Lab Assignment 2
Team 14-2

Gulnoza Khakimova (ID: 28)
Pranitha Saroj Karumanchi (ID: 24)

Part 1

Objective

The purpose of this task was to create Android application which has login and sign up page. registration page should use any of cloud services to store credentials. Home activity should use Machine learning API. Also application should use smart phone hardware features.

Features

- Application has login page and login with social media feature which uses Facebook to log in
- Application has sign up page which integrates with Firebase and stores credentials which helps user to login in into application.
- Application uses camera to take a photo and uses Deep Learning API - Clarifai API to detect what is displayed on image.
- Application has user friendly interface.

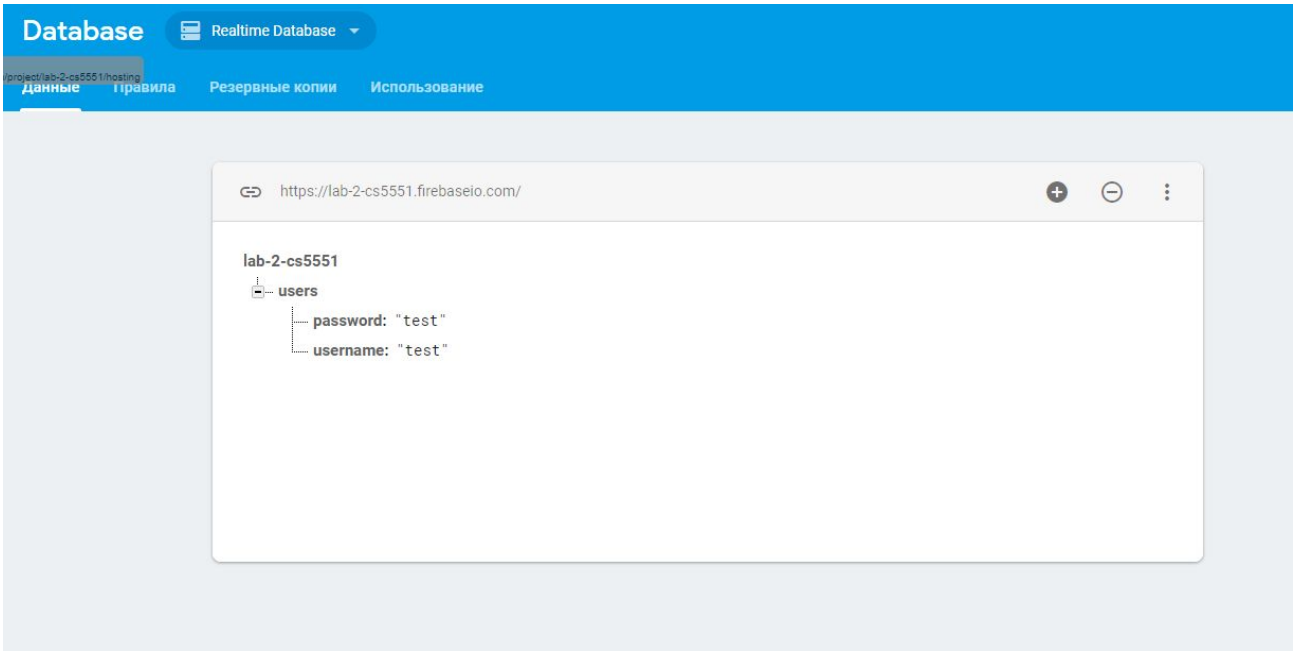
Styling/UI

Since UI is very important for user and success of application we did not avoided spending time on designing our application. User interface of application is user friendly. It is simple to navigate through application. We used custom settings for buttons, text fields and set neutral background image. Each page has consistent layout and design. We created Login layout, Sign Up Layout and Home Layout.

DataBase

We used Firebase database for our project wich stores user's credentials when they sign up to use our application. We store username and password in our database so when user tried to login in we check our database to make sure that they present in it. Application makes a call to Firebase Realtime database to perform login validation. If user does not have account, they can click on sign up button and register by filling out required information. After that we store their e-mail addresses and passwords to RealTime database so next time user

can use those credentials to login into application. Please see screenshot of our database below:



Source Code

Our application is Android application which has Login Activity, Sign Up Activity, Facebook Activity and HomePage Activity and each Activity has related to them layout.

