

# **Design Report 04**

---

EEI3269 introduction to Mobile Application Development

**R.G.B. C. RAJAGALGODA  
S92066157  
WD-OL-G7**

## **Contents**

Introduction.....	01
Source-code of App.js .....	02
Screenshot of Application .....	23

## **Introduction**

This is the design class 04 design report. There are a few special notes and assumptions to inform.

- 1) All images in this application are directly obtained from some websites. ( URL attached. )
- 2) Included 3 Book categories in this application.
  - a. **Action and Adventure Books**
  - b. **Comic Books**
  - c. **Detective Books**
- 3) All the book title was randomly selected.
- 4) All the book details were obtained from random websites such as Google, Wikipedia, Amazon etc
- 5) Source code of App.js will be included in this report. **App.js** file will be uploaded to Google drive & shareable link of that file will be attached. ( The whole application is in one .js file. )

**Link to App.js :**

[https://drive.google.com/file/d/1Ds2lqD4YL\\_HjCHmtZn7zqOr3TBNrZWlY/view?usp=sharing](https://drive.google.com/file/d/1Ds2lqD4YL_HjCHmtZn7zqOr3TBNrZWlY/view?usp=sharing)

**Link to High Quality Screenshots :**

[https://drive.google.com/drive/folders/15K69M5HnKMJgj6qK\\_mM0Gr4wAqgvGvUE?usp=sharing](https://drive.google.com/drive/folders/15K69M5HnKMJgj6qK_mM0Gr4wAqgvGvUE?usp=sharing)

## Source Code of App.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, TextInput, Image, Button, FlatList, Alert, ScrollView } from 'react-native';
import { NavigationContainer, StackActions, TabActions } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import react, {useState} from 'react';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';

const Stack = createNativeStackNavigator();

const App =() => {
  return(
    <View style = {styles.container}>
<NavigationContainer>
  <Stack.Navigator>
    <Stack.Screen name = "Welcome Screen" component={Welcome} options={{ title: "Welcome to Book Reader", headerTitleAlign: 'center' }} />
    <Stack.Screen name = "Registration" component={Reg} options={{ title: "User Registration", headerTitleAlign: 'center' }} />
    <Stack.Screen name = "Log in" component={Login} options={{ title: "User Log in", headerTitleAlign: 'center' }} />
    <Stack.Screen name = "Registration Complete" component={RegistrationComplete} options={{headerTitleAlign: 'center'}}/>
    <Stack.Screen name = "Reset PW" component={Reset} options={{ title: "Reset Login Password", headerTitleAlign: 'center' }} />
    <Stack.Screen
      name="Home"
      component={Home}
      options={{ headerShown: false }}
    />
  </Stack.Navigator>
</NavigationContainer>

  </View>
)
}

const Tab = createBottomTabNavigator();

function Home() {
  return (
    <Tab.Navigator
      screenOptions={{
        tabBarLabelPosition: "beside-icon",
        tabBarLabelStyle: {
          fontWeight: "700",
        }
      }}
    <Tab.Screen
      name="Home"
      component={Home}
      options={{ headerShown: false }}
    />
  </Tab.Navigator>
)

}

function Welcome() {
  return (
    <View style = {styles.container}>
      <Text>Welcome to Book Reader</Text>
      <Text>Please log in or register</Text>
    </View>
  )
}

function Reg() {
  return (
    <View style = {styles.container}>
      <Text>Registration</Text>
      <Text>Please enter your details</Text>
    </View>
  )
}

function Login() {
  return (
    <View style = {styles.container}>
      <Text>Log in</Text>
      <Text>Please enter your credentials</Text>
    </View>
  )
}

function RegistrationComplete() {
  return (
    <View style = {styles.container}>
      <Text>Registration Complete</Text>
      <Text>Please check your email for verification code</Text>
    </View>
  )
}

function Reset() {
  return (
    <View style = {styles.container}>
      <Text>Reset PW</Text>
      <Text>Please enter your email</Text>
    </View>
  )
}
```

```

        fontSize: 10
    },
    tabBarIconStyle: { display: "none" },
})
>
    <Tab.Screen name = 'Dashboard' component={Dashboard} options={{ headerShown: false }} />
    <Tab.Screen name = 'ActionBooks' component={ActionBooks} options={{ headerShown: false }}/>
    <Tab.Screen name = 'ComicBooks' component={ComicBooks} options={{ headerShown: false }}/>
    <Tab.Screen name = 'DetectiveBooks' component={DetectiveBooks} options={{ headerShown: false }}/>
</Tab.Navigator>
);
}

const Welcome = ({navigation}) => {
    return (
<View style = { styles.container}>

<View style = {styles.container2}>

<Image source={{
height: 310,
width:390,
uri: "http://clipart-library.com/images/pi78gKzxT.png"} } />

<View style = {styles.button}>
    <Button
        onPress = { () => navigation.navigate('Registration')}
        title = "Register"
        color = "#266e73"
    />
</View>
<View style = {styles.button}>
    <Button
        onPress = { () => navigation.navigate('Log in')}
        title = "Log in"
        color = "#266e73"
    />
</View>
</View>
</View>
);
}

const Reset = ({navigation}) => {

```

```

const [email, setEmail] = useState('');
const [pw, setPW] = useState('');
const [rpw, setRPW] = useState('');
const [pin, setPIN] = useState('');

const Go = (text) => {

let mail = /^[^\w+([.-]?[\w+])*@\w+([.-]?[\w+])*(\.\w\w+)+$/;
if (mail.test(text) === false) {
  Alert.alert("Error", "Invalid Email Address !");
}
else {
  Alert.alert("Success", "Check your email for PIN and Enter Received PIN Below!");
} }

const validate =() => {

if(pin == ''){
  Alert.alert("Error", "Enter PIN");
}
else {
  Alert.alert("Prompt", "Enter your new password below !");
} }

const validatepw =() => {
if (pw == '' || rpw == '') {
  Alert.alert("Error", "Enter Password and Re - Password");
}
else if( pw != rpw) {
  Alert.alert("Error", "Password and Re - Password must be the same");
}
else {
  Alert.alert("Success", "Password Reset Successfully. You will be redirected to log in screen !",
    [{text: 'OK' ,onPress:()=> navigation.navigate('Log in'),}],);
} }
return (

```

<View style = {styles.container}>

<View style = {styles.container2}>

<TextInput style={styles.inputcontent}  
placeholder="Enter Email"  
onChangeText={newEmail => setEmail(newEmail)}  
defaultValue=""  
/>

```

<View style ={styles.button}>
  <Button
    onPress = {() => Go(email)}
    title = "Request PIN"
    color = "#266e73" />
</View>

<View style = {{ marginTop: 40}}>
  <TextInput style={styles.inputcontent}
    placeholder="Enter Received PIN "
    onChangeText={text => setPIN(text)}
    defaultValue=""
  />
</View>

<View style ={styles.button}>
  <Button
    onPress = {() => validate()}
    title = "PIN Validate"
    color = "#266e73" />
</View>

<View style = {{ marginTop: 40}}>
  <TextInput style={styles.inputcontent}
    secureTextEntry={true}
    placeholder="Enter Password"
    onChangeText={newText => setPW(newText)}
    defaultValue=""
  />

  <TextInput style={styles.inputcontent}
    secureTextEntry={true}
    placeholder="Re - Enter Password"
    onChangeText={newText => setRPW(newText)}
    defaultValue=""
  />
</View>

<View style ={styles.button}>
  <Button
    onPress = {() => validatepw()}
    title = "Reset Password"
    color = "#266e73" />
</View>

```

