MINI PROJECT

NAME : HARSHIT AGARWAL
REG. NO. : 18BCE0647
SLOT : G1
FACULTY : PROF. JAISANKAR N

TOPIC : JAVA BASED TEXT EDITOR

PROJECT ABSTRACT :

A text editor is a type of computer program that edits plain text. Such programs are sometimes known as "notepad" software, following the naming of Microsoft Notepad. Text editors are provided with operating systems and software development packages, and can be used to change files such as configuration files, documentation files and programming language source code.

In this project, I have created a Text Editor called "MyEditor" with the Java Programming Language. MyEditor can be used to edit and modify any sorts of files. The speciality of MyEditor is that it can be used to write, compile and run Java Code too. It provides features for creating, opening and saving files. We can :

- **modify and edit files**
- **Find and replace sequences of strings**
- **can read live website url html code**
- **Modify multiple files at once (I.e.provides multiple tabs)**
- **Show/hide the output dialog box**
- **write English translated to Hindi**
- **Write, compile and run Java code**

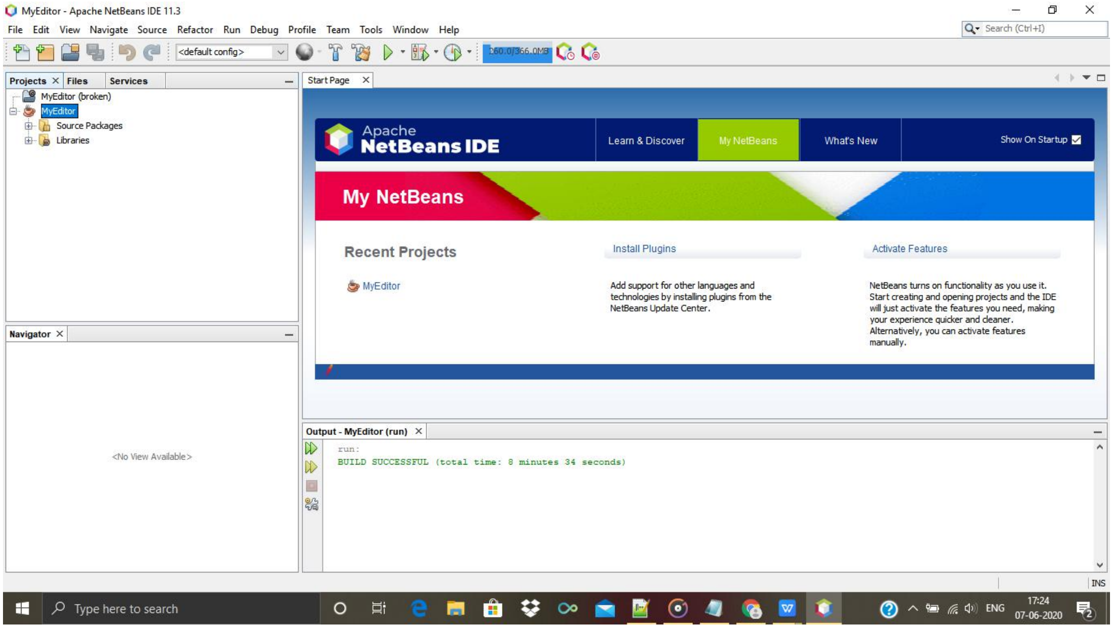
LINK FOR THE PROJECT :

https://drive.google.com/drive/folders/1uv6QasR6RM9_vdx_p_RPj5uzpfsi2nam?usp=sharing

HOW TO OPEN AND RUN THE PROJECT :

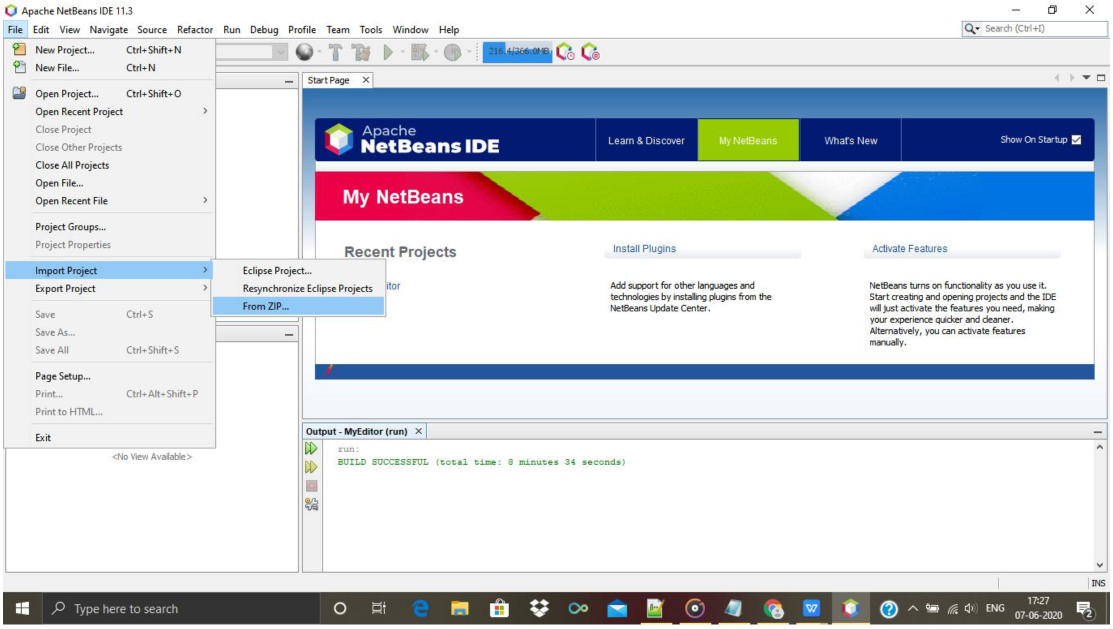
1. Download the file "18BCE0647 JAVA MINI PROJECT.zip" present in this folder.

2. Open Apache NetBeans IDE .


MyEditor - Apache NetBeans IDE 11.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects Files Services
MyEditor (broken)
MyEditor
Source Packages
Libraries
Navigator
<No View Available>
Start Page
Apache NetBeans IDE
Learn & Discover
My NetBeans
What's New
Show On Startup
My NetBeans
Recent Projects
MyEditor
Install Plugins
Add support for other languages and technologies by installing plugins from the NetBeans Update Center.
Activate Features
NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.
Output - MyEditor (run)
INS
Type here to search
ENG
17:24
07-06-2020

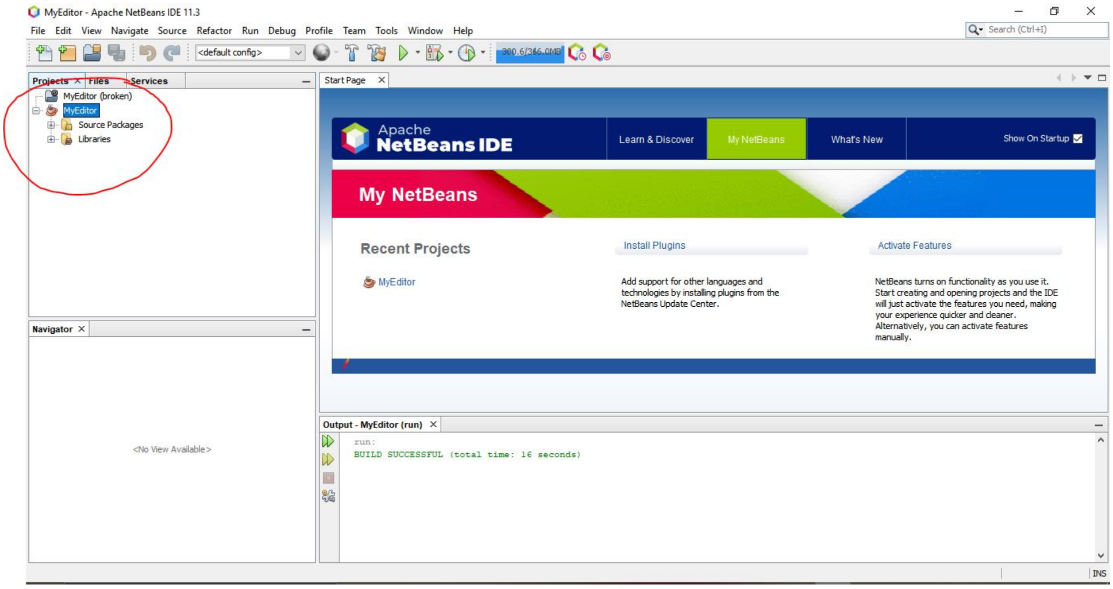
```
run:
BUILD SUCCESSFUL (total time: 8 minutes 34 seconds)
```

3. Click on File and then click on Import Project.
4. Select From ZIP ….
5. The screen would look like this :



```
run:
BUILD SUCCESSFUL (total time: 8 minutes 34 seconds)
```

