

MY WIKIPEDIA SEARCH

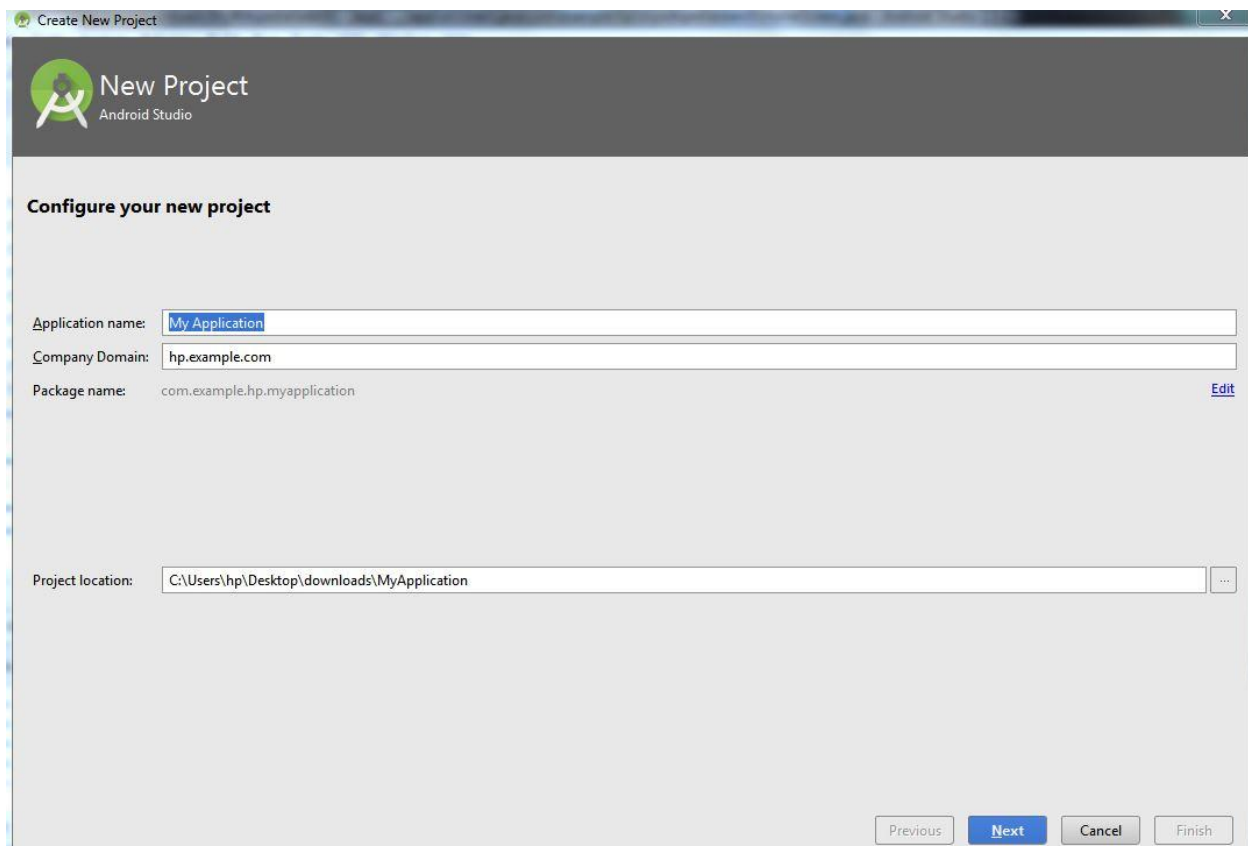
Introduction:

My Wikipedia Search is an android app to display the brief description of the search. It uses Wikipedia API to produce this desired result. On launching the app it asks provides login screen and after successful it redirects to the search screen.

