UNIVERSITY OF MISSOURI-KANSAS CITY

Programming Web/Cloud/Mobile

Lab Assignment 3

Submitted by

Avinash Banala (Class Id.: 02)

Pranoop Mutha (Class Id.: 19)

Part-1

Question 1:

Visualise the twitter friends list through the D3.js

Objective:

The below each question will follow different approaches to solve. Coding is done to perform the evaluation of each individual snippet to build a web app that populates the Twitter Friend List by using the visualization with D3 and the android app is developed to enable the social login to the app.

Explanation:

In this, the followers are visualized from the twitter account and visualized through the Data Driven Documents (D3) to get the certain visualisations.

We assign the consumer Api keys and secret token keys to generate the graph assigned to the friends count.

```
var source;
var root;
var cb = new Codebird;
cb.setConsumerKey("o0uqYhTxHbSyxZnkiolch8nZ6", "cV4wBlnKPJ8LMZEyjV3aTRIaWoOtBMjgAMJjMB8IjIaRuV72kG ");
cb.setToken("912448569348632577-hcZXmrfS20fW3AZMaZmoTV2U3dBZCzU", "AHtJ9NfD86AHqTMiBBnrmeOyHUujX9INOmy8O8vKKPe98");
cb.__call("followers_list", {}, function(result1) {
    //console.log(result1);
    for(var i=0;i<result1.users.length;i++)
    {
      jsons=jsons+'{"name":"'+result1.users[i].name+'","parent":"sampath"}';
      if(i!=result1.users.length-1)
      {
         jsons=jsons+",";
      }
    }
    jsons=jsons+"]}";
    //console.log(jsons);
    //console.log(JSON.parse(jsons));
    source=JSON.parse(jsons);
     root=source;
    update(source);
    // ...
});

// ************* Generate the tree diagram  ****************
var margin = {top: 20, right: 120, bottom: 20, left: 120},
    width = 960 - margin.right - margin.left,
    height = 500 - margin.top - margin.bottom;
```

In this height and width is assigned to set the values right such that the graph is visualised. Translate is used for the margins at the top and left.

```
// ************** Generate the tree diagram  *****************
var margin = {top: 20, right: 120, bottom: 20, left: 120},
    width = 960 - margin.right - margin.left,
    height = 500 - margin.top - margin.bottom;

var i = 0;

var tree = d3.layout.tree()
    .size([height, width]);

var diagonal = d3.svg.diagonal()
    .projection(function(d) { return [d.y, d.x]; });

var svg = d3.select("body").append("svg")
    .attr("width", width + margin.right + margin.left)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

// load the external data
//d3.json("treeData.json", function(error, treeData) {
  //  root = treeData[0];

    //update(root);
//});
```

```
function update(source) {
    console.log(root);
    // Compute the new tree layout.
    var nodes = tree.nodes(root).reverse(),
        links = tree.links(nodes);

    // Normalize for fixed-depth.
    nodes.forEach(function(d) { d.y = d.depth * 180; });

    // Declare the nodes…
    var node = svg.selectAll("g.node")
        .data(nodes, function(d) { return d.id || (d.id = ++i); });

    // Enter the nodes.
    var nodeEnter = node.enter().append("g")
        .attr("class", "node")
        .attr("transform", function(d) {
            return "translate(" + d.y + "," + d.x + ")"; });

    nodeEnter.append("circle")
        .attr("r", 10)
        .style("fill", "#fff");

    nodeEnter.append("text")
        .attr("x", function(d) {
            return d.children || d._children ? -13 : 13; })
        .attr("dy", ".35em")
        .attr("text-anchor", function(d) {
            return d.children || d._children ? "end" : "start"; })
```

Java Script:

It is characterized by the dynamic, weakly typed and programming scripted elements. In this javascript is used to get the json elements and also to visualize the graph.

```
}, {
    key: "_url",
    value: function _url(data) {
        if (/boolean|number|string/.test(typeof data === "undefined" ? "undefined" : _typeof(data))) {
            return encodeURIComponent(data).replace(/!/g, "%21").replace(/'/g, "%27").replace(/\(/g, "%28").replace(/\)/g, "%29").replace(/\*/g, "%2A
        } else {
            return "";
        }
    }
```

Part – 2

Objective:

Create a login application in Android which has the Login and Register Functionality. Also include the OAuth Login.