```

        </View>
    </View>
);
}

const Login = ({navigation}) => {
    const [userName, setUserName] = useState('');
    const [pw, setPW] = useState('');

    const Go = (text) => {

        let mail = /^[\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w\w+)+$/;
        if (mail.test(text) === false) {
            Alert.alert("Error", "Invalid Email Address !")
        }

        else if(userName == '' || pw == '') {
            Alert.alert("Error", "Check your inputs")
        }

        else {
            navigation.navigate('Home')
        }
    }
    return (
        <View style = {styles.container}>
            <View style = {styles.container2}>
                <TextInput style={styles.inputcontent}
                    placeholder="Enter Email"
                    onChangeText={newText => setUserName(newText)}
                    defaultValue=""
                />
                <TextInput style={styles.inputcontent}
                    secureTextEntry={true}
                    placeholder="Enter Password"
                    onChangeText={newText => setPW(newText)}
                    defaultValue=""
                />
            <View style = {styles.button}>
                <Button

                    onPress = {()=> Go(userName)}
                    title = "Log in"

```

```

        color = "#266e73"
    />

    </View>
    <View style = {styles.button}>
        <Button
            color = "#266e73"
            onPress = {() => navigation.navigate('Welcome Screen')}
            title = "Back"
        />

    </View>

    <View style = {styles.button}>
        <Button
            color = "#266e73"
            onPress = {() => navigation.navigate('Reset PW')}
            title = "Reset Password"
        />

    </View>

    </View>
</View>
);
}

const Reg = ({navigation}) => {
    const [userName, setUserName] = useState('');
    const [email, setEmail] = useState('');
    const [pw, setPW] = useState('');
    const [rpw, setRPW] = useState('');
    const [age, setAge] = useState('');

    const Go = (text) => {

        let mail = /^[^\w+([.-]?[\w+])*@\w+([.-]?[\w+])*(\.\w\w+)+$/;
        if (mail.test(text) === false) {
            Alert.alert("Error", "Invalid Email Address !")
        }

        else if (isNaN(age)) {
            Alert.alert("Error", "Age must be numeric !")
        }

        else if(userName == '' || email == '' || age == '' || pw == '' || rpw == '') {
            Alert.alert("Error", "All Input field are Required !")
        }
    }
}

```

```

}

else if ( pw != rpw) {
  Alert.alert("Error", "Password and Re - Password must be the same !")
}

else {
  navigation.navigate('Registration Complete')
}
}

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

<View style = {styles.container2}>

<TextInput style={styles.inputcontent}
  placeholder="Enter Name"
  onChangeText={newName => setUserName(newName)}
  defaultValue=""
/>

<TextInput style={styles.inputcontent}
  placeholder="Enter Age"
  keyboardType='numeric'
  onChangeText={newText => setAge(newText)}
  defaultValue=""
/>

<TextInput style={styles.inputcontent}
  placeholder="Enter Email"
  onChangeText={newEmail => setEmail(newEmail)}
  defaultValue=""
/>

<TextInput style={styles.inputcontent}
  secureTextEntry={true}
  placeholder="Enter Password"
  onChangeText={newText => setPW(newText)}
  defaultValue=""
/>

<TextInput style={styles.inputcontent}
  secureTextEntry={true}
  placeholder="Re Enter Password"
  onChangeText={newText => setRPW(newText)}
  defaultValue=""
/>

```

```

        />

        <View style ={styles.button}>
          <Button
            onPress = {() => Go(email)}
            title = "Register"
            color = "#266e73" />
        </View>

        <View style ={styles.button}>
          <Button style = {styles.button}
            onPress = {() => navigation.navigate('Welcome Screen')}
            title = "Back"
            color = "#266e73" />
        </View>

      </View>
    </View>
  );
}

const RegistrationComplete = ({navigation}) => {
  return (
<View style = {styles.container}>

<View style = {styles.container2}>

  <Image source={{
height: 200,
width: 200,
uri: "https://www.ap3n.africa/wp-content/uploads/2020/09/check.png"
}} />

<Text style = {{ fontSize: 30, marginTop: 20, fontWeight: 'bold',}}>
  Registration Success!
</Text>

  <View style = {styles.button}>
    <Button style = {styles.button}
      onPress = {()=>{navigation.navigate('Log in')}}
      title = "Back to Log in"
      color = "#266e73"
    />
  </View>
</View>
</View>
);
}

```

```

}

const Dashboard = ({navigation}) => {
  const [dataFromState, setData] = useState(dataFromState)

  const [book, setBook] = useState([
    {Book: '1', Name : 'Harry Potter'},
    {Book: '2', Name : 'The Lord of the Rings'},
    {Book: '3', Name : 'The Golden Compass'},
    {Book: '4', Name : 'The Lightning Thief '},
    {Book: '5', Name : 'Shadow and Bone'},
    {Book: '6', Name : 'Action and Adventure Books'},
  ]);
}

const action=() => {
  setBook ([{Book: '1', Name : 'Harry Potter'},
  {Book: '2', Name : 'The Lord of the Rings'},
  {Book: '3', Name : 'The Golden Compass'},
  {Book: '4', Name : 'The Lightning Thief '},
  {Book: '5', Name : 'Shadow and Bone'},
  {Book: '6', Name : 'Action and Adventure Books'},]);
  Alert.alert('Promt', 'Do you want more information about Action books ?',
  [{text: 'Yes' ,onPress:()=> navigation.navigate('ActionBooks'),}, {text: 'No'}]);
}

const comic =() => {
  setBook ([{Book: '1', Name : 'Maus'},
  {Book: '2', Name : 'Batman: The Killing Joke'},
  {Book: '3', Name : 'All-star Superman'},
  {Book: '4', Name : 'Kingdom Come'},
  {Book: '5', Name : 'Ghost Circles'},
  {Book: '6', Name : 'Comic Books'},])
  Alert.alert('Prompt', 'Do you want more information about Comic Books ?',
  [{text: 'Yes', onPress:() => navigation.navigate('ComicBooks'),}, {text: 'No'}])
}

const detective = () => {
  setBook ([{Book: '1', Name : 'One of Us Is Lying'},
  {Book: '2', Name : 'Inheritance Games'},
  {Book: '3', Name : 'We Were Liars'},
  {Book: '4', Name : 'The Cellar'},
  {Book: '5', Name : 'Edgewood '},
  {Book: '6', Name : 'Detective Books'},]);
  Alert.alert('Prompt', 'Do you want more information about Detective Books ?',
  [{text: 'Yes', onPress:()=> navigation.navigate('DetectiveBooks'),}, {text: 'No'}]);
}

const olddata = [

```

```

{key: 'Harry Potter'},
{key: 'The Lord of the Rings'},
{key: 'The Golden Compass'},
{key: 'The Lightning Thief'},
{key: 'Shadow and Bone'},
{key: 'Maus'},
{key: 'Batman: The Killing Joke'},
{key: 'All-star Superman'},
{key: 'Kingdom Come'},
{key: 'Ghost Circles'},
{key: 'One of Us Is Lying '},
{key: 'Inheritance Games'},
{key: 'We Were Liars'},
{key: 'The Cellar'},
{key: 'Edgewood'},
]

const searchName=(input)=> {
let data = olldata
let searchData = data.filter((item) => {
  return item.key.toLowerCase().includes(input.toLowerCase())
});
setData(searchData) }

return(
<View style = {styles.BookDetails}>
  <View style ={styles.container2}>
    <Text style = {styles.head}>
      Book Categories
    </Text>

    <View style ={styles.new}>
      <Text style = {styles.bookhead1}
        onPress ={ action }> Action </Text>