6. Choose the downloaded .zip file from the device.
7. Select MyEditor from Projects pane.

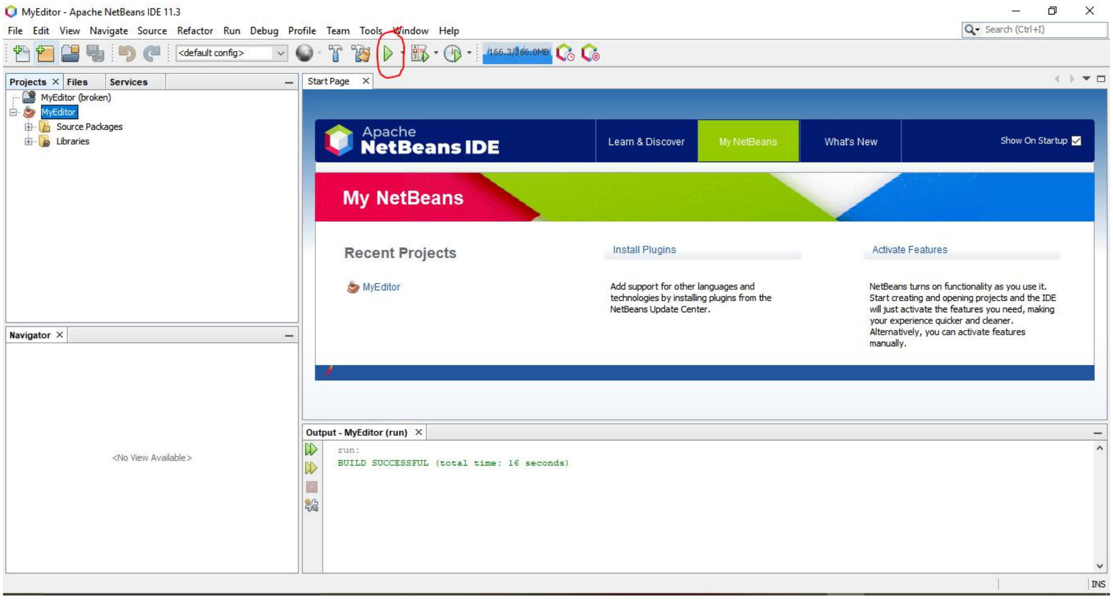


Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.

```
run:
BUILD SUCCESSFUL (total time: 16 seconds)
```

8. Finally click on the Run Button.

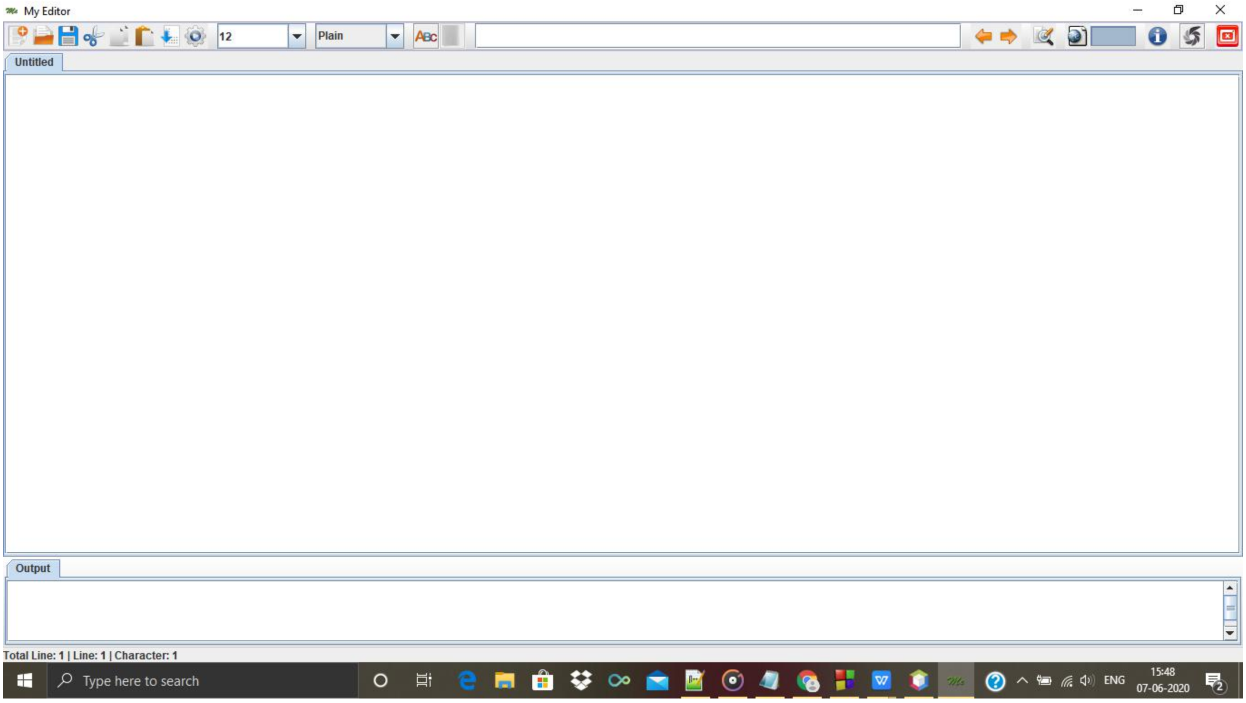


Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.

```
run:
BUILD SUCCESSFUL (total time: 16 seconds)
```

9. This screen shall appear.



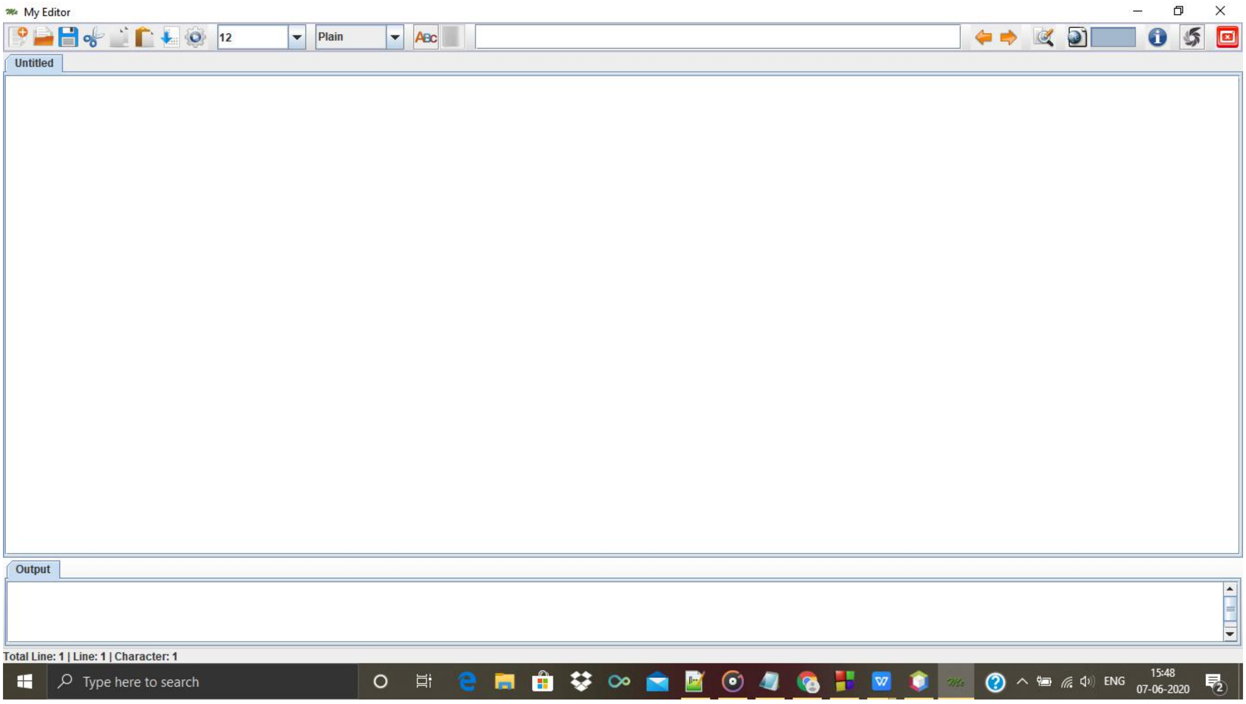
12

Plain

Total Line: 1 | Line: 1 | Character: 1

10. The user can also refer to the README FILE.docx in the folder that opens through the link mentioned above.

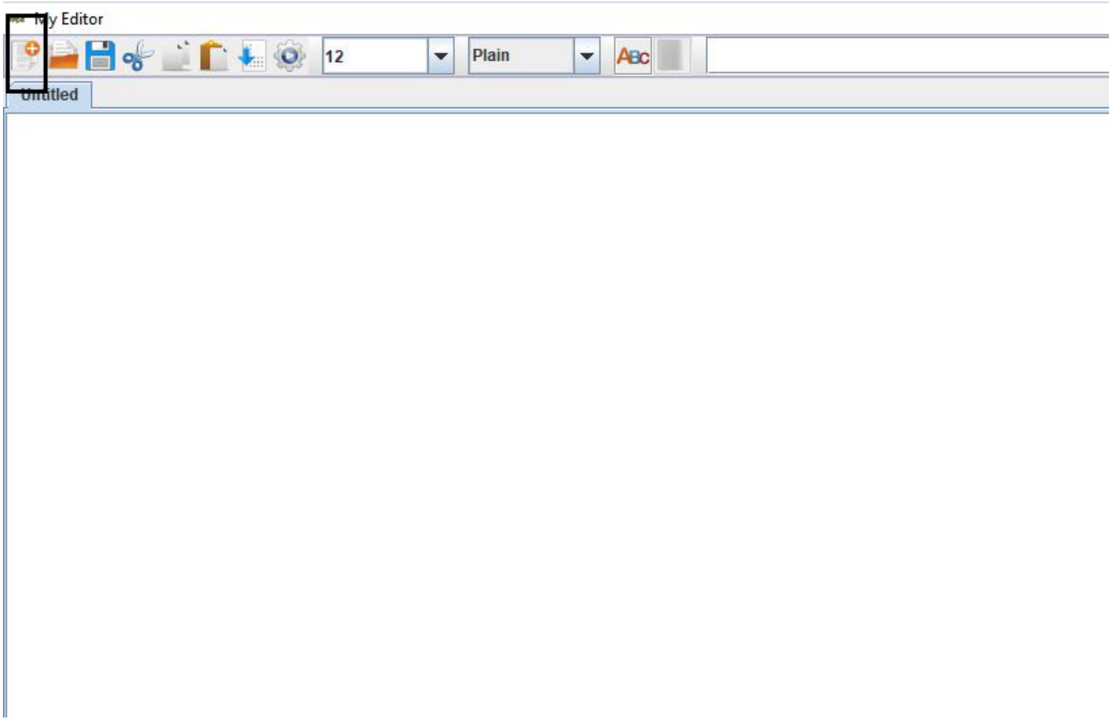
MAIN SCREEN :



