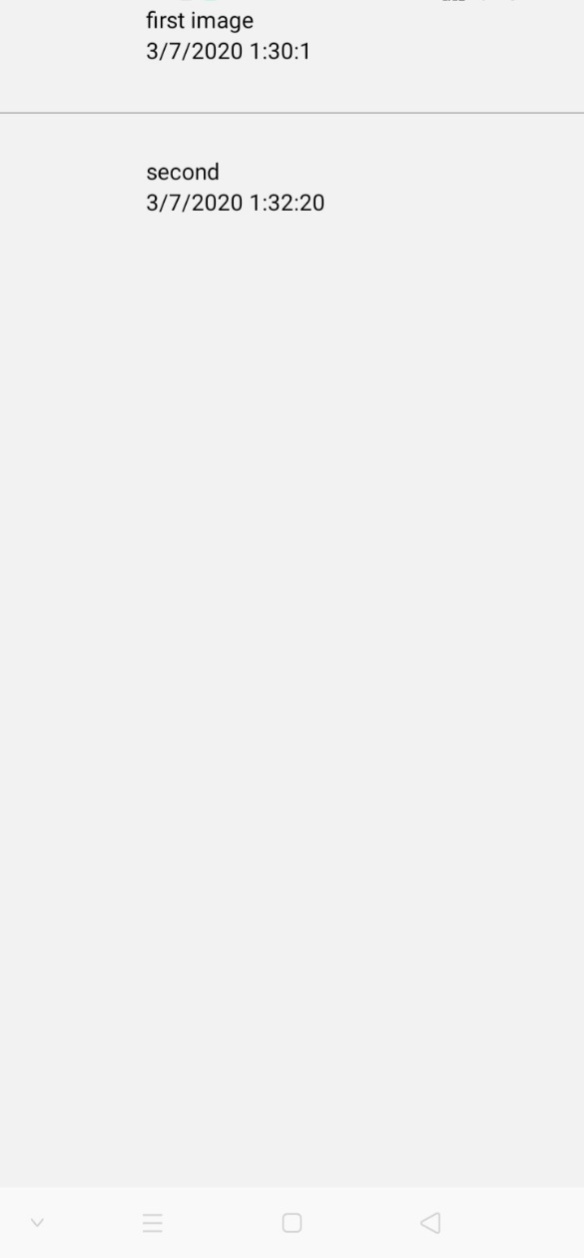
**ACTUAL SCREENS**

**Main screen-Drawer-**



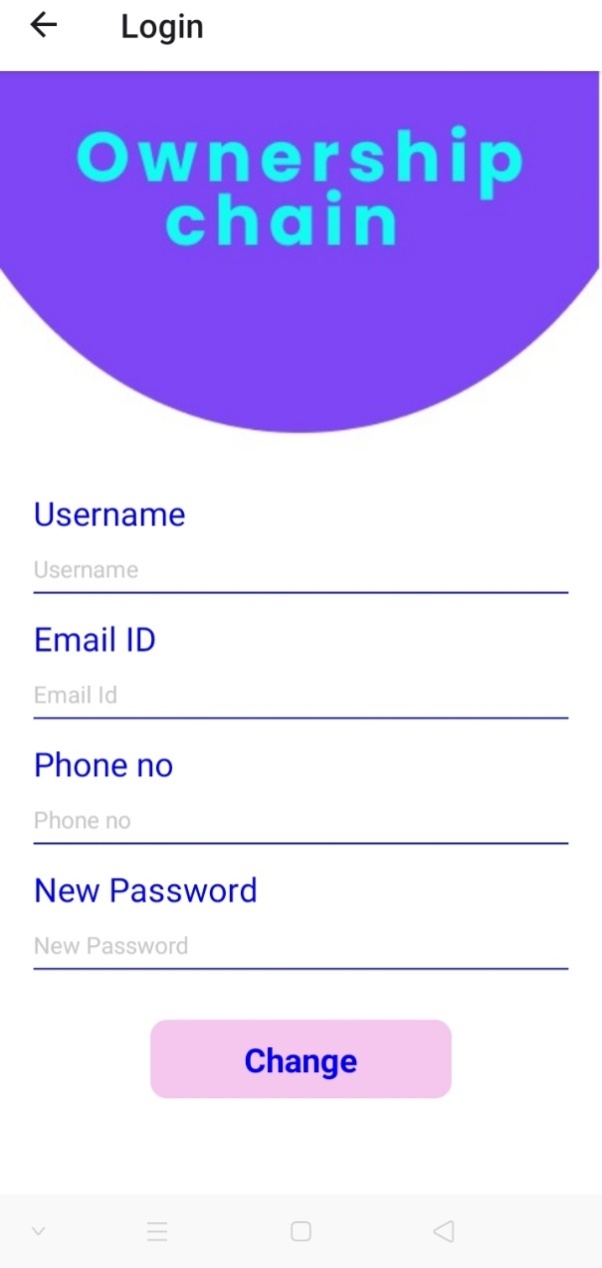
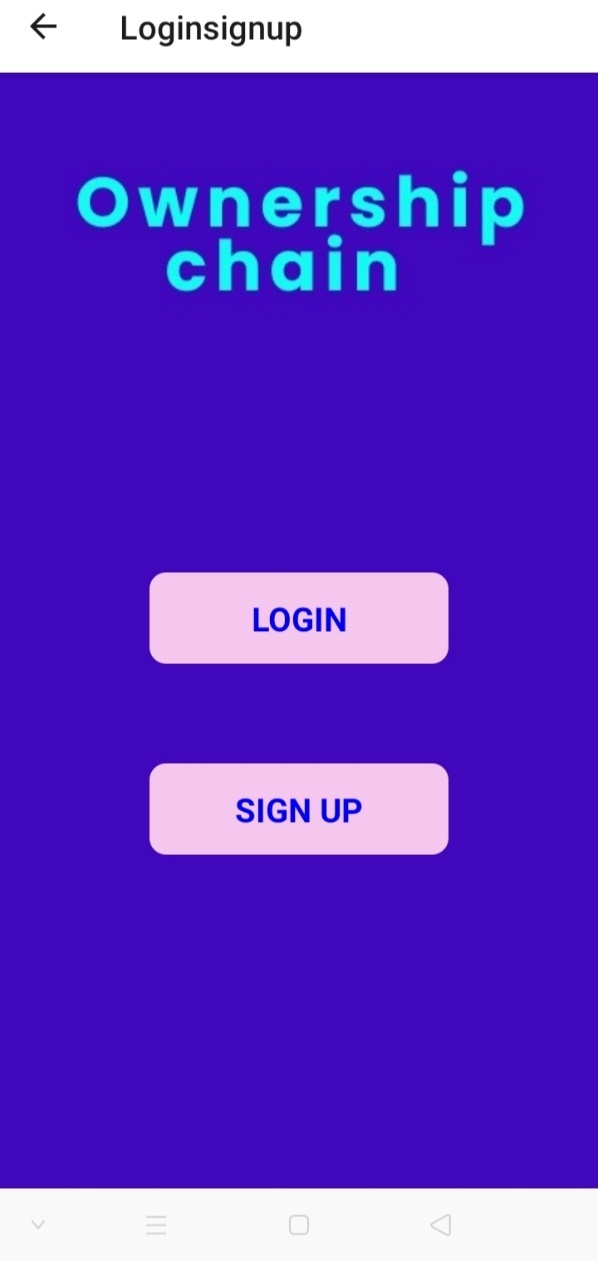