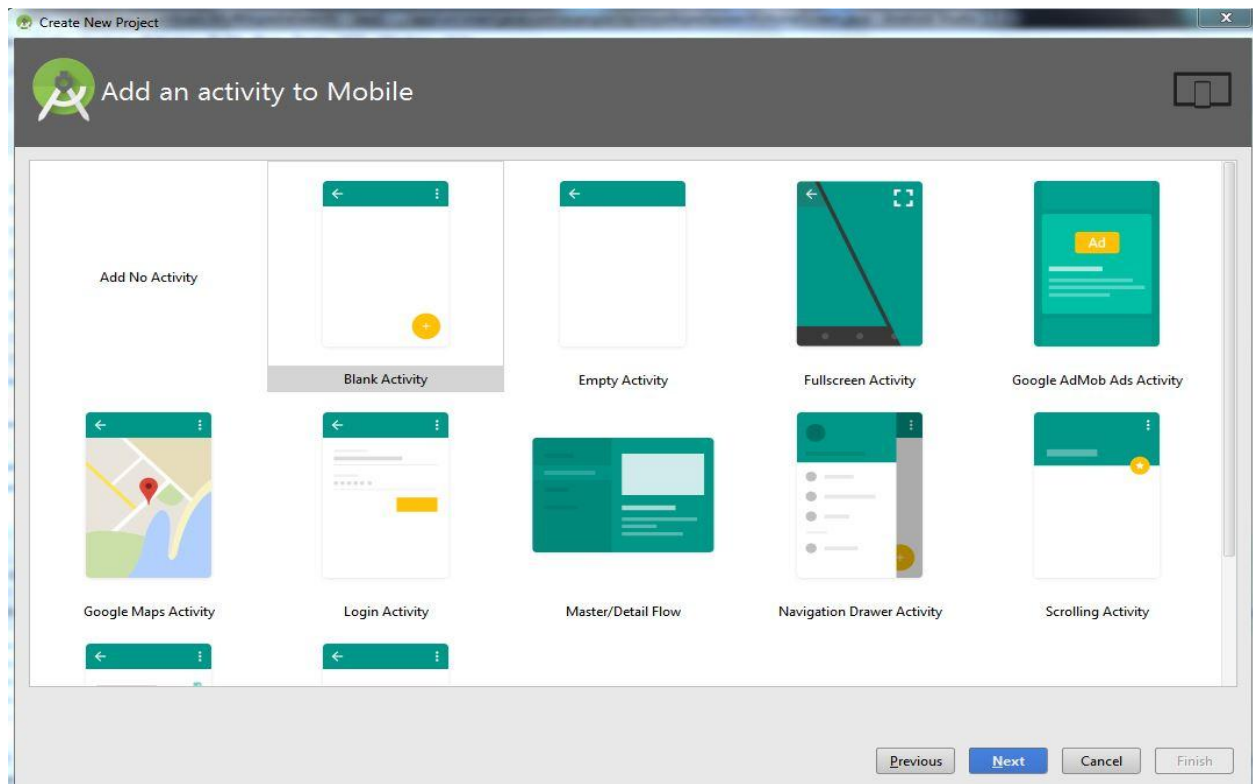
Documentation:

The tool used for the development of this android application is ANDROID STUDIO.

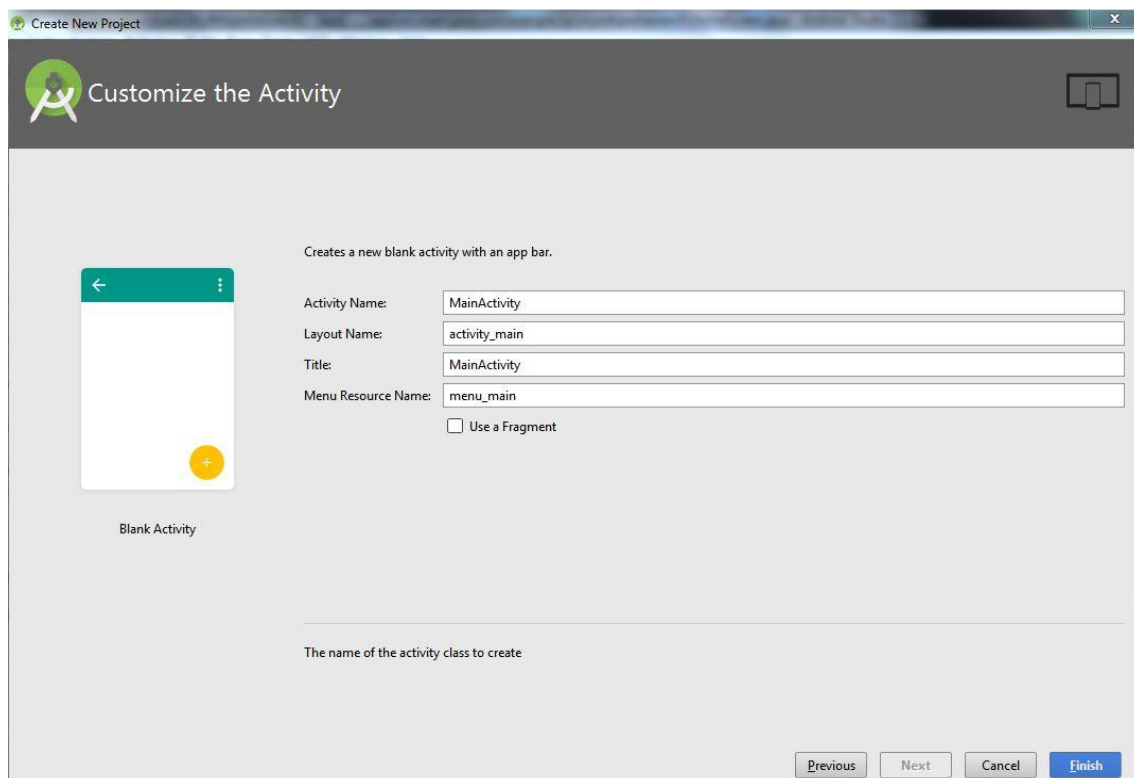
After opening Android Studio, Click on File and select New>New Project. Fill the application Name and choose the Project location.



