Name: Lakshmana Kumar Mettu

Class ID:11

Introduction:

- Implementation of user authentication web application.
- Developing an android application using social logins like Gmail, Facebook etc.

Programming elements:

- HTML5
- CSS
- Bootstrap
- Java Script
- Java

Platforms Required:

- Jet brains web storm
- Android Studio

Connections and API's:

- MongoDB
- Facebook oauth
- Google oauth

Task-1:

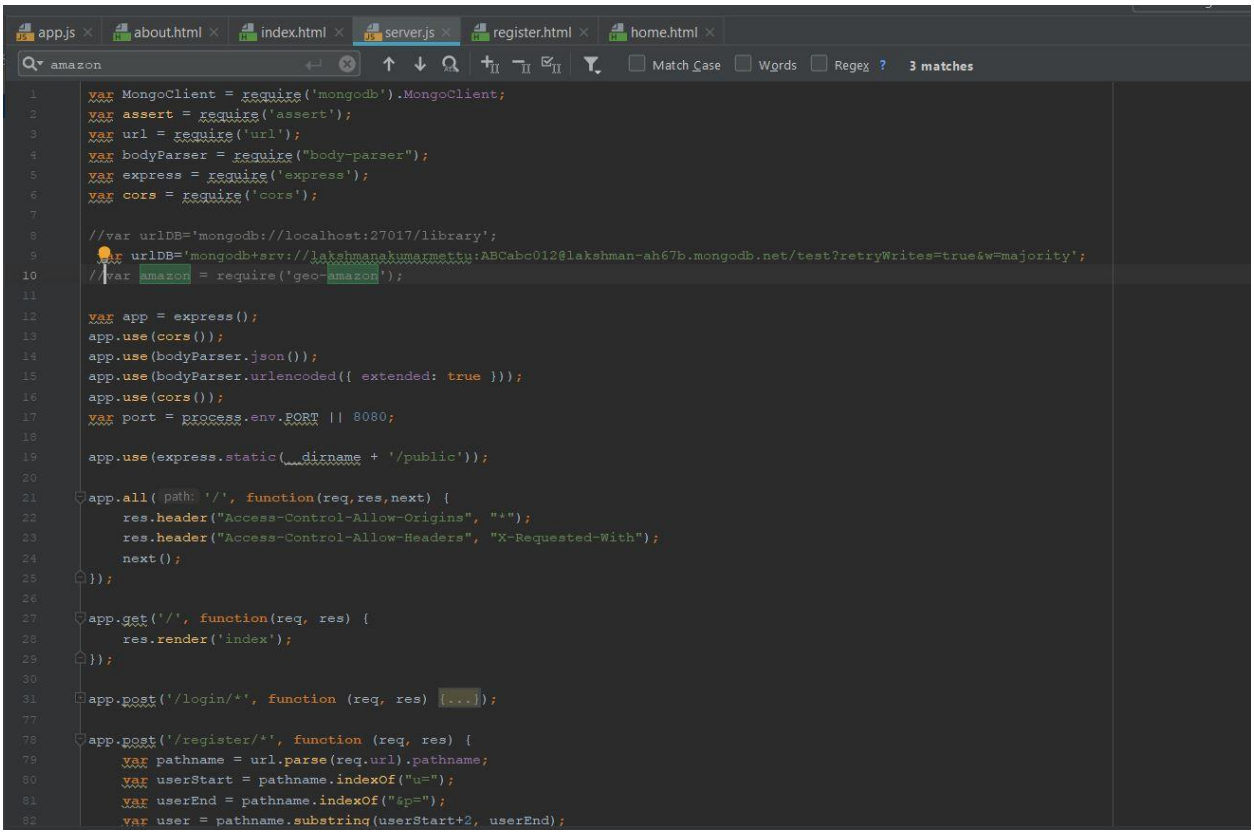
Objective:

- The main objective of task-1 is to develop mean stack web application.
- User should Register
- User can login into home page using registered details

- User should Save their data
- login status when they are moving to other pages
- Output screen changing depending on the moving pages
- Logout session by user
- Corresponding Code snippets for this task are given below

Approaches and Workflow:

- The main component for this task is server.js where user implement url connection to mongo DB which is shown below.



- Here DB name is library and have created collection as lab-2.
- After establishing the connection user can register their details and for each case pop-up will be shown.
- And update and delete functionalities are implemented for database shown in code snippet

```
app.post('/get/*', function (req, res) {
    var pathname = url.parse(req.url).pathname;
    var userStart = pathname.indexOf("u=");
    var user = pathname.substring(userStart+2, pathname.length);

    MongoClient.connect(urlDB,  options: function(err, client) {
        var db = client.db('library');

        if (err) {
            res.write("Failed, Error while connecting to Database");
            res.end();
        }

        db.collection('lab2').findOne({Username: user}, function (err, doc) {
            if (doc == null) {
                client.close();
                console.log("fail");
                res.send("fail");
            }
            assert.equal(err,  expected: null);

            if (doc != null) {
                res.send(doc);
            }
        });
    });
});

app.listen(port, function() {
    console.log('app running')
});
console.log('Client Server running at http://127.0.0.1:8080/');
```

- User will redirect to local host and register and login into pages using saved details.

```
app.all( path: '/', function(req,res,next) {
    res.header("Access-Control-Allow-Origins", "*");
    res.header("Access-Control-Allow-Headers", "X-Requested-With");
    next();
});

app.get('/', function(req, res) {
    res.render('index');
});

app.post('/login/*', function (req, res) {...});

app.post('/register/*', function (req, res) {
    var pathname = url.parse(req.url).pathname;
    var userStart = pathname.indexOf("u=");
    var userEnd = pathname.indexOf("&p=");
    var user = pathname.substring(userStart+2, userEnd);
    var password = pathname.substring(userEnd+3, pathname.length);
    res.setHeader( header: 'Access-Control-Allow-Methods',  value: 'GET, POST, OPTIONS, PUT, PATCH, DELETE');

    console.log(user);
    //console.log(password);

    MongoClient.connect(urlDB,  options: function(err, client) {
        console.log(client);
        var db = client.db('library');
        //res.setHeader('Access-Control-Allow-Origin', '*');
        //res.setHeader('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, PATCH, DELETE');
```

- Now app.js having functions for login, register and displaying pop-up messages depending on the user status.

```
app.controller('loginCtrl', function ($scope, $http, $window)
{
    $scope.login = function ()
    {
        var userid = document.getElementById( elementId: "txt_userid").value;
        var userpass = document.getElementById( elementId: "txt_userpass").value;
        localStorage.setItem("user", userid);

        if (userid != "" && userpass != "")
        {

            $http({url:"http://localhost:8080/login/u=" + userid + "&p=" + userpass , method: 'POST'}).then(function(data, status){

                if(data.data == "fail") {
                    alert("User not found");
                }
                else if(data.data == "noMatch")
                {
                    alert("User and/or Password do not match");
                }
                else
                {
                    //alert(data.data);
                    $window.location.href = "home.html";
                }
            }).catch( onrejected: function() {
            // handle errors
                alert("hi");
        });
        }
        else
        {
```

```
app.controller('registerController',function($scope,$http,$window){
    $scope.register = function() {
        console.log('check');
        var userid = document.getElementById( elementId: "txt_userid").value;
        var userpass = document.getElementById( elementId: "txt_userpass").value;

        if (userid != "" && userpass != "")
        {

            $http({url:"http://localhost:8080/register/u=" + userid + "&p=" + userpass, method: 'POST'}).then(function(data, status){

                if(data.data == "fail") {
                    alert("Insertion Failed");
                }
                else
                {
                    alert(data.data);
                    $window.location.href = "index.html";
                }
            }).catch( onrejected: function() {
                // handle errors
                alert("hi"+status);

            });
        }
        else
        {
            alert("Please enter a Username and Password");
        }
    }
});

app.controller('homeCtrl',function($scope,$http){
    $scope.update = function() {
```