**Main Dashboard-Login screen-**



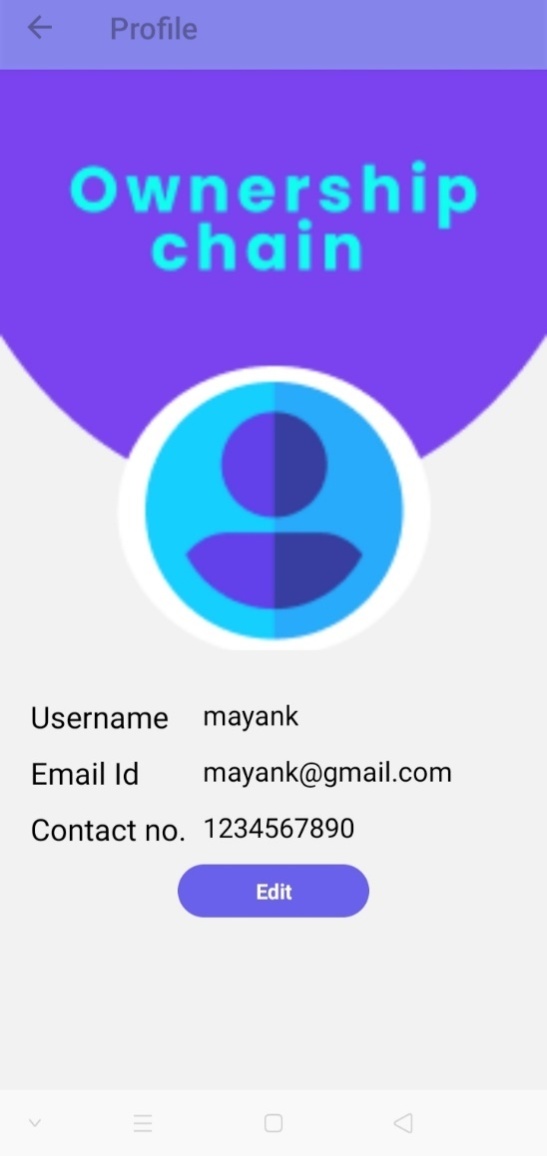
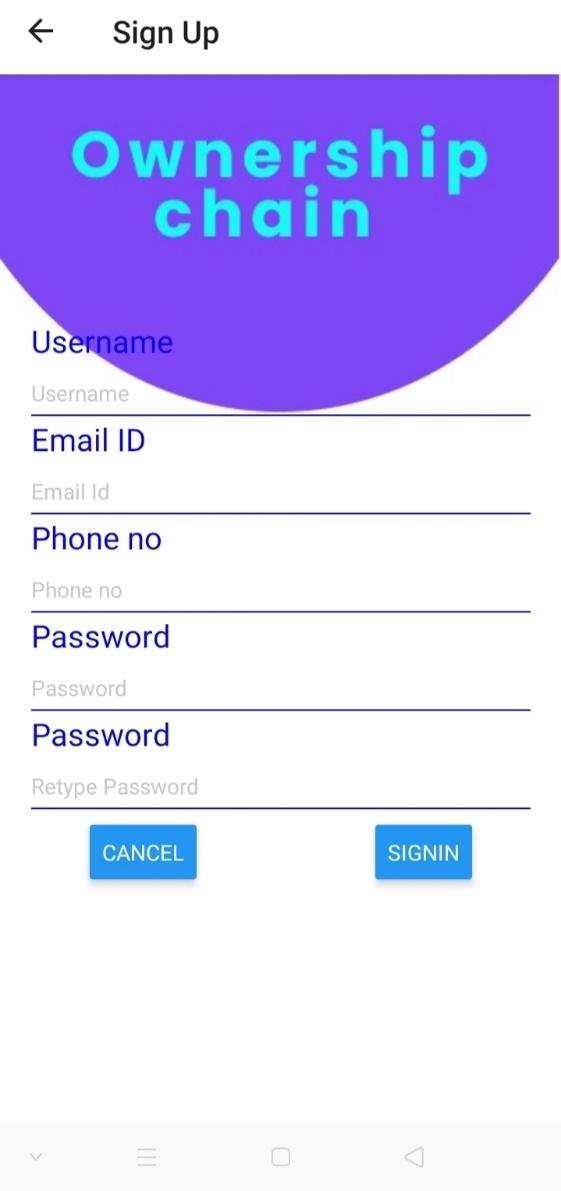


