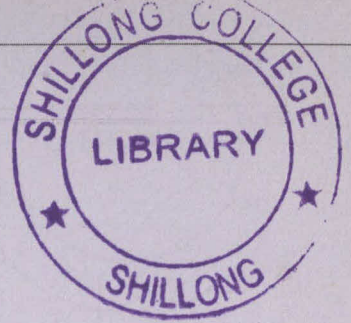




ANDROID CUSTOM BROWSER

*A project report submitted in partial fulfillment of the requirements for the degree
of*

**Bachelor of Computer Applications (BCA)
OF
North Eastern Hill University (NEHU)
2017**



Submitted by
Name: VINOD SHARMA
Roll No.: P1500079
Reg. No.: 12353 of 2014-15



**DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS
SHILLONG COLLEGE
SHILLONG 793003
MEGHALAYA.**



CERTIFICATE

This is to certify that the Project work titled Android Web Browser is a bonafide work done by VINOD SHARMA University Roll No.: **P1500079**, Registration No.: 12353 of **2014-15**, under my guidance during the final year of the course.

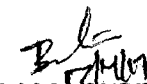
~~Sir.~~ 
(Donald Jefferson Thabab)

The Project Seminar was held on: _____ at Shillong College Shillong.


17/4/17

Smt. Airom Mitri

HOD


External examiner

Dept. Of Comp. Sci.&App.

Shillong College

Shillong, Meghalaya.

793003

ACKNOWLEDGMENT

I thank the almighty God for the blessings he bestowed upon me.

I would like to express my gratitude to all those who provided me the possibility to complete this Project. A special gratitude to sir. Donald Jefferson Thabah for his guidance. I also wish to express my gratitude to Smt. Aiom Mitri, HOD and other faculty members of Department Of Computer Science and Applications, Shillong College, who rendered their help during the period of my project work.

I would also take the opportunity to thank Dr. K.D Ramsej, Principal, Shillong College for Providing me the opportunity to work on this project.

Finally, I would like to thank my parents and friends who helped me a lot in finalizing this Project within the limited time frame



CONTENTS

1.ABSTRACT

2.FEATURES OF SIMPLE BROWSER

3.COMPONENTS AND WIDGETS

4.REQUIREMENTS

5.TEMPLATE(THE STEPS FOR MAKING THIS APPLICATION)

(a) Activity Main

(b) Manifest XML

(c) Main Activity JAVA

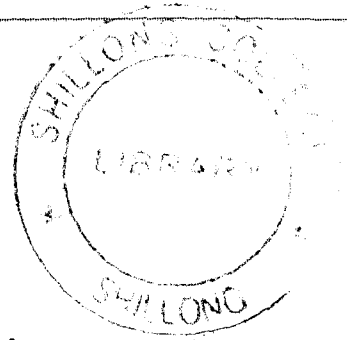
(d) SplashScreen JAVA

(e) SplashScreen Layout

(f) Execute the Application

6. Conclusion

ABSTRACT



Web browsing Android apps are great if We have a nice website and we want to make our brand's application with just a minimal amount of code. At the same time, it is a great way to learn more about Android application development, and it doesn't require a background in application development.

So here we are going to create our own simple web browser which will have a simple layout with the SplashScreen and Menu Settings.

Features

Smart address bar that autocompletes search results and browsing history as you type, as well as suggestions if you've made a typo or error.

Form auto-fill that can populate email addresses and other information without you having to type it all in manually

Customizable text scaling for accessibility purposes, and the option to override page instructions to disable zoom

we have added menu bar also so that anytime we can go back, forward and home screen.

COMPONENTS AND WIDGETS:

1. EditText:

EditText is used to get URL from user like google.com, android-examples.com etc.

2. Button:

Setting up on click listener function on button so each

& every time after clicking on button it will get EditText value and set that URL value into webview.

3. WebView :

WebView is used to display websites.

4. Featured Buttons:

- (a) Go
- (b) Next
- (c) Back
- (d) Refresh
- (e) Exit

Requirements:

We need to have the following installed and configured on our development machine:

- **Android SDK and platform tools**
- **ANDROID STUDIO 2.2.3**
- **An Emulator or Android device running Android 2.3 or higher**

TEMPLATE

(BUILDING STEPS FOR CREATING OUR OWN SIMPLE BROWSER)

1. START ANDROID STUDIO AND TYPE THE PROJECT NAME

 **New Project**
Android Studio

Configure your new project

Application name:

Company Domain:

Package name: [Edit](#)

☐ Include C++ support

Project location: ...

[Previous](#) [Next](#) [Cancel](#) [Finish](#)

Now Choose API AND SDK

Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK

API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available.
By targeting API 15 and later, your app will run on approximately 98.3% of the devices that are active on the Google Play Store.
[Help me choose](#)

Stats load failed. Value may be out of date.

☐ Wear

Minimum SDK

API 21: Android 5.0 (Lollipop)

☐ TV

Minimum SDK

API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass (Not Available)

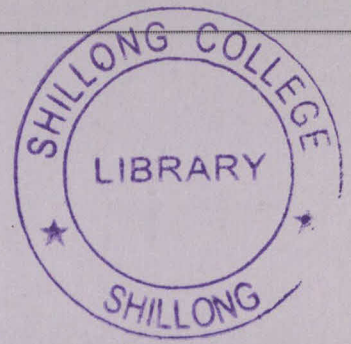
Minimum SDK

Previous

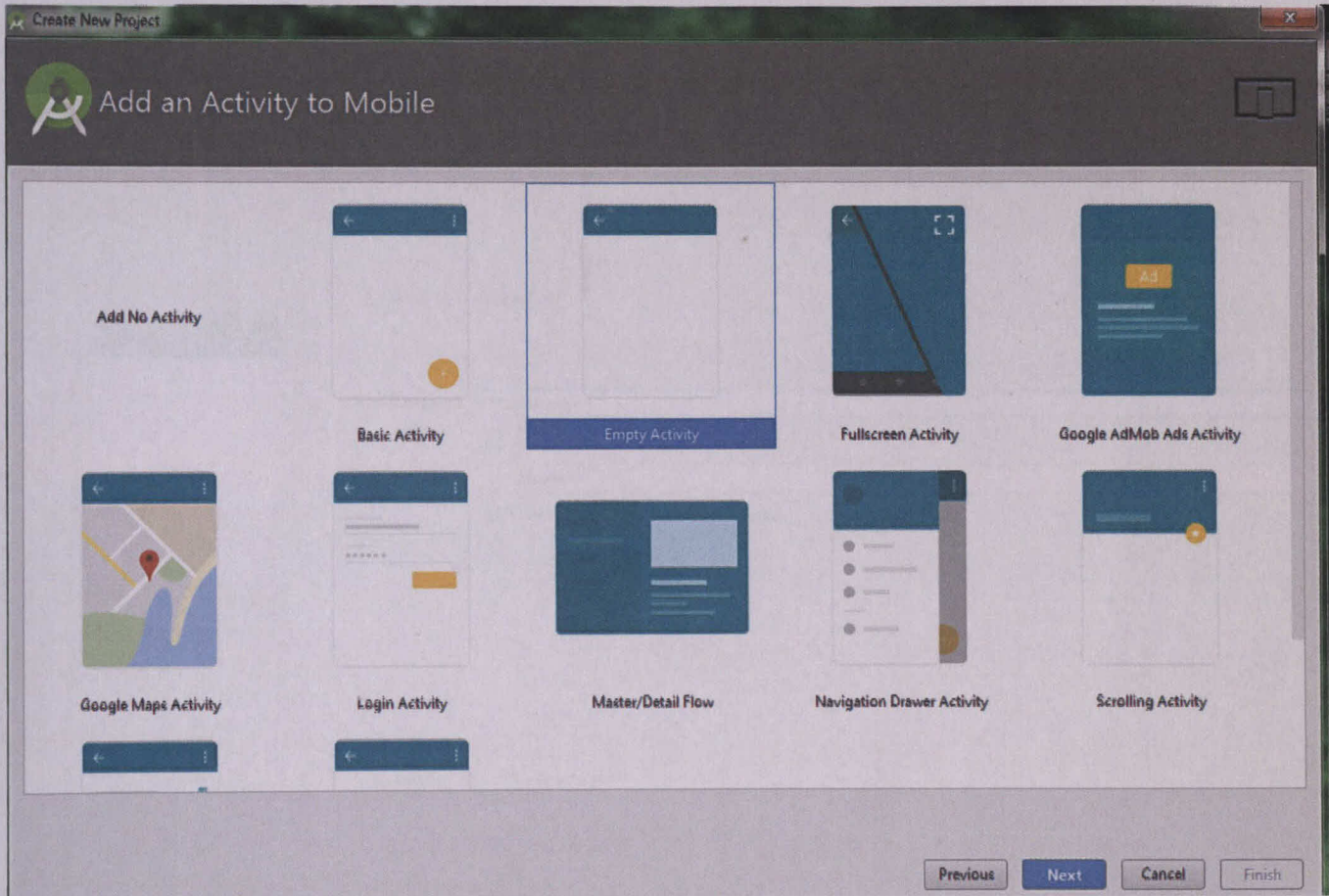
Next

Cancel

Finish



Now choose Empty Activity



And click finish

Create New Project

 Customize the Activity

Creates a new empty activity



Empty Activity

Activity Name:

☒ Generate Layout File

Layout Name:

☒ Backwards Compatibility (AppCompat)

The name of the activity class to create

2. Now we have to edit the activity_main layout.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="43dp"
        android:orientation="vertical" >

    </LinearLayout>

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="12"
        android:hint="PLEASE TYPE URL" />

    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button4"
        android:layout_alignBottom="@+id/button4"
        android:layout_alignParentRight="true"
        android:textSize="10sp"
        android:text="Exit" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Next"
        android:textSize="10sp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
    android:text="Back"
    android:textSize="10sp"
    android:layout_alignParentBottom="true"
    android:layout_alignRight="@+id/linearLayout1"
    android:layout_alignEnd="@+id/linearLayout1" />
```

<Button

```
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Refresh"
    android:textSize="10sp"
    android:layout_alignParentBottom="true"
    android:layout_toRightOf="@+id/button2"
    android:layout_toEndOf="@+id/button2" />
```

<Button

```
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="GO"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
```

<WebView

```
    android:id="@+id/webView1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/editText1"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

</RelativeLayout>

3. Now we have to edit the manifest xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.personal.webbrowser">

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="24" />
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE">

    </uses-permission>

    <application
        android:allowBackup="true"
        android:icon="@drawable/coool"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity
            android:name=".SplashScreen"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MainActivity"></activity>
    </application>

</manifest>
```

4. step 4 is to edit the MainActivity.java code

```
package com.example.personal.webbrowser;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MotionEvent;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends Activity {
    WebView wv;
    TextView tv;
    Button btn1, btn2, btn3, btn4, btn5;
    EditText et;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btn1 = (Button) findViewById(R.id.button1);

        et = (EditText) findViewById(R.id.editText1);

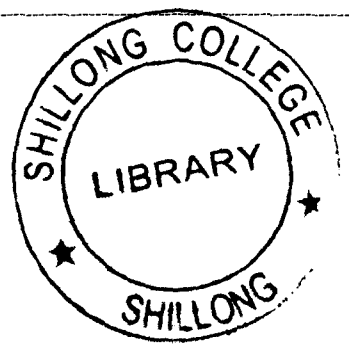
        wv = (WebView) findViewById(R.id.webView1);
        wv.loadUrl("file:///android_asset/web.html");
        wv.setWebViewClient(new WebViewClient());

        btn1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                String url = et.getText().toString();
                wv.loadUrl("http://" + url);
            }
        });
        tv = (TextView) findViewById(R.id.editText1);

        btn5 = (Button) findViewById(R.id.button5);
        btn5.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
```



```
// TODO Auto-generated method stub
finish();
    }
});

btn4 = (Button) findViewById(R.id.button4);
btn4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        wv.reload();
    }
});

btn3 = (Button) findViewById(R.id.button3);
btn3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        wv.goForward();
    }
});

btn2 = (Button) findViewById(R.id.button2);
btn2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        wv.goBack();
    }
});

wv.requestFocus(View.FOCUS_DOWN);
wv.setOnTouchListener(new View.OnTouchListener() {

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        // TODO Auto-generated method stub
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
            case MotionEvent.ACTION_UP:
                if (!v.hasFocus()) {
                    v.requestFocus();
                }
                break;
        }
        return false;
    }
});

}
```

```
class webCont extends WebViewClient {  
    @Override  
    public boolean shouldOverrideUrlLoading(WebView view, String  
url) {  
        // TODO Auto-generated method stub  
        view.loadUrl(url);  
  
        return true;  
    }  
}  
}
```

5. Now its time for adding SplashScreen. For that we need to create empty activity for java and edit the following codes. Here we are naming our activity SplashScreen Activity.

```
package com.example.personal.webbrowser;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class SplashScreen extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        Thread myThread =new Thread() {
            @Override
            public void run() {
                try {
                    sleep(5000);
                    Intent intent = new
                        Intent(getApplicationContext(),
MainActivity. class);
                    startActivity(intent);
                    finish();
                }
                catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        };
        myThread.start();
    }
}
```

6. Now we have to edit the splash screen layout.

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_splash_screen"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    tools:context="com.example.personal.webbrowser.SplashScreen"
    android:weightSum="1">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/image"
        android:id="@+id/imageView"/>

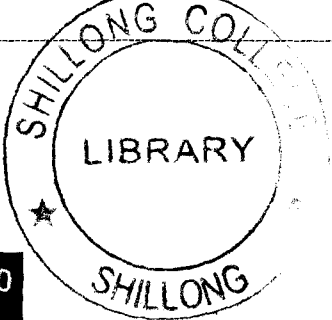
</LinearLayout>
```

7. Step 7 is the final step to excute our newly created android browser.

First screen (SplashScreen)

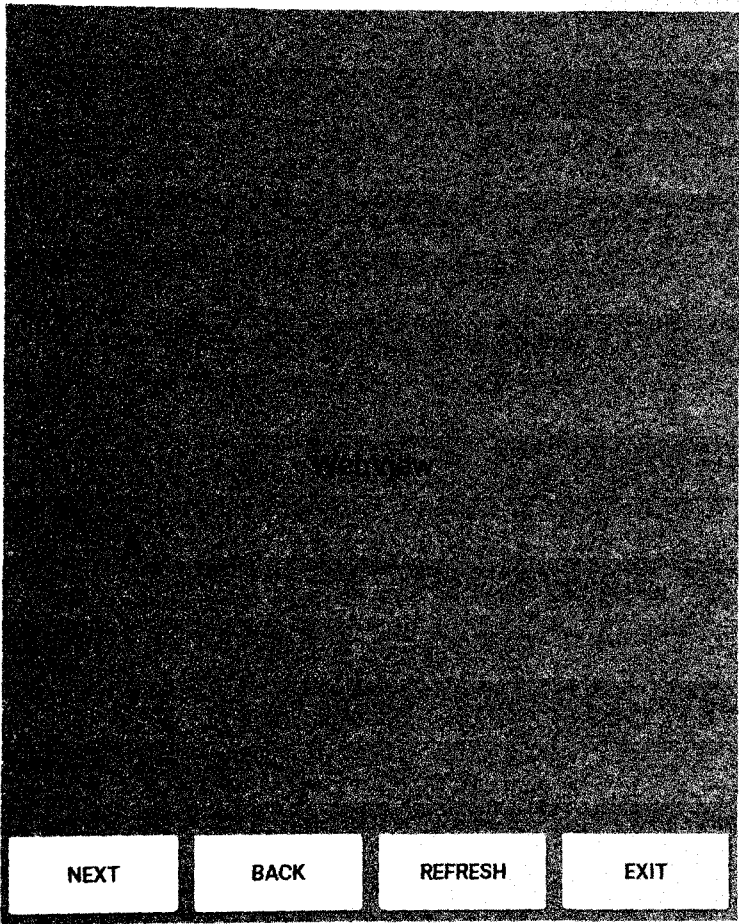


Second Screen we will see is:



PLEASE TYPE URL

GO



And now its time to type the URL for example (google.co.in) and the page finally loads and it appears like this along with the featured buttons.



CONCLUSION

Android web Browser are used to surf websites on android phones. With the help of web browser we can visit web site, download content from other websites and do much more. The primary purpose of a web browser is to bring information resource to the user. An information resource is identified by a Uniform Resource Identifier and may be a web page, image, video or other piece of content. Hyperlinks present in resources enable users to easily navigate their browsers to related resources. Although browser are primarily intended to access the world wide web, they can also be used to access information provided by Web Servers in private Networks or file Systems.

REFERENCES:

1. Android Apps For Absolute Beginners by Wallace Jackson
2. Google Search engine for the purpose of gathering extra information
3. Wikipedia for detail information.
3. Java The Complete Reference by Herbert Schildt