- It has functions for updating the dish details shown below.

```
    }

    $scope.init = function ()
    {
        var user = localStorage.getItem( key: "user");

        if (user != "") {

            $http({url:"http://localhost:8080/get/u=" + user, method: 'POST'}).then(function(data, status){
                if(data.data == "fail") {
                    alert("Reteival Failed");
                }
                else
                {
                    //alert(data.data.Username);
                    $scope.user = data.data.Username;
                    $scope.color = data.data.Color;
                    $scope.height = data.data.Height;
                    $scope.weight = data.data.Weight;
                    $scope.amazon = data.data.Amazon;
                }

            });
        }
    }
});
```

- Moving onto register page where user register their details user name and password.

```
<head >
    <meta charset="utf-8">
    <title>Register Page Lab2</title>
    <link rel="stylesheet" href="app.css">
</head>
<body ng-controller="registerController">
    <div id="registerContainer">
        <h1>Register:</h1>
        <div class="center">
            <input type="text" id="txt_userid" placeholder="Enter Username"/>
        </div>
        <div class="center">
            <input type="password" id="txt_userpass" placeholder="Enter Password"/>
        </div>
        <div class="centerButton">
            <button id="btn_register" ng-click="register()" class="centerButton">Register</button>
        </div>
    </div>
    <script type="text/javascript" src="http://code.angularjs.org/1.6.4/angular.js"></script>
    <script src="app.js"></script>
</body>
</html>
```

- User can login with registered details using login page given below.

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Login Page Lab2</title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="app.css">

</head>
<body class="body">
  <div  ng-controller="loginCtrl">
    <div id="LoginContainer">
      <h1>Login:</h1>
      <div class="center">
        <input type="text" id="txt_userid" placeholder="Enter Username"/>
      </div>
      <div class="center">
        <input type="password" id="txt_userpass" placeholder="Enter Password"/>
      </div>
      <div class="centerButton">
        <button id="btn_login" ng-click="login()" class="centerButton">Login</button>
      </div>
    </div>
    <div id="registerLinkContainer">
      <a href="register.html">Register</a>
    </div>
  </div>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.js"></script>

    <script src="app.js" type="text/javascript"></script>
</body>
</html>
```

- Home page having item details shown below.

```
        </form>
    </div>

    <div id="homeDeleteContainer">
        <h1>User Details:</h1>
        <div id="homeItems">
            <h2>{{user}}</h2>
        </div>
        <div id="homeColor">
            <h2>{{item}}</h2>
        </div>
        <div id="homeHeight">
            <h2>{{cost}}</h2>
        </div>
        <div id="homeWeight">
            <h2>{{rating}}</h2>
        </div>
        <div id="homeAmazon">
            <h2>Bawarchi</h2>
        </div>
        <div class="centerButton">
            <button id="btn_delete" ng-click="delete()">Delete Info</button>
        </div>
    </div>
    <script type="text/javascript" src="http://code.angularjs.org/1.6.4/angular.js"></script>
    <script src="app.js"></script>
</body>
```

- Finally profile page is also implemented with user details.

```
<nav class="navbar navbar-default">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Lakshmana Kumar Mettu</a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav navbar-right">
        <li><a href="#">Gmail:lmknd@edu</a></li>
        <li><a href="#">contact me:12345678</a></li>