Login Activity

Login activity has two fields username and password. Where user can enter their credentials. When Login button is clicked we perform username and password validation - we make sure that they are not empty and that they are registered users by checking it in Firebase database.
Here on create we are getting hash key do we can perform Facebook login also we are initializing buttons:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    FacebookSdk.sdkInitialize(getApplicationContext());
    callbackManager = CallbackManager.Factory.create();
    try {
        PackageInfo info = getPackageManager().getPackageInfo(
                packageName: "edu.umkc.anonymous.lab3",
                PackageManager.GET_SIGNATURES);
        for (android.content.pm.Signature signature : info.signatures) {
            MessageDigest md = MessageDigest.getInstance("SHA");
            md.update(signature.toByteArray());
            Log.d( tag: "KeyHash:", Base64.encodeToString(md.digest(), Base64.DEFAULT));
        }
    } catch (PackageManager.NameNotFoundException e) {
    } catch (NoSuchAlgorithmException e) {
    }
    setContentView(R.layout.activity_login);

    Button myFab = (Button) findViewById(R.id.fb);
    myFab.setOnClickListener((v) -> {
            Intent redirect = new Intent( packageContext: LoginActivity.this, FacebookActivity.class);
            startActivity(redirect);
    });
    Button log = (Button) findViewById(R.id.btnLogin);
    log.setOnClickListener((v) -> { checkCredentials(v); });
```

```
    Button myFab = (Button) findViewById(R.id.fb);
    myFab.setOnClickListener((v) -> {
            Intent redirect = new Intent( packageContext: LoginActivity.this, FacebookActivity.class);
            startActivity(redirect);
    });
    Button log = (Button) findViewById(R.id.btnLogin);
    log.setOnClickListener((v) -> { checkCredentials(v); });

    Button signup = (Button) findViewById(R.id.btnSignup);
    signup.setOnClickListener((v) -> { signup(v); });
}
```

'`checkCredentials`' function is called to validate user name and password. We are getting username and password from EditText fields and comparing them agins usernames and passwords stored in Firebase Realtime database by looping through childs, if they are present we set validation flag to true:

```
public void checkCredentials(View v) {
    EditText usernameCtrl = (EditText)findViewById(R.id.editTextUser);
    EditText passwordCtrl = (EditText) findViewById(R.id.editTextPassword);
    TextView errorText = (TextView) findViewById(R.id.lbl_Error);
    final String userName = usernameCtrl.getText().toString();
    final String password = passwordCtrl.getText().toString();
    temp = v;
    // Verify username and password not empty
    if(!userName.isEmpty() && !password.isEmpty()) {

        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot snapshot : dataSnapshot.getChildren()){
                    String name = snapshot.child("username").getValue().toString();
                    String pass = snapshot.child("password").getValue().toString();
                    if(name.equals(userName) && pass.equals(password)){
                        validationFlag = true;
                        break;
                    }
                }
                check(temp);
```

Below code shows how we navigate between pages by calling Intent:

```
    public void check(View v){
        if(!validationFlag) {
            // errorText.setVisibility(View.VISIBLE);
        }
        else {
            redirectToHomePage(v);
        }
    }

    public void signup (View v){
        Intent intent = new Intent( packageContext: this,SignupActivity.class);
        startActivity(intent);
    }

    public void redirectToHomePage(View v) {
        Intent redirect = new Intent( packageContext: LoginActivity.this, MainActivity.class);
        startActivity(redirect);
    }

    public void gotofb() {
        Intent redirect = new Intent( packageContext: LoginActivity.this, FacebookActivity.class);
        startActivity(redirect);
    }

}
```

Facebook activity which perform login by using social media:

```
public class FacebookActivity extends Activity {
    private TextView info;
    private LoginButton loginButton;
    private CallbackManager callbackManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        Log.d( tag: "Timur", msg: "1");
        super.onCreate(savedInstanceState);
        FacebookSdk.sdkInitialize(getApplicationContext());
        callbackManager = CallbackManager.Factory.create();
        setContentView(R.layout.activity_facebook);
        info = (TextView)findViewById(R.id.info);
        loginButton = (LoginButton)findViewById(R.id.login_button);        // If using in a fragment
//        mFacebookSignInButton.setFragment(this);

        loginButton.registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
            @Override
            public void onSuccess(LoginResult loginResult) {
//                info.setText(
//                        "User ID: "
//                                + loginResult.getAccessToken().getUserId()
//                                + "\n" +
//                                "Auth Token: "
//                                + loginResult.getAccessToken().getToken()
//                );
                Intent redirect = new Intent( packageContext: FacebookActivity.this, MainActivity.class);
                startActivity(redirect);
//
            }
```

Sign up activity which stores username and password to our database:

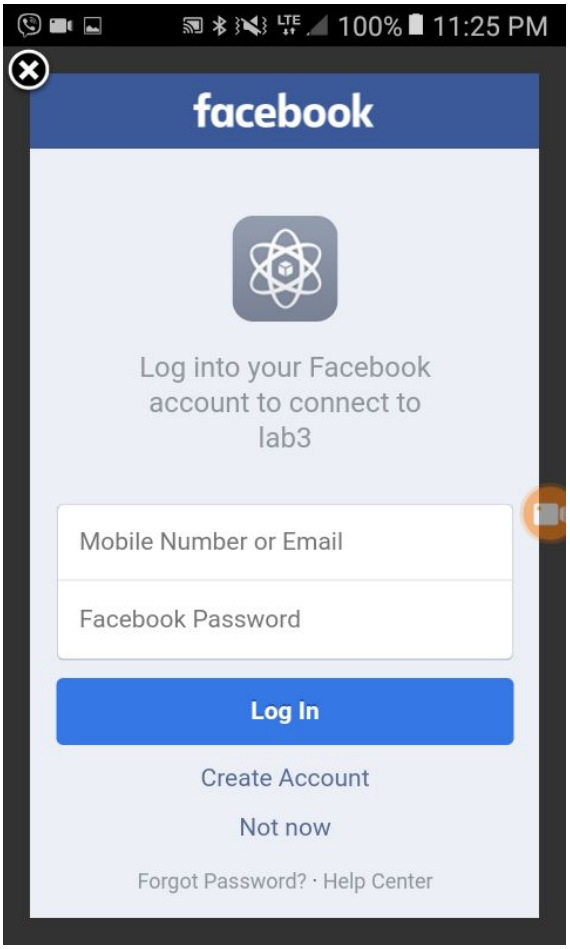
```
public class SignupActivity extends AppCompatActivity {
    private FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();
    private DatabaseReference myRef = firebaseDatabase.getReference();
EditText email;
EditText pass;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        email = (EditText)findViewById(R.id.email);
        pass = (EditText)findViewById(R.id.password);
    }