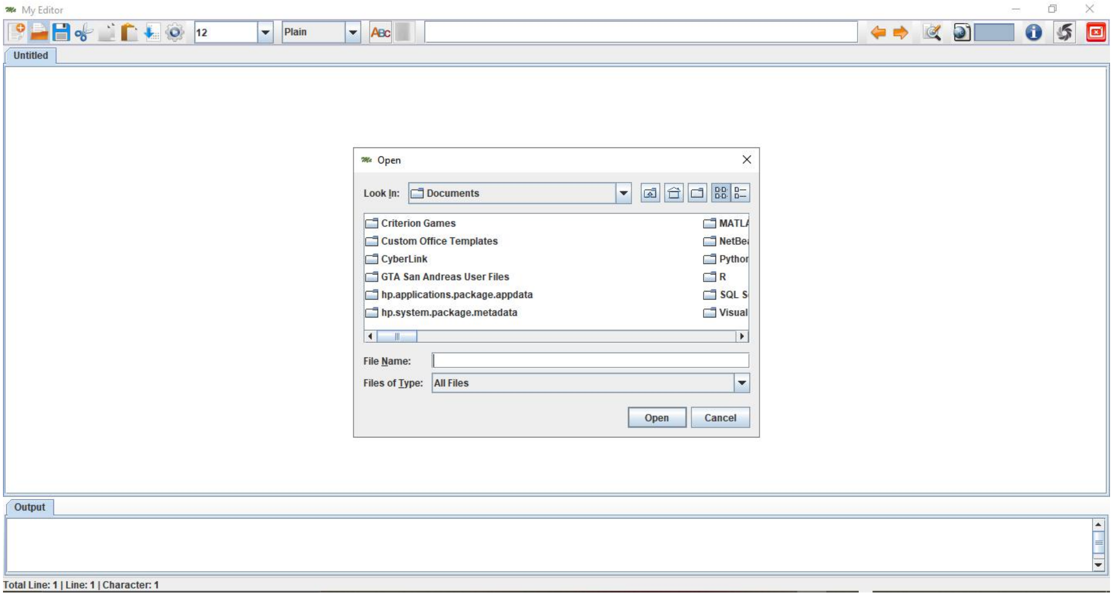
Total Line: 1 | Line: 1 | Character: 1

THE VARIOUS COMPONENTS AND THEIR FEATURES :

1. NEW FILE : Used to create a new file.



2. OPEN FILES : Used to open existing files and lets the user browse the files in the system.



12

Plain

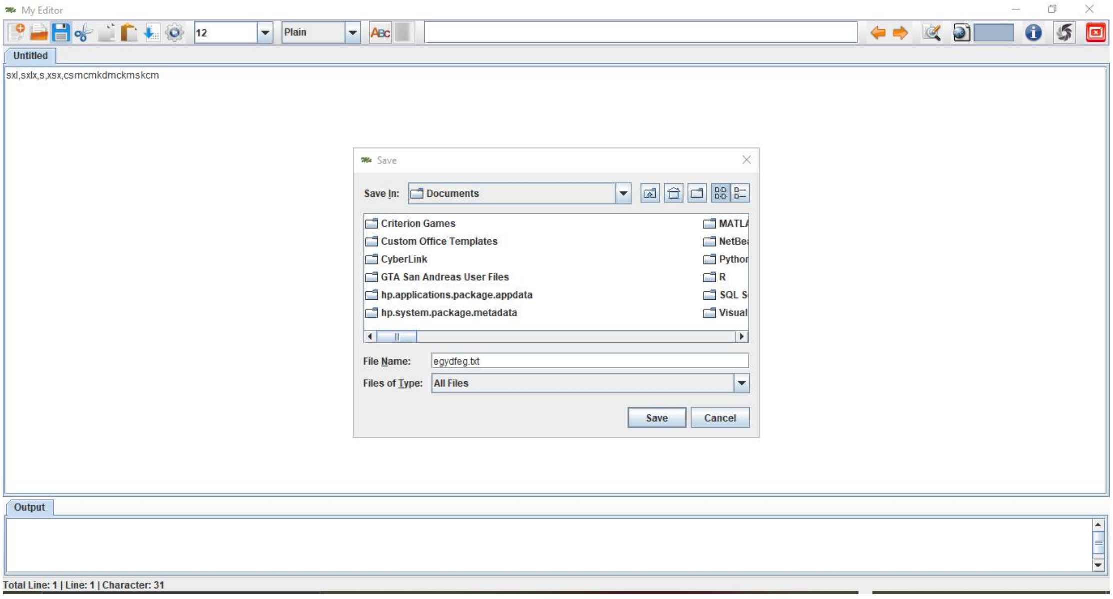
Look In: Documents

File Name:

Files of Type: All Files

Total Line: 1 | Line: 1 | Character: 1

3. **SAVE FILE :** Used to save files.



12

Plain

sxl,sxlx,s,xsx,csmcmkdmckmskcm

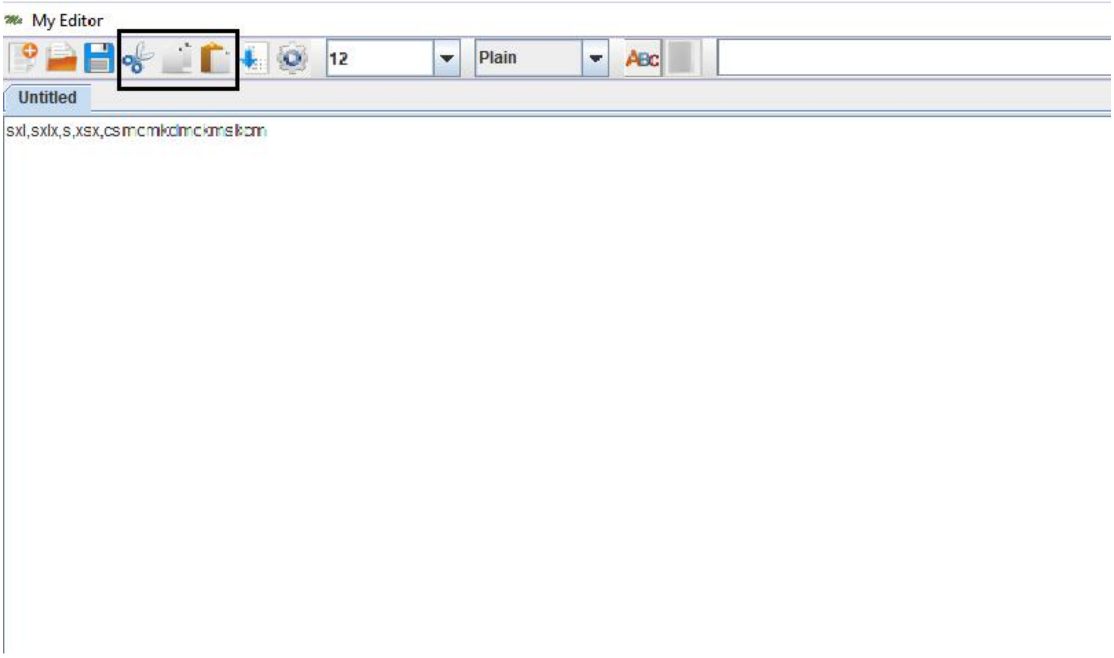
Save In: Documents

File Name: egydfeg.txt

Files of Type: All Files

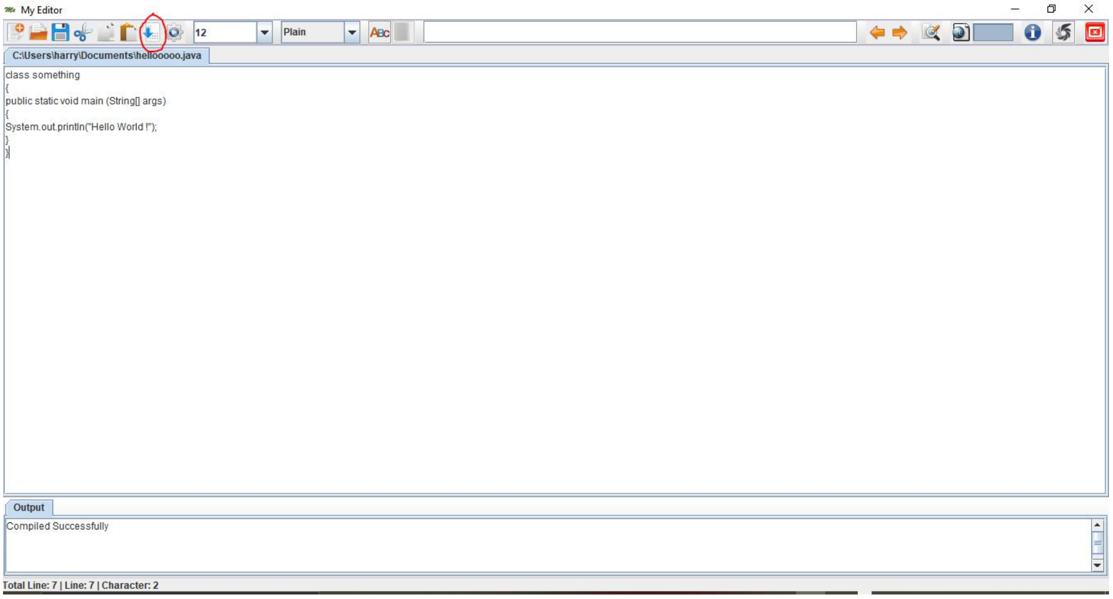
Total Line: 1 | Line: 1 | Character: 31

4. CUT, COPY AND PASTE : Used to edit text and apply operations such as Cut, Copy and Paste on the text.



sxl,sxlx,s,xsx,csmcmkdmckmslkcm

5. **COMPILE BUTTON :** Used to compile Java code.