      <Text style = {styles.bookhead1}
        onPress ={ comic }> Comic </Text>

      <Text style = {styles.bookhead1}
        onPress ={ detective }> Detective </Text>

    </View>

    <View style = {styles.container2}>
      <Text style ={styles.head}>
        {book[5].Name}
      </Text>

```

```

        </View>
        <View style ={styles.container2}>
            <View>
                <Text style = {styles.content}>1. {book[0].Name}</Text>
                <Text style = {styles.content}>2. {book[1].Name}</Text>
                <Text style = {styles.content}>3. {book[2].Name}</Text>
                <Text style = {styles.content}>4. {book[3].Name}</Text>
                <Text style = {styles.content}>5. {book[4].Name}</Text>
            </View>

        </View>

        </View>

        <View style = {styles.new }>
            <Text style = {styles.bookhead3}>SEARCH BOOKS : </Text>
            <TextInput style = {styles.bookhead3}
                placeholder="Enter book's name here to search.. "
                onChangeText={(input) => {searchName(input)} }
            />
        </View>

        <View style = {styles.container5}>
            <FlatList
                data = {dataFromState}
                renderItem={({item }) => <Text style = {styles.bookhead3}>{item.key}</Text>} />
        </View>

        <View style = {styles.container2}>
            <View style = {styles.button}>
                <Button style = {styles.button}
                    onPress = {()=>{navigation.popToTop()} }
                    title = "Logout"
                    color = "#266e73"
                />
            </View>
        </View>
    </View>
);

}

const ActionBooks = () => {
    const [book, setBook] = useState({
        Name: 'Harry Potter', Price: '90' , Author : 'J.K Rowlings',
        Pic : "https://www.jkrowling.com/wp-content/uploads/2016/10/HPATPS_Hero_OnGrey.png",

```

```

    info: 'The novels chronicle the lives of a young wizard, Harry Potter, and his
friends Hermione Granger and Ron Weasley, all of whom are students at Hogwarts School of
Witchcraft and Wizardry.'
});

const [num, setNum] = useState(0);
return (
<View style = {styles.BookDetails}>

<View style = {styles.container2}>

<View style = {styles.bookdetails}>
<Image source={{{
height: 200,
width: 117,
uri: book.Pic
}} />

<Text style = {styles.content}>
Book Name : {book.Name}
</Text>
<Text style = {styles.content}>
Author : {book.Author}
</Text>
<Text style = {styles.content}>
Price (INR) : {book.Price}
</Text>
<Text style = {styles.content}>
Description : {book.info}
</Text>

</View>

<View style = {styles.container4}>

<Text style = {styles.bookhead2}
onPress={()=> setBook({
Name: 'Harry Potter', Price: '90' , Author : 'J.K Rowlings',
Pic : "https://www.jkrowling.com/wp-
content/uploads/2016/10/HPATPS_Hero_OnGrey.png",
info: 'The novels chronicle the lives of a young wizard, Harry Potter, and his
friends Hermione Granger and Ron Weasley, all of whom are students at Hogwarts School of
Witchcraft and Wizardry.'})}
> Harry Potter </Text>
```

```

<Text style = {styles.bookhead2}
    onPress={()=> setBook({
        Name: "Lord of the Rings", Author: "J. R. R. Tolkien", Price : '100',
        Pic: "https://images-na.ssl-images-amazon.com/images/I/51kffFS5-
fnL._SX332_BO1,204,203,200_.jpg",
        info: "The Lord of the Rings is the saga of a group of sometimes reluctant
heroes who set forth to save their world from consummate evil. Its many worlds and
creatures were drawn from Tolkien's extensive knowledge of philology and folklore."
    })}
> Lord of the Rings </Text>

<Text style = {styles.bookhead2}
    onPress={()=> setBook({ Name: "The Golden Compass", Author: "Philip Pullman",
Price : '80',
        Pic : "https://images-na.ssl-images-amazon.com/images/I/91uLA-C1-kL.jpg",
        info: "The story line has all the hallmarks of a myth: brought up ignorant of her
true identity, 11-year-old Lyra goes on a quest from East Anglia to the top of the world
in search of her kidnapped playmate Roger and her imprisoned uncle, Lord Asriel."
    })}
> The Golden Compass </Text>

<Text style = {styles.bookhead2}
    onPress={()=> setBook({ Name: "The Lightning Thief", Author: "Rick Riordan", Price
: '110',
        Pic : "https://www.readriordan.com/wp-
content/uploads/2022/02/FINALPJ_LightningThief_rpkg_CVR.jpg",
        info: "The Lightning Thief is a light-hearted fantasy about a modern 12-year-old
boy who learns that his true father is Poseidon, the Greek god of the sea. Percy sets out
to become a hero by undertaking a quest across the United States to find the entrance to
the Underworld and stop a war between the gods."
    })}
> The Lightning Thief </Text>

<Text style = {styles.bookhead2}
    onPress={()=> setBook({ Name: "Shadow and Bone", Author: "Leigh Bardugo", Price :
'120',
        Pic : "https://prodimage.images.bn.com/pimages/9781250777881_p0_v4_s550x406.jpg",
        info: "Orphaned and expendable, Alina Starkov is a soldier who knows she may not
survive her first trek across the Shadow Fold—a swath of unnatural darkness crawling with
monsters. But when her regiment is attacked, Alina unleashes dormant magic not even she
knew she possessed."
    })}
> Shadow and Bone </Text>

```

```

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

const ComicBooks = () => {
  const [book, setBook] = useState({
    Name: 'Maus', Price: '90' , Author : 'Art Spiegelman',
    Pic :
    "https://upload.wikimedia.org/wikipedia/en/7/7d/Maus_%28volume_1%29_cover.jpg",
    info: "Maus is a nonfiction book presented in the graphic novel style, written by American cartoonist Art Spiegelman. Serialized from 1980 to 1991, it depicts Spiegelman interviewing his father about his experiences as a Polish Jew and Holocaust survivor."
  });

  const [num, setNum] = useState(0);
  return (
    <View style = {styles.BookDetails}>

      <View style = {styles.container2}>

        <ScrollView>
          <View style = {styles.bookdetails}>
            <Image source={{ height: 250, width: 167, uri: book.Pic }} />

            <Text style = {styles.content}>
              Book Name : {book.Name}
            </Text>
            <Text style = {styles.content}>
              Author : {book.Author}
            </Text>
            <Text style = {styles.content}>
              Price(INR) : {book.Price}
            </Text>
            <Text style = {styles.content}>
              Description : {book.info}
            </Text>

        </View>
    
```

```

<View style = {styles.container4}>

<Text style = {styles.bookhead2}
    onPress={()=> setBook({
        Name: 'Maus', Price: '90' , Author : 'Art Spiegelman',
        Pic :
        "https://upload.wikimedia.org/wikipedia/en/7/7d/Maus_%28volume_1%29_cover.jpg",
        info: "Maus is a nonfiction book presented in the graphic novel style, written by American cartoonist Art Spiegelman. Serialized from 1980 to 1991, it depicts Spiegelman interviewing his father about his experiences as a Polish Jew and Holocaust survivor."
    })}

    > Maus </Text>

<Text style = {styles.bookhead2}
    onPress={()=> setBook({
        Name: 'Batman: The Killing Joke', Author: 'Alan Moore', Price : '100',
        Pic: 'https://pictures.abebooks.com/isbn/9781401216672-us.jpg',
        info: 'Batman: The Killing Joke is a 1988 DC Comics one-shot graphic novel featuring the characters Batman and the Joker written by Alan Moore and illustrated by Brian Bolland. The Killing Joke provides an origin story for the supervillain the Joker, loosely adapted from the 1951 story arc "The Man Behind the Red Hood!," which was written by Batman co-creator Bill Finger.'
    })}