      </ul>
    </div>
  </div>
</nav>

<!-- First Container -->
<div class="container-fluid bg-1 text-center">
  <h3 class="margin">My Profile</h3>
  <img src="bird.jpg" class="img-responsive img-circle margin" style="display:inline" alt="Bird" width="350" height="350">
  <h3>This is Lakshmana Kumar Mettu</h3>
  <h3>Masters in Datascience</h3>
  <h3>UMKC</h3>
</div>
```

Output:

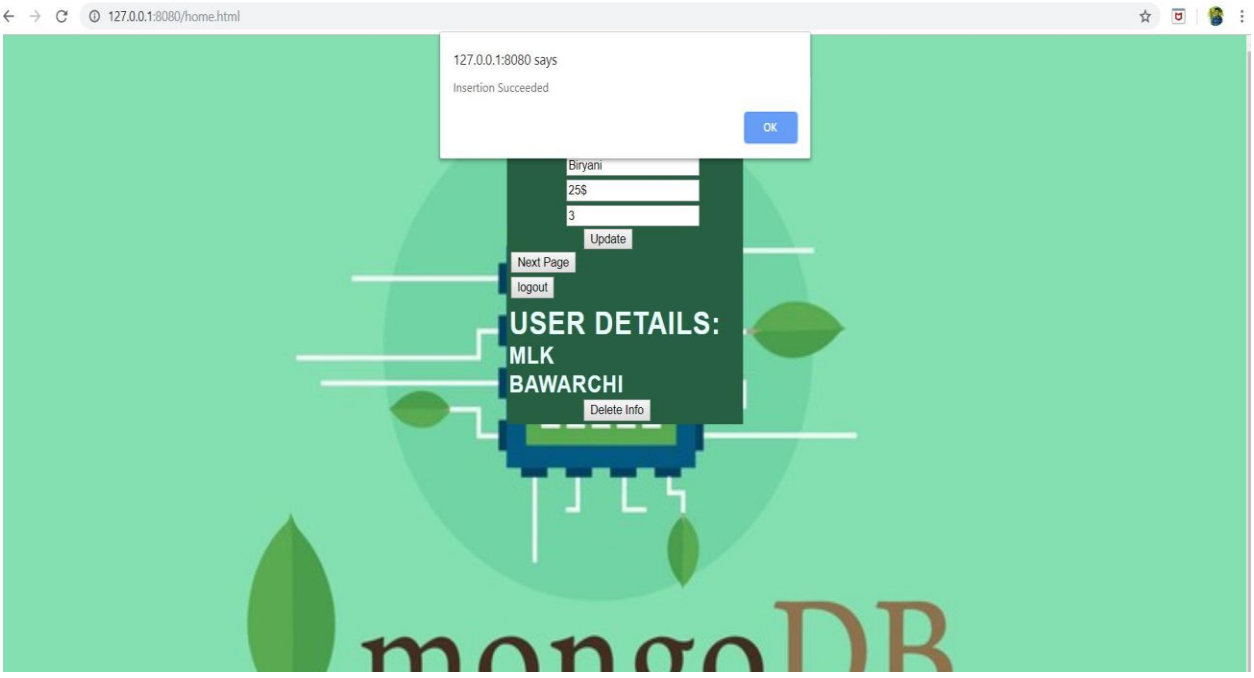
- Connecting to server shown below.

- Register page

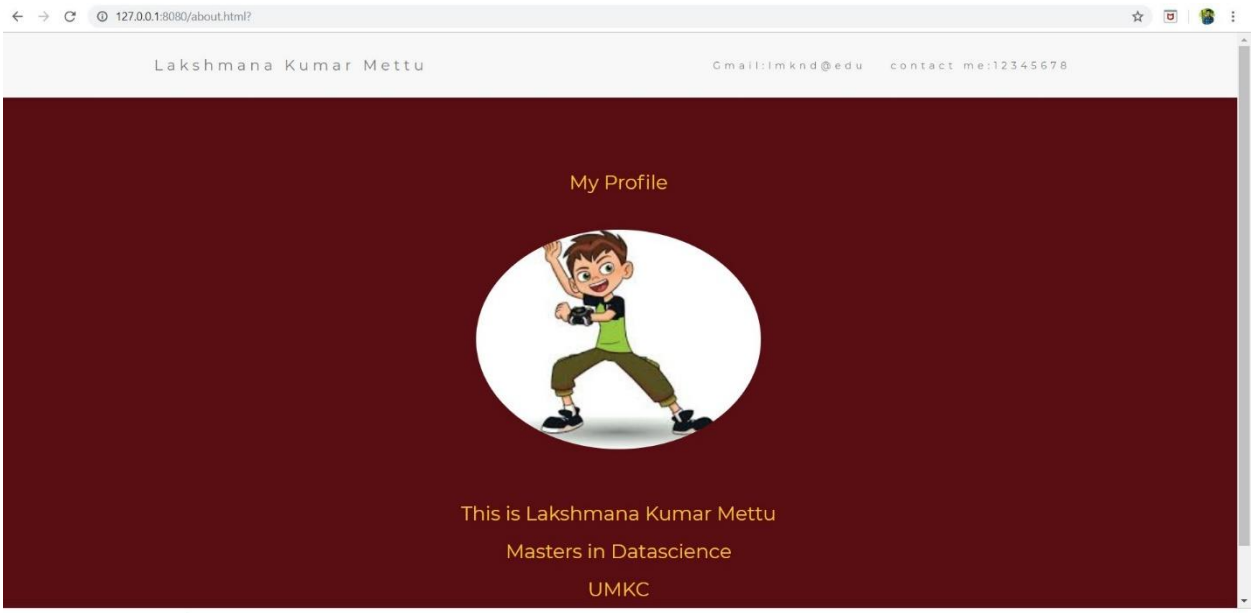
- Login page

- when user login with saved details they will re-direct to home page shown below.

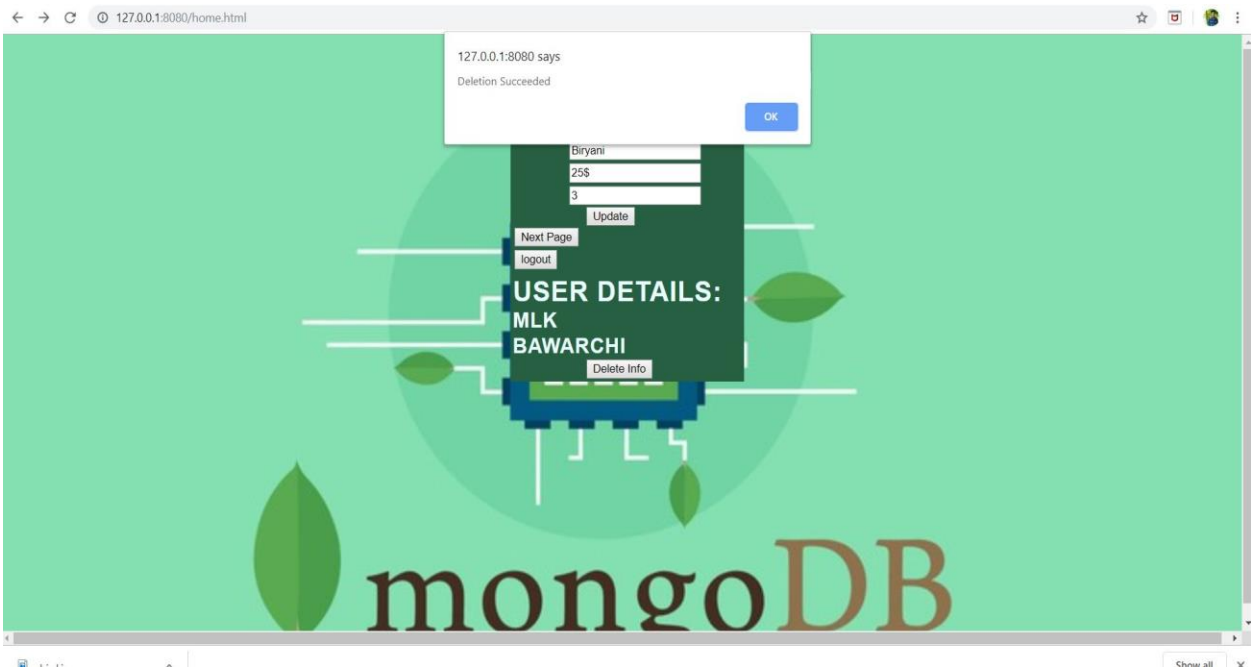
- Details inserting and updating in home page



- When user click on next page, redirects to profile page.

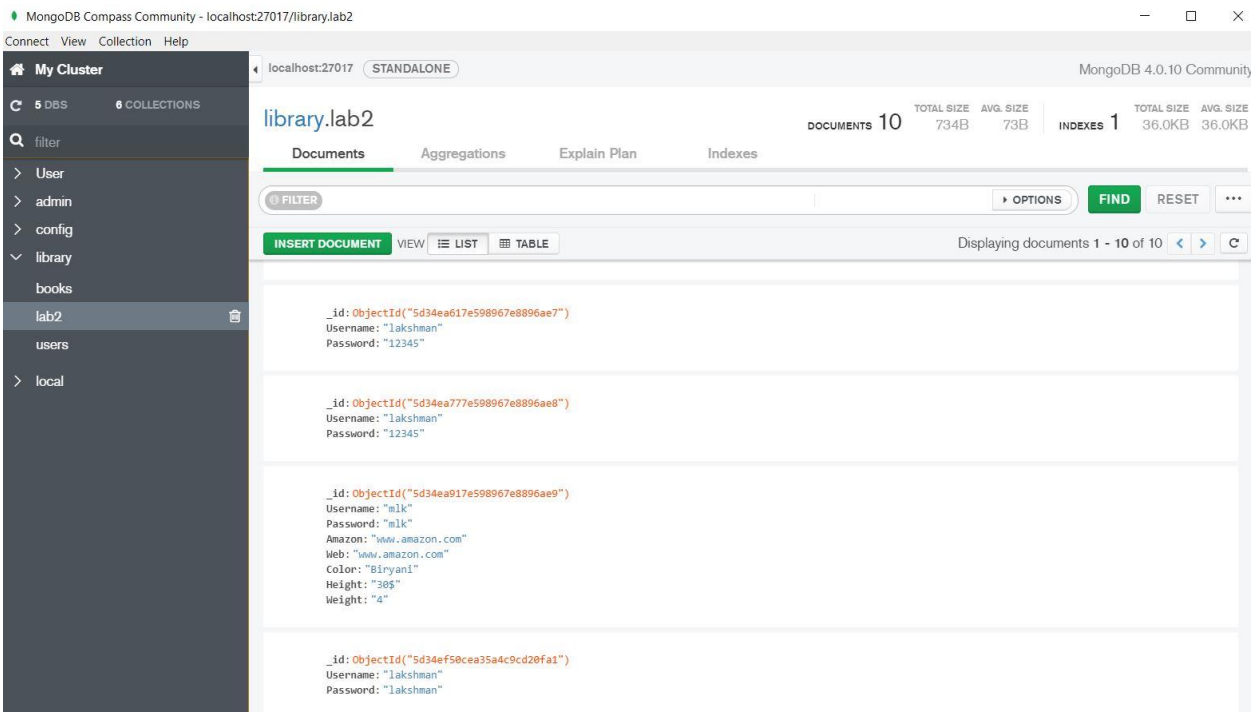


- Details deletion in database

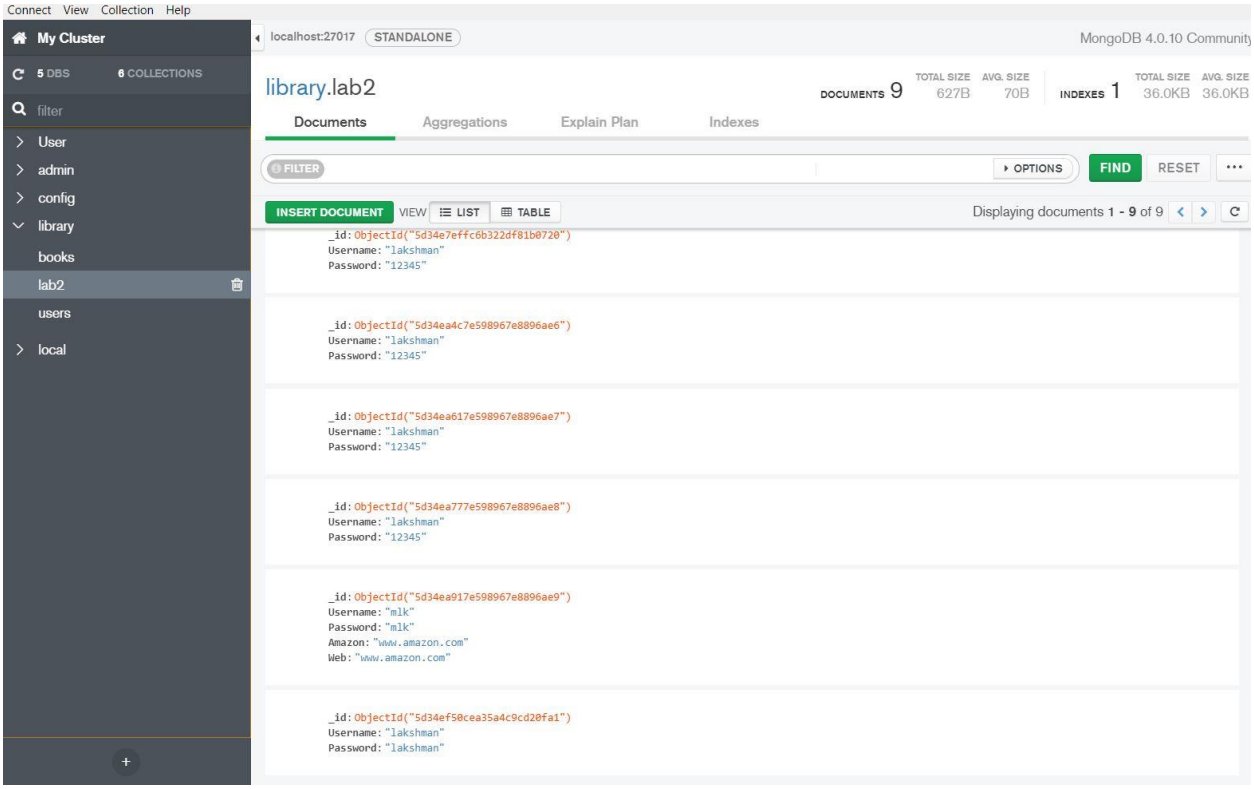


- when user click on logout button, it will redirect to login page given here.

- Showing updated details in database



- Screenshot to show details are deleted in database



Task-2:

Objective:

- To design an android application
- It has normal login and signup
- It also has social login like Facebook, Gmail etc.
- It should display the user details like DOB, mail, location etc. in the welcome page.
- It has logout button

Approaches and Workflow:

- Coming to the code part it has two activities.

1. MainActivity.java
2. HomeActivity.java

- Main activity has normal login button, user login with their given username and password.
- It also has social logins Gmail and Facebook oauth.
- Using Json parser Facebook details are validated
- And by googleAPI oauth class Gmail credentials are validated.
- If user enters wrong username or password It shows wrong credentials.
- The corresponding code snippets are shown below.