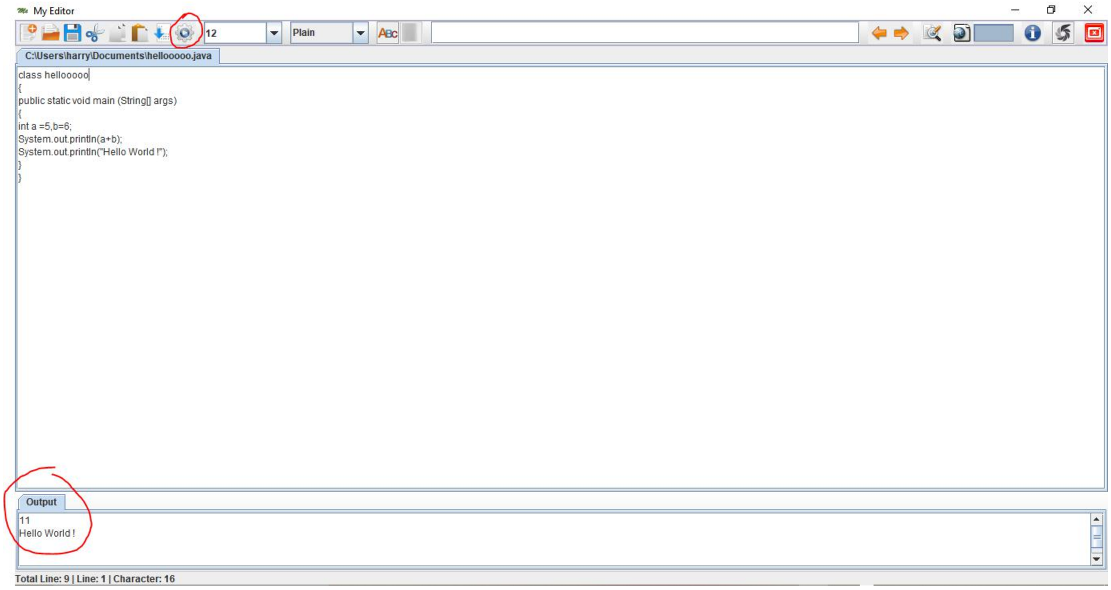


```
class something
{
public static void main (String[] args)
{
System.out.println("Hello World !");
}
}
```

```
Compiled Successfully
```

Total Line: 7 | Line: 7 | Character: 2

6. **RUN BUTTON :** To run the successfully compiled Java Code.

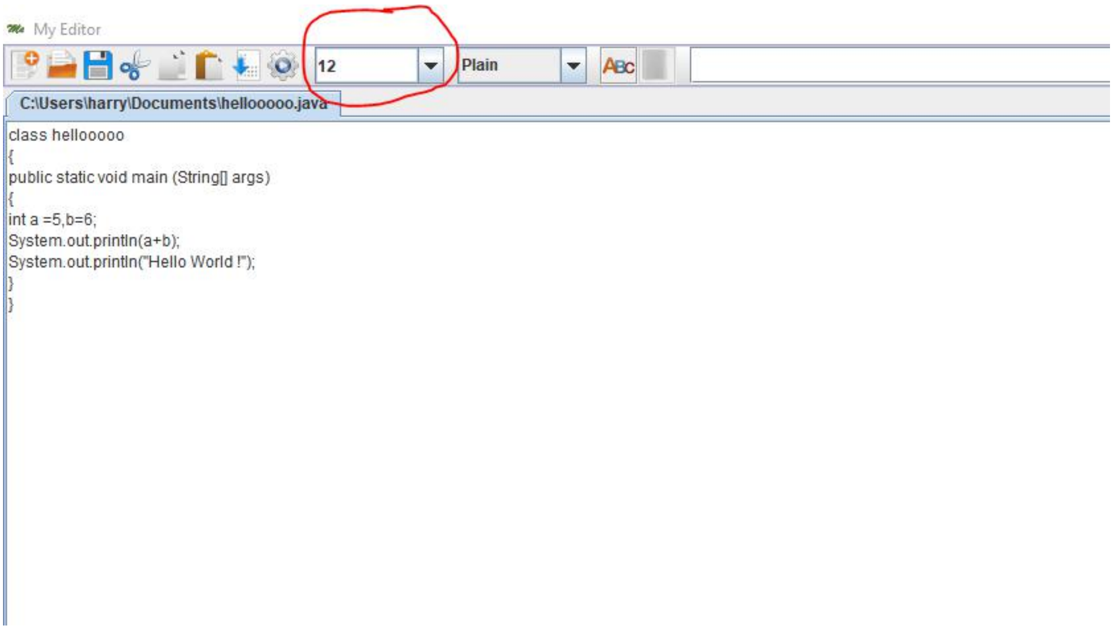


```
class hellooooo
{
public static void main (String[] args)
{
int a =5,b=6;
System.out.println(a+b);
System.out.println("Hello World !");
}
}
```

```
11
Hello World !
```

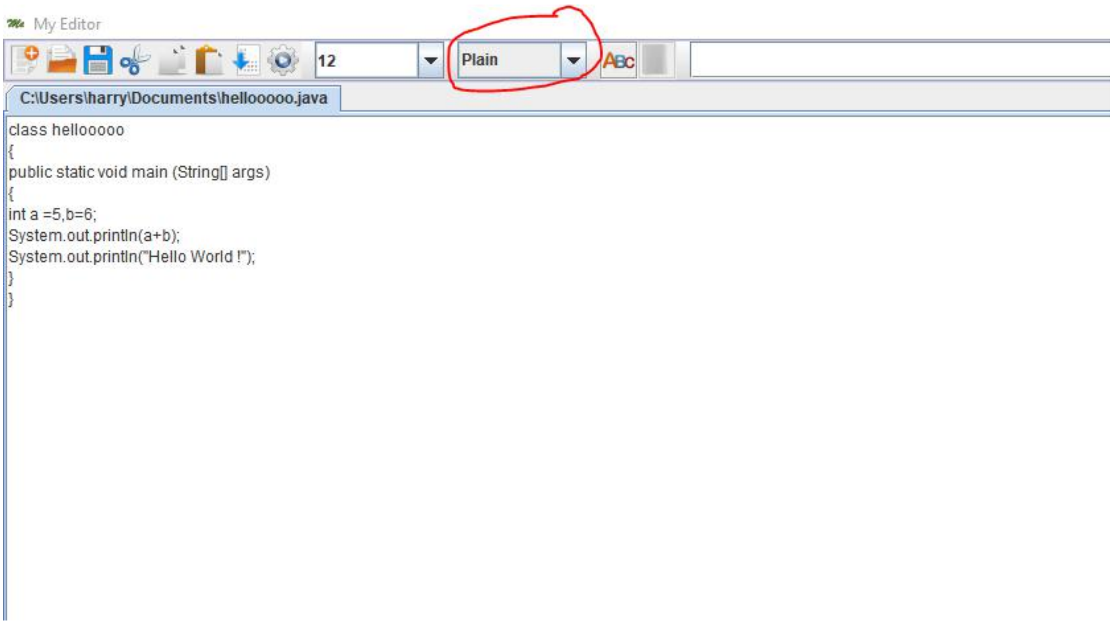
Total Line: 9 | Line: 1 | Character: 16

7. **FONT SIZE :** Used to select Font Size.