    > Batman: The Killing Joke </Text>

<Text style = {styles.bookhead2}
    onPress={()=> setBook({ Name: 'All-star Superman', Author: ' Frank Quitely', Price : '80',
        Pic: 'https://m.media-amazon.com/images/I/517EwXhW4oL.jpg',
        info: "After being poisoned by sun radiation, a dying Superman decides to fulfill his lifelong dreams while Lex Luthor has his own agenda. After being poisoned by sun radiation, a dying Superman decides to fulfill his lifelong dreams while Lex Luthor has his own agenda."
    })}

    > All-star Superman</Text>

<Text style = {styles.bookhead2}
    onPress={()=> setBook({ Name: 'Kingdom Come', Author: 'Mark Waid', Price : '110',
        Pic: 'https://images-na.ssl-images-amazon.com/images/I/81DgBmpvp-L.jpg',
        info: "A Novel. A violent novel filled with insidious twists, Kingdom Come follows the exploits of Richard Pearson, a rebellious, unemployed advertising executive, whose father is gunned down by a deranged mental patient in a vast shopping mall outside Heathrow Airport."
    })}
```

```

        })}

    > Kingdom Come </Text>

    <Text style = {styles.bookhead2}
        onPress={()=> setBook({ Name: 'Ghost Circles', Author: 'Jeff Smith', Price :
'130',
        Pic: 'https://m.media-amazon.com/images/I/51GmrXI02hL.jpg',
        info: " A long-dormant volcano explodes, blacking out the sun, mowing down trees,
and filling the land with soot and ash. The Lord of the Locusts is released with the
eruption. Against this apocalyptic backdrop, the Bone cousins along with Thorn and Grandma
Ben struggle to reach safe haven in the city of Atheia."
        })}

    > Ghost Circles</Text>

    </View>
    </ScrollView>
    </View>
    </View>
);

}

const DetectiveBooks = () => {
    const [book, setBook] = useState({
        Name: 'One of Us Is Lying', Author: 'Karen M. McManus', Price : '100',
        Pic: 'https://images-na.ssl-images-amazon.com/images/I/51mgwiNMAgL.jpg',
        info: "One of Us is Lying follows the gripping story of Bronwyn, Addy, Nate, and
Cooper as suspects in the murder of Simon Kelleher. Each of the high school students have
secrets that they would do anything to protect, so how far would they go to make sure
they're kept out of the spotlight?"
    });

    const [num, setNum] = useState(0);
    return (

```

```

<View style = {styles.BookDetails}>

<View style = {styles.container2}>
<ScrollView>
<View style = {styles.bookdetails}>
<Image source={{
    height: 250,
    width: 167,
    uri: book.Pic
}} />

<Text style = {styles.content}>
```

```

Book Name : {book.Name}
</Text>
<Text style = {styles.content}>
Author : {book.Author}
</Text>
<Text style = {styles.content}>
Price(INR) : {book.Price}
</Text>
<Text style = {styles.content}>
Description : {book.info}
</Text>

</View>

<View style = {styles.container4}>

<Text style = {styles.bookhead2}
onPress={()=> setBook({
Name: 'One of Us Is Lying', Author: 'Karen M. McManus', Price : '100',
Pic: 'https://images-na.ssl-images-amazon.com/images/I/51mgwiNMAgL.jpg',
info: "One of Us is Lying follows the gripping story of Bronwyn, Addy, Nate, and
Cooper as suspects in the murder of Simon Kelleher. Each of the high school students have
secrets that they would do anything to protect, so how far would they go to make sure
they're kept out of the spotlight?"
})}

> One of Us Is Lying</Text>

<Text style = {styles.bookhead2}
onPress={()=> setBook({
Name: 'Inheritance Games', Author: 'Jennifer Lynn Barnes', Price : '90',
Pic: 'https://i.gr-
assets.com/images/S/compressed.photo.goodreads.com/books/1587396413l/52439531.jpg',
info: "An utterly addictive and twisty thriller, full of dark family secrets and
deadly stakes. Perfect for fans of One of Us is Lying and Knives Out. She came from
nothing. Avery has a plan: keep her head down, work hard for a better future. Then an
eccentric billionaire dies, leaving her almost his entire fortune."
})}

> Inheritance Games </Text>

<Text style = {styles.bookhead2}
onPress={()=> setBook({ Name: 'We Were Liars', Author: 'E. Lockhart', Price :
'110',
Pic: 'https://images-na.ssl-images-amazon.com/images/I/916kdpY0YRL.jpg',
info: "Avery Grambs has a plan for a better future: survive high school, win a
scholarship, and get out. But her fortunes change in an instant when billionaire Tobias
Hawthorne dies and leaves Avery virtually his entire fortune"
})

```

```

        })}

    > We Were Liars </Text>

    <Text style = {styles.bookhead2}
          onPress={()=> setBook({ Name: 'The Cellar', Author: 'Natasha Preston', Price :
'80',
          Pic: 'https://images-na.ssl-images-
amazon.com/images/I/4180ICPt5YL._SX265_B01,204,203,200_.jpg',
          info: "Sixteen-year old Summer Robinson doesn't think anything ever happens in
the small town of Long Thorpe, England. But she strays from a party and is snatched by a
man who calls himself "Clover." He forces Summer into his cellar, where she finds three
other young women who were kidnapped just like her."
        })}

    >The Cellar </Text>

    <Text style = {styles.bookhead2}
          onPress={()=> setBook({ Name: 'Edgewood', Author: 'Karen McQuestion', Price :
'120',
          Pic: 'https://images-na.ssl-images-amazon.com/images/I/41FiciQpJRL.jpg',
          info: "Wandering the dark streets at night is Russ Becker's way of dealing with
his relentless insomnia and the angst of life. But that changes forever the night he
witnesses a strange astronomical event, then discovers he's developed incredible
superpowers. And he's not alone."
        })}> Edgewood</Text>

        </View>
        </ScrollView>
    </View>
    </View>
);

}

const styles = StyleSheet.create({
container: {
  flex: 1,
  backgroundColor: 'orange',
},
BookDetails: {
  flex: 1,
  backgroundColor: 'orange',
  marginTop: 20,
},
container2: {
  display: 'flex',

```

```
    alignItems: 'center',
    marginTop: 10,
    marginBottom: 10,
  },

  container3 : {
    margin: 5,
    display: 'flex',
    flexDirection: 'row',
    justifyContent: 'space-around',
    width: 370,
  },
}

container4: {
  display: 'flex',
  alignItems: 'center',
  marginTop: 10,
  marginBottom: 10,
  width: 400,
},
}

container5: {
  display: 'flex',
  marginLeft: 20,
  marginTop: 10,
  marginBottom: 10,
  height: 200,
},
}

btncontainer: {
  display: 'flex',
  alignItems: 'center',
  flexDirection: 'column',
  marginTop: 50,
  width: 300,
},
}

button: {
  marginTop: 50,
  width: "80%",
  backgroundColor: 'red',
},
}

listhead: {
  fontSize: 20,
  backgroundColor: 'red',
  alignItems: 'center',
},
```

```
content: {
  fontSize: 16,
  marginTop: 5,
  marginRight: 10,
  marginLeft: 10,
  textAlign: 'justify',
},