**Login Signup Screen-Forget Password Screen-**



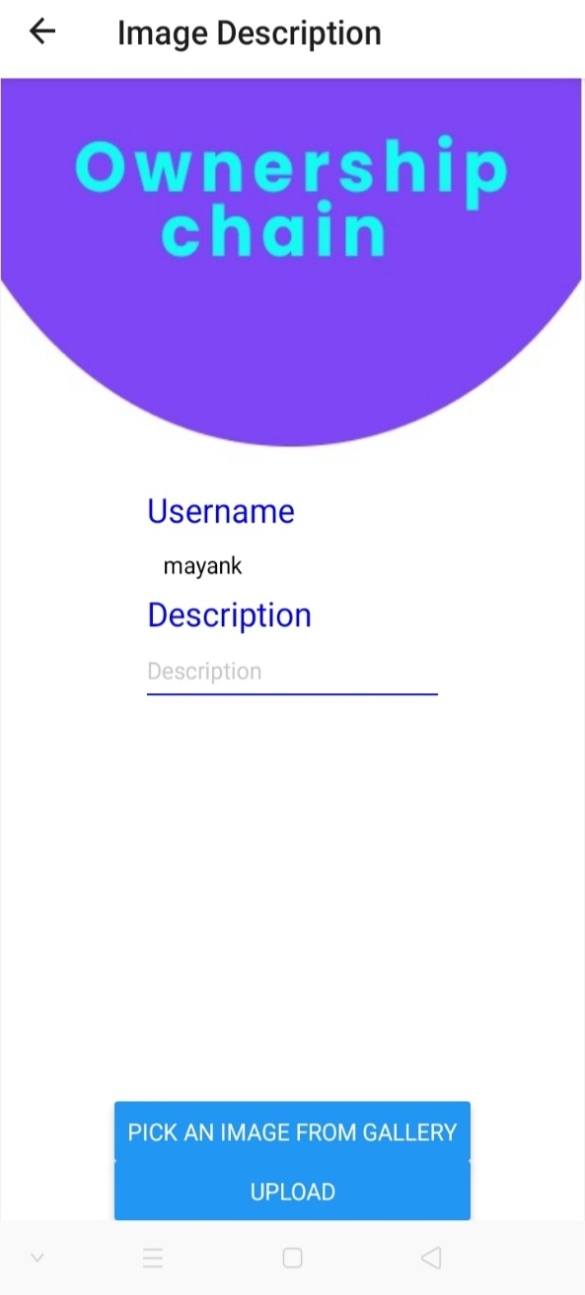
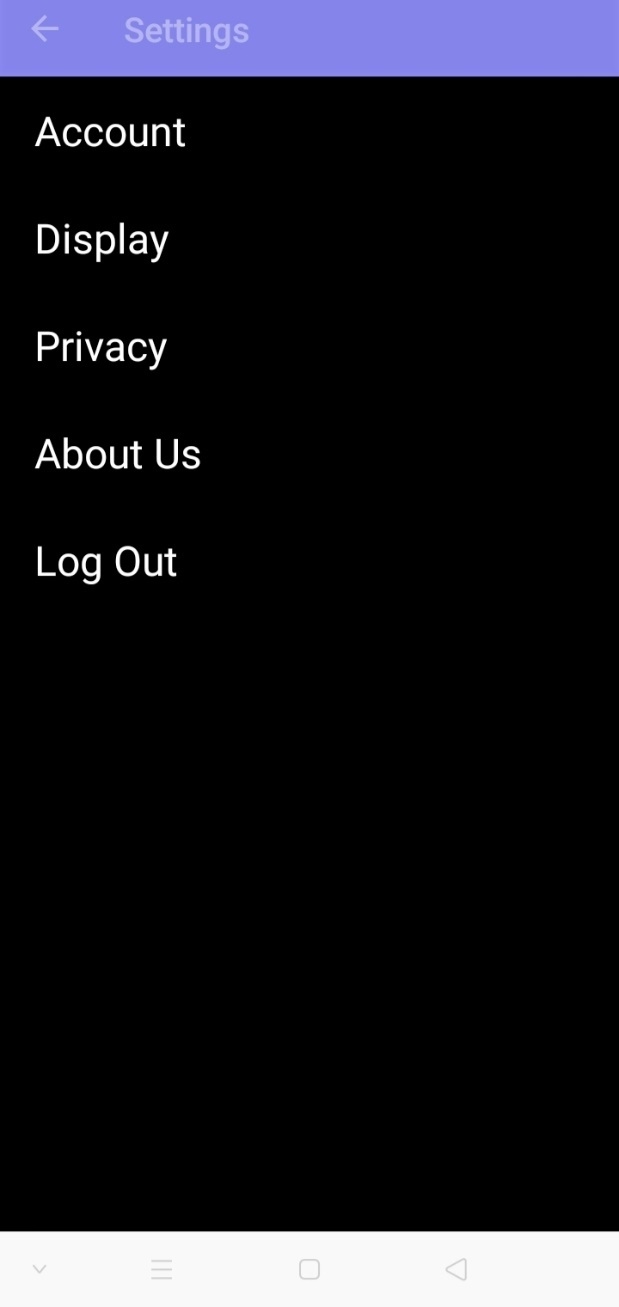


**Signup Screen-Profile Screen-**





**Setting Screen-Attach Image Screen-**





**SCREEN WISE CODE**

* 1. **App.js-**

import 'react-native-gesture-handler';

import React ,{useEffect} from 'react';

import {ScrollView,Animated,View,Image,Dimensions,Text} from 'react-native';

import SignUpp from './components/signup1';

import home from './components/home';

import Login from './components/loginScreen';

import Loginsignup from './components/loginSignupScreen';

import CameraScreen from './components/cameraScreen';

import Gallery from './components/gallery';

import ImageDescription from './components/imageDes';

import AddImage from './components/addImage';

import SettingScreen from './components/settingScreen';

import ChangePassword from './components/changePassword1';

import { NavigationContainer } from '@react-navigation/native';

import {createStackNavigator} from '@react-navigation/stack';

import {createDrawerNavigator} from '@react-navigation/drawer';

import ProfileScreen from './components/profileScreen';

const Signupp=({navigation})=>{return <SignUpp navigation={navigation}/>}

// const imageDes=({navigation,route})=>{

//   const {image}=route.params;

//   console.log(image)