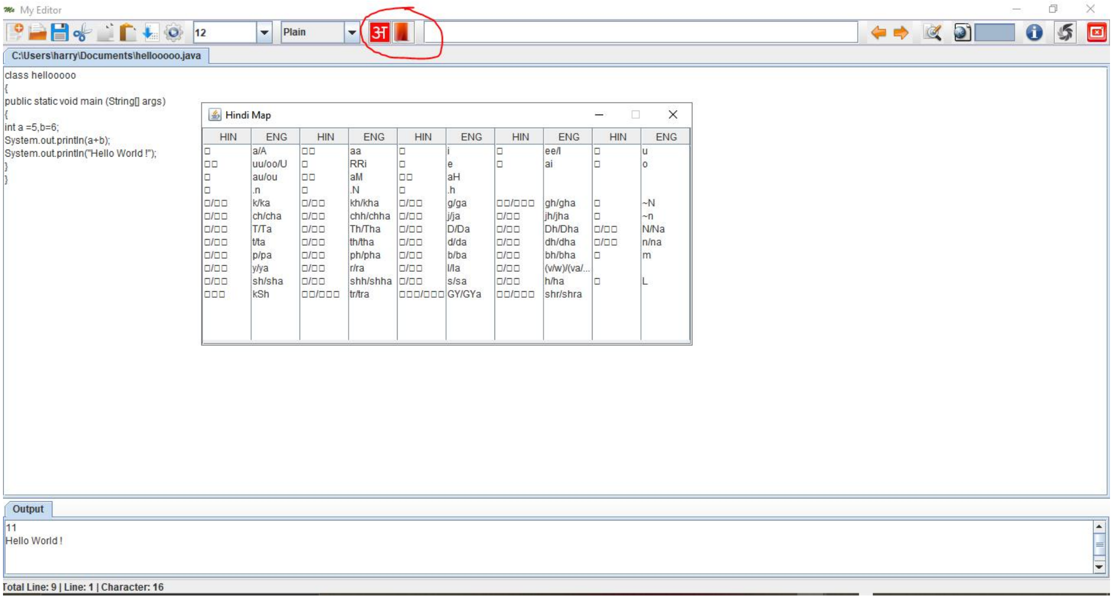
```
class hellooooo
{
public static void main (String[] args)
{
int a =5,b=6;
System.out.println(a+b);
System.out.println("Hello World !");
}
}
```

8. **FONT STYLE :** Used to select font style such as Plain, Bold or Italic.



```
class hellooooo
{
public static void main (String[] args)
{
int a =5,b=6;
System.out.println(a+b);
System.out.println("Hello World !");
}
}
```

9. **HINDI TRANSLATOR :** Used to type in Hindi.



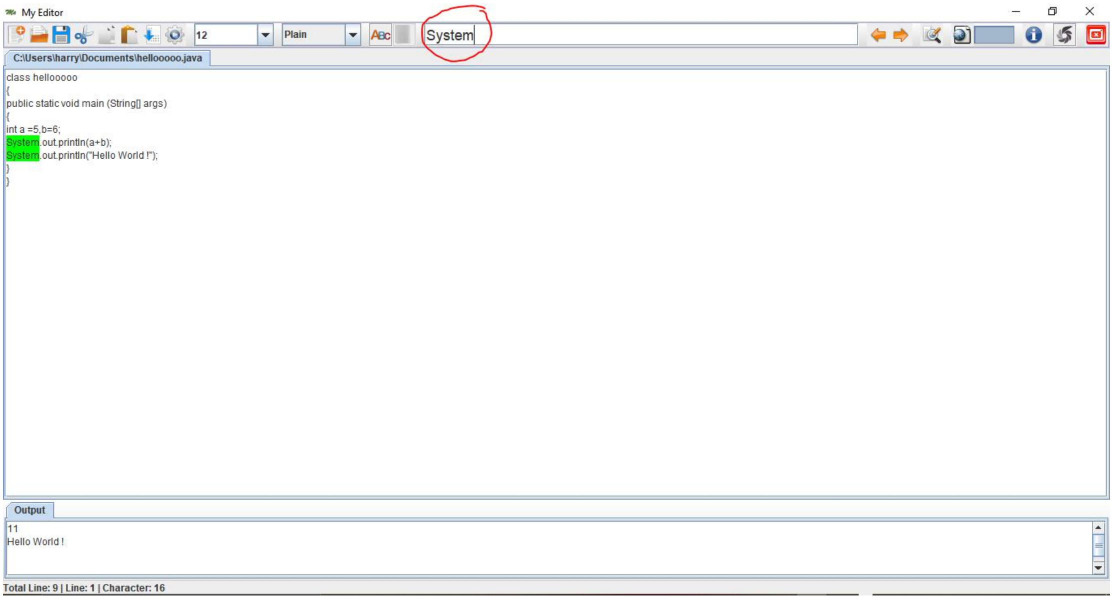
```
class hellooooo
{
public static void main (String[] args)
{
int a =5,b=6;
System.out.println(a+b);
System.out.println("Hello World !");
}
}
```

HIN	ENG	HIN	ENG	HIN	ENG	HIN	ENG	HIN	ENG
□	a/A	□□	aa	□	i	□	ee/I	□	u
□□	uu/oo/U	□	RRi	□	e	□	ai	□	o
□	au/ou	□□	aM	□□	aH				
□	.n	□	.N	□	.h				
□/□□	k/ka	□/□□	kh/kha	□/□□	g/ga	□□/□□□	gh/gha	□	~N
□/□□	ch/cha	□/□□	chh/chha	□/□□	j/ja	□/□□	jh/jha	□	~n
□/□□	T/Ta	□/□□	Th/Tha	□/□□	D/Da	□/□□	Dh/Dha	□/□□	N/Na
□/□□	t/ta	□/□□	th/tha	□/□□	d/da	□/□□	dh/dha	□/□□	n/na
□/□□	p/pa	□/□□	ph/pha	□/□□	b/ba	□/□□	bh/bha	□	m
□/□□	y/ya	□/□□	r/ra	□/□□	l/la	□/□□	(v/w)/(va/...		
□/□□	sh/sha	□/□□	shh/shha	□/□□	s/sa	□/□□	h/ha	□	L
□□□	kSh	□□/□□□	tr/tra	□□□/□□□	GY/GYa	□□/□□□	shr/shra		

```
11
Hello World !
```

Total Line: 9 | Line: 1 | Character: 16

10. **FIND TEXT FIELD :** Used to find text within the document.



C:\Users\harry\Documents\hellooooo.java

```
class hellooooo
{
public static void main (String[] args)
{
int a =5,b=6;
System.out.println(a+b);
System.out.println("Hello World !");
}
}
```

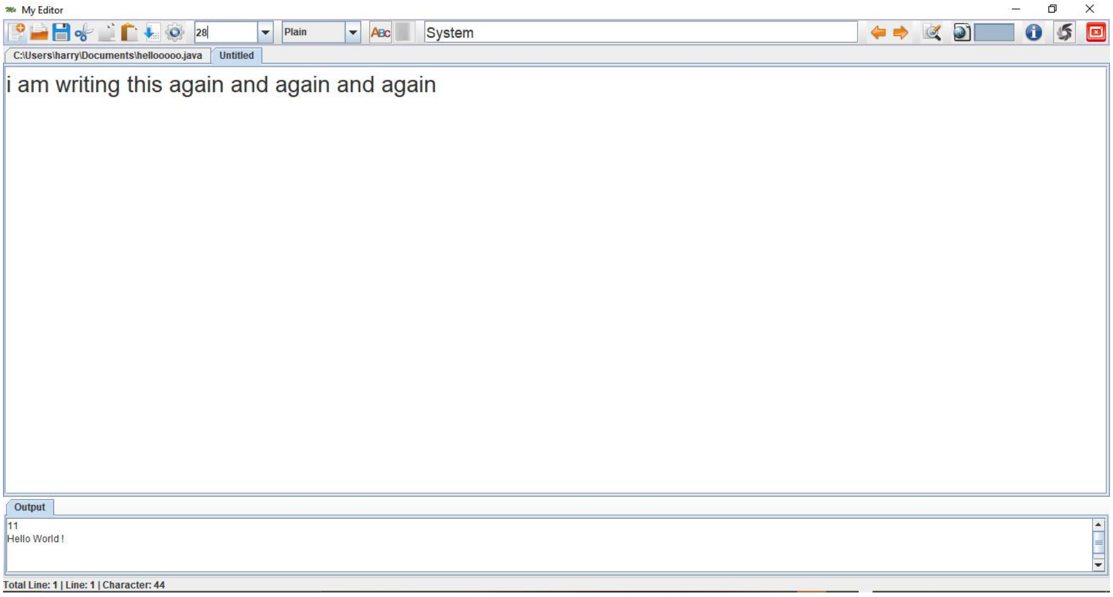
Output

```
11
Hello World !
```

Total Line: 9 | Line: 1 | Character: 16

11. FIND AND REPLACE : Used to enter a search string and replace it with other string.

BEFORE :



C:\Users\harry\Documents\hellooooo.java | Untitled

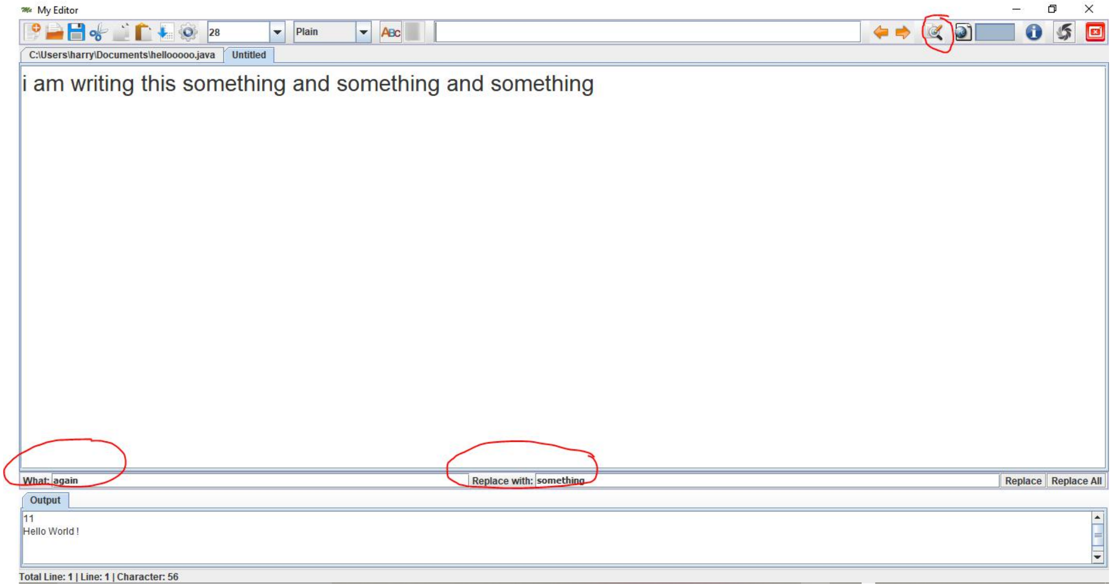
i am writing this again and again and again