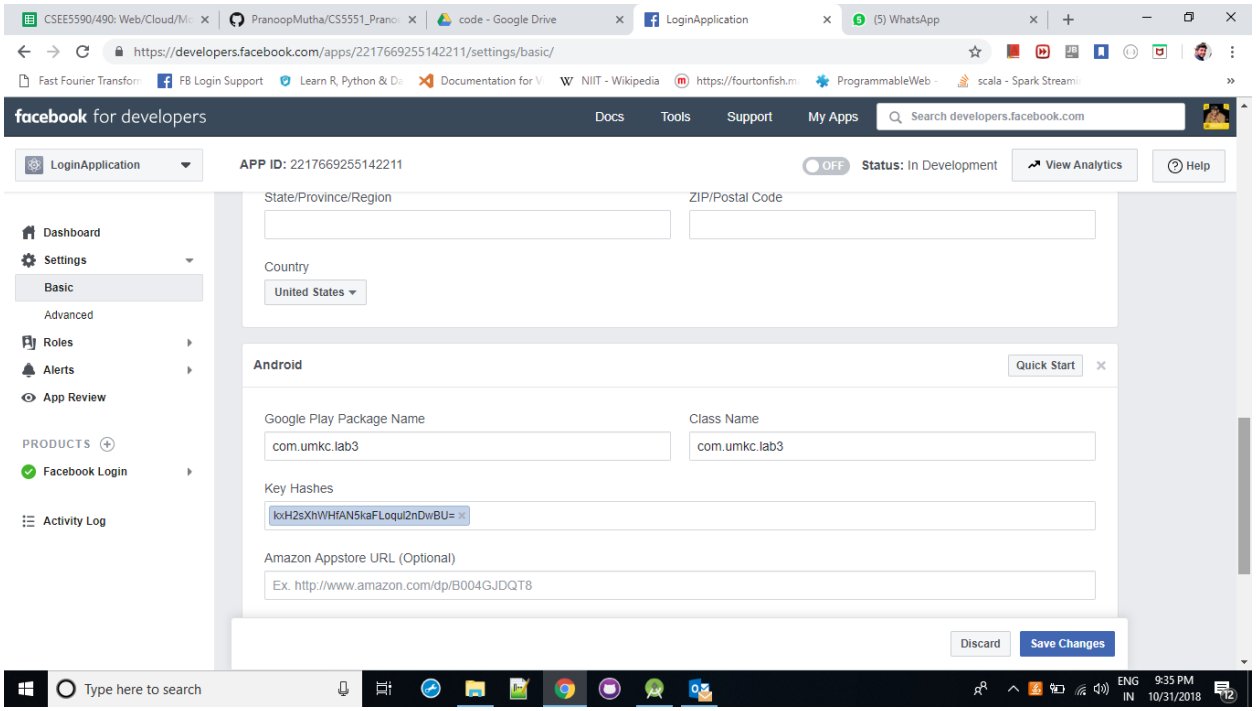
Overview:

1. In this lab assignment, we have included 3 xml pages, one for register, one for the login and other for home page, we have corresponding java activities for each of the page.
2. We have used the FB login over here.

Implementation:

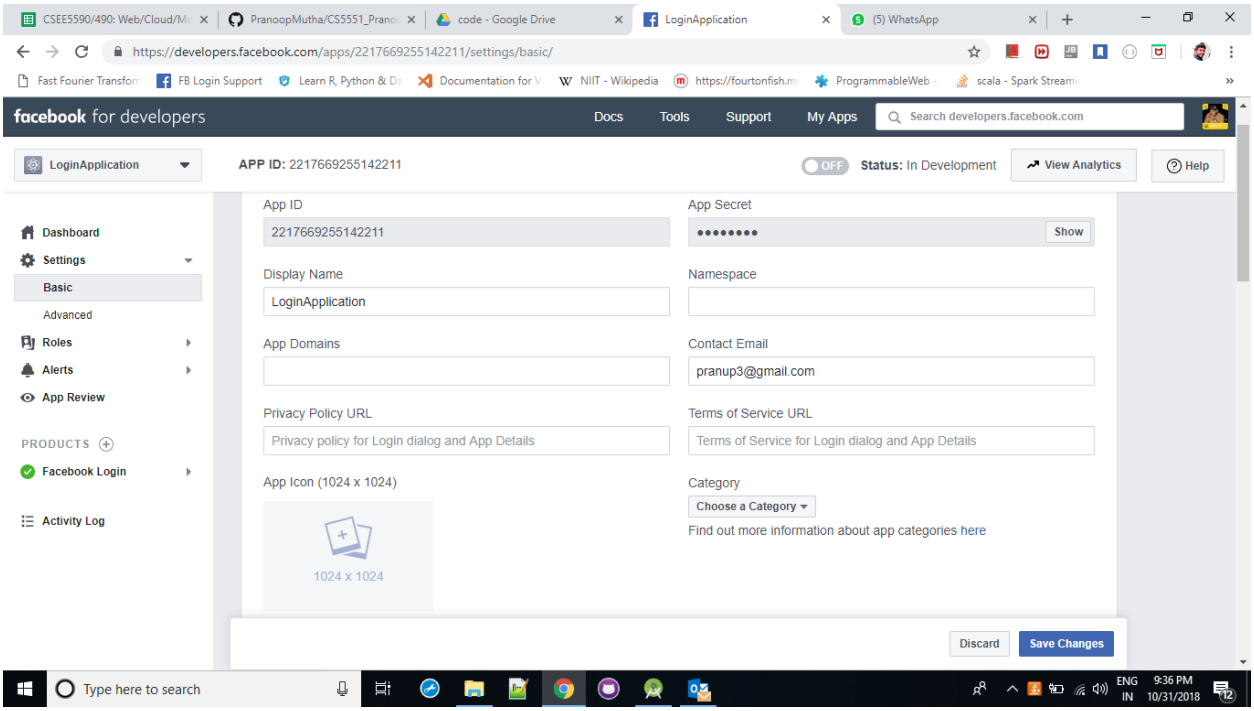
For the Facebook login, we follow these steps:

1. Login to developers.facebook.com
2. Create a new app and give the package name of your project and the Main activity.
3. Then we need to create a key hash using the key store from android folder and openssl library.
4. Then we need to paste it in the developers.facebook.com to authorize our login through android.