Click Next and the below screen appears.



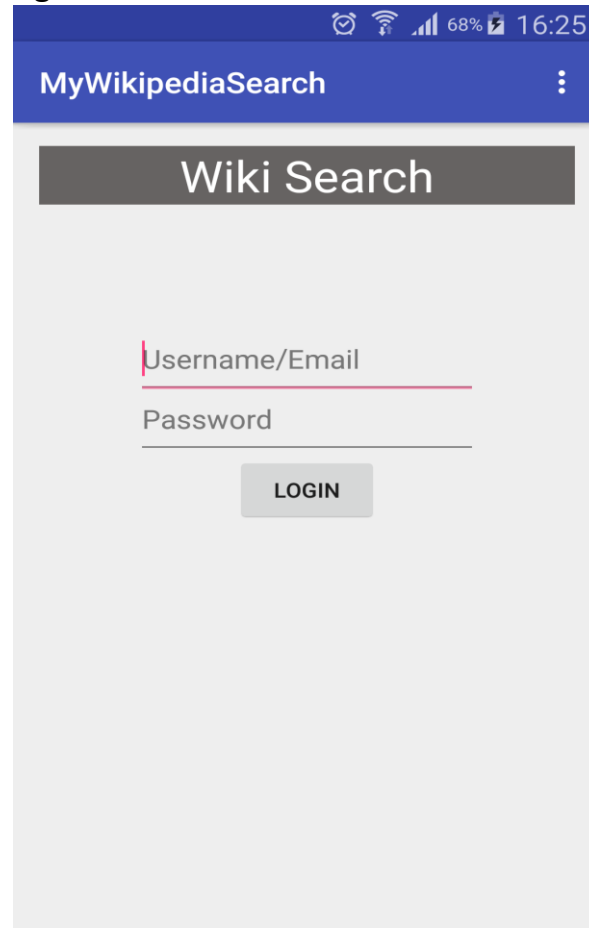
Complete the following details and Click Finish.



Design:

Home Screen:

The following screen on successful launch of the application.



This is HomeScreen(Activity_Home_Screen.xml)

This screen contains

A TextView to display the title.

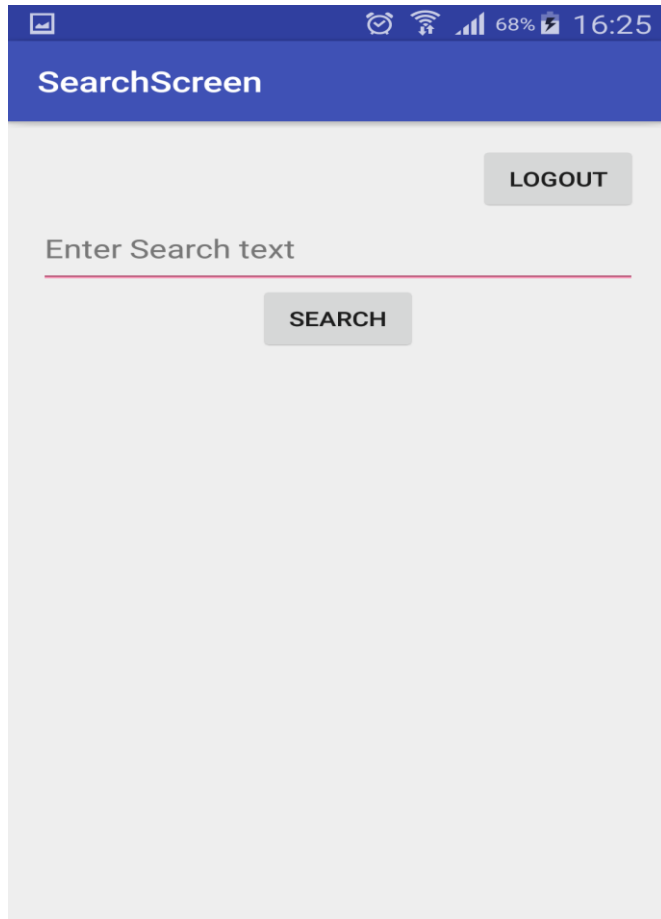
2 EditText widgets for authentication. One for username and the other for password.

One Button for LOGIN.

A textView to show the login failure details.

Search Screen:

After successful login the following screen appears.



This screen contains

A logout button which on clicking revert back to the home screen

An Edit text widget to search.

A Search button on clicking gives the desired Wikipedia result.

Implementation:

The main implementation of the application is done in two Java Files.

1) HomeScreen

```
public class HomeScreen extends AppCompatActivity implements View.OnClickListener {

    Button btlogin;
    EditText etusername;
    EditText etpassword;
    TextView tvloginfail;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home_screen);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        btlogin = (Button) findViewById(R.id.btlogin);
        etusername = (EditText) findViewById(R.id.etusername);
        etpassword = (EditText) findViewById(R.id.etpassword);
        tvloginfail = (TextView) findViewById(R.id.tvloginfail);

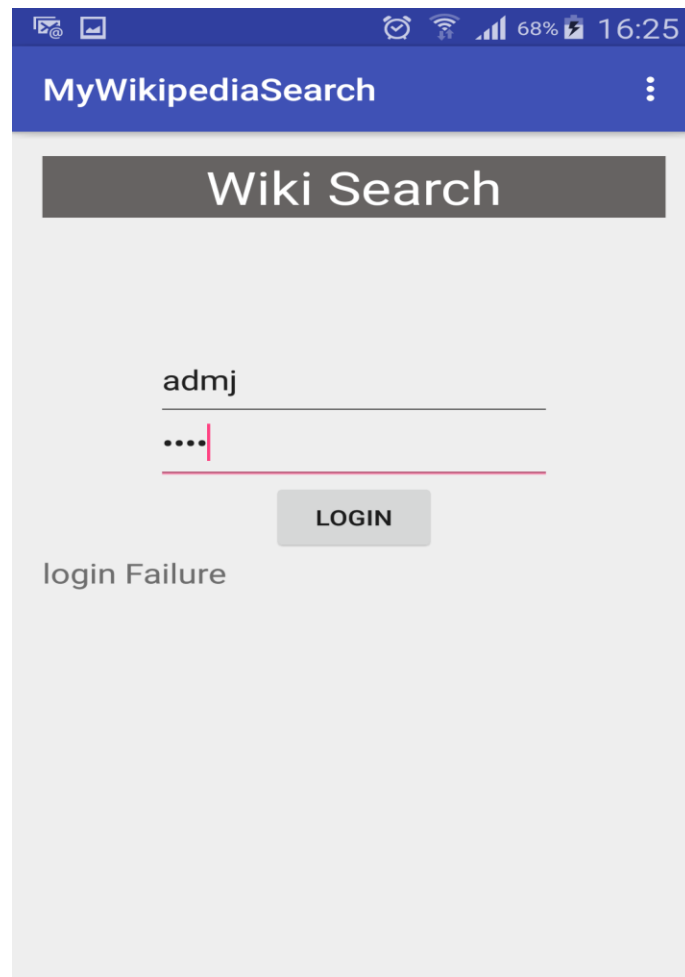
        btlogin.setOnClickListener(this);
    }
}
```

This contains the declaration of all variables(widgets) from Home Screen and link to their .xml counterparts.

```
@Override
public void onClick(View v) {
    String username = etusername.getText().toString();
    String password = etpassword.getText().toString();

    if (!username.isEmpty() && !password.isEmpty()) {
        if (username.equals("admin") && password.equals("admin")) {
            Intent redirect = new Intent(HomeScreen.this, SearchScreen.class);
            startActivity(redirect);
        }
        else{
            tvloginfail.setVisibility(View.VISIBLE);
            tvloginfail.setText("Username or Password must be incorrect");
        }
    }
}
```

The above code validates the username and password. On successful login it redirects to Search Screen. On login failure it displays Failure text in the TextView.



2) Search Screen

```
public class SearchScreen extends AppCompatActivity {

    TextView outputTextView;
    Context mContext;
    EditText etsearchtext;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mContext = getBaseContext();
        setContentView(R.layout.activity_search_screen);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        etsearchtext = (EditText) findViewById(R.id.etsearchtext);
        outputTextView = (TextView) findViewById(R.id.tvResult);
        Button btlogout = (Button) findViewById(R.id.btlogout);
        Button btsearch = (Button) findViewById(R.id.btsearch);

        btlogout.setOnClickListener((v) -> {
            Intent redirect = new Intent(SearchScreen.this, HomeScreen.class);
            startActivity(redirect);
        });

        btsearch.setOnClickListener((v) -> { new RetrieveFeedTask().execute(); });
    }
}
```

The above screenshot contains the class SearchScreen which shows declaration of all variables(widgets) from Search Screen and link to their .xml counterparts.

```
class ExtractWikiData extends AsyncTask<Void, Void, String> {
    String sourceText = etsearchtext.getText().toString();
    private Exception exception;
    protected void onPreExecute() {
        outputTextView.setText("");
    }

    protected String doInBackground(Void... urls) {
        try {
            URL url = new URL("https://en.wikipedia.org/w/api.php?format=json&action=query&prop=extracts" +
                "&exintro=&explaintext=&titles="+sourceText);
            HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
            try {
                BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));
                StringBuilder stringBuilder = new StringBuilder();
                String line;
                while ((line = bufferedReader.readLine()) != null) {
                    stringBuilder.append(line).append("\n");
                }
                bufferedReader.close();
                return stringBuilder.toString();
            } finally {
                urlConnection.disconnect();
            }
        } catch (Exception error) {
            Log.e("ERROR", error.getMessage(), error);
            return null;
        }
    }
}
```

The above screenshot contains the ExtractWikiData class. This class contains the method doInBackground() which uses Wikipedia Web API(URL) to extract data from Wikipedia.

```
protected void onPostExecute(String response) {  
    if (response == null) {  
        response = "Give a text to search";  
    }  
  
    Log.i("INFO", response);  
    runOnUiThread(new Runnable() {  
        @Override  
        public void run() {  
            hideKeyboard(outputTextView);  
        }  
    });  
    try {  
        JSONObject object = new JSONObject(response);  
  
        JSONObject jarray = object.getJSONObject("query");  
        JSONObject jarray2 = jarray.getJSONObject("pages");  
        String extract = jarray2.getString("extract");  
        int i=2;  
        String result ="";  
        while(extract.charAt(i)!='') {  
            result = result+extract.charAt(i);  
            i++;  
        }  
        String newExtract = result;  
        JSONObject jarray3 = jarray2.getJSONObject(newExtract);  
        String finalExtract = jarray3.getString("extract");  
  
        outputTextView.setText(finalExtract);  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
}
```

The above piece of code is to parse and extract the desired data using JSON object.

It also contains the code to hide keyboard on search.

The implementation is described in class HomeScreen and calling is done in ExtractWikiData class.

```
private void hideKeyboard(View editableView) {  
    InputMethodManager keyboard = (InputMethodManager)mContext  
        .getSystemService(Context.INPUT_METHOD_SERVICE);  
    keyboard.hideSoftInputFromWindow(editableView.getWindowToken(), 0);  
}
```


Finally the output of the App is shown below.