```
public class MainActivity extends AppCompatActivity  implements View.OnClickListener, GoogleApiClient.OnConnectionFailedListener {
    String usernames[] = new String[5];
    String passwords[] = new String[5];
    LoginButton loginButton;
    TextView textView;
    CallbackManager callbackManager;

    private LinearLayout profile_section;
    private Button signout;
    private SignInButton signin;
    private TextView name, email;
    private ImageView profile_picture;
    public static GoogleApiClient googleApiClient;
    private static final int REQ_CODE = 9001;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        FacebookSdk.sdkInitialize(getApplicationContext());
        setContentView(R.layout.activity_main);
        TextView textviewCtrl = (TextView) findViewById(R.id.textView);
        textviewCtrl.setTextColor(Color.RED);
        textviewCtrl.setText("");
        usernames[0]="lakshman";
        passwords[0]="12345";
```

```
loginButton = (LoginButton)findViewById(R.id.fb_login_btn);
textView = (TextView)findViewById(R.id.textView2);
callbackManager = CallbackManager.Factory.create();
final String[] name = new String[7];
signin = (SignInButton)findViewById(R.id.google_login_btn);
signin.setOnClickListener(this);
GoogleSignInOptions signInOptions = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN).requestScopes(new Scope(Scopes.PLUS_LOGIN)).requestEmail().requestProfile()
googleApiClient = new GoogleApiClient.Builder( context: this).enableAutoManage( fragmentActivity: this, onConnectionFailedListener: this).addApi(Auth.GOOGLE_SIGN_IN_API,signInOptions).build();
loginButton.setReadPermissions(Arrays.asList("public_profile,email"));
loginButton.registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) {
        GraphRequest request = GraphRequest.newMeRequest(
                loginResult.getAccessToken(),
                new GraphRequest.GraphJSONObjectCallback() {
                    @Override
                    public void onCompleted(
                            JSONObject object,
                            GraphResponse response) {
                        try {
                            final FacebookRequestError error = response.getError();
                            if (error != null) {
                                String message = error.getErrorMessage();
                                if (message == null) {
                                    message = "Error staging photo.";
                                }
                                throw new FacebookGraphResponseException(response, message);
                            }
                            final JSONObject data = response.getJSONObject();
                            if (data == null) {
                                throw new FacebookException("Error staging photo.");
                            }
```

```
                                }
                                throw new FacebookGraphResponseException(response, message);
                            }
                            final JSONObject data = response.getJSONObject();
                            if (data == null) {
                                throw new FacebookException("Error staging photo.");
                            }
                            name[0] = data.optString( name: "name");
                            name[1] = data.optString( name: "birthday");
                            name[2] = data.optString( name: "email");
                            final JSONObject location = data.optJSONObject("location");
                            name[3] = location.optString( name: "name");
                            name[4] = data.optString( name: "id");
                            name[5] = "https://graph.facebook.com/" + name[4] + "/picture?width=400&height=400";
                            name[6] = data.optString( name: "gender");
                        } catch (Exception e) {
                            Log.v( tag: "Login", String.valueOf(e));
                        }
                        Intent redirect = new Intent( packageContext: MainActivity.this, HomeActivity.class);
                        redirect.putExtra( name: "USER_NAME", name[0]);
                        redirect.putExtra( name: "BIRTHDAY", name[1]);
                        redirect.putExtra( name: "EMAIL", name[2]);
                        redirect.putExtra( name: "LOCATION",name[3]);
                        redirect.putExtra( name: "ID",name[4]);
                        redirect.putExtra( name: "IMAGE_URL",name[5]);
                        redirect.putExtra( name: "GENDER",name[6]);
                        MainActivity.this.startActivity(redirect);