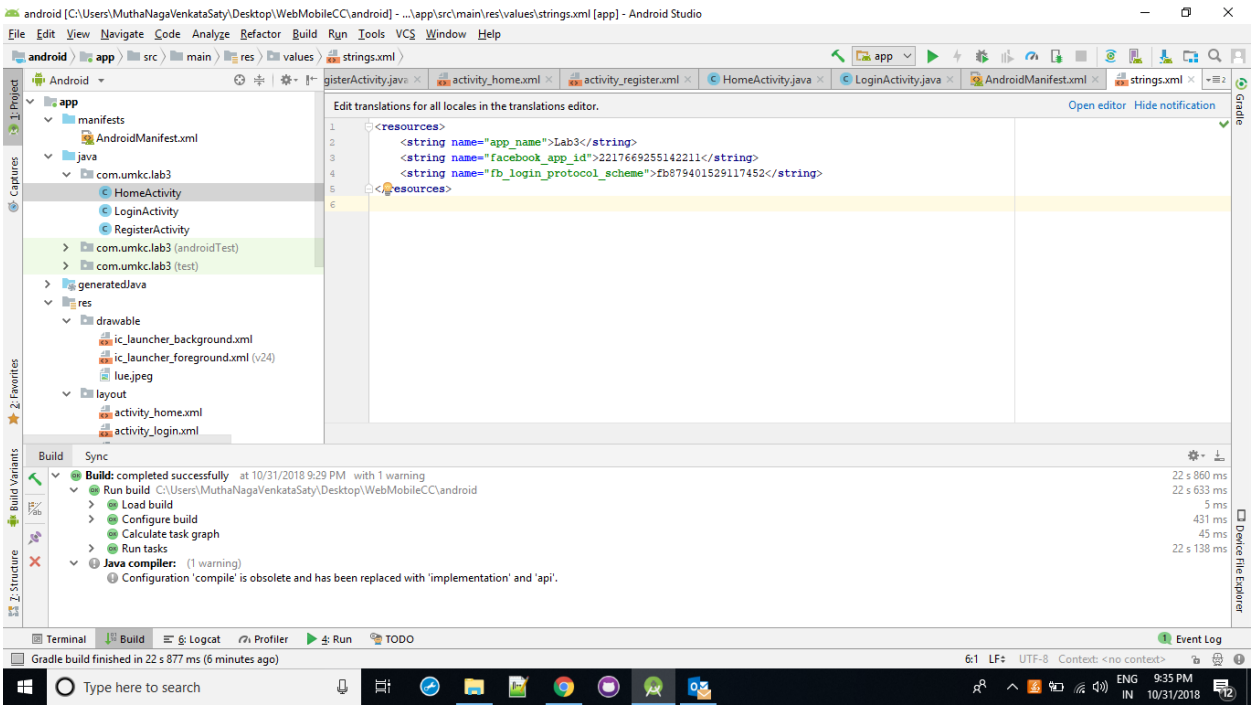


5. The facebook button widget code will also be included in the development, we need not create a new button for that and we also need not create a on click functionality also.

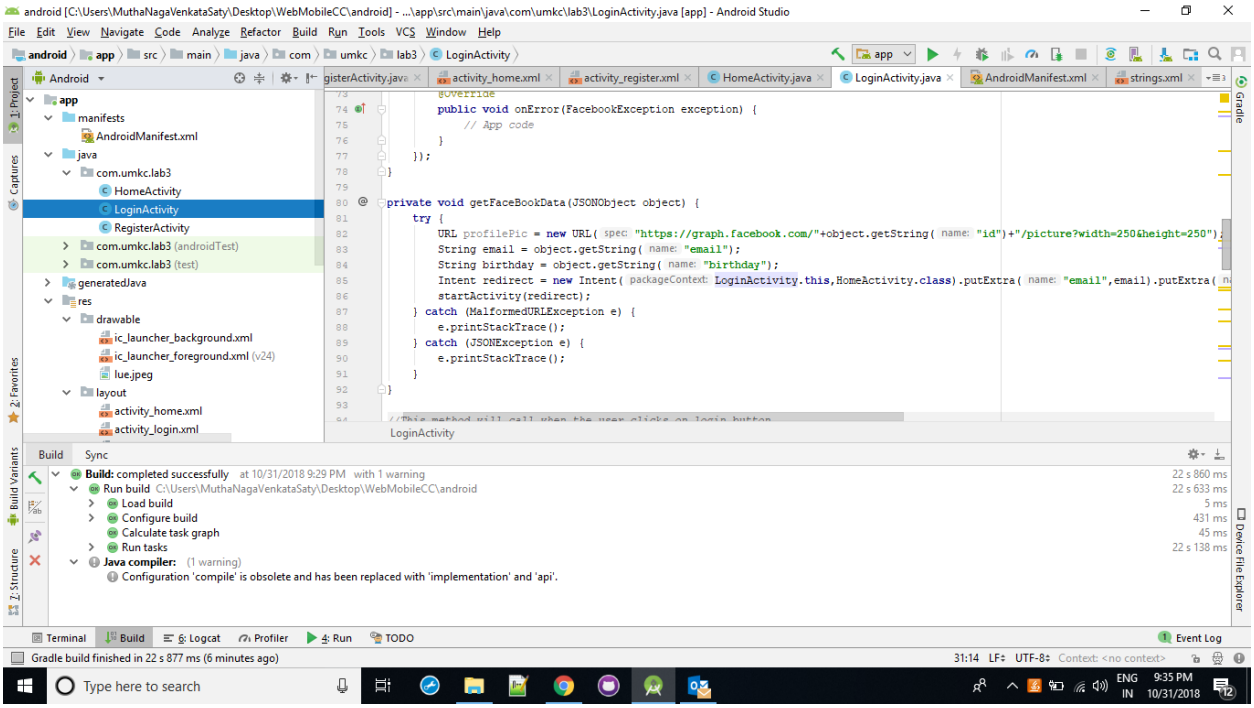
6. Once we create the app, we get an AppID, which we need to use it in the android to fetch the details of the user.



7. Coming to the code part, in strings.xml, we use the facebook app id and give in the manifest.xml and we validate it using the Facebook SDK.



We retrieve the facebook data using the Facebook graph and display the objects in the home page which comes after successful login.