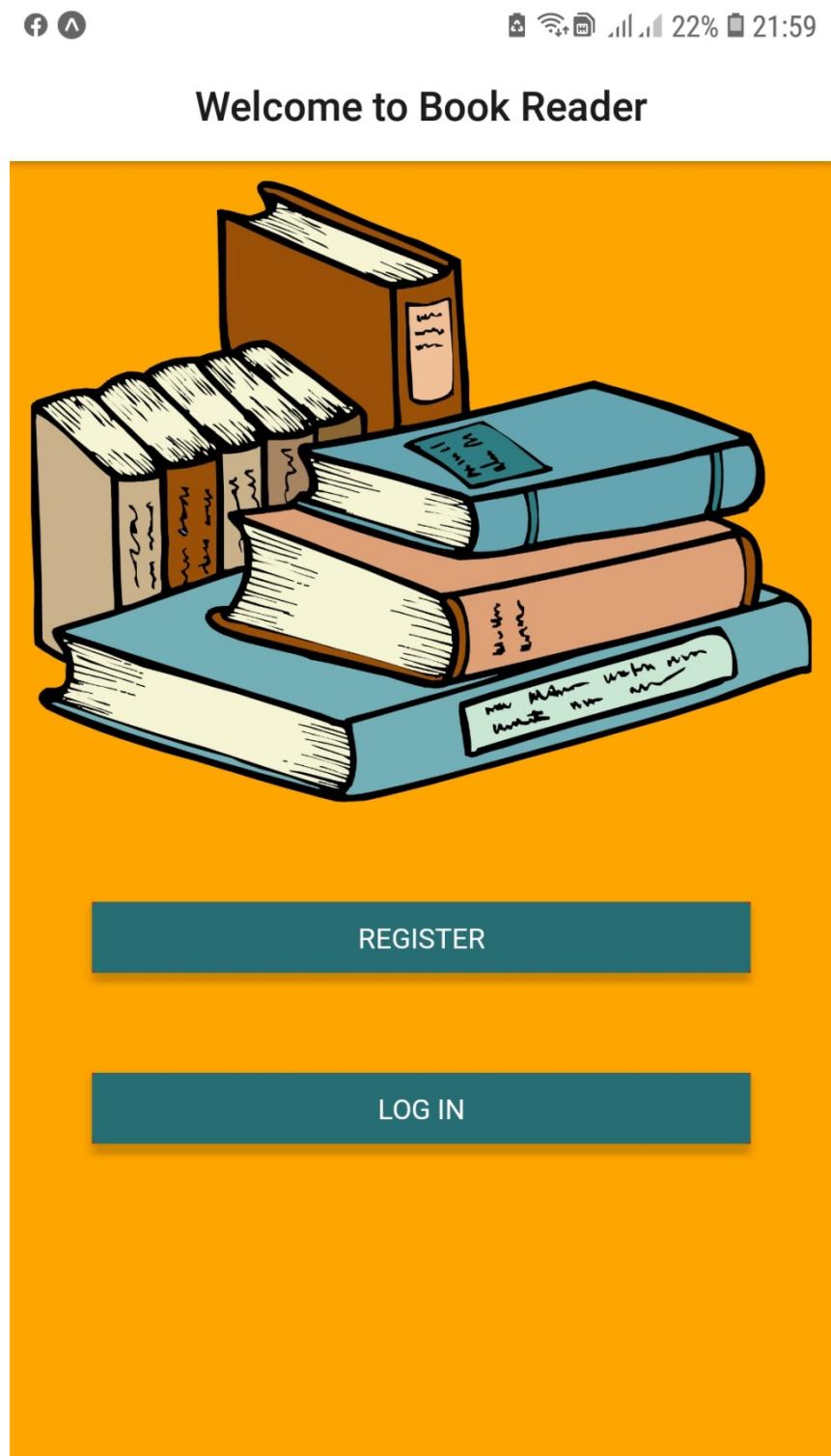
inputcontent: {
  marginTop: 10,
  fontSize: 20,
  textAlign: 'justify',
  backgroundColor: '#C6ECCF',
  width: 250,
  height: 40,
  textAlign: 'center',
  borderRadius: 20,
},

head: {
  fontSize: 20,
  fontWeight: 'bold',
},
listhead: {
  fontSize: 20,
  marginBottom: 10,
  fontWeight: 'bold',
  marginTop: 10,
  backgroundColor: 'pink',
},
image: {
  justifyContent: 'center',
  alignItems: 'center',
  paddingBottom: 10,
},
bookdetails: {
  display: 'flex',
  flexDirection: 'column',
  justifyContent: 'center',
  alignItems: 'center',
  paddingBottom: 10,
},
```

```
bookhead : {
  backgroundColor: 'lightgreen',
  fontSize: 16,
  padding: 5,
  margin: 5,
  width: 90,
  height: 30,
  textAlign: 'center',
},
bookhead1 : {
  backgroundColor: '#007acd',
  color: 'white',
  fontSize: 16,
  padding: 5,
  margin: 5,
  width: 100,
  height: 30,
  textAlign: 'center',
},
bookhead2 : {
  backgroundColor: 'lightgreen',
  fontSize: 16,
  padding: 5,
  margin: 5,
  width: 200,
  height: 30,
  textAlign: 'center',
},
bookhead3 : {
  fontSize: 16,
  fontWeight: 'bold',
},
new: {
  width: 410,
  height: 40,
  backgroundColor: 'lightgreen',
  alignItems: 'center',
  justifyContent: 'center',
  display: 'flex',
  flexDirection: 'row',
},
label: {
  display: 'flex',
  flexDirection: 'row',
```

```
    width: 300,  
    marginBottom: 10,  
    alignItems: 'center',  
,  
});  
  