                    }
                });
```

- Validating the credentials

```
public  void validate(View view){


    EditText usernameCtrl = (EditText) findViewById(R.id.editText3);
    EditText passwordCtrl = (EditText) findViewById(R.id.editText4);

    String username = usernameCtrl.getText().toString().toLowerCase();
    String password = passwordCtrl.getText().toString();
    int i;
    for(i=0;i<usernames.length;i++) {
        if ( username.equals(usernames[i]) && password.equals(passwords[i])) {
            Intent redirect = new Intent( packageContext: MainActivity.this, HomeActivity.class);
            redirect.putExtra( name: "USER_NAME",  value: "Lakshman");
            redirect.putExtra( name: "BIRTHDAY",  value: "10/07/1995");
            redirect.putExtra( name: "EMAIL",  value: "lakshmankumarmettu@gmail.com");
            redirect.putExtra( name: "LOCATION", value: "kansas");
            redirect.putExtra( name: "ID", ((String) null));
            redirect.putExtra( name: "IMAGE_URL", ((String) null));
            redirect.putExtra( name: "GENDER", value: "male");
            MainActivity.this.startActivity(redirect);
            break;
        }
    }
```

- Gmail login implementation

```
private void signIn(){
    Intent intent = Auth.GoogleSignInApi.getSignInIntent(googleApiClient);
    startActivityForResult(intent,REQ_CODE);
}

private void handleResult(GoogleSignInResult result)
{
    if(result.isSuccess())
    {
        GoogleSignInAccount account = result.getSignInAccount();
        String name = account.getDisplayName();
        String email = account.getEmail();
        Uri img_uri = account.getPhotoUrl();
        String acc = account.getAccount().toString();
        String image_url = null;
        if(img_uri!=null)
        {
            image_url = img_uri.toString();
        }
        Log.v( tag: "Login google", msg: "12");
        Intent redirect = new Intent( packageContext: MainActivity.this, HomeActivity.class);
        redirect.putExtra( name: "USER_NAME", name);
        redirect.putExtra( name: "BIRTHDAY",  value: "null");
        redirect.putExtra( name: "EMAIL", email);
        redirect.putExtra( name: "LOCATION", value: "null");
        redirect.putExtra( name: "ID", value: "null");
        redirect.putExtra( name: "IMAGE_URL",image_url);
        redirect.putExtra( name: "GENDER", value: "null");
        MainActivity.this.startActivity(redirect);
    }
}
public static void signOutFromGoogle() {
    Auth.GoogleSignInApi.signOut(googleApiClient).setResultCallback(
        (ResultCallback) (status) -> { return; });
```

- Coming to the Home activity, where user details are implemented.
- It also containing logout button
- code snippet for home activity is given below

```
import ...

public class HomeActivity extends AppCompatActivity {
    // private GoogleApiClient googleApiClient;
     GoogleApiClient mGoogleApiClient;
     GoogleSignInClient mGoogleSignInClient;
     private Button signout;
     @Override
     protected void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.welcome);
         Intent intent=getIntent();
         String username = intent.getStringExtra( name: "USER_NAME");
         String birthday = intent.getStringExtra( name: "BIRTHDAY");
         String email = intent.getStringExtra( name: "EMAIL");
         String location = intent.getStringExtra( name: "LOCATION");
         String gender = intent.getStringExtra( name: "GENDER");
         String image_url = intent.getStringExtra( name: "IMAGE_URL");
         TextView textviewCtrl = (TextView) findViewById(R.id.textView2);
         TextView textviewCtrl2 = (TextView) findViewById(R.id.textView3);
         TextView textviewCtrl3 = (TextView) findViewById(R.id.textView4);
         TextView textviewCtrl4 = (TextView) findViewById(R.id.textView5);
         TextView textviewCtrl5 = (TextView) findViewById(R.id.textView6);
         signout = (Button) findViewById(R.id.button3);
         textviewCtrl.append(" "+username);
         textviewCtrl2.append(": "+birthday);
         textviewCtrl3.append(": "+email);
         textviewCtrl4.append(": "+location);
         textviewCtrl5.append(": "+gender);
         if(image_url!=null){
         new DownloadImageTask((ImageView) findViewById(R.id.imageView))
                 .execute(image_url);}
     }
```

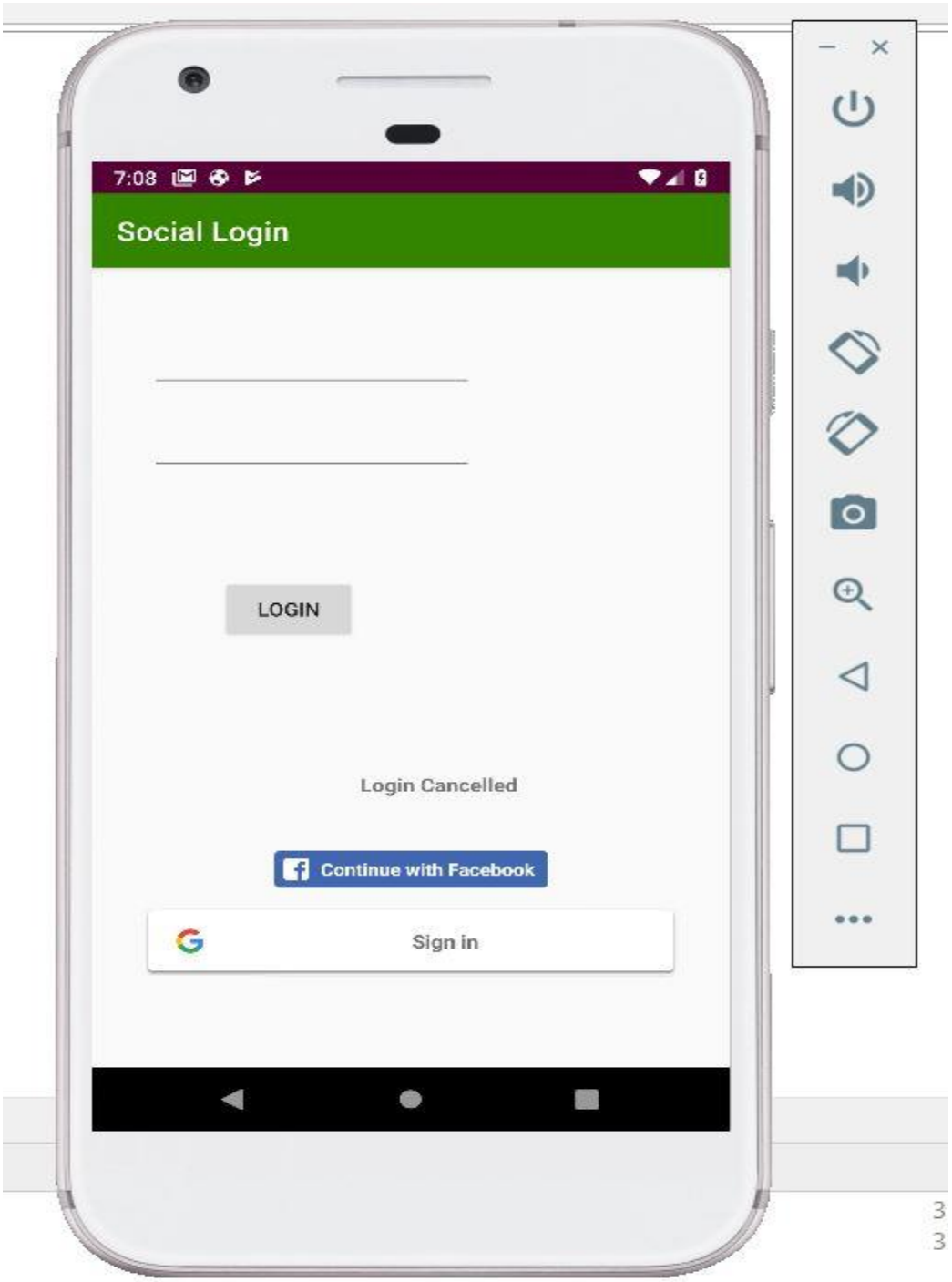
```
    public void logout(View view){
        if (MainActivity.googleApiClient.isConnected()) {
            MainActivity.googleApiClient.connect();
            signout.setOnClickListener((v) → {
                    MainActivity.signOutFromGoogle();
            });
        }
        LoginManager.getInstance().logOut();
        Intent redirect = new Intent( packageContext: HomeActivity.this,MainActivity.class);
        HomeActivity.this.startActivity(redirect);
    }