    public void save(View v) {
        String e = email.getText().toString();
        String p = pass.getText().toString();
       // Map<String,String> users = new HashMap<>();
        myRef.child("users").child("username").setValue(e);
        myRef.child("users").child("password").setValue(p);
       // users.put("username",e);
        //users.put("password",p);

        runOnUiThread(() -> {

                if (!isFinishing()) {
                    new AlertDialog.Builder( context: SignupActivity.this)
                            .setTitle("Registration Completed")
                            .setMessage("You are successfully signed up!")
                            .setCancelable(false)
                            .setPositiveButton( text: "ok", (dialog, which) -> {
                                    Intent redirect = new Intent( packageContext: SignupActivity.this, LoginActivity.class);
                                    startActivity(redirect);
                            }).show();
                }
```

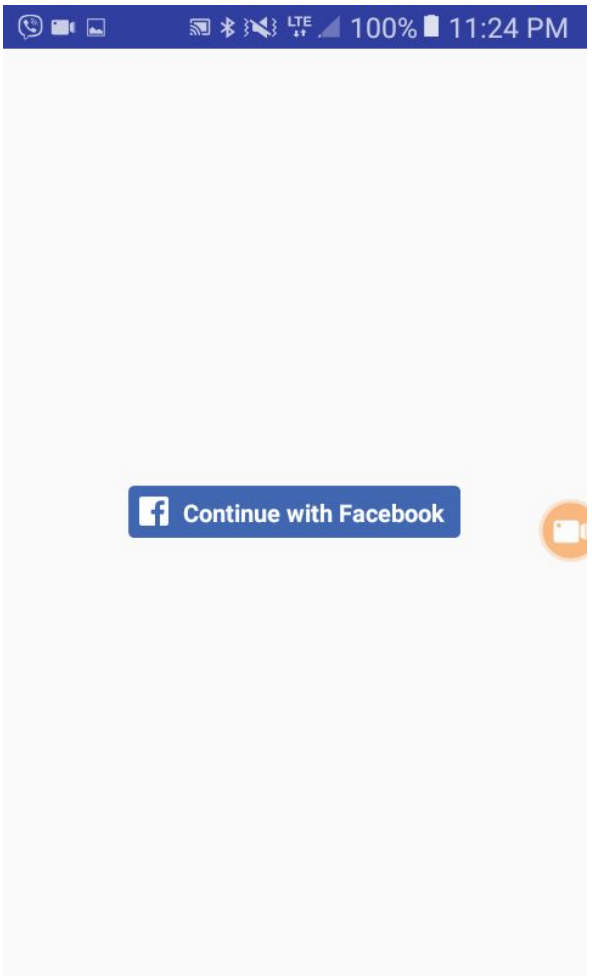
Main Activity perform API call to Clarifai to get make image recognition by url:

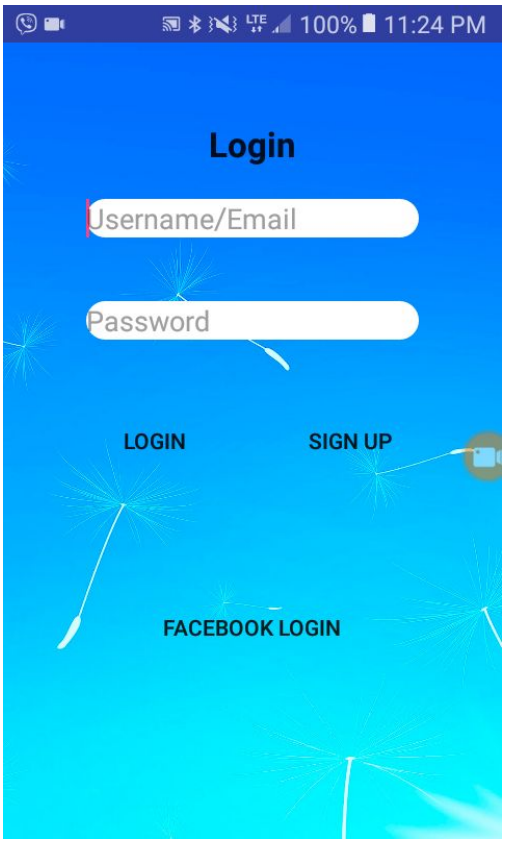
```
private class myAsyncTask extends AsyncTask<String, Integer, String> {
    String mTAG = "myAsyncTask";
    EditText url = (EditText) findViewById(R.id.url);
    String imageUrl = url.getText().toString();

    @Override
    protected void onPreExecute() {
    }

    @Override
    protected String doInBackground(String... arg) {

        final ClarifaiClient client = new ClarifaiBuilder( apiKey: "814464bfd2914a3b8ba9e6f7138ff131").buildSync();
        Model<Concept> generalModel = client.getDefaultModels().generalModel();

        PredictRequest<Concept> request = generalModel.predict().withInputs(
                ClarifaiInput.forImage(imageUrl)
        );
        List<ClarifaiOutput<Concept>> result = request.executeSync().get();
final String results = "Results: "+result.get(0).data().get(0).name().toString() +", "+result.get(0).data().get(1).name().toString()+", "+result.get(0).data(
        runOnUiThread(new Runnable() {

            @Override
            public void run() {

                TextView TV = (TextView) findViewById(R.id.clarifai);
                TV.setText(results);

            }
        });
        return "ok";
    }
```

Steps

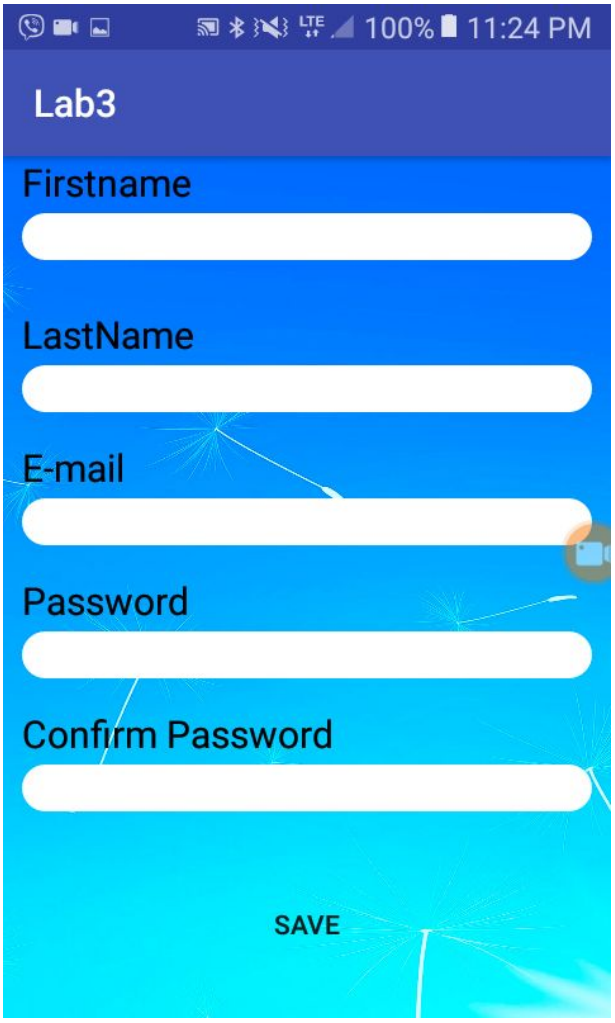
First page of application is login page, user needs to input their username and password then click login button or login can be performed by clicking login using Facebook button. If user want to login using their social media they need to provide their credentials which gets validated and user is redirected to home page.


100% 11:25 PM
facebook
Log into your Facebook account to connect to lab3
Mobile Number or Email
Facebook Password
Log In
Create Account
Not now
Forgot Password? · Help Center

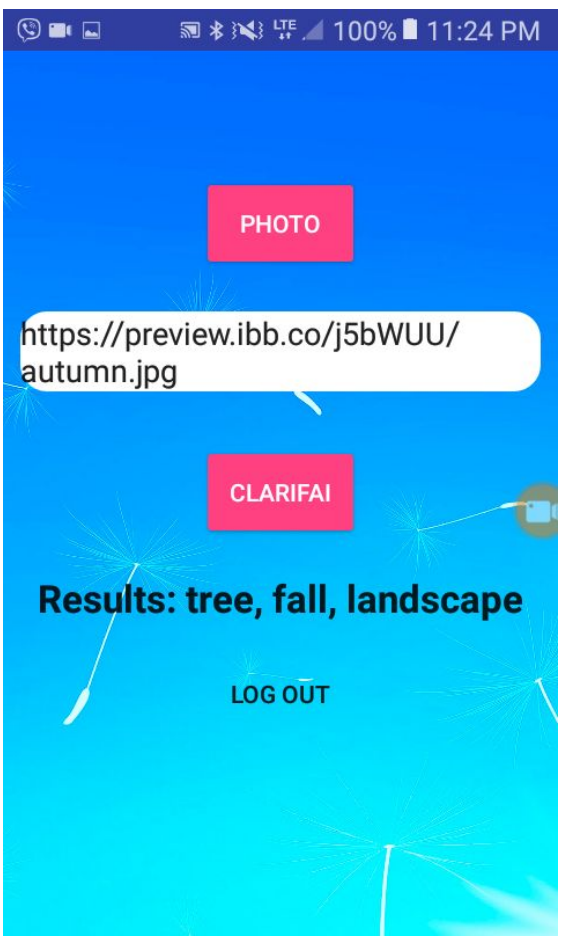

100% 11:24 PM
Continue with Facebook

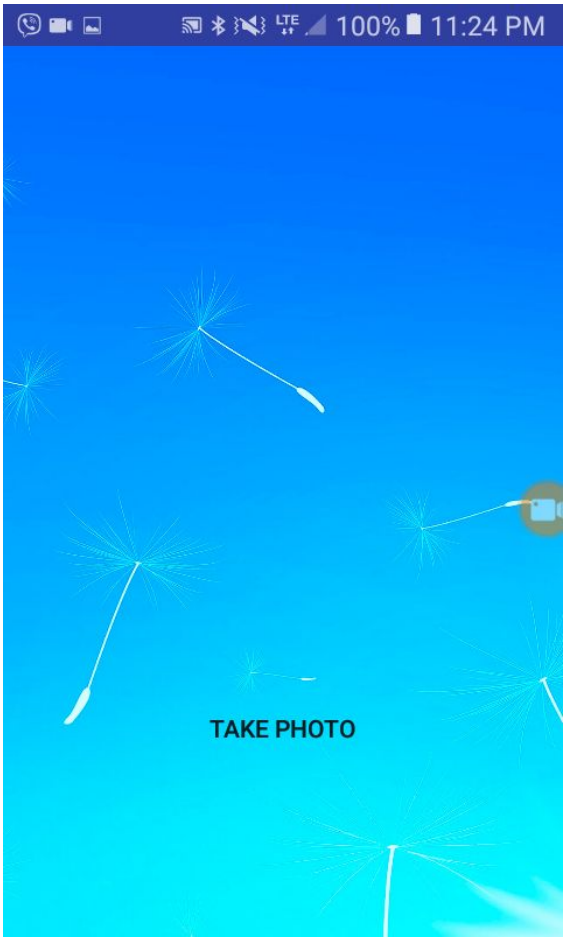

100% 11:24 PM
Login
Username/Email
Password
LOGIN
SIGN UP
FACEBOOK LOGIN

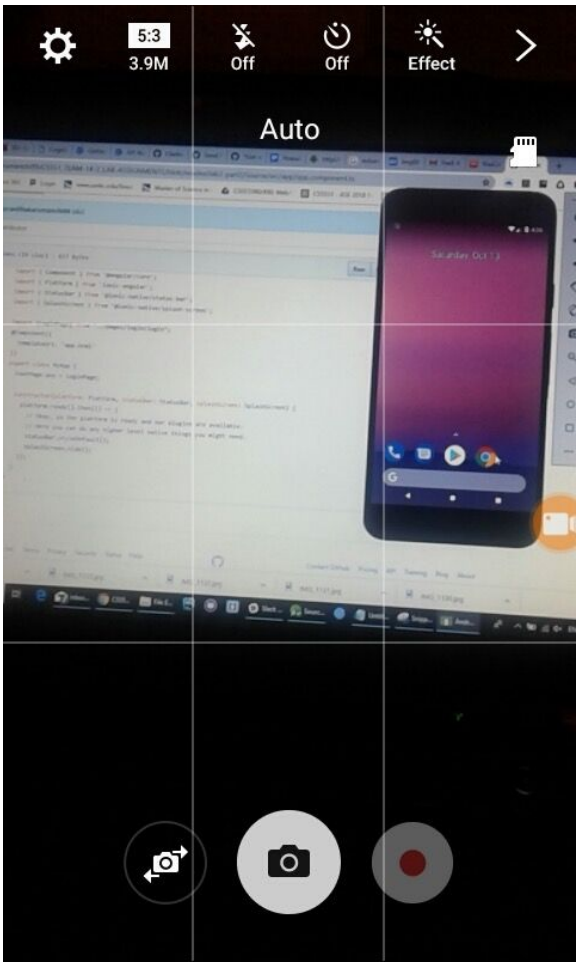
If user does not have an account they can sign up by clicking sign up button, which will take them to registration page where user is prompt to fill out required information. After registration user can log in using their email and password.

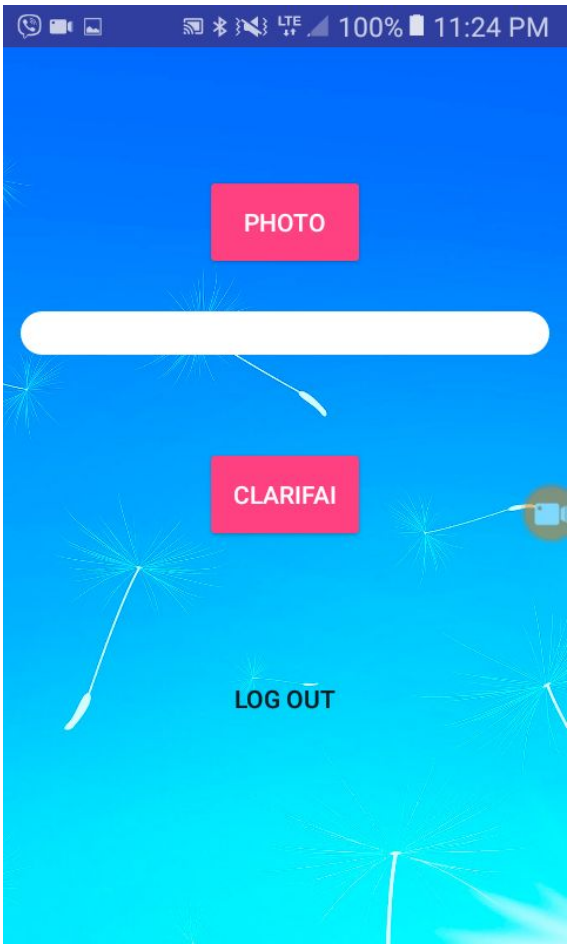


Home page has a button Photo which opens up camera and take a photo.
Home page also have Field where user can enter image url so it gets detected by Clarifai API and image classification will be outputted on the screen.


100% 11:24 PM
PHOTO
https://preview.ibb.co/j5bWUU/autumn.jpg
CLARIFAI
Results: tree, fall, landscape
LOG OUT


100% 11:24 PM
TAKE PHOTO





APIs

We used Clarifai API deep learning API to classify an image. Url of the image is send to the server where model runs and perform image recognition. Then result is returned in Json format. In order to make API call we need to get Key which will be given to us when we create account in Clarifai.

References

1. https://clarifai.com/developer/guide/android-sdk#android-sdk
2. https://developer.android.com/reference/java/net/HttpURLConnection
3. https://console.firebase.google.com

Part 2

Objectives

Objective of task 2 was to create Ionic application which runs both on Android platform and Web Application. Application should use any of Machine Learning APIs and should have good design and implementation.

Features

- Application work on both Web and Android
- Application has login and sign up pages
- Application perform Face detection to predict gender and age
- Application has text to speech functionality

Styling/UI

Application has easy to navigate and good design interface which is same for both Web and Android application. It is user friendly and user custom features such as text fields with rounded corners and custom buttons. User will have good experience while using our application.

Source Code

We created ionic application which will work both on Android and Web browser. Here we are defining our app.ts:

```
import { Component } from '@angular/core';
import { Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import {LoginPage} from "../pages/login/login";
@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage:any = LoginPage;

  constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {
    platform.ready().then(() => {
      // Okay, so the platform is ready and our plugins are available.
      // Here you can do any higher level native things you might need.
      statusBar.styleDefault();
      splashScreen.hide();
    });
  }
}

```

Declaring out modules such as login, registration and main page. Defining bootsarps and providers:

```
16
17  @NgModule({
18    declarations: [
19      MyApp,
20      LoginPage,
21      RegisterPage,
22      MainPage
23    ],
24    imports: [
25      BrowserModule,
26      IonicModule.forRoot(MyApp),
27      HttpClientModule,
28      HttpModule
29    ],
30    bootstrap: [IonicApp],
31    entryComponents: [
32      MyApp,
33      LoginPage,
34      RegisterPage,
35      MainPage
36    ],
37    providers: [
38      StatusBar,
39      SplashScreen,
40      Camera,
41      HttpClientModule,
42      TextToSpeech,
43      Media,
44      {provide: ErrorHandler, useClass: IonicErrorHandler}
45    ]
46  })
47  export class AppModule {}
```

Here is our index.html page where we are setting application layout:

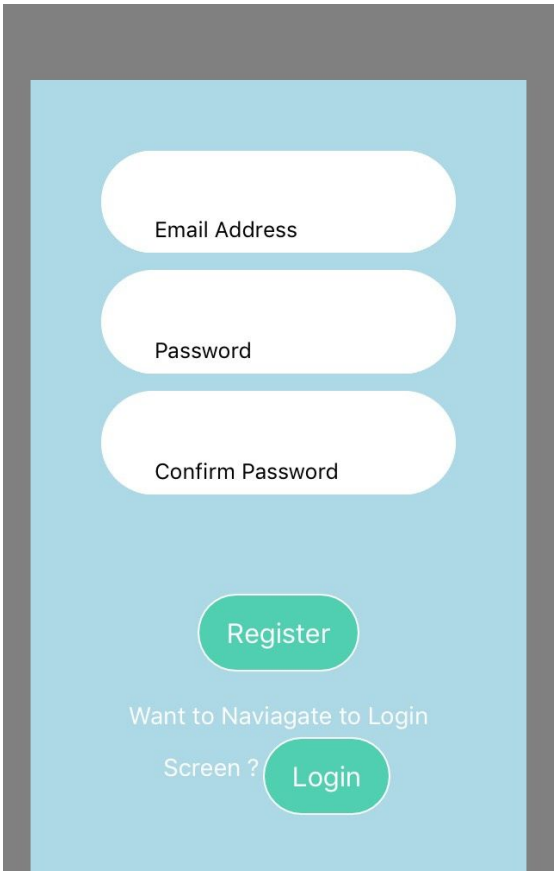
```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
  <meta charset="UTF-8">
  <title>Ionic App</title>
  <meta name="viewport" content="viewport-fit=cover, width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-sca
  <meta name="format-detection" content="telephone=no">
  <meta name="msapplication-tap-highlight" content="no">

  <link rel="icon" type="image/x-icon" href="assets/icon/favicon.ico">
  <link rel="manifest" href="manifest.json">
  <meta name="theme-color" content="#4e8ef7">

  <!-- add to homescreen for ios -->
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">

  <!-- cordova.js required for cordova apps (remove if not needed) -->
  <script src="cordova.js"></script>

  <!-- un-comment this code to enable service worker
  <script>
    if ('serviceWorker' in navigator) {
      navigator.serviceWorker.register('service-worker.js')
        .then(() => console.log('service worker installed'))
        .catch(err => console.error('Error', err));
    }
  </script>-->

  <link href="build/main.css" rel="stylesheet">

</head>
<body>
```

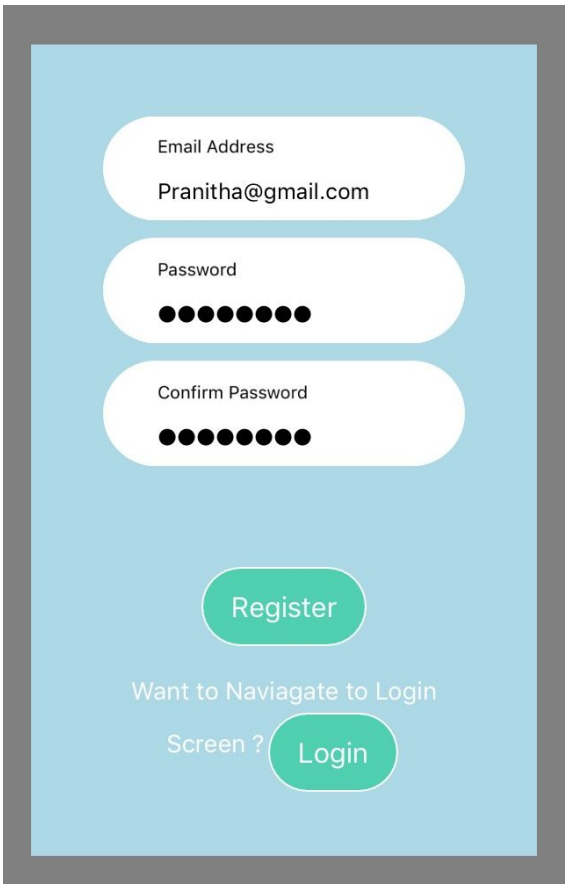
```
30      <link href="build/main.css" rel="stylesheet">
31
32    </head>
33    <body>
34
35      <!-- Ionic's root component and where the app will load -->
36      <ion-app></ion-app>
37
38      <!-- The polyfills js is generated during the build process -->
39      <script src="build/polyfills.js"></script>
40
41      <!-- The vendor js is generated during the build process
42           It contains all of the dependencies in node_modules -->
43      <script src="build/vendor.js"></script>
44
45      <!-- The main bundle js is generated during the build process -->
46      <script src="build/main.js"></script>
47
48    </body>
49    </html>
```

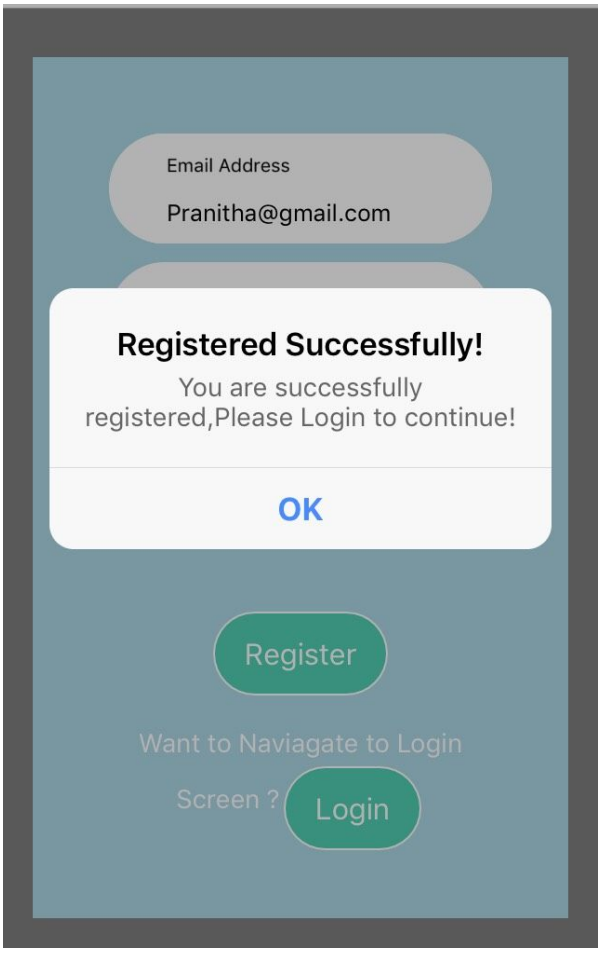
Steps

First page is login page where user is asked to enter their username and password if user does not have an account they can go to registration page and create an account.

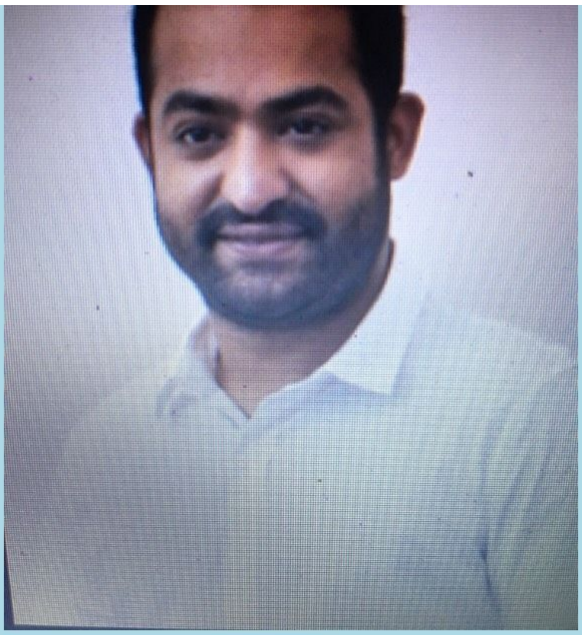
Registration - in order to create an account we need to fill out required information such as: Email address and password. Alert will pop up saying that registration was successful.


Email Address
Password
Confirm Password
Register
Want to Naviagate to Login
Screen ?
Login


Email Address
Pranitha@gmail.com
Password
Confirm Password
Register
Want to Naviagate to Login
Screen ?
Login


Email Address
Pranitha@gmail.com
Registered Successfully!
You are successfully registered,Please Login to continue!
OK
Register
Want to Naviagate to Login
Screen ?
Login

Home page has take a picture button which open camera and takes a picture. Then when Get Analysis button is click application perform Face detection and outputs result to the screen.



Get Face Analysis

Face Detection Details
Age : 33
Gender : Male

Text to Speech

--:--

Logout Take Picture

Logout Take Picture

Picture Taken

Get Face Analysis

Result is also represented by text to speech which will read result to a user.
In order to logout we need to click on Logout button.

APIs

We are using Kairos API to perform Face detection which gets our image send to the server where image runs through Machine Learning model and makes a prediction. Then we get result back in Json format. In order to use Kairos API we need to create account and get application key and ID key so we can use it when making post request. If we get successful respond result gets outputed on the screen.

Unit Testing

Test Cases

Case	Test Case Description	Expected Result	Actual Result	Result
New User Registration	Unique Username and Password	Error Message should pop up, stating that invalid credentials and Re-enter valid credentials.	Error Message should pop up, stating that invalid credentials and Re-enter valid credentials.	Pass
Login, User Authentication	Valid Registered Username and Valid Password	Error Message should pop up, stating that invalid credentials and Re-enter valid credentials.	Error Message should pop up, stating that invalid credentials and Re-enter valid credentials.	Pass
Login, User Authentication	Valid Registered Username and Valid Password	Error Message should pop up, stating that invalid credentials and Re-enter valid credentials.	Error Message should pop up, stating that invalid credentials and Re-enter valid credentials.	Pass
Login Page, Validation	Should not accept blank spaces	Error Message should pop up, stating that please enter valid credentials	Error Message should pop up, stating that please enter valid credentials	Pass

References

1. https://www.kdnuggets.com/2017/02/machine-learning-data-science-apis-updated.html
2. https://www.kairos.com/docs/api/
3. https://ionicframework.com

Contribution

Gulnoza Khakimova - Task 1, Report, Wiki page, Video - 50%
Pranitha Saroj Karumanchi - Task 2, Report, Wiki Page, Video - 50%