export default App;
```

## Screenshot of Application



**Figure 1 - Welcome Screen**

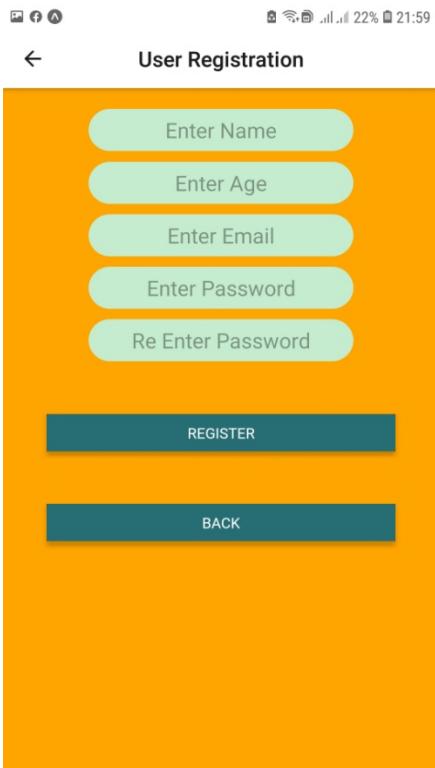


Figure 2 - User Registration

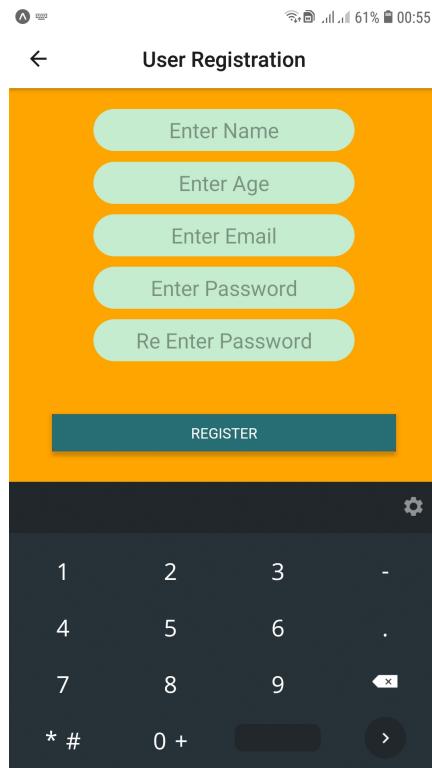


Figure 3 – Numeric input for age

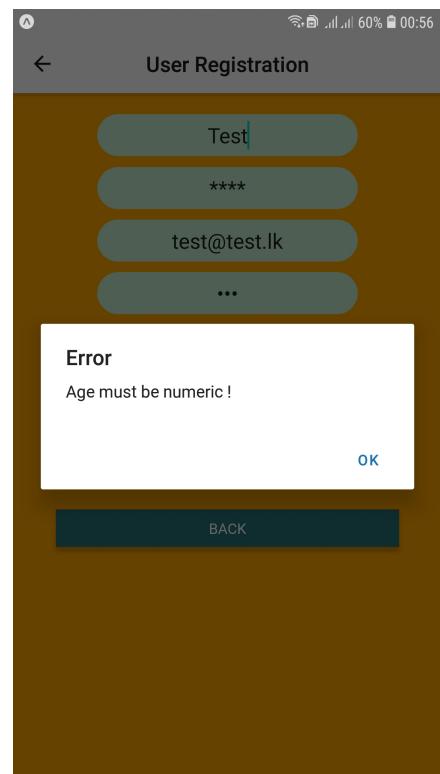


Figure 4 – Age is numeric

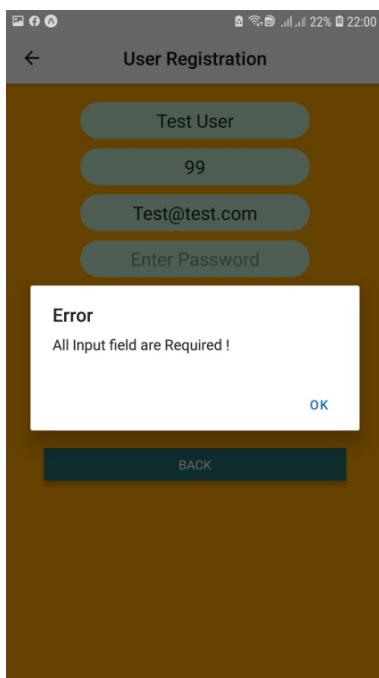


Figure 5 - All field required - Registration

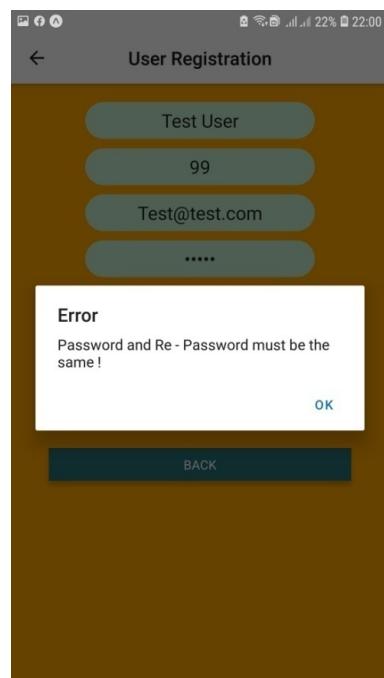


Figure 6- Same password - Registration

- When entering age it will give a numeric keyboard, but a numerically validate alert is also added. – **Figure 4**
- All input fields are required in the user registration form. It will alert if one is empty. – **Figure 5**
- Password and Re-Password must same. If one of them is different it will notify by the alert. – **Figure 6**

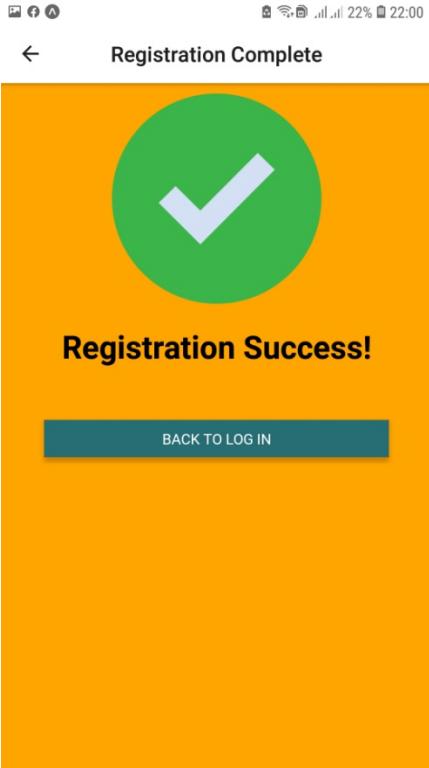


Figure 7 - Registration Success

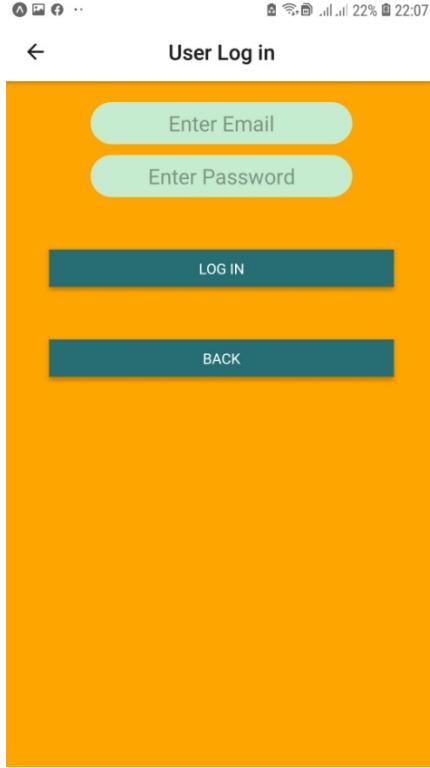


Figure 8 - Log in Screen

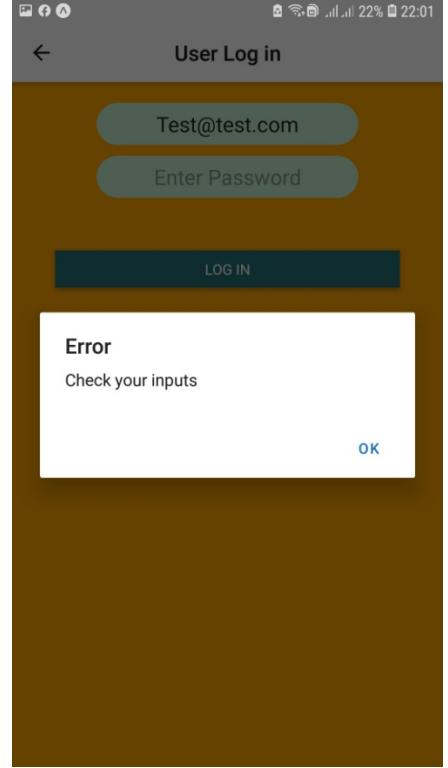


Figure 9 - All field Required

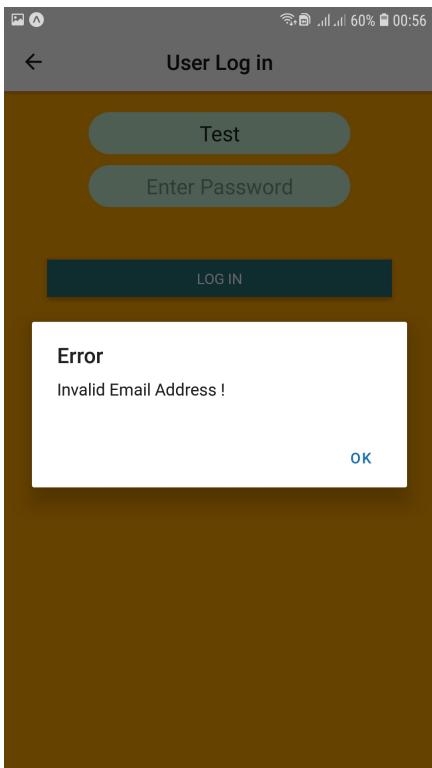
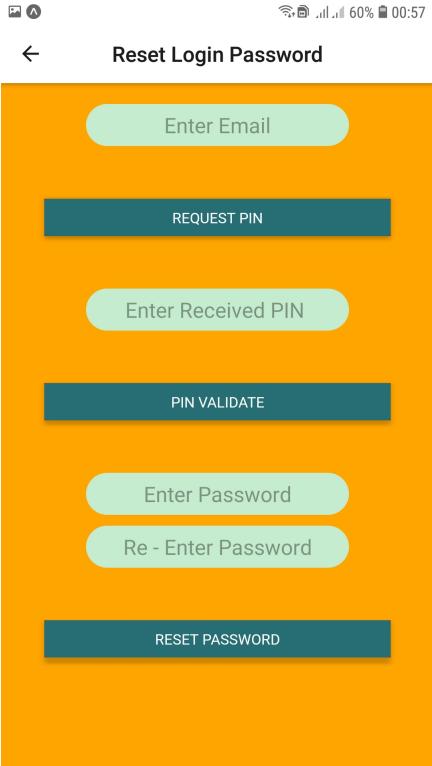
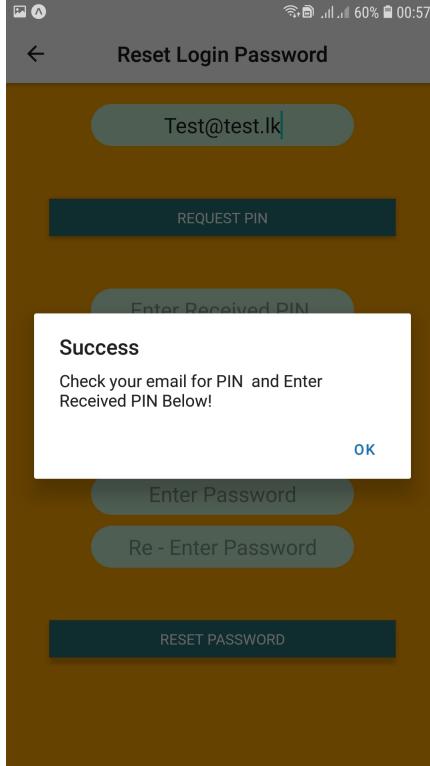


Figure 10 Invalid Email Address

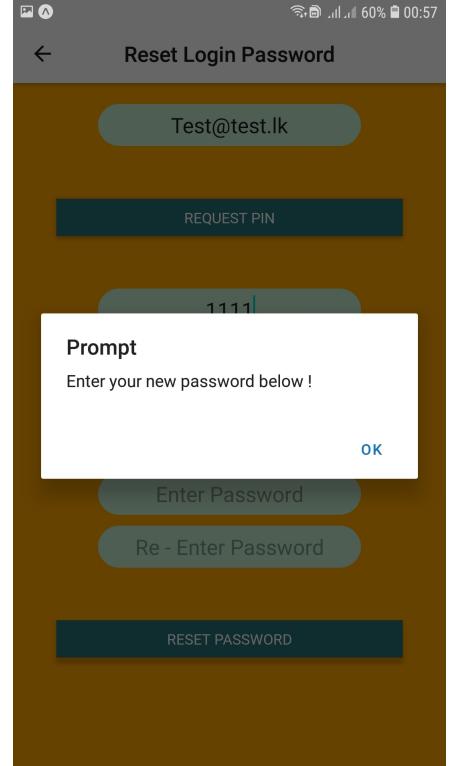
- After successful registration, The user will get this screen – **Figure 7**
- User must enter email and password to log in. The user will get this alert if field is empty. – **Figure 9**