}

final  class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {
    ImageView bmImage;

    public DownloadImageTask(ImageView bmImage) { this.bmImage = bmImage; }

    protected Bitmap doInBackground(String... urls) {
        String urldisplay = urls[0];
        Bitmap mIcon11 = null;
        try {
            InputStream in = new java.net.URL(urldisplay).openStream();
            mIcon11 = BitmapFactory.decodeStream(in);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return mIcon11;
    }

    protected void onPostExecute(Bitmap result) { bmImage.setImageBitmap(result); }
}
```

- Google-oauth class shown below.

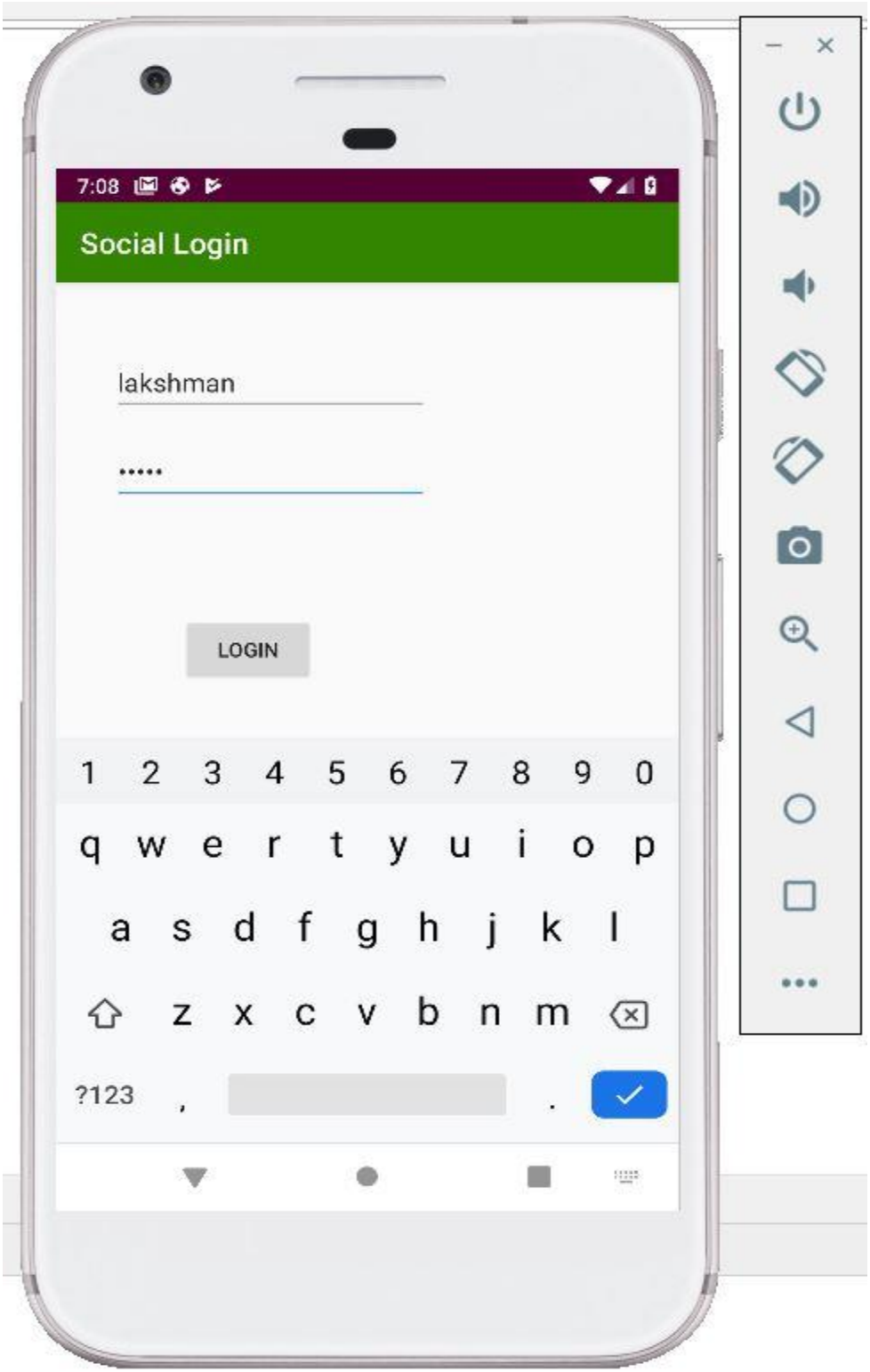
```
    public final Bundle toBundle() {
        Bundle var1;
        (var1 = new Bundle()).putString("consumer_package", (String)null);
        var1.putBoolean("force_save_dialog", this.zzm);
        return var1;
    }

    /** @deprecated */
    @Deprecated
    public static class Builder {
        protected Boolean zzn = false;

        public Builder() {
        }

        public Auth.AuthCredentialsOptions.Builder forceEnableSaveDialog() {
            this.zzn = true;
            return this;
        }

        public Auth.AuthCredentialsOptions zzc() {
            return new Auth.AuthCredentialsOptions( var1: this);
        }
    }
}
```

Output:

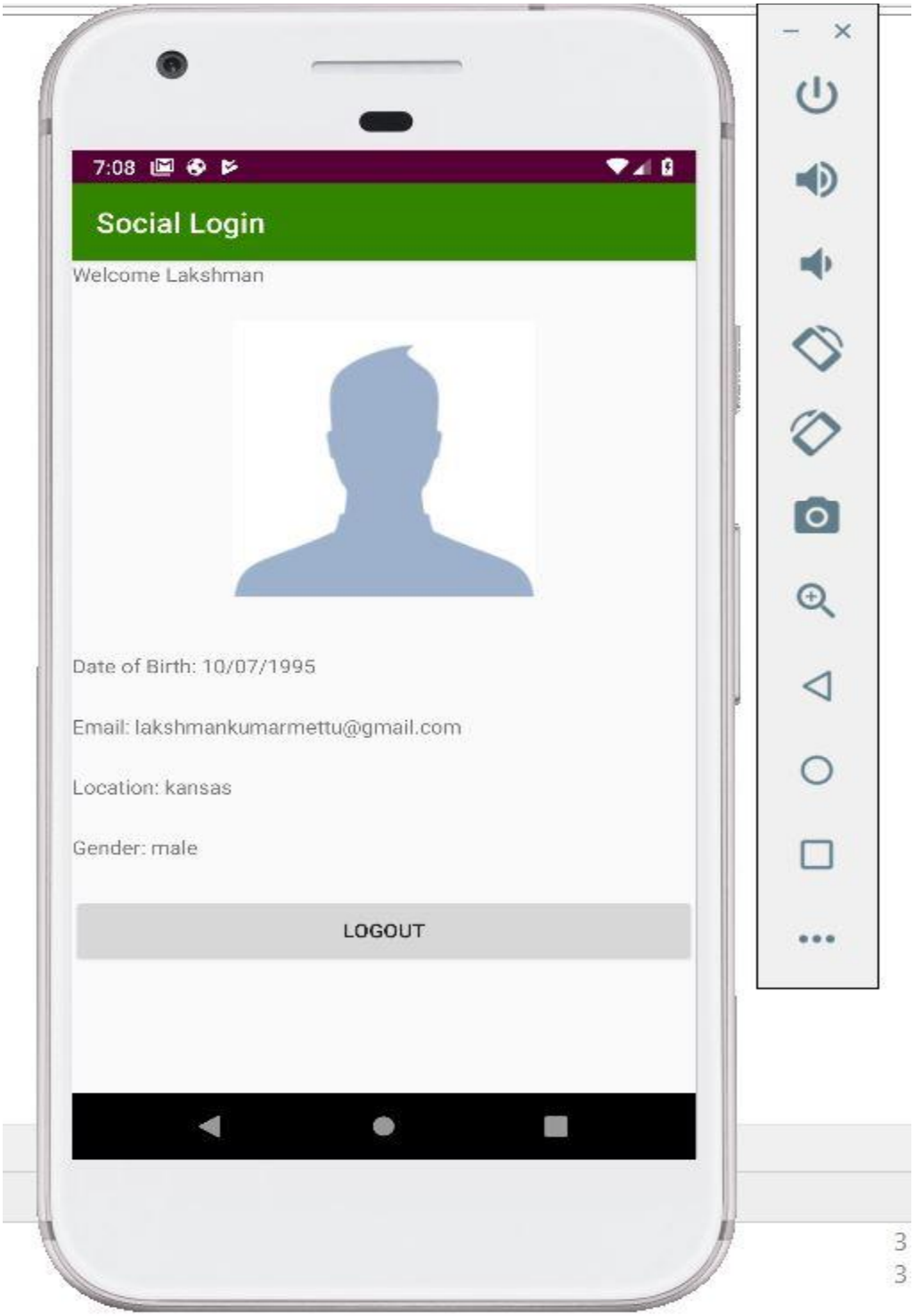
- Main activity having login, sign up and social logins Facebook and Gmail.



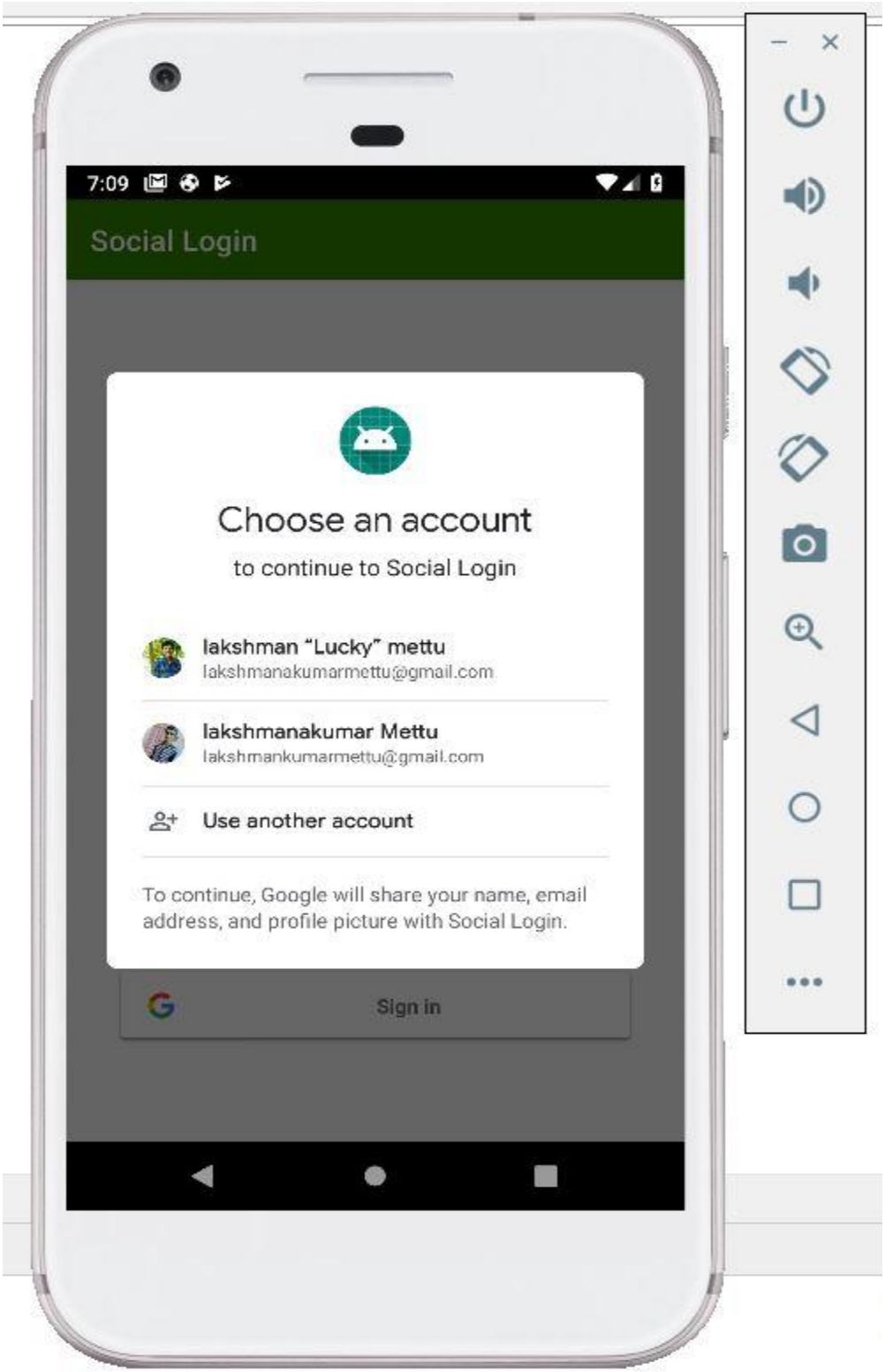
- Normal login



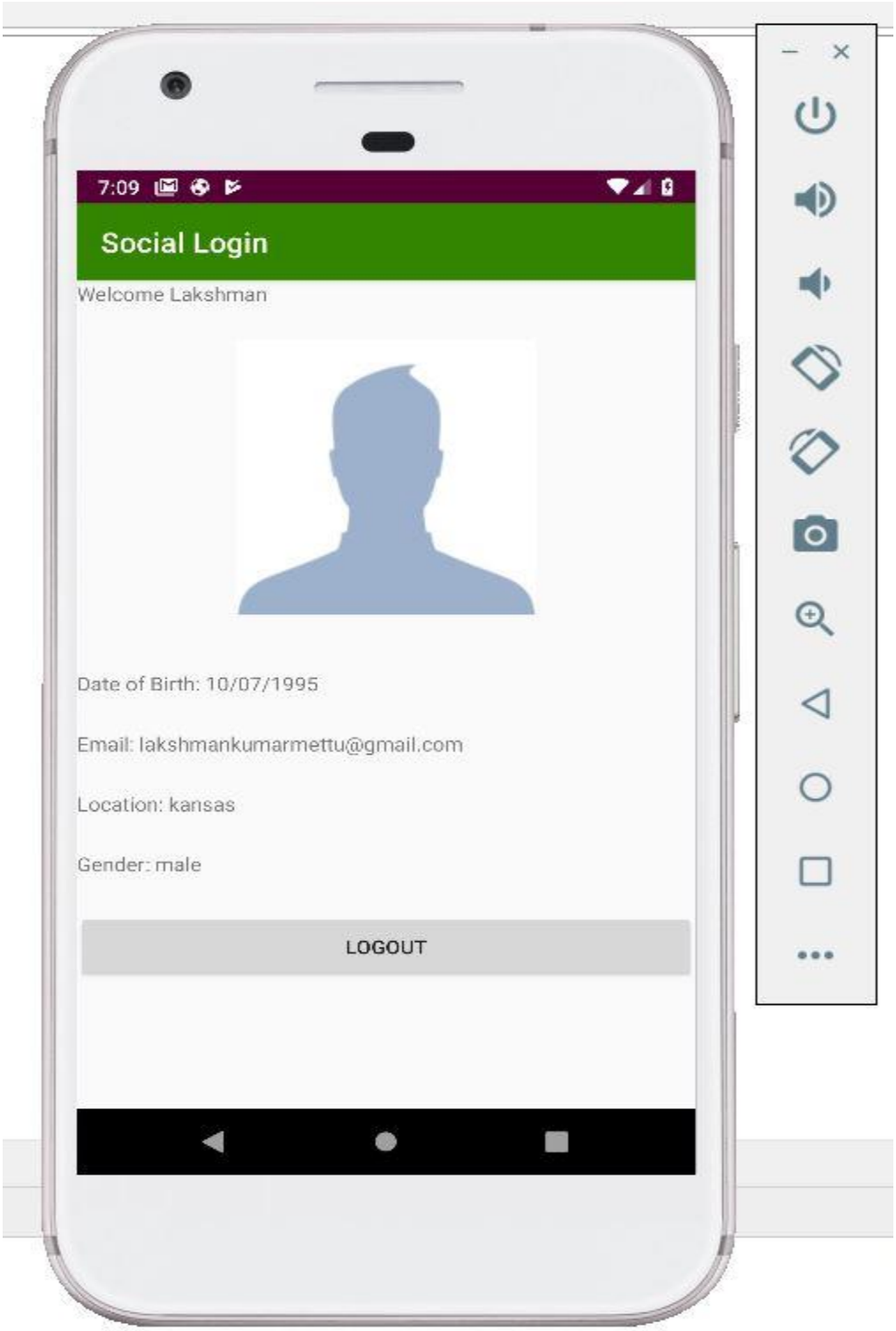
- Welcome screen showing the user details



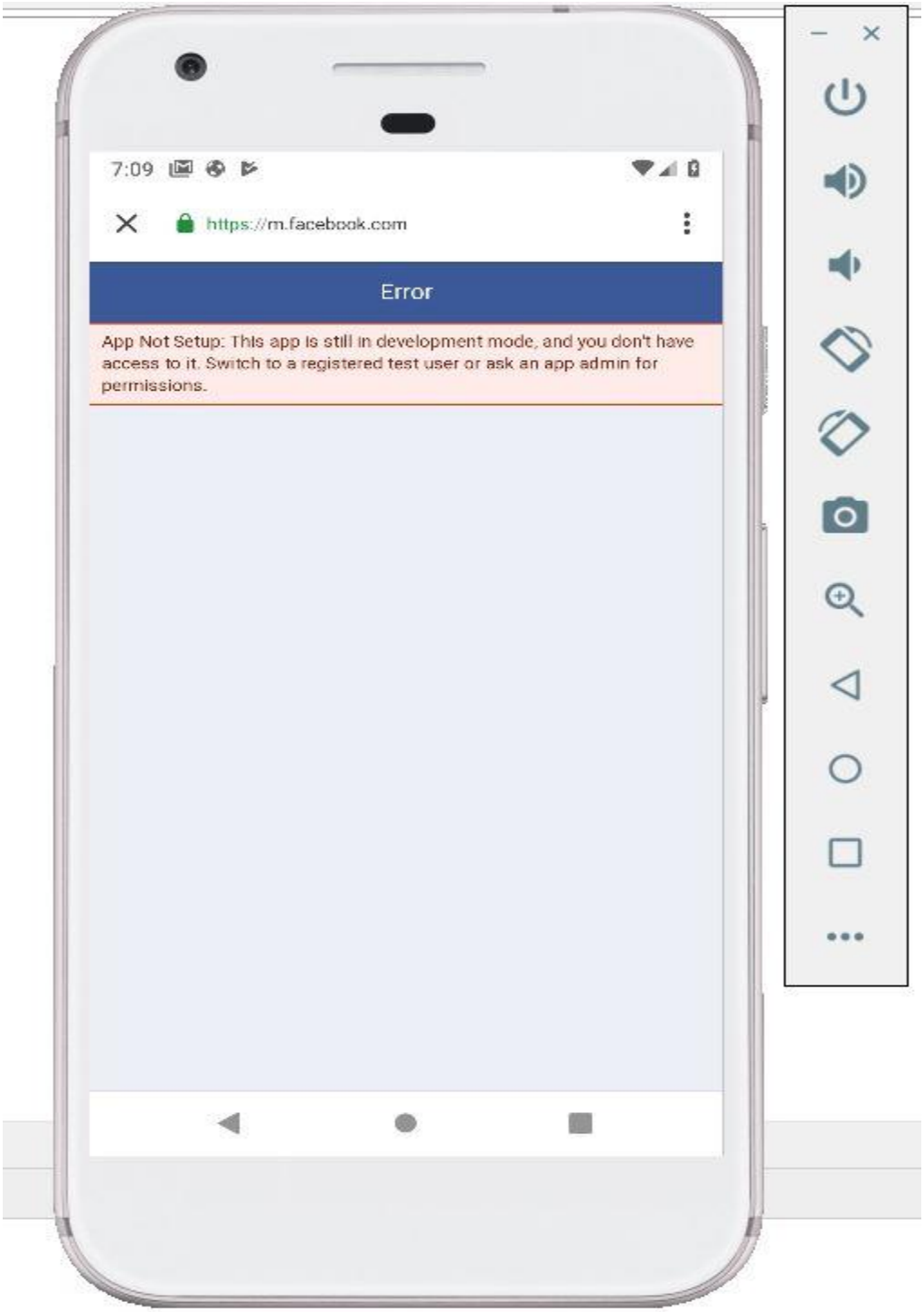
- Gmail login


7:09
Social Login
Choose an account
to continue to Social Login
lakshman "Lucky" mettu
lakshmanakumarmettu@gmail.com
lakshmanakumar Mettu
lakshmankumarmettu@gmail.com
Use another account
To continue, Google will share your name, email address, and profile picture with Social Login.
Sign in

- Redirecting to welcome screen with gmail login



- Facebook login
- Since password attempts are exceeded so it is showing error message otherwise it redirects to welcome page.



Conclusion:

- Hence Mean stack application as per the requirements has been completed
- Also, Implementation of android application using social login and signup with normal details has accomplished
- Learnt real time database usage in web as well as in android.