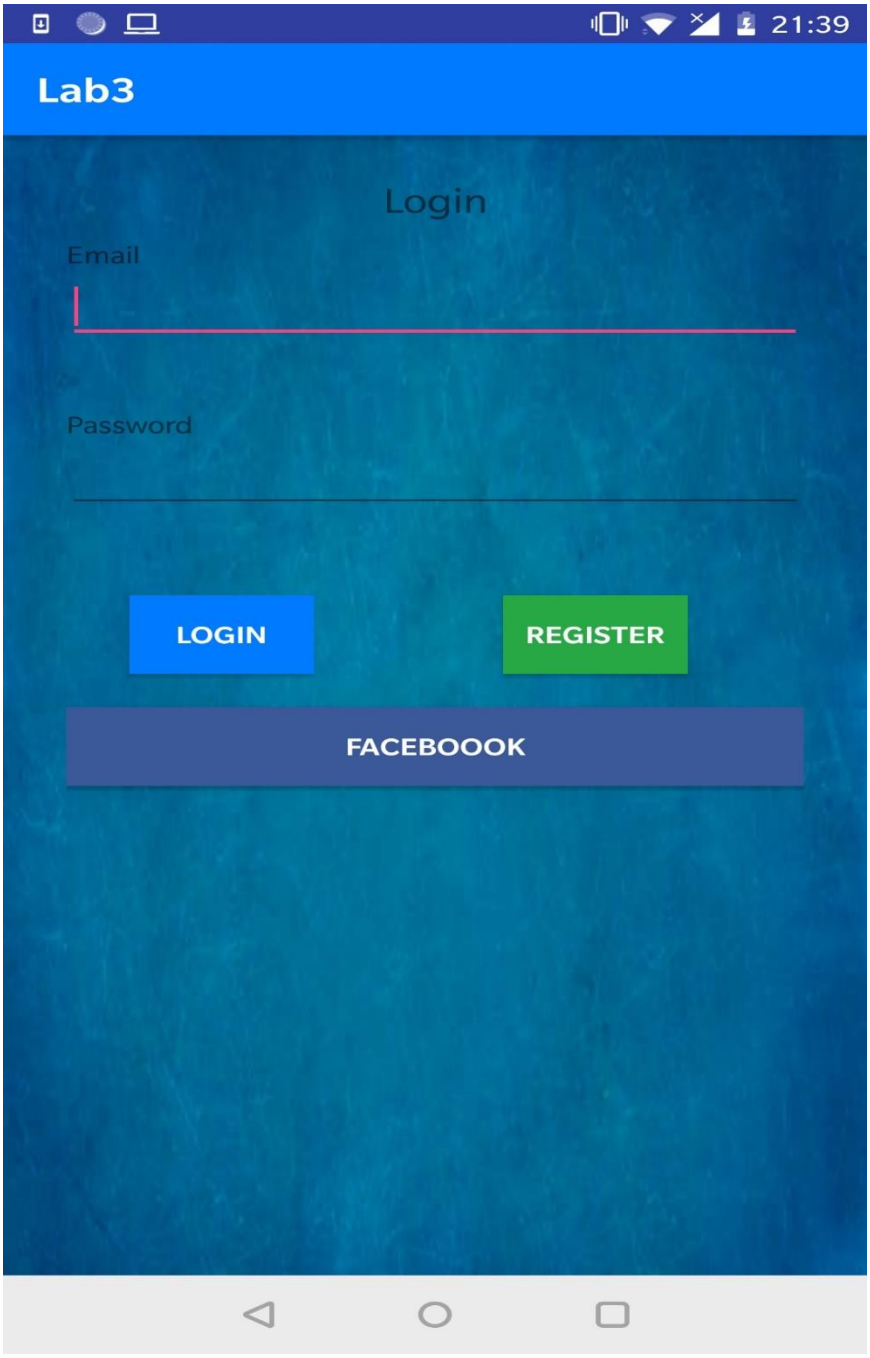


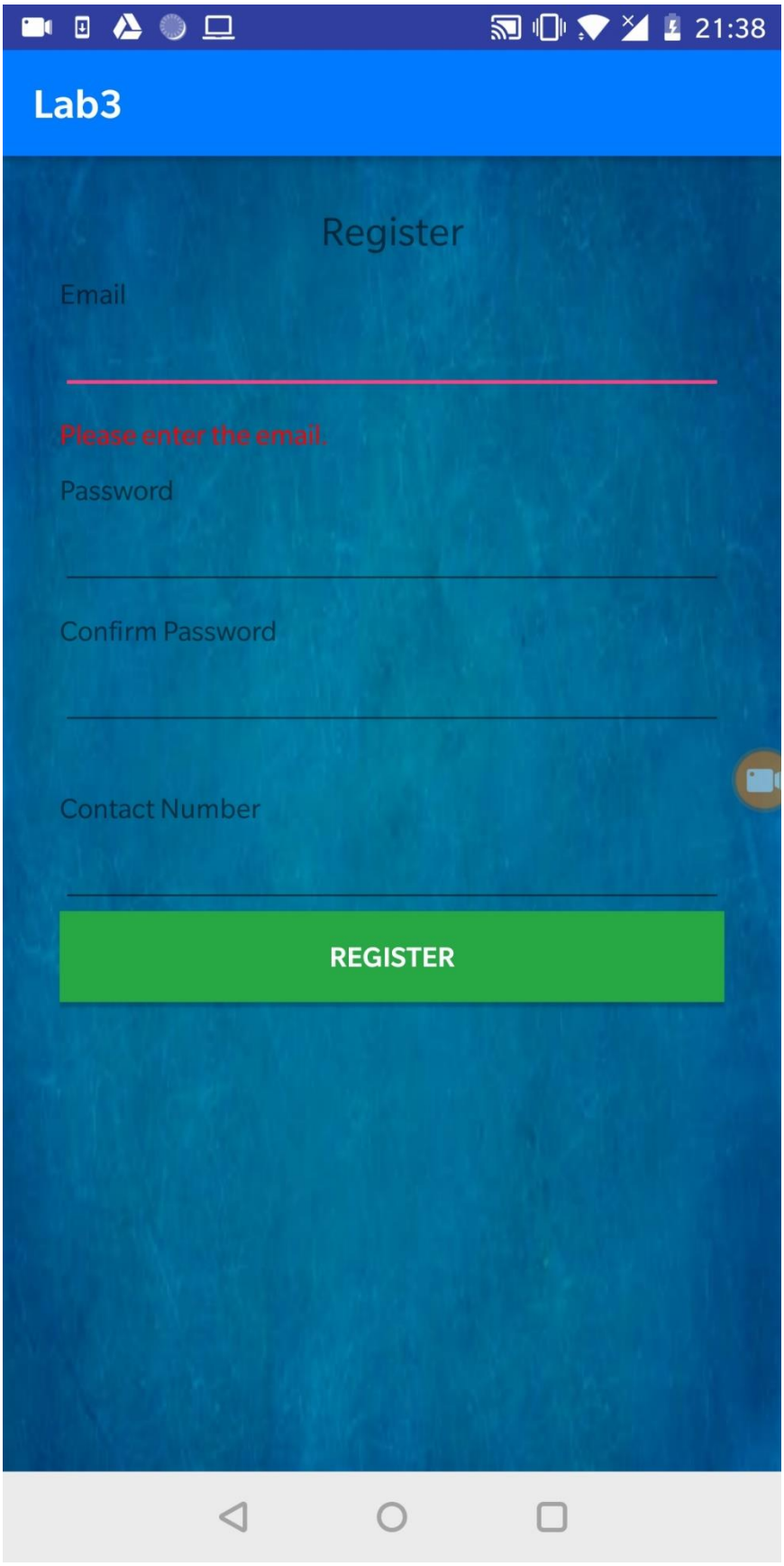
For login and register, these are the steps:

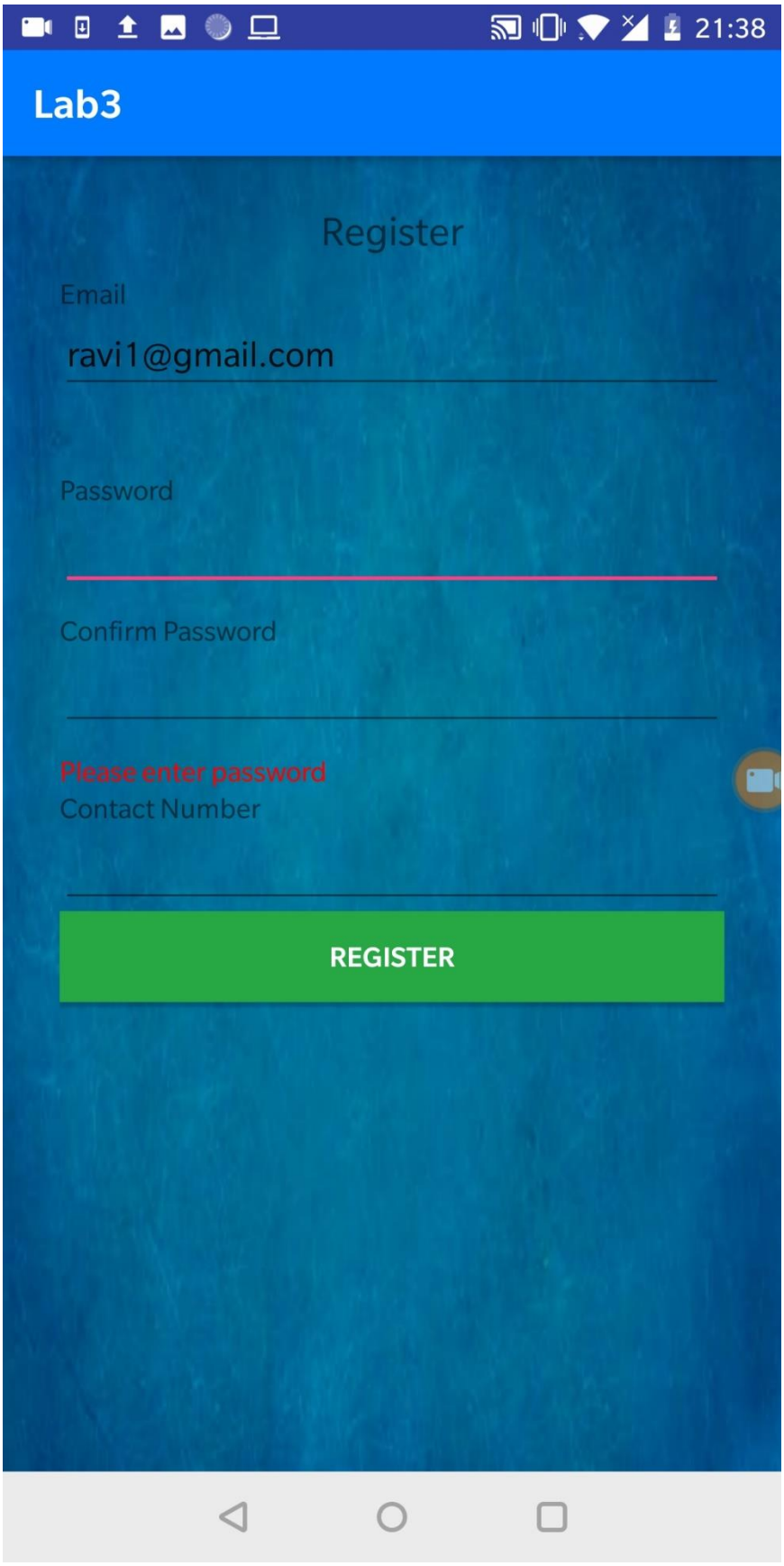
1. Once the user clicks the register button, it goes to the register page and asks for the details of the user like email id, password, confirm password and contact number.
2. We have several validations like
    a. email or password or confirm password cannot be empty.
    b. Password and confirm password should be equal.
3. Once the details are valid, it registers the user.

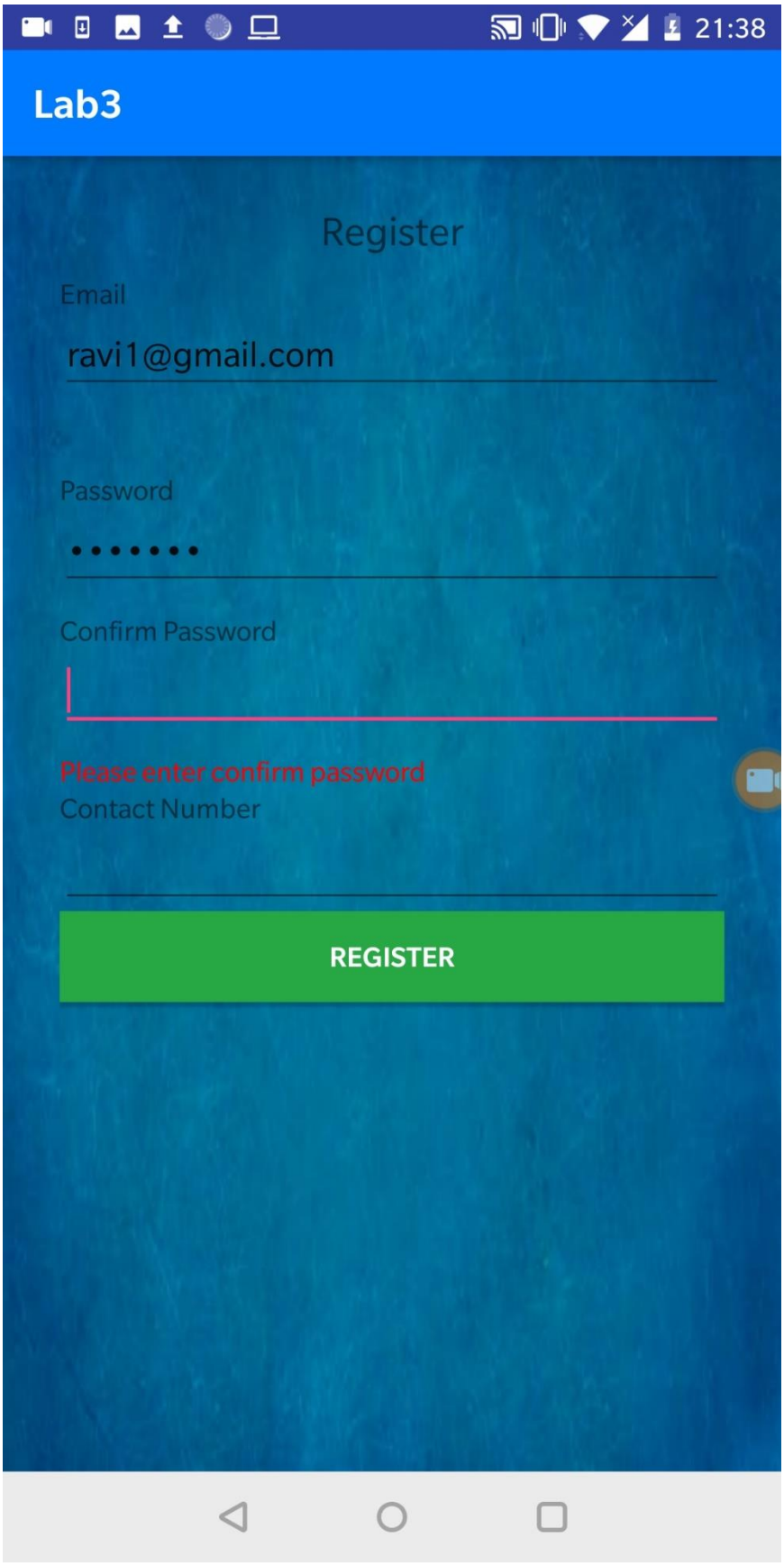
4. Once you click login, it asks to enter the Email id and password.
5. It validates the username and password combination and logs in.
6. Once it logged on, it navigates to the Home Page and displays the user details.

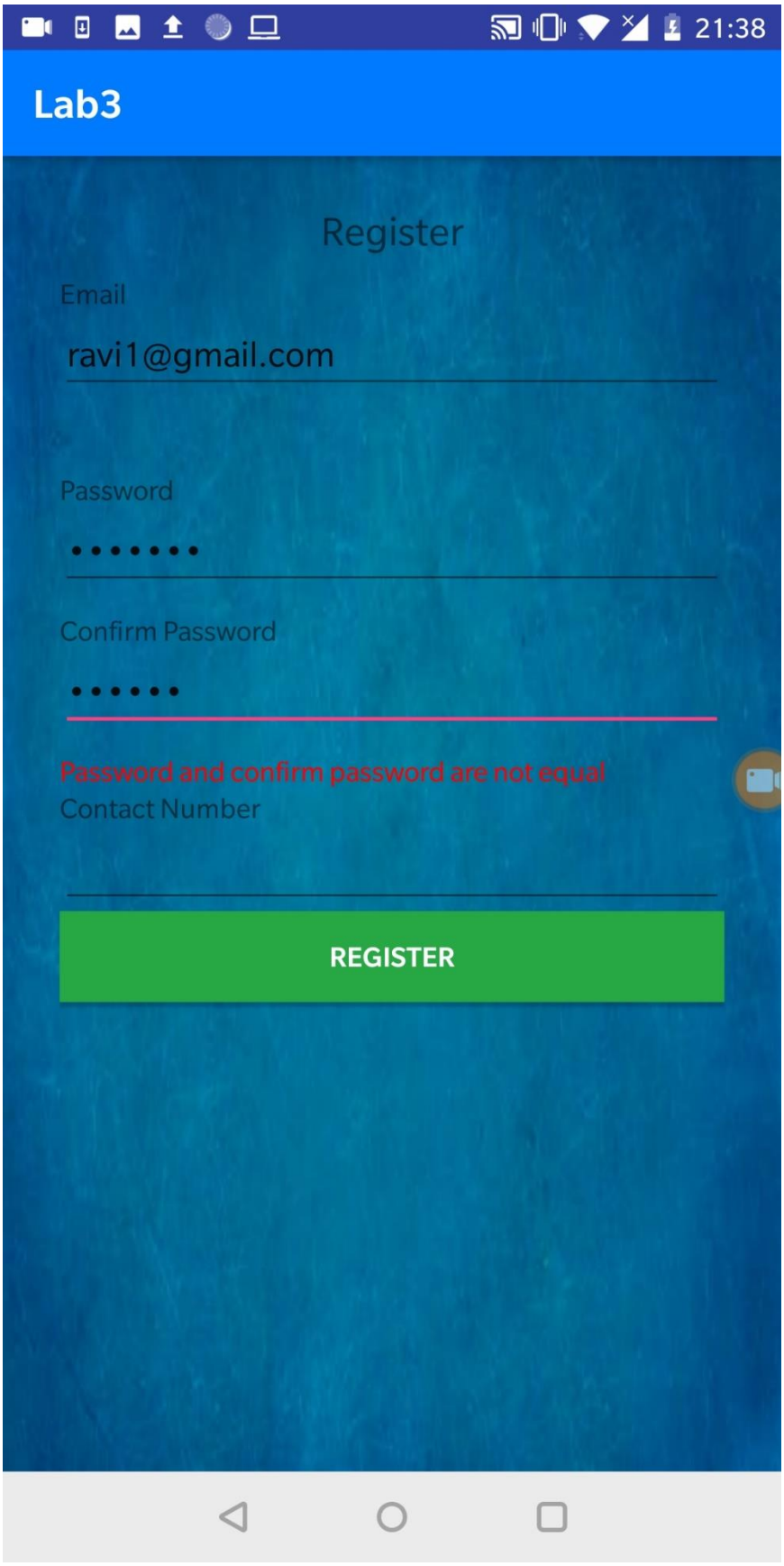
The output from Android are below:

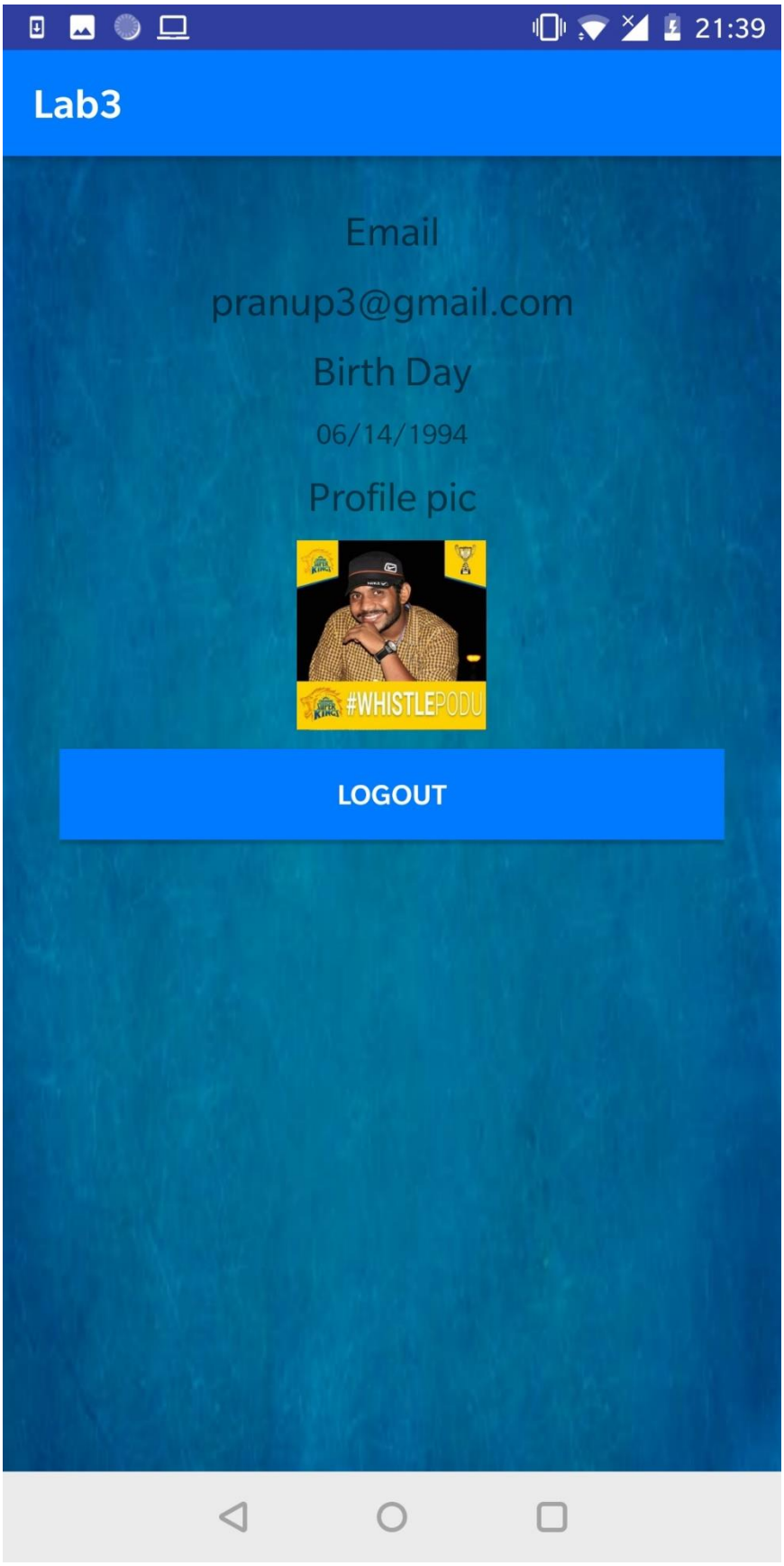


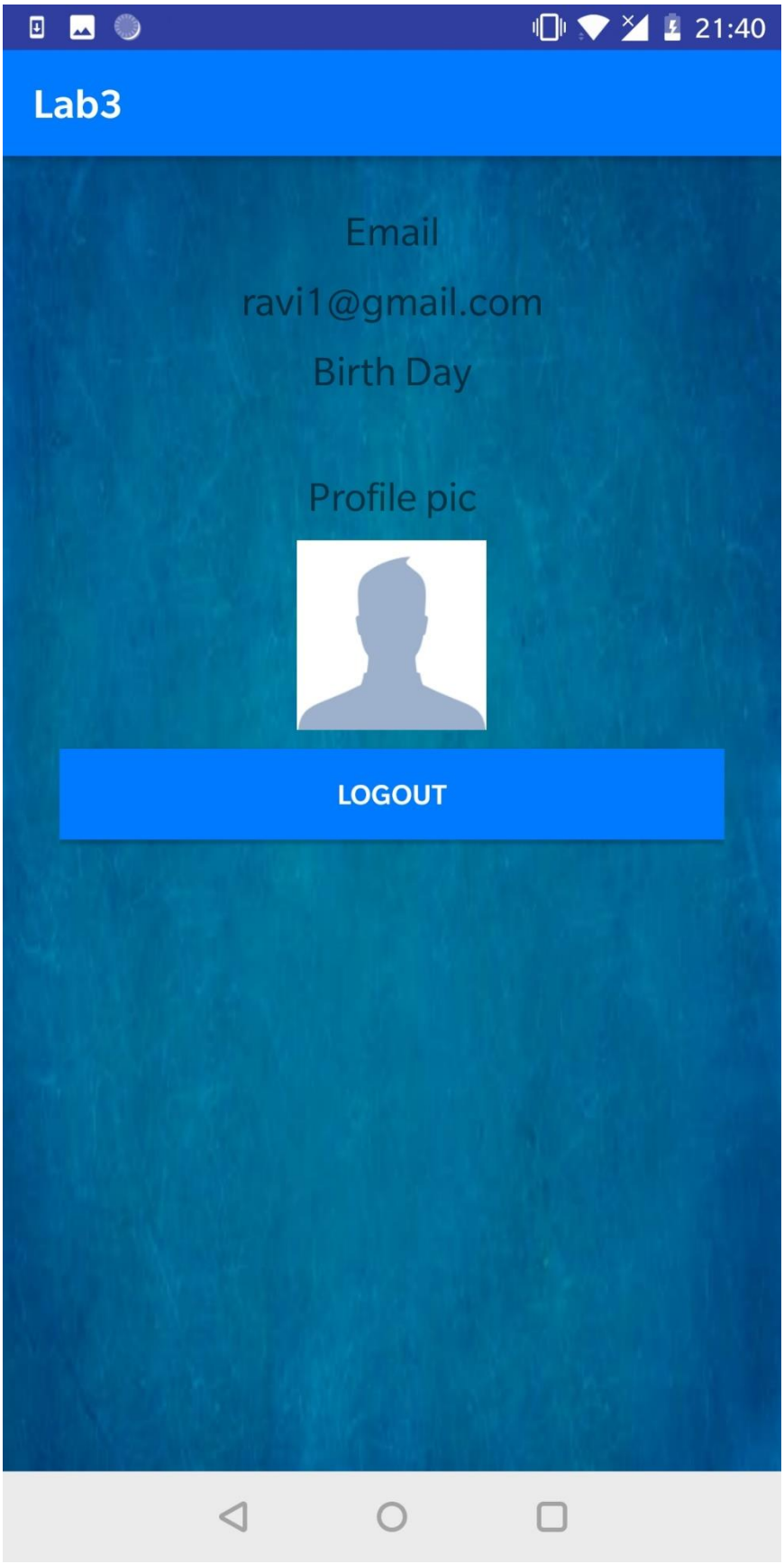

21:38
Lab3
Register
Email
Please enter the email.
Password
Confirm Password
Contact Number
REGISTER


21:38
Lab3
Register
Email
ravi1@gmail.com
Password
Confirm Password
Please enter password
Contact Number
REGISTER


21:38
Lab3
Register
Email
ravi1@gmail.com
Password
Confirm Password
Please enter confirm password
Contact Number
REGISTER


21:38
Lab3
Register
Email
ravi1@gmail.com
Password
Confirm Password
Password and confirm password are not equal
Contact Number
REGISTER


21:39
Lab3
Email
pranup3@gmail.com
Birth Day
06/14/1994
Profile pic
#WHISTLEPODU
LOGOUT


21:40
Lab3
Email
ravi1@gmail.com
Birth Day
Profile pic
LOGOUT