- If users enter an invalid email address, This alert will alert them – **Figure 10**



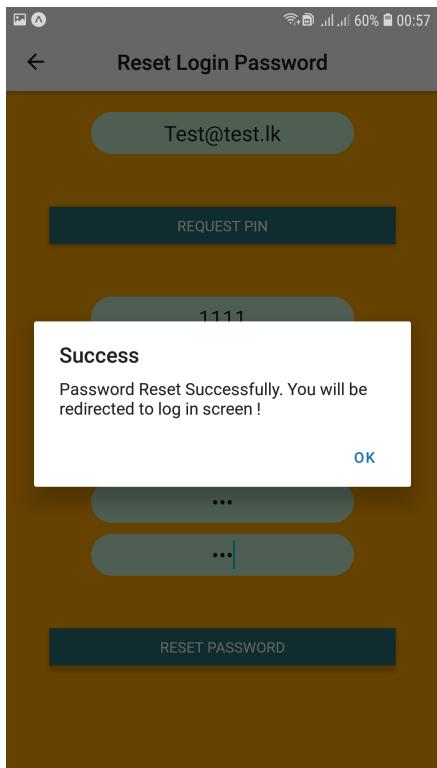
**Figure 11** Reset Login Password



**Figure 12** Password reset – Alert 1

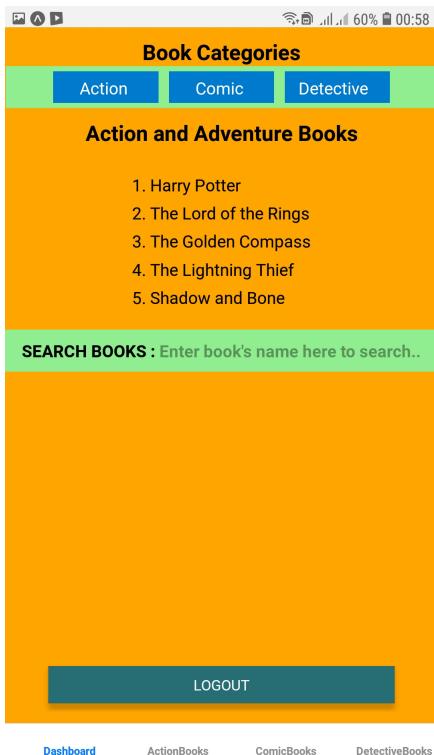


**Figure 13** Password reset – Alert 2



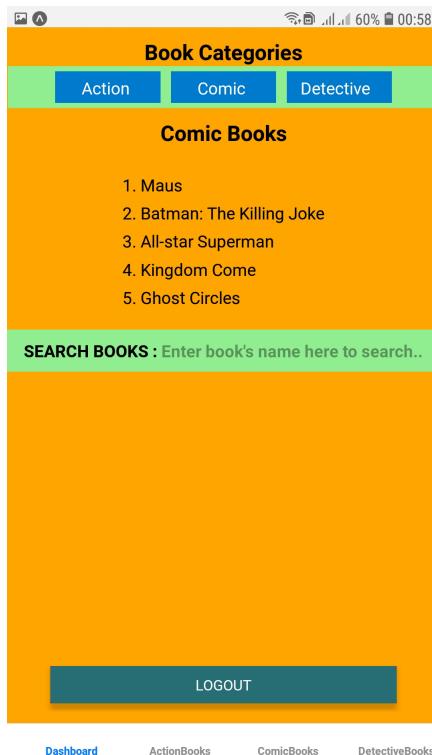
**Figure 14** Password reset – Alert 3

- If user forgot log in password, user can reset password using this screen.
- User should enter email that provided on registration.
- User will get a password reset pin for email. After get this pin, user must enter that pin in text input and should tap on pin validate button
- After tapping pin validate button, user can set new password as their log in password. After successful password change user will automatically redirected on to log in screen. – **Figure 14**



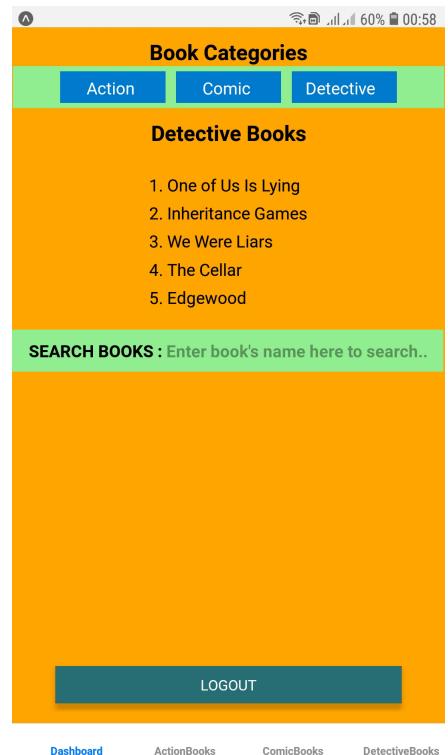
Dashboard ActionBooks ComicBooks DetectiveBooks

**Figure 15**



Dashboard ActionBooks ComicBooks DetectiveBooks

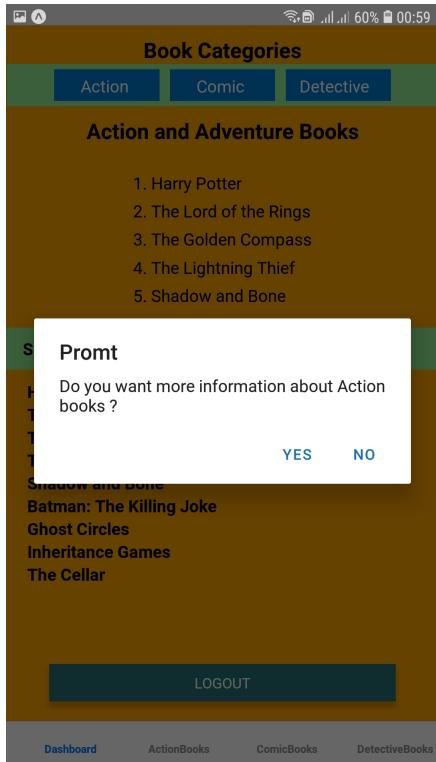
**Figure 16**



Dashboard ActionBooks ComicBooks DetectiveBooks

**Figure 17**

- User can tap on each button ( Category ) and get a book list from each category.
  - Action
  - Comic
  - Detective

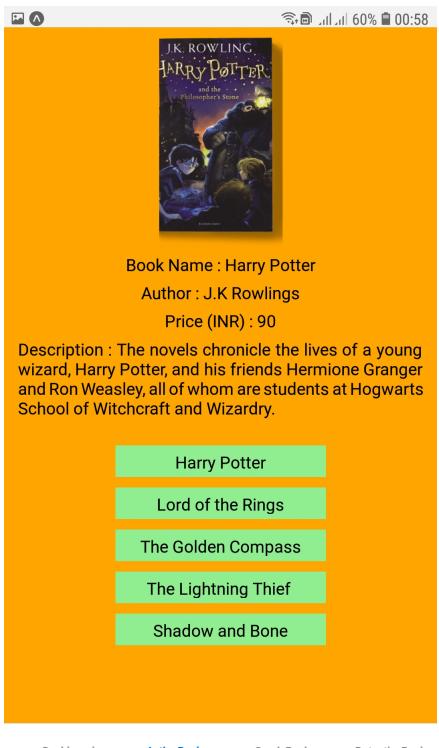


**Figure 18 Prompt to more information**

- When selecting each category, the user will get an alert that asks if the user needs more information about books in each category.-

**Figure 18**

- If the user needs more information about those books such as author, price, and description, then the user needs to tap on yes.
- After tapping on yes, the user will get a screen that includes all details about those books ( **Figure 19 / 20/ 21**  )
- User can also use **bottom tab navigation** bar for access those details screens



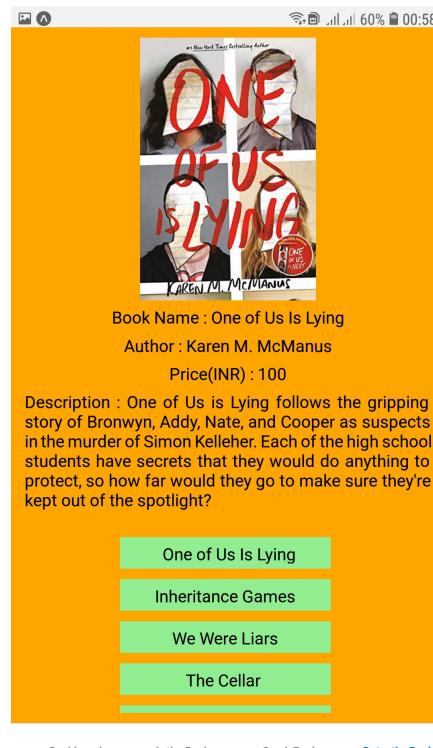
Dashboard    ActionBooks    ComicBooks    DetectiveBooks

**Figure 19 - Details– Action**



Dashboard    ActionBooks    **ComicBooks**    DetectiveBooks

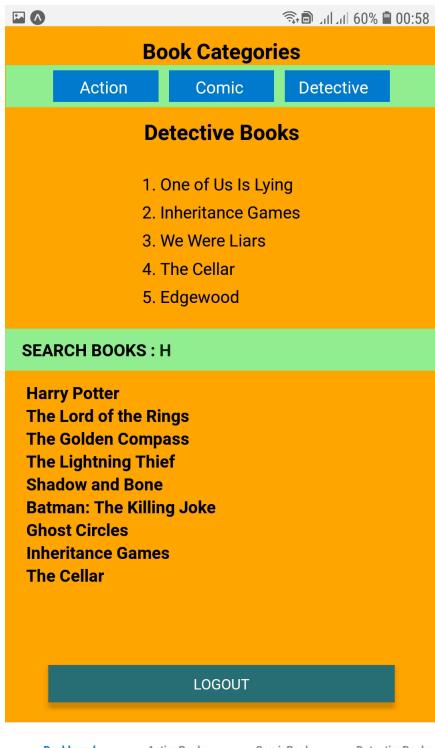
**Figure 20 – Details - Comic**



Dashboard    ActionBooks    ComicBooks    DetectiveBooks

**Figure 21 – Details - Detective**

- In these screens, users can tap on each button that contains book names and get more information about each book such as Book name, Author, price, and description



Dashboard    ActionBooks    ComicBooks    DetectiveBooks

**Figure 22- Search book by book's name**

- Users can search a book by name from the dashboard screen.
- User needs to enter part of book's name in **search books text field**. Then search results will be displayed below as **Figure 22**
- User can logout from application by tapping logout button at bottom in dashboard

**END OF DESIGN REPORT 04**

**THANK YOU !**