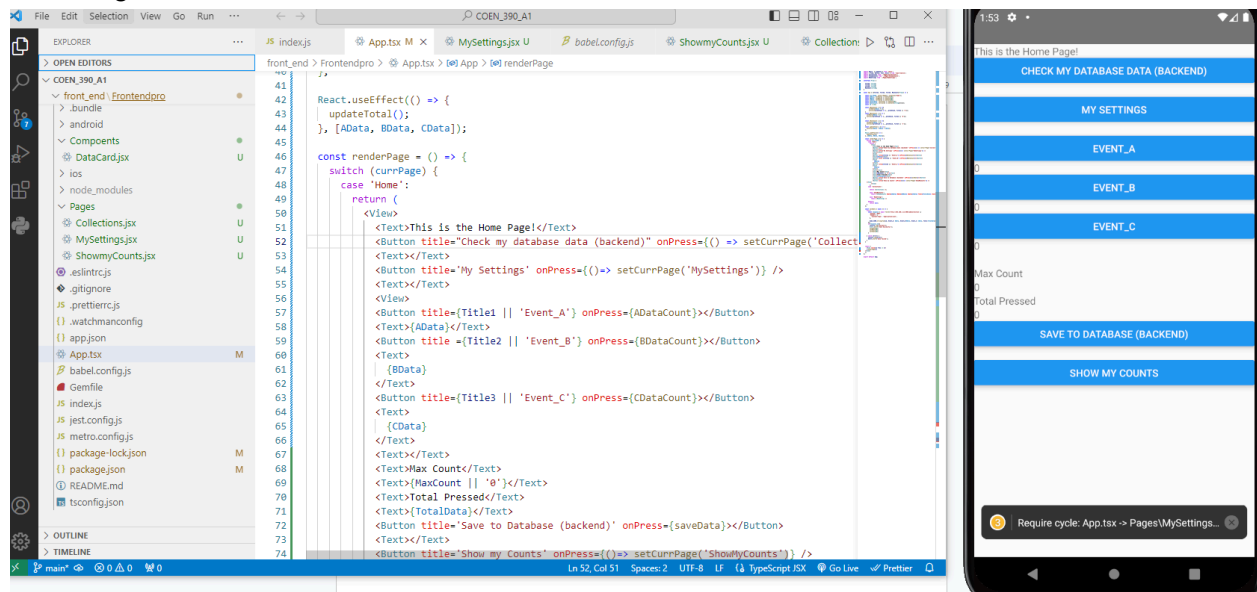


This application consists of React native, mongodb, node.js, express.js and android studio connection.

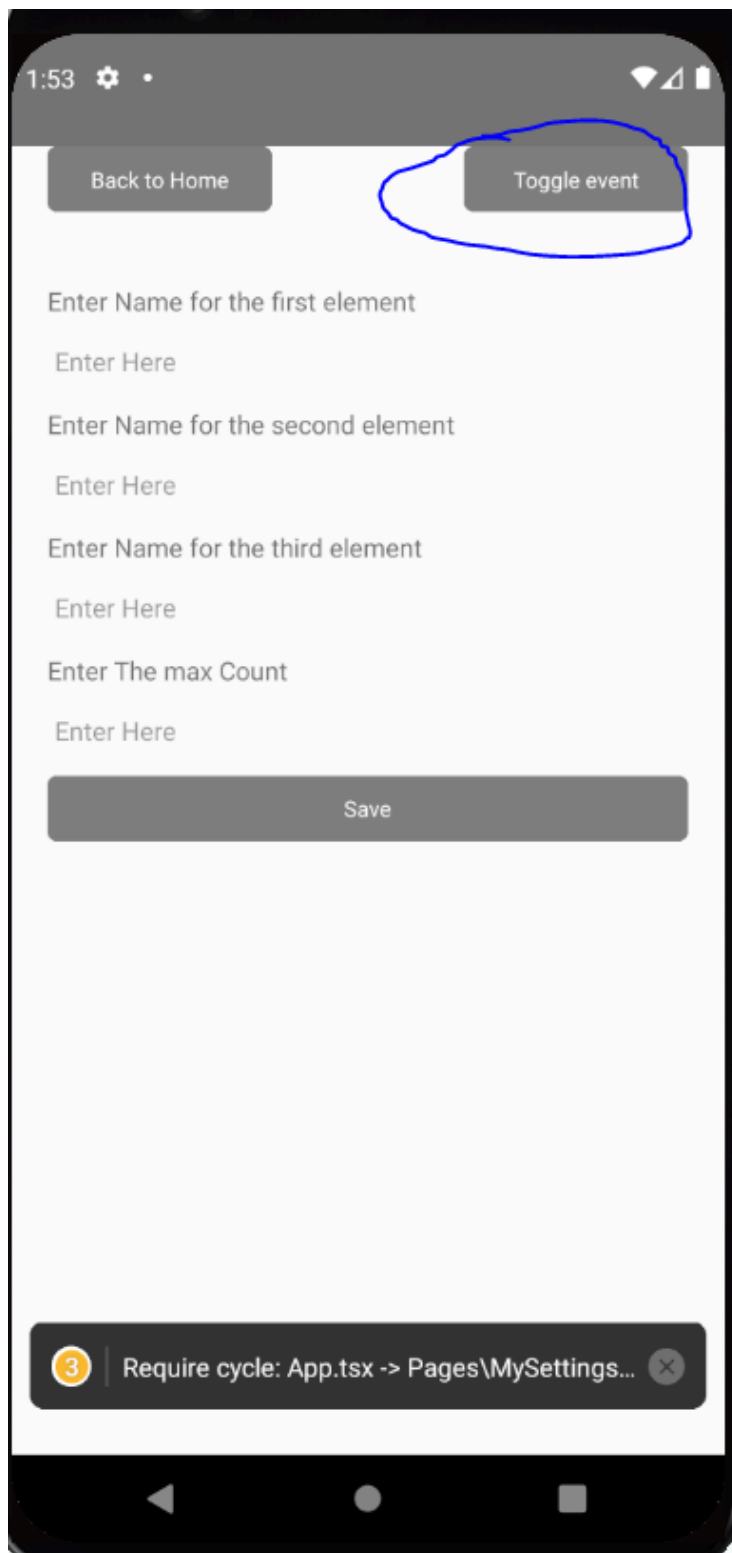
To set up the android studio emulator to the react native development, you need to go through a few steps \_> <https://reactnative.dev/docs/environment-setup>

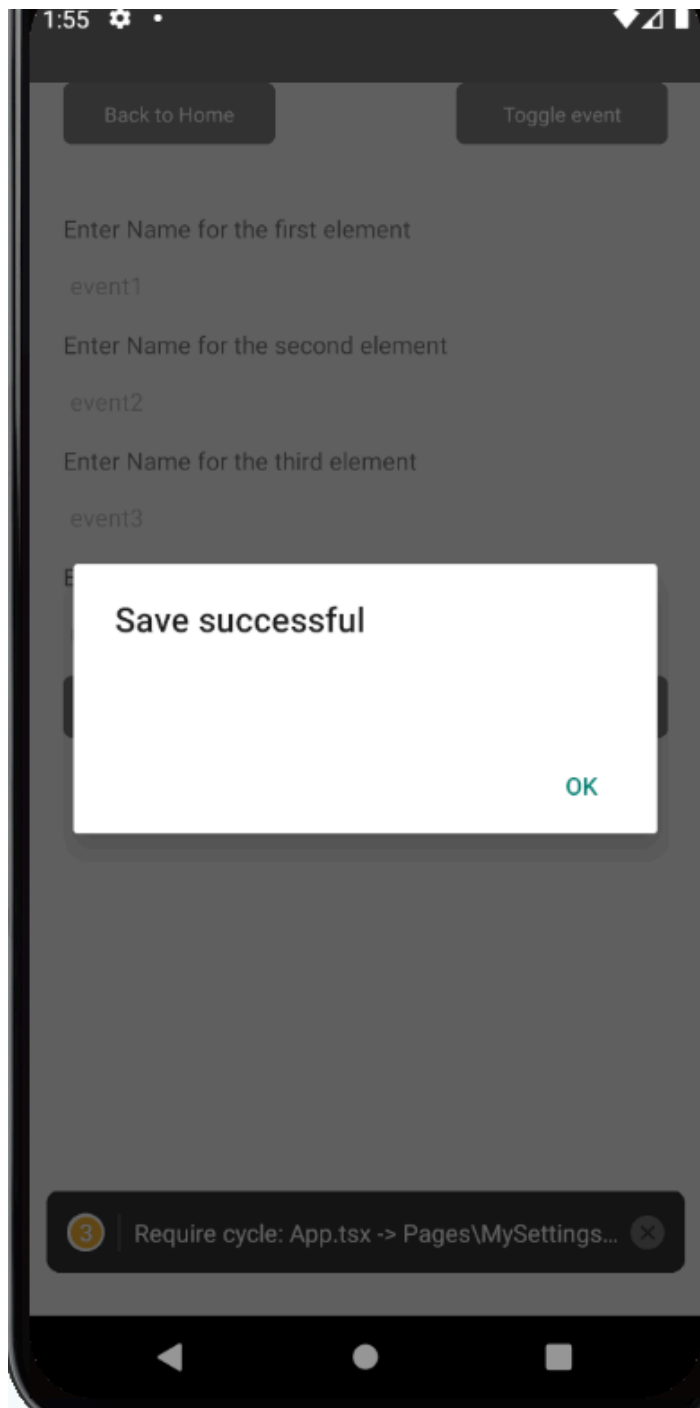
The link should provide you with an easy step to set up the environment. Or you can watch some youtube video to make it work.

Main Page ->

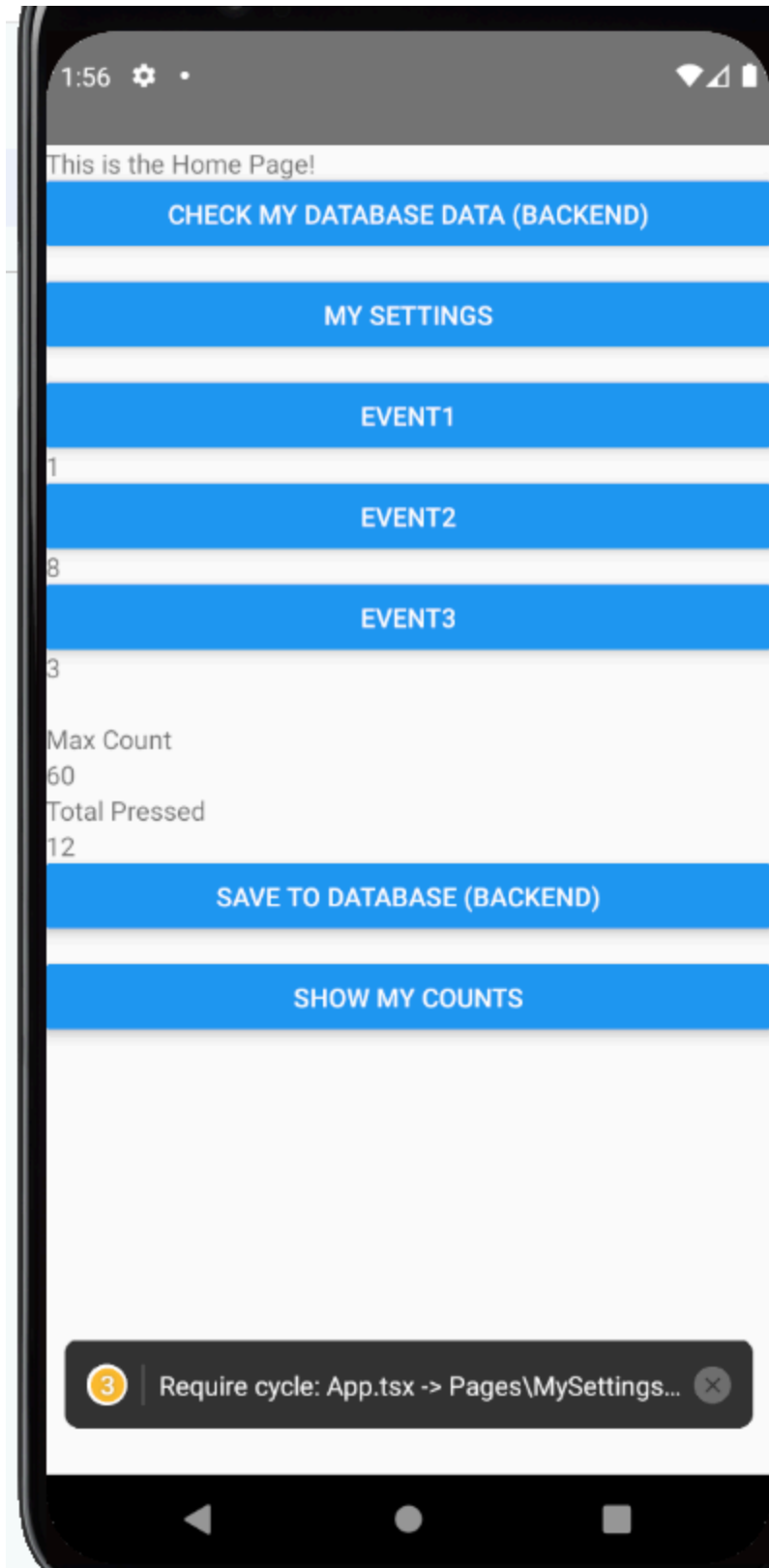


My setting page, you can only be able to enter the name and max count once you click this toggle event.

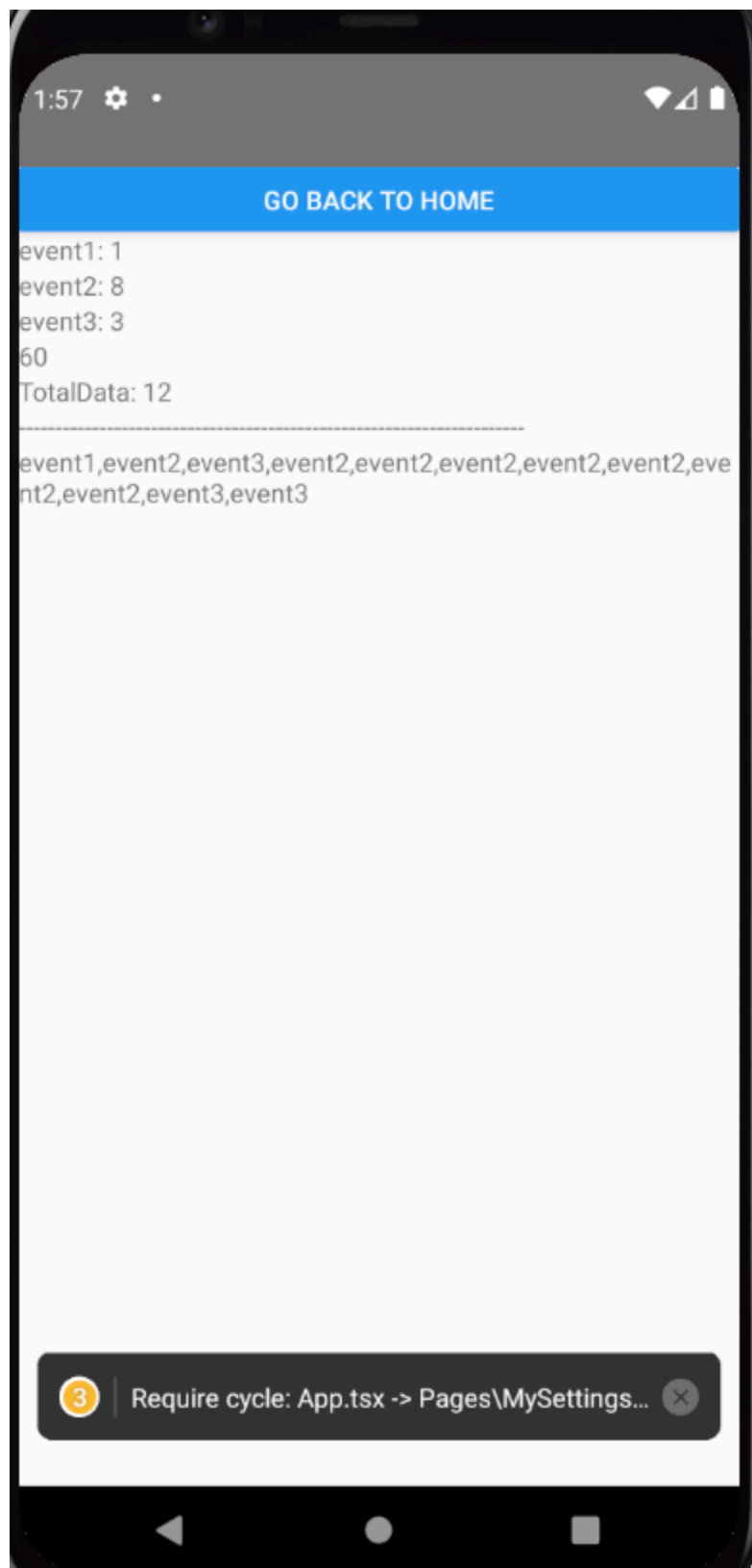




Once you click the save, it will save all the necessary items. Then you can return to home page to see the result.

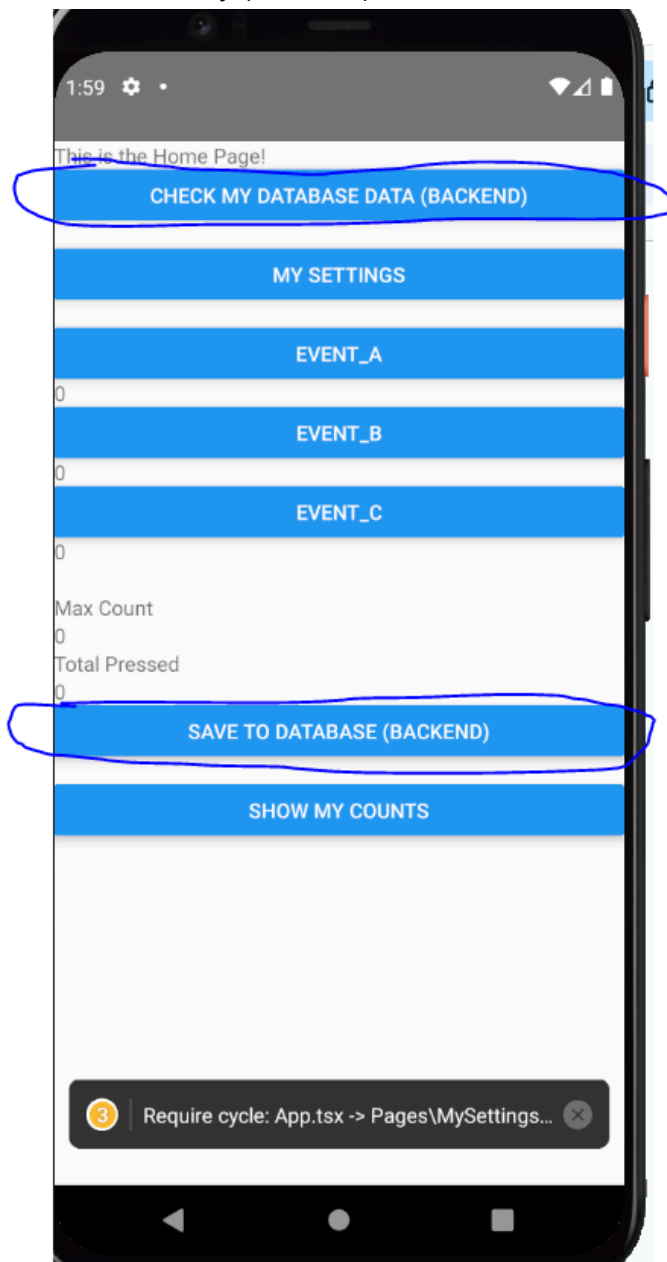


When you press the button, you can clearly see the number will increased, and then once you enter the show my counts page -> (screen shot at the next page)



You will see the activity of your press and count.

Extra functionality (backend)



In order for these two buttons to work, you need to install the Mongodb compass and paste your own links in

```
mongoose.connect("mongodb://localhost:27017/COEN_390_db", {useNewUrlParser: true, useUnifiedTopology: true})
```

Then on the frontend side you need to modify your IPV4 endpoint in your network here ->

```
const Collections = () =>{
  const [currPage, setCurrPage] = useState('Collections');
  const [Data, setData] = useState(null);
  const getData = async () => {
    try {
      const response = await fetch(`http://192.168.1.10:3001/getcollections`);
      if (!response.ok) {
        throw new Error(`HTTP error! Status: ${response.status}`);
      }
      const json = await response.json();
      setData(json);
    } catch (error) {
      console.error('Error fetching data:', error);
    }
  }
  React.useEffect(() => {
    getData();
  })
}
```



Collections.jsx

```
const saveData = async () => {
  try {
    const response = await fetch(`http://192.168.1.10:3001/addcollection`,{
      'method':'POST',
      'headers':{
        'Content-Type': 'application/json',
      },
      body:JSON.stringify({id, Event_A: AData, Event_B:BData, Event_C: CData, Total
    });
    if(response.ok){
      console.log("Success");
      Alert.alert('Save Successful');
      setAData(0);
      setBData(0);
      setCData(0);
    }
  } catch (error) {
    console.log(error);
    Alert.alert('Save failed');
  }
}
```



App.tsx

then you need to run the terminal in the backend folder -> npm install -> nodemon index.js to run the backend server.

Once you do all these steps correctly, the app should work with no issue.