//   return <ImageDescription image={image}/>

// }

const addImg=({navigation})=>{

  const Stack=createStackNavigator();

  return(

    <Stack.Navigator>

      <Stack.Screen name="Add image camera or gallery" component={AddImage}/>

      <Stack.Screen name="Add image using Camera" component={CameraScreen} />

      <Stack.Screen name="Import image from gallery" component={Gallery} />

      <Stack.Screen name='Image Description' component={ImageDescription}/>

    </Stack.Navigator>

  )

  }

const profile=()=>{return <ProfileScreen />}

// const changePassword1=({route,navigation})=>

// {

//   const {userName}=route.params;

//   const user=JSON.stringify(userName);

//   return(<View>

//     <ChangePassword1 navigation={navigation}/>

//     </View>)

// }

const settings=()=>{

  const Stack=createStackNavigator();

return (

  <Stack.Navigator>

  <Stack.Screen

    name="Settings"

    component={SettingScreen}

    options={{

      headerTintColor: 'white',

      headerStyle: { backgroundColor: 'blue',opacity:0.4 },

    }}

  />

  <Stack.Screen

    name="Profile"

    component={profile}

    options={{ headerStyleInterpolator: forFade , headerStyle: { backgroundColor: 'blue',opacity:0.4 }}}

  />

</Stack.Navigator>

)

}

const forFade = ({ current, next }) => {

  const opacity = Animated.add(

    current.progress,

    next ? next.progress : 0

  ).interpolate({

    inputRange: [0, 1, 2],

    outputRange: [0, 1, 0],

  });

  return {

    leftButtonStyle: { opacity },

    rightButtonStyle: { opacity },

    titleStyle: { opacity },

    backgroundStyle: { opacity },

  };

};

const login=({navigation})=>{

  const Stack=createStackNavigator();

  return (

    <Stack.Navigator>

    <Stack.Screen

      name="Login"

      component={Login}

      options={{

        headerTintColor: 'black',

      }}

    />

    <Stack.Screen

      name="Forget Password"

      component={ChangePassword}

      options={{ headerStyleInterpolator: forFade, headerStyle: { backgroundColor: 'blue',opacity:0.4 } }}

    />

  </Stack.Navigator>

  )}

const mainDashboard=({navigation,route})=>{

  const Drawer=createDrawerNavigator();

return(

  <Drawer.Navigator initialRouteName="Home">

         <Drawer.Screen name="Home" component={home} options={{ title: 'my home' }}/>

         <Drawer.Screen name="Add image" component={addImg}

    />

         <Drawer.Screen name="Settings" component={settings} />

       </Drawer.Navigator>

)

}

 const mainScreenTimeout=({navigation})=>{

  const w=Math.round(Dimensions.get('window').width);

const h=Math.round(Dimensions.get('window').height);

useEffect(()=>{const timer=setTimeout(()=>{return navigation.navigate('Loginsignup')},1000)},[]);

  return(

    <View style={{ backgroundColor: 'white' }}>

    <View style={{ flex: 1, paddingTop: 0}}>

        <Image source={require('./components/images/mainScreen.png')} style={{width:w,height:h}} />

    </View>

 </View>

  )

}

const App=(props)=>{

const Stack=createStackNavigator();

  return(

         <NavigationContainer>

          <Stack.Navigator initialRouteName="Main Screen" >

            <Stack.Screen name="Main Screen" component={mainScreenTimeout} />

            <Stack.Screen name="Loginsignup" component={Loginsignup} />

            <Stack.Screen name="Login" component={login} />

            <Stack.Screen name="Sign Up" component={Signupp} />

            <Stack.Screen name="Forget Password" component={ChangePassword} />

            <Stack.Screen name="Home" component={mainDashboard}  options={{headerShown:false}}/>

          </Stack.Navigator>

       </NavigationContainer>

)

}

 export default App;

* 1. **loginScreen.js-**

import React, { Component } from 'react';

import { StyleSheet, Text, View, ImageBackground,Button,TouchableOpacity,TouchableHighlight, TextInput, Alert} from 'react-native';

import { username1 } from './signup1';

let username2=''

class Login extends Component{

    constructor(props){

        super(props);

        this.state={username:'',password:''};

        this.passwordChange=this.passwordChange.bind(this);

        }

    passwordChange()

    {

        this.props.navigation.navigate('Forget Password')

    }

    search1=(responseJson)=>{

        if(responseJson=='Searched'){

            username2=this.state.username

            this.props.navigation.navigate('Home')

        }

        else{

            Alert.alert('ERROR','INCORRECT DETAILS')

        }

    }

    search = () => {

        // alert('ok');

        fetch('http://192.168.43.167:8080/ppphp/ownership\_chain\_users\_search.php', {

            method: 'post',

            header: {

                'Accept': 'application/json',

                'Content-type': 'application/json'

            },

            body: JSON.stringify({ username: this.state.username, password: this.state.password })

        }).then((response) => response.json()).then((responseJson) => {this.search1(responseJson) }).catch((error) => { console.error(error); })

    }

    render(){

    return (

        <View style={styles.container}>

            <ImageBackground source={require('./images/Login.jpeg')} style={{ flex:1,justifyContent: 'center'}}>

                <View style={styles.container}>

                    <Text style={styles.label}>

                        Username or Email ID

                    </Text>

                    <TextInput placeholder="Username or Email ID" onChangeText={(value)=>this.setState({username:value})} style={styles.textInput} value={this.state.username}/>

                    <Text style={{...styles.label,marginTop:10}}>

                        Password

                    </Text>

                    <TextInput secureTextEntry={true} placeholder="Password" style={styles.textInput} onChangeText={(value)=>this.setState({password:value})} value={this.state.password}/>

                    <TouchableHighlight underlayColor="black" style={{marginLeft:20,marginTop:20,marginRight:'45%',}} >

                        <Text style={styles.text} onPress={this.passwordChange}>

                            Forget Password ?

                        </Text>

                        </TouchableHighlight>

                        <TouchableHighlight underlayColor="white" style={{marginLeft:20,marginTop:20,marginRight:'45%',}} >

                        <Text style={styles.text} onPress={()=>{navigation.navigate('Sign Up')}}>

                            Sign Up

                        </Text>

                    </TouchableHighlight>

                    <TouchableOpacity activeOpacity={0.7} style={{paddingVertical:10,backgroundColor:'#f5c6ee',marginLeft:90,marginRight:90,borderRadius:10,marginTop:30}}>

                    <View>

                        <Text style={{textAlign:'center',color:'blue',fontWeight:'bold',fontSize:20}} onPress={this.search}>

                            LOGIN

                        </Text>

                    </View>

                </TouchableOpacity>

                </View>

            </ImageBackground>

        </View>

    )}

}

const styles=StyleSheet.create({

    textInput:{

        marginLeft:20,

        height:30,

        borderBottomWidth:1,

        marginRight:20,

        borderBottomColor:'blue',

    },

    label:{

        marginLeft:20,

        fontSize:20,

        color:'blue',

        marginBottom:5,

        marginTop:'50%',

    },

    container:{

        flex: 1,

        justifyContent:'center',

    },

    text:{

        color:'blue',

        fontSize:20

    }

})

export default Login;

export {username2}



* 1. **loginSignupScreen.js-**

import React, { Component } from 'react';

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

class Loginsignup extends Component {

    constructor(props){

        super(props);

    }

    render() {

        return (

            <View style={{ flex: 1, flexDirection: "column" }}>

                <ImageBackground source={require('./images/12.jpeg')} style={{ flex: 1, justifyContent: 'center' }}>

                    <TouchableOpacity activeOpacity={0.7} style={{ borderColor: 'blue', backgroundColor: '#f5c6ee', paddingVertical: 14, marginTop: 100, marginBottom: 10, marginLeft: 90, marginRight: 90, borderRadius: 10 }} onPress={() =>this.props.navigation.navigate("Login")}>

                        <View>

                            <Text style={{ textAlign: 'center', color: 'blue', fontWeight: 'bold', fontSize: 20 }}>

                                LOGIN

                        </Text>

                        </View>

                    </TouchableOpacity>

                    <TouchableOpacity activeOpacity={0.7} style={{ borderColor: 'blue', backgroundColor: '#f5c6ee', paddingVertical: 14, marginLeft: 90, marginRight: 90, borderRadius: 10, marginTop: 50 }} onPress={() =>this.props.navigation.navigate("Sign Up")}>

                        <View>

                            <Text style={{ textAlign: 'center', color: 'blue', fontWeight: 'bold', fontSize: 20 }}>

                                SIGN UP

                        </Text>

                        </View>

                    </TouchableOpacity>

                </ImageBackground>

            </View>

        );

    }

}

export default Loginsignup

* 1. **signup1.js-**

import React, { Component } from 'react';

import { View, StyleSheet, Text, ImageBackground, TouchableHighlight, TouchableOpacity, TextInput, Alert,Button } from 'react-native';

let username1='u'

let email1='u'

let phoneno1='u'

class Signupp extends Component {

    constructor(props) {

        super(props);

        this.state = { username: '', emailid: '', phoneno: '', password: '', repassword: '' };

    }

    check = () => {

        if (this.state.password==this.state.repassword){

            username1=this.state.username

            email1=this.state.emailid

            phoneno1=this.state.phoneno

            this.InsertDataToServer()

        }

        else{

            Alert.alert('ERROR','WRONG PASSWORD')

        }

    }

    InsertDataToServer = () => {

        fetch('http://192.168.43.167:8080/ppphp/ownership\_chain\_users\_save.php', {

            method: 'POST',

            headers: {

                'Accept': 'application/json',

                'Content-Type': 'application/json',

            },

            body: JSON.stringify({

                username: this.state.username,

                emailid: this.state.emailid,

                phoneno: this.state.phoneno,

                password: this.state.password

            })

        }).then((response) => response.json())

            .then((responseJson) => {

                // Showing response message coming from server after inserting records.

                console.log(responseJson)

            }).catch((error) => {

                console.error(error);

            });

            this.props.navigation.navigate('Login')

    }

    render() {

        return (

            <View style={styles.container}>

                <ImageBackground source={require('./images/Login.jpeg')} style={{ flex: 1, justifyContent: 'center' }}>

                    <View style={styles.container}>

                        <Text style={styles.label}>

                            Username

                        </Text>

                        <TextInput placeholder="Username" onChangeText={(value) => this.setState({ username: value })} style={styles.textInput} value={this.state.username} />

                        <Text style={styles.label}>

                            Email ID

                        </Text>

                        <TextInput placeholder="Email Id" onChangeText={(value) => this.setState({ emailid: value })} style={styles.textInput} value={this.state.emailid} />

                        <Text style={styles.label}>

                            Phone no

                        </Text>

                        <TextInput placeholder="Phone no" onChangeText={(value) => this.setState({ phoneno: value })} style={styles.textInput} value={this.state.phoneno} />

                        <Text style={styles.label}>

                            Password

                        </Text>

                        <TextInput secureTextEntry={true} placeholder="Password" onChangeText={(value) => this.setState({ password: value })} style={styles.textInput} value={this.state.password} />

                        <Text style={styles.label}>

                            Password again

                        </Text>

                        <TextInput secureTextEntry={true} placeholder="Retype Password" onChangeText={(value) => this.setState({ repassword: value })} style={styles.textInput} value={this.state.repassword} />

                        <View style={{ flexDirection: 'row-reverse', justifyContent: 'space-around', marginTop: 10 }}>

                            <Button title='Signin' onPress= {this.check} />

                            <Button title='Cancel' onPress={()=>this.props.navigation.navigate('Login')} />

                        </View>

                    </View>

                </ImageBackground>

            </View>

        );

    }

}

export default Signupp;

export {username1,email1,phoneno1}

const styles = StyleSheet.create({

    textInput: {

        marginLeft: 20,

        height: 30,

        borderBottomWidth: 1,

        marginRight: 20,

        borderBottomColor: 'blue',

    },

    label: {

        marginLeft: 20,

        fontSize: 20,

        color: 'blue',

        marginBottom: 5,

        marginTop: 1,

    },

    container: {

        flex: 1,

        justifyContent: 'center',

    },

    text: {

        color: 'blue',

        fontSize: 20

    }

})



* 1. **home.js-**

import React,{Component} from 'react';

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

import { FlatList } from 'react-native-gesture-handler';

import {username2} from './loginScreen'

class List extends Component{

    constructor(props){

        super(props);

        this.state={

            dataPeople:[],

            isLoading:true,

            // username:'',

        };

    }

    componentDidMount=()=>{

        this.fetchItem()

    }

    fetchItem=()=>{

        fetch('http://192.168.43.167:8080/ppphp/ownership\_chain\_imagesearch.php',{method:'POST',header:{

            'Accept': 'application/json',

            'Content-type': 'application/json'

        },

        body:JSON.stringify({

            username:username2,

        })

    }).then((response)=>response.json()).then((responseJson)=>{

        console.log(responseJson);

        this.setState({

            isLoading:false,

            dataPeople:responseJson.item

        });

    }).catch((error)=>{

        console.error(error)

    })

    };

    itemComponent=({item})=>{

        return(

            <View style={{flex:1,flexDirection:'row',marginBottom:3}}>

                <Image style={{width:80,height:80,margin:5}} source={{uri:item.photo}}/>

                <View style={{flex:1,justifyContent:'center'}}>

                    <Text styel={{fontsize:18,color:'green',marginBottom:15}}>

                        {item.description}

                    </Text>

                    <Text styel={{fontsize:16,color:'red'}}>

                        {item.date\_time}

                    </Text>

                </View>

            </View>

        )

    }

    seperatorComponent=()=>{

        return(

            <View style={{backgroundColor:'grey',height:0.5}}></View>

        )

    }

    render(){

        return(

            <FlatList

            data={this.state.dataPeople}

            renderItem={this.itemComponent}

            keyExtractor={(item,index)=>index.toString()}

            ItemSeparatorComponent={this.seperatorComponent}/>

        )

    }

}

export default List;

* 1. **profileScreen.js-**

import React from 'react';

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

import {username1,email1,phoneno1} from './signup1';

const w=Math.round(Dimensions.get('window').width);

const h=Math.round(Dimensions.get('window').height);

const ProfileScreen =()=>{

    return (

        <View>

          <Image source={require('./images/profileLL.png')} style={{resizeMode:'cover',width:w,height:h/3+140}} />

    <View style={{marginTop:30}}>

        <View style={{flexDirection:'row'}}>

            <Text style={{flex:1,marginLeft:20,marginBottom:10,fontSize:20,fontWeight:'400'}}>Username</Text>

    <Text style={{flex:2,marginBottom:10,fontSize:18,fontWeight:'200'}}>{username1}</Text>

        </View>

        <View style={{flexDirection:'row',justifyContent:'space-around'}}>

            <Text style={{flex:1,marginLeft:20,marginBottom:10,fontSize:20,fontWeight:'400'}}>Email Id</Text>

    <Text style={{flex:2,marginBottom:10,fontSize:18,fontWeight:'200'}}>{email1}</Text>

        </View>

        <View style={{flexDirection:'row',justifyContent:'space-around'}}>

            <Text style={{flex:1,marginLeft:20,marginBottom:10,fontSize:20,fontWeight:'400'}}>Contact no.</Text>

    <Text style={{flex:2,marginBottom:10,fontSize:18,fontWeight:'200'}}>{phoneno1}</Text>

        </View>

        <View style = {styles.changeButton}>

        <TouchableOpacity activeOpacity={0.95} style={styles.button}>

                      <Text style={styles.buttonText}

                      >Edit</Text>

                  </TouchableOpacity>

      </View>

    </View>

        </View>

    )

}

const styles=StyleSheet.create({

  button:{

    borderRadius:50,

    width:"35%",

    height:35,

    backgroundColor: "#6960EC",

    alignItems: "center",

    justifyContent: "center"

  },

  buttonText: {

    color: "white",

    fontWeight: "bold",

    padding: 8

  },

  changeButton: {

    alignItems: "center",

  },

})

export default ProfileScreen;

* 1. **settingScreen.js-**

import React from 'react';

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

export default function SettingScreen(props)

{const navigation=props.navigation;

    return (

        <View style={styles.conatiner}>

            <Text onPress={()=>{navigation.navigate('Profile')}} style={styles.text}>Account</Text>

            <Text style={styles.text}>Display</Text>

            <Text style={styles.text}>Privacy</Text>

            <Text style={styles.text}>About Us</Text>

            <Text onPress={()=>{navigation.navigate('Login')}} style={styles.text}>Log Out</Text>

        </View>

    )

}

const styles=StyleSheet.create({

    conatiner:{

        flex:1,

        backgroundColor:'black',

    },

    text:{

        marginVertical:15,

        fontSize:24,

        color:'white',

        marginHorizontal:20

    }

})

* 1. **changePassword1.js-**

import React,{Component} from 'react';

import {View,StyleSheet,Text,ImageBackground,TouchableHighlight,TouchableOpacity,TextInput,Alert} from 'react-native';

class ChangePassword extends Component{

    constructor(props){

        super(props);

        this.state = { username: '',email:'',phoneno:'',newpassword1: '' };

    }

    Next=()=>{

        Alert.alert('Change the password')

        this.props.navigation.navigate('Login')

    }

    InsertDataToServer = () =>{

       fetch('http://192.168.43.167:8080/ppphp/ownership\_chain\_forget\_password.php', {

         method: 'POST',

         headers: {

           'Accept': 'application/json',

           'Content-Type': 'application/json',

         },

         body: JSON.stringify({

           username:this.state.username ,

           emailid: this.state.emailid,

           phoneno: this.state.phoneno,

           password1:this.state.newpassword1

         })

       }).then((response) => response.json())

             .then((responseJson) => {

       // Showing response message coming from server after inserting records.

               this.Next()

             }).catch((error) => {

               console.error(error);

             });

         }

    render(){

        return(

            <View style={styles.container}>

                <ImageBackground source={require('./images/Login.jpeg')} style={{ flex: 1, justifyContent: 'center' }}>

                    <View style={{marginTop:'50%'}}>

                        <Text style={styles.label}>

                            Username

                        </Text>

                        <TextInput placeholder="Username" onChangeText={(value) => this.setState({ username: value })} style={styles.textInput} value={this.state.username} />

                        <Text style={styles.label}>

                            Email ID

                        </Text>

                        <TextInput placeholder="Email Id" onChangeText={(value) => this.setState({ emailid: value })} style={styles.textInput} value={this.state.emailid} />

                        <Text style={styles.label}>

                            Phone no

                        </Text>

                        <TextInput placeholder="Phone no" onChangeText={(value) => this.setState({ phoneno: value })} style={styles.textInput} value={this.state.phoneno} />

                        <Text style={styles.label}>

                            New Password

                        </Text>

                        <TextInput placeholder="New Password" onChangeText={(value) => this.setState({ newpassword1: value })} style={styles.textInput} value={this.state.newpassword1} />

                        <TouchableOpacity activeOpacity={0.7} style={{ paddingVertical: 10, backgroundColor: '#f5c6ee', marginLeft: 90, marginRight: 90, borderRadius: 10, marginTop: 30 }}>

                            <View>

                                <Text style={{ textAlign: 'center', color: 'blue', fontWeight: 'bold', fontSize: 20 }} onPress={this.InsertDataToServer}>

                                    Change

                        </Text>

                            </View>

                        </TouchableOpacity>

                    </View>

                </ImageBackground>

            </View>

        );

    }

}

export default ChangePassword;

const styles = StyleSheet.create({

    textInput: {

        marginLeft: 20,

        height: 30,

        borderBottomWidth: 1,

        marginRight: 20,

        borderBottomColor: 'blue',

    },

    label: {

        marginLeft: 20,

        fontSize: 20,

        color: 'blue',

        marginBottom: 5,

        marginTop: 13,

    },

    container: {

        flex: 1,

        justifyContent: 'center',

    },

    text: {

        color: 'blue',

        fontSize: 20

    }

})

* 1. **addImage.js-**

import React from 'react';

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

export default function AddImage({navigation})

{

    return(

        <View style={{flex:1}}>

            <View>

            <Text>Adding Image </Text>

            </View>

            <View>

                <Button title="Camera" onPress={()=>{navigation.navigate('Add image using Camera')}} />

                <Text>Or</Text>

                <Button title="Import from Gallery" onPress={()=>{navigation.navigate('Image Description')}} />

            </View>

        </View>

    )

}

* 1. **cameraScreen.js-**

import React, { Component } from 'react';

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

import { Camera } from 'expo-camera';

import Toolbaar from './toolbaarCamera';

 const {width:winWidth,height:winHeight}=Dimensions.get('window');

 export default class CameraScreen extends Component {

   constructor(props)

   {

     super(props);

     this.state={

       hasCameraPermission:null,

       captures:[],

       flashMode:Camera.Constants.FlashMode.off,

       capturing:null,

       cameraType:Camera.Constants.Type.back,

       camera:null,

     }

     this.handleShortCapture=this.handleShortCapture.bind(this);

   }

    handleShortCapture=async()=>{

      const photoData=await this.state.camera.takePictureAsync();

     this.setState({capturing:false});

     console.log(photoData);

      this.setState({captures:[photoData,...this.state.captures]});

    }

    async componentDidMount() {

      this.setState({camera:await Camera.requestPermissionsAsync()});

      this.setState({hasCameraPermission :(this.state.camera.status === 'granted'?true:false)});

  };

  render()

  {

         if (this.state.hasCameraPermission === null) {

            return <View />;

        } else if (this.state.hasCameraPermission === false) {

            return <Text>Access to camera has been denied.</Text>;

        }

        return (

            <View style={{ flex: 1,flexDirection:'column',alignContent:'center',justifyContent:'center' }}>

          <View style={{height:600,width:320,backgroundColor:"black",alignSelf:'center'}}>

          <Camera style={{flex:6}} ref={camera => this.state.camera = camera}>

          </Camera>

          <View style={{flex:1}}>

          <Toolbaar capturing={this.state.capturing} flashMode={this.state.flashMode} cameraType={this.state.cameraType} setFlashMode={(flashMode)=>this.setState({flashMode})} setCameraType={(cameraType)=>this.setState({cameraType})}onShortCapture={this.handleShortCapture} />

          </View>

          </View>

          {/\* <View style={{flex:1}}>

          {this.state.captures.length>0 && <Gallery captures={this.state.captures}/>} \*/}

            {/\* </View> \*/}

            </View>

        );

  }

}



* 1. **gallery.js-**

import React,{Component} from 'react';

import { Button, Image, View } from 'react-native';

import \* as ImagePicker from 'expo-image-picker';

import Constants from 'expo-constants';

import \* as Permissions from 'expo-permissions';

export default class Gallery extends Component {

  constructor(props)

  {super(props)

    this.state = {

      image: null,

    };

  }

  render() {

    let { image } = this.state;

const navigation=this.props.navigation;

    return (

      <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>

        <Button title="Pick an image from gallery" onPress={this.\_pickImage} />

        {image && <Image source={{ uri: image }} style={{ width: 200, height: 200 }} />}

        <Button title="save" onPress={()=>{navigation.navigate('Image Description',{image:image})}} />

      </View>

    );

  }

  componentDidMount() {

    this.getPermissionAsync();

  }

  getPermissionAsync = async () => {

    if (Constants.platform.ios) {

      const { status } = await Permissions.askAsync(Permissions.CAMERA\_ROLL);

      if (status !== 'granted') {

        alert('Sorry, we need camera roll permissions to make this work!');

      }

    }

  };

  \_pickImage = async () => {

    try {

      let result = await ImagePicker.launchImageLibraryAsync({

        mediaTypes: ImagePicker.MediaTypeOptions.All,

        allowsEditing: true,

        aspect: [4, 3],

        quality: 1,

      });

      if (!result.cancelled) {

        this.setState({ image: result.uri });

        fetch()

      }

      console.log(result);

    } catch (E) {

      console.log(E);

    }

  };

}

* 1. **imageDes.js-**

import \* as React from 'react';

import { Button, Image, View, Text, TextInput, StyleSheet, ImageBackground, Alert } from 'react-native';

import \* as ImagePicker from 'expo-image-picker';

import Constants from 'expo-constants';

import \* as Permissions from 'expo-permissions';

import {username2} from './loginScreen'

export default class ImageDescription extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      username: '',

      image: null,

      description: ''

    }

  };

  render() {

    let { image } = this.state;

    return (

      <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>

        <ImageBackground source={require('./images/Login.jpeg')} style={{ flex: 1, justifyContent: 'center',paddingLeft:70,paddingRight:70 }}>

          <View style={styles.container}>

            <Text style={styles.label}>

              Username

            </Text>

            <TextInput style={{marginLeft:30}} defaultValue={username2} editable={false} />

            <Text style={styles.label}>

              Description

            </Text>

            <TextInput placeholder="Description" onChangeText={(value) => this.setState({ description: value })} style={styles.textInput} value={this.state.password} />

          </View>

          <Button title="Pick an image from gallery" onPress={this.\_pickImage}  />

          <Button title="Upload" onPress={this.\_pickupload} />

          {image && <Image source={{ uri: image }} style={{ width: 200, height: 200 }} />}

        </ImageBackground>

      </View>

    );

  }

  \_pickupload = () => {

    this.state.username=username2

    var date = new Date().getDate(); //Current Date

    var month = new Date().getMonth() + 1; //Current Month

    var year = new Date().getFullYear(); //Current Year

    var hours = new Date().getHours(); //Current Hours

    var min = new Date().getMinutes(); //Current Minutes

    var sec = new Date().getSeconds(); //Current Seconds

    let date1=date + '/' + month + '/' + year + ' ' + hours + ':' + min + ':' + sec;

    let data=new FormData();

    data.append('username',this.state.username)

    let s=this.state.image;

    const uriPart = s.split('.');

    const fileExtension = uriPart[uriPart.length - 1];

    data.append('image', {

      uri: this.state.image,

      name: this.state.image+`.${fileExtension}`,

      type: `image/${fileExtension}`

    })

    // data.append('image',this.state.image);

    data.append('description',this.state.description)

    data.append('date',date1)

    // this.\_picksave

    fetch('http://192.168.43.167:8080/ppphp/ownership\_chain\_imagesave.php',{method: 'POST',body:data}).then(response=>{

      if(!response.ok){

        throw new Error('oops something is wrong');

      }

    }).catch((err)=>{

      console.log(err);

    });

    this.props.navigation.navigate('Home')

  }

  componentDidMount() {

    this.getPermissionAsync();

  }

  getPermissionAsync = async () => {

    if (Constants.platform.ios) {

      const { status } = await Permissions.askAsync(Permissions.CAMERA\_ROLL);

      if (status !== 'granted') {

        alert('Sorry, we need camera roll permissions to make this work!');

      }

    }

  };

  \_pickImage = async () => {

    try {

      let result = await ImagePicker.launchImageLibraryAsync({

        mediaTypes: ImagePicker.MediaTypeOptions.All,

        allowsEditing: true,

        aspect: [4, 3],

        quality: 1,

      });

      if (!result.cancelled) {

        this.setState({ image: result.uri });

      }

      console.log(result);

    } catch (E) {

      console.log(E);

    }

  };

}

const styles = StyleSheet.create({

  textInput: {

    marginLeft: 20,

    height: 30,

    borderBottomWidth: 1,

    marginRight: 20,

    borderBottomColor: 'blue',

  },

  label: {

    marginLeft: 20,

    fontSize: 20,

    color: 'blue',

    marginBottom: 5,

    marginTop: 1,

  },

  container: {

    flex: 1,

    justifyContent: 'center',

  },

  text: {

    color: 'blue',

    fontSize: 20

  }

})

* 1. **toolbarCamera.js-**

import React,{useState} from 'react';

import { Camera } from 'expo-camera';

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

import { Col, Row, Grid } from "react-native-easy-grid";

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

const {width:winWidth,height:winHeight}=Dimensions.get('window');

export default function Toolbaar({capturing=false,cameraType=Camera.Constants.Type.back,flashMode=Camera.Constants.FlashMode.off,setFlashMode,setCameraType,onShortCapture})

{

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

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

     return (

    <Grid style={styles.bottomToolbar}>

    <Row>

        <Col style={styles.alignCenter}>

            <TouchableOpacity onPress={() => setFlashMode(

                flashMode === Camera.Constants.FlashMode.off ? Camera.Constants.FlashMode.on : Camera.Constants.FlashMode.on

            )}>

                <Ionicons

                    name={flashMode == Camera.Constants.FlashMode.on ? "md-flash" : 'md-flash-off'}

                    color="white"

                    size={30}

                />

            </TouchableOpacity>

        </Col>

        <Col size={2} style={styles.alignCenter}>

            <TouchableWithoutFeedback

                onPress={onShortCapture}>

                <View style={[styles.captureBtn, capturing && styles.captureBtnActive]}>

                    {capturing && <View style={styles.captureBtnInternal} />}

                </View>

            </TouchableWithoutFeedback>

        </Col>

        <Col style={styles.alignCenter}>

            <TouchableOpacity onPress={() => setCameraType(

                cameraType === Camera.Constants.Type.back ? Camera.Constants.Type.front : Camera.Constants.Type.back

            )}>

                <Ionicons

                    name="md-reverse-camera"

                    color="white"

                    size={30}

                />

            </TouchableOpacity>

        </Col>

    </Row>

</Grid>

     )

}

const styles=StyleSheet.create({

    alignCenter: {

        flex: 1,

        alignItems: 'center',

        justifyContent: 'center',

    },

    bottomToolbar: {

        width: winWidth,

        position: 'absolute',

        alignItems:'center',

        height: 90,

        bottom: 0,

        marginLeft:-18

    },

    captureBtn: {

        width: 60,

        height: 60,

        borderWidth: 2,

        borderRadius: 60,

        borderColor: "#FFFFFF",

    },

    captureBtnActive: {

        width: 80,

        height: 80,

    },

    captureBtnInternal: {

        width: 76,

        height: 76,

        borderWidth: 2,

        borderRadius: 76,

        backgroundColor: "red",

        borderColor: "transparent",

    },

})



**REFERENCES**

* <https://reactnative.dev/>
* <https://reactnavigation.org/>