Output

```
11
Hello World !
```

Total Line: 1 | Line: 1 | Character: 44

AFTER REPLACING again WITH something :



28

C:\Users\harry\Documents\hellooooo.java | Untitled

i am writing this something and something and something

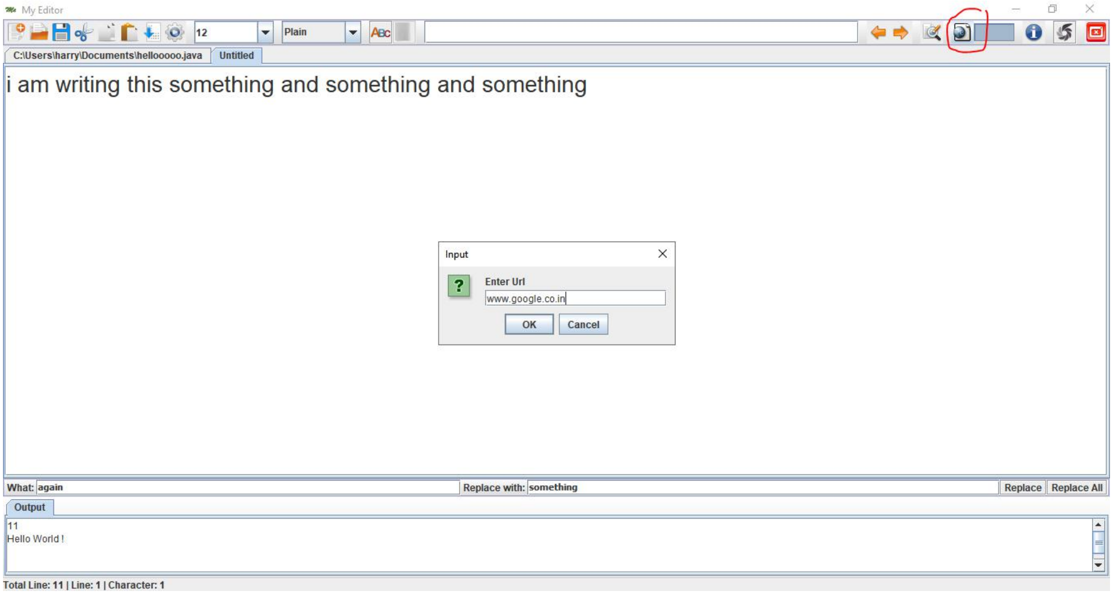
What: again

Replace with: something

```
11
Hello World !
```

Total Line: 1 | Line: 1 | Character: 56

12. FOR READING URL : Used to read URL.



12

C:\Users\harry\Documents\hellooooo.java | Untitled

i am writing this something and something and something

Enter Url

www.google.co.in

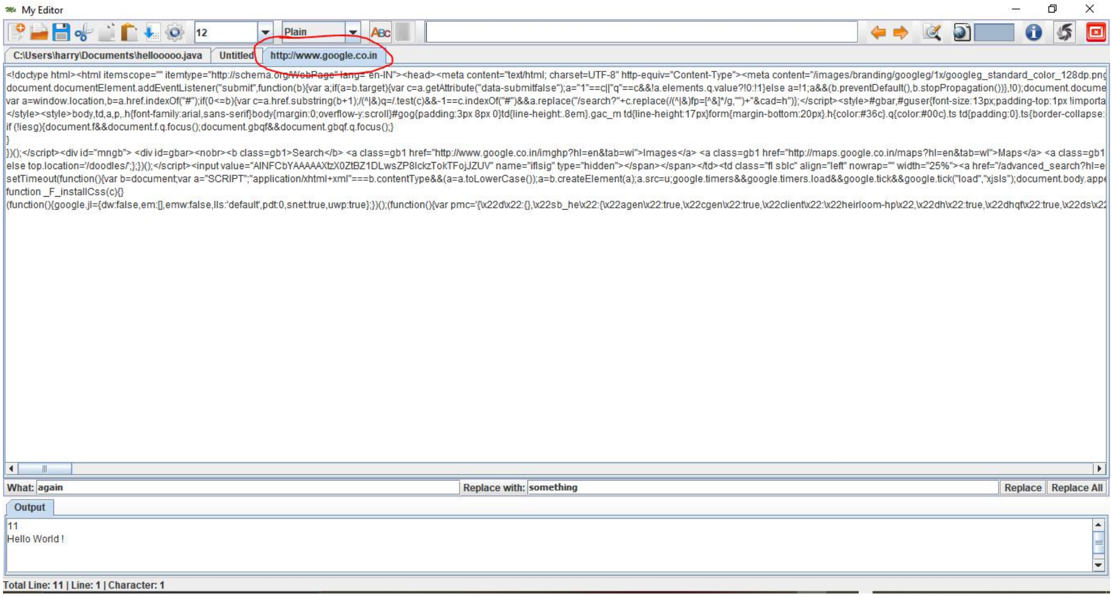
What: again

Replace with: something

```
11
Hello World !
```

Total Line: 11 | Line: 1 | Character: 1

THE HTML CODE OF THE ENTERED URL :



C:\Users\harry\Documents\hellooooo.java | Untitled | http://www.google.co.in

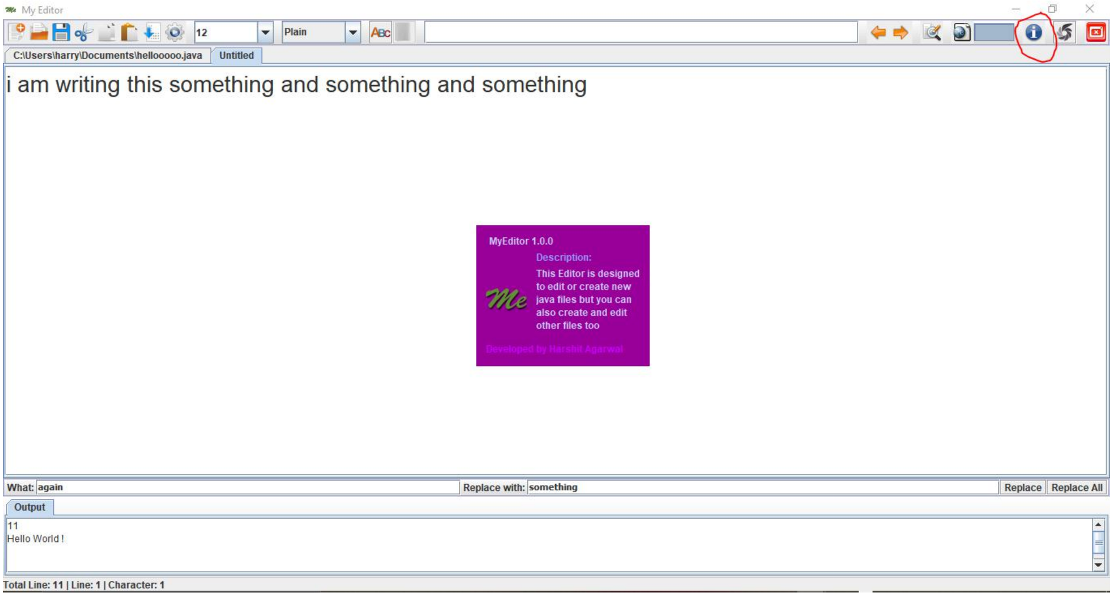
```
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a="1"==c||"q"==c&&!a.elements.q.value?!0:!1}else a=!1;a&&(b.preventDefault(),b.stopPropagation())},!0);document.docume
var a=window.location,b=a.href.indexOf("#");if(0<=b){var c=a.href.substring(b+1);/(^|&)q=/.test(c)&&-1==c.indexOf("#")&&a.replace("/search?"+c.replace(/(^|&)fp=[^&]*/g,"")+"&cad=h")};</script><style>#gbar,#guser{font-size:13px;padding-top:1px !importa
</style><style>body,td,a,p,.h{font-family:arial,sans-serif}body{margin:0;overflow-y:scroll}#gog{padding:3px 8px 0}td{line-height:.8em}.gac_m td{line-height:17px}form{margin-bottom:20px}.h{color:#36c}.q{color:#00c}.ts td{padding:0}.ts{border-collapse:
if (!iesg){document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();}
}
})();</script><div id="mngb"> <div id=gbar><nobr><b class=gb1>Search</b> <a class=gb1 href="http://www.google.co.in/imghp?hl=en&tab=wi">Images</a> <a class=gb1 href="http://maps.google.co.in/maps?hl=en&tab=wl">Maps</a> <a class=gb1
else top.location='/doodles/';};})();</script><input value="AINFCbYAAAAAXtzX0ZtBZ1DLwsZP8IckzTokTFojJZUV" name="iflsig" type="hidden"></span></span></td><td class="fl sblc" align="left" nowrap="" width="25%"><a href="/advanced_search?hl=e
setTimeout(function(){var b=document;var a="SCRIPT";"application/xhtml+xml"===b.contentType&&(a=a.toLowerCase());a=b.createElement(a);a.src=u;google.timers&&google.timers.load&&google.tick&&google.tick("load","xjsls");document.body.appe
function _F_installCss(c){}
(function(){google.jl={dw:false,em:[],emw:false,lls:'default',pdt:0,snet:true,uwp:true};})();(function(){var pmc='{\x22d\x22:{},\x22sb_he\x22:{\x22agen\x22:true,\x22cgen\x22:true,\x22client\x22:\x22heirloom-hp\x22,\x22dh\x22:true,\x22dhqt\x22:true,\x22ds\x22
```

What: again | Replace with: something

```
11
Hello World !
```

Total Line: 11 | Line: 1 | Character: 1

13. ABOUT BUTTON : Gives information about the editor.



C:\Users\harry\Documents\hellooooo.java | Untitled

i am writing this something and something and something

MyEditor 1.0.0

Description:

This Editor is designed to edit or create new java files but you can also create and edit other files too

Developed by Harshit Agarwal

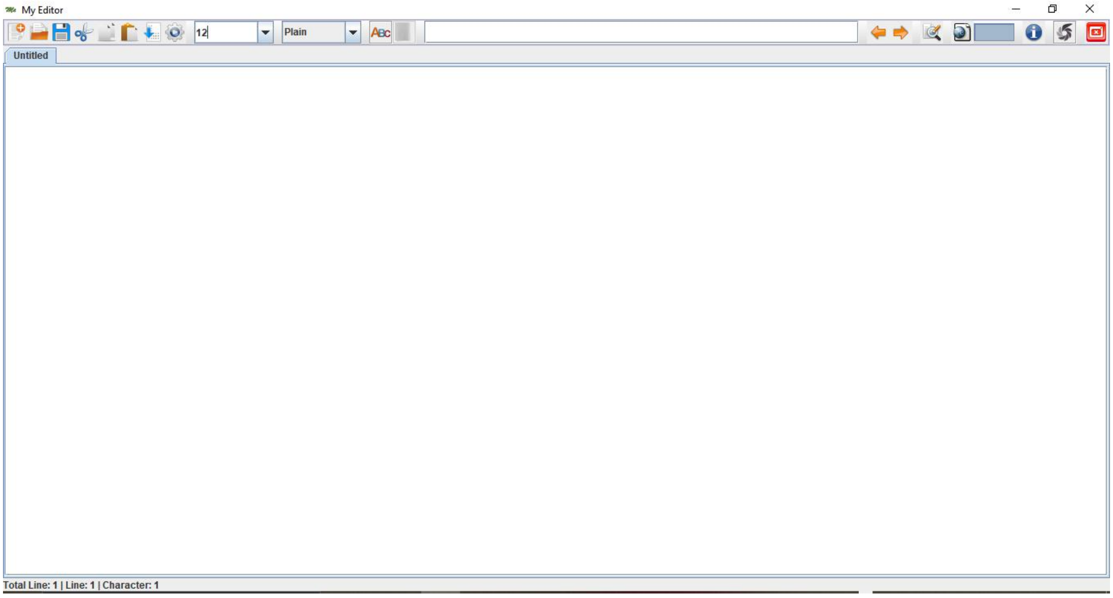
What: again | Replace with: something

```
11
Hello World !
```

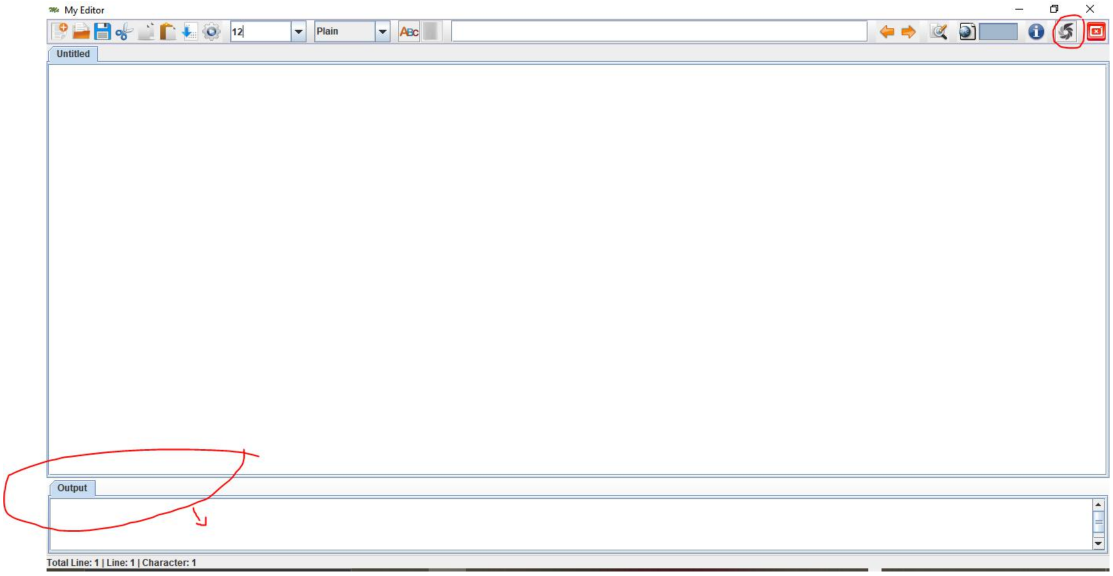
Total Line: 11 | Line: 1 | Character: 1

14. SHOW/HIDE OUTPUT BUTTON : Used to show / hide output area at the bottom.

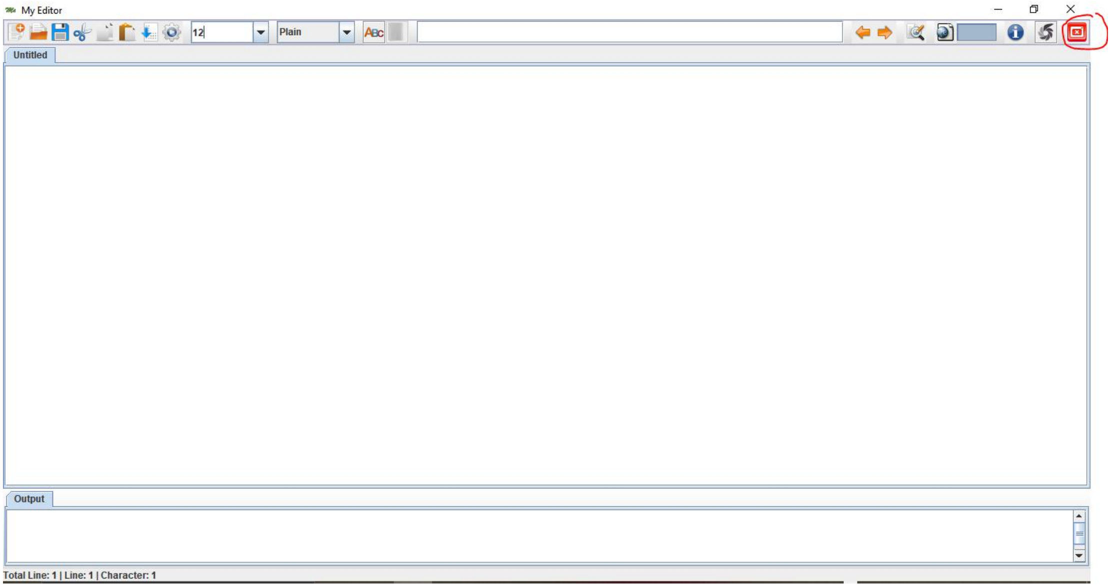
BEFORE :



AFTER :

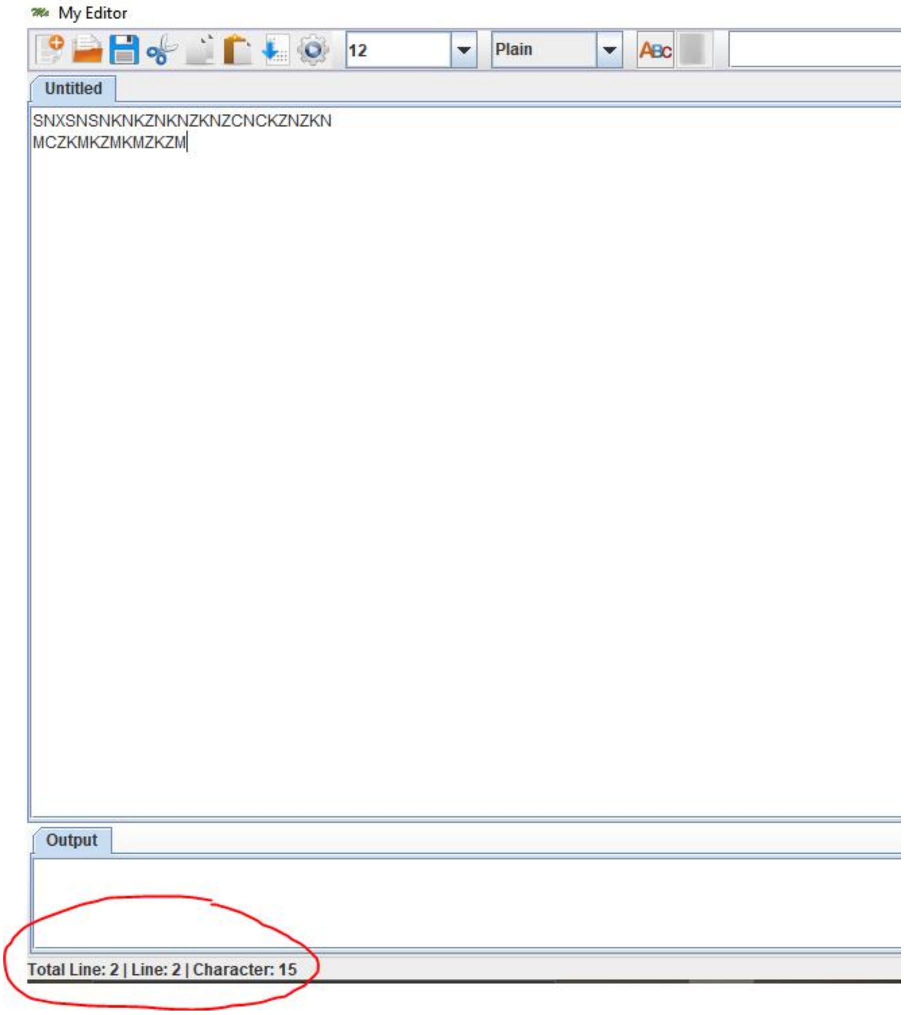


15. CLOSE BUTTON : Used to close the Current Tab.



Total Line: 1 | Line: 1 | Character: 1

16. INDICATION OF LINE AND CHARACTERS :



```
SNXSNSNKNKZNKNZKNZCNCKZNZKN
MCZKMKZMKMZKZM
```

Total Line: 2 | Line: 2 | Character: 15

SOURCE CODE

About.java :

```
package jeditor;

/**
 *
 * @author Harshit Agarwal
 */
public class About extends javax.swing.JFrame {

    /** Creates new form sprite */
    public About() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    //          <editor-fold          defaultstate="collapsed"          desc="Generated
Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel();
        jLabel6 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setAlwaysOnTop(true);
        setBackground(new java.awt.Color(255, 204, 204));
        setLocationByPlatform(true);
        setUndecorated(true);

        jPanel2.setBackground(new java.awt.Color(153, 0, 153));

        jLabel6.setForeground(new java.awt.Color(204, 204, 255));
        jLabel6.setText("java files but you can");

        jLabel3.setForeground(new java.awt.Color(153, 153, 255));
        jLabel3.setText("Description:");

        jLabel4.setForeground(new java.awt.Color(204, 204, 255));
        jLabel4.setText("This Editor is designed");

        jLabel5.setForeground(new java.awt.Color(204, 204, 255));
        jLabel5.setText("to edit or create new");

        jLabel7.setForeground(new java.awt.Color(204, 204, 255));
        jLabel7.setText("also create and edit");

        jLabel1.setForeground(new java.awt.Color(204, 204, 255));
        jLabel1.setText("MyEditor 1.0.0");

        jLabel2.setForeground(new java.awt.Color(204, 0, 255));
        jLabel2.setText("Developed by Harshit Agarwal");

        jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/logo.png"))); // NOI18N

        jLabel8.setForeground(new java.awt.Color(204, 204, 255));
        jLabel8.setText("other files too");

        javax.swing.GroupLayout          jPanel2Layout          =          new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.Group
Layout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addGroup(jPanel2Layout.createParallelGroup(javax.sw
ing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel2Layout.createSequentialGroup()
                                .addGap(4, 4, 4)
                                .addComponent(jLabel1))
                            .addComponent(jLabel2))
                        .addGap(15, 15, 15))
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel9)
                        .addPreferredGap(javax.swing.LayoutStyle.Component
Placement.UNRELATED)
                        .addGroup(jPanel2Layout.createParallelGroup(javax.sw
ing.GroupLayout.Alignment.LEADING)
```

```
                                   .addComponent(jLabel6,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                   .addComponent(jLabel5,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                   .addComponent(jLabel4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                   .addGroup(jPanel2Layout.createSequentialGroup()
                                        .addGroup(jPanel2Layout.createParallelGrou
p(javax.swing.GroupLayout.Alignment.LEADING)
                                             .addComponent(jLabel3)
                                             .addComponent(jLabel7,
javax.swing.GroupLayout.PREFERRED_SIZE,                              126,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                             .addComponent(jLabel8,
javax.swing.GroupLayout.PREFERRED_SIZE,                              126,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                        .addGap(0, 0, Short.MAX_VALUE)))))
                    .addContainerGap())
          );
          jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(jPanel2Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE,                               14,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                    .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE,                               14,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.Group
Layout.Alignment.TRAILING)
                         .addGroup(jPanel2Layout.createSequentialGroup()
                              .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE,                               14,
javax.swing.GroupLayout.PREFERRED_SIZE)
                              .addGap(2, 2, 2)
                              .addComponent(jLabel5,
javax.swing.GroupLayout.PREFERRED_SIZE,                               14,
javax.swing.GroupLayout.PREFERRED_SIZE)
                              .addGap(2, 2, 2)
```

```
                        .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE,                                  14,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(2, 2, 2)
                        .addComponent(jLabel7,
javax.swing.GroupLayout.PREFERRED_SIZE,                                  14,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addComponent(jLabel9))
                .addGap(2, 2, 2)
                .addComponent(jLabel8,
javax.swing.GroupLayout.PREFERRED_SIZE,                                  14,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
UNRELATED)
                .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE, 20, Short.MAX_VALUE)
                .addContainerGap())
        );

        javax.swing.GroupLayout          layout          =          new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel2,  javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel2,  javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    /**
     * @param args the command line arguments
     */
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
```

```
        private javax.swing.JLabel jLabel9;
        private javax.swing.JPanel jPanel2;
        // End of variables declaration//GEN-END:variables
}
```

Area.java

```
package jeditor;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextPane;
import javax.swing.border.EmptyBorder;
import javax.swing.text.SimpleAttributeSet;

/**
 *
 * @author Harshit Agarwal
 */
class Area extends JScrollPane implements KeyListener {

        JTextPane jtp = new JTextPane();
        JPanel panel = new JPanel();
        SimpleAttributeSet attr = new SimpleAttributeSet();
        Font font;
        boolean saved = false;
        boolean edited;
        boolean hindi = false;
        public int caretPosition,len;

        public boolean isEdited() {
                return edited;
        }

        public void setEdited(boolean edited) {
                this.edited = edited;
        }

        public void setSaved(boolean saved) {
                this.saved = saved;
        }
	public boolean isSaved() {
		return saved;
	}

	public boolean isHindi() {
		return hindi;
	}

	public void setHindi(boolean flag) {
		hindi = flag;
	}

	public Area() {
		initComponent();
	}

	private void initComponent() {
		jtp.setBorder(new EmptyBorder(1, 1, 1, 1));//same border for synch list and
jtp lines
		panel.setLayout(new BorderLayout());
		panel.add(jtp, BorderLayout.CENTER);
		setViewportView(panel);
		setFont("Aerial", Font.PLAIN, 12);
		jtp.addKeyListener(this);


	}

	public JTextPane getArea() {
		return jtp;
	}


	public void setFont(String face, int style, int size) {
		font = new Font(face, style, size);
		jtp.setFont(font);
	}

	@Override
	public Font getFont(){
		return font;
	}
	@Override
	public void keyTyped(KeyEvent e) {
	//	jtp.getHighlighter().removeAllHighlights();
	}

	@Override
	public void keyPressed(KeyEvent e) {
```

```
            //calculate caret position
            caretPosition = jtp.getCaretPosition();
            len = jtp.getText().length();
            setEdited(true);
    }


    @Override
    public void keyReleased(KeyEvent e) {
         //line count and update line numbers
            if (e.getKeyCode() != KeyEvent.VK_SPACE && e.getKeyCode() !=
KeyEvent.VK_LEFT && e.getKeyCode() != KeyEvent.VK_RIGHT &&
e.getKeyCode() != KeyEvent.VK_UP && e.getKeyCode() != KeyEvent.VK_DOWN
&& e.getKeyCode() != KeyEvent.VK_HOME && e.getKeyCode() !=
KeyEvent.VK_END && e.getKeyCode() != KeyEvent.VK_SHIFT && hindi) {
                String translated = Translator.toHindi(jtp.getText());
                jtp.setText(translated);
                int len2 = translated.length();
                if(caretPosition < jtp.getCaretPosition()){
                     jtp.setCaretPosition(caretPosition+(len2-len));
                }
            }

    }

}
```

Editor.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * @author Harshit Agarwal
 *
 * Created on Aug 12, 2012, 9:16:46 PM
 */
package jeditor;

import java.awt.Color;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.URL;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.text.DefaultHighlighter;
import javax.swing.text.Highlighter;

/**
 *
 * @author Rajesh
 */
public class Editor extends javax.swing.JFrame implements WindowListener {

    /**
     * Creates new form Editor
     */
    public Editor() {
        setIconImage(new
ImageIcon(getClass().getResource("/img/logo.png")).getImage());
        initComponents();
        area = new Area();
        addTab("Untitled", area);
        addWindowListener(this);
        output.setVisible(false);
        startCounter();
    }

    private void startCounter(){
        new Thread(){
            @Override
            public void run(){
                String text="";
                int lastp = 0;
                while(running){
                    if(area!=null){
                        try{
                        text = "Total Line: "+
area.getArea().getText().split("\n").length+" | ";
                        text += "Line: "+area.getArea().getText(0,
area.getArea().getCaretPosition()).split("\n").length+" | ";
                            if (area.getArea().getText().lastIndexOf("\n",
area.getArea().getCaretPosition()) > 0) {
```

```
                                lastp = area.getArea().getText(0,
area.getArea().getCaretPosition()).lastIndexOf("\n");
                                text += "Character: " +
(area.getArea().getCaretPosition() - lastp);
                            } else {
                                text += "Character: " +
(area.getArea().getCaretPosition()+1);
                            }
                        linecurL.setText(text);
                        Thread.sleep(10);
                        }catch(Exception e){}
                    }
                }
            }
        }.start();
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        topToolbar = new javax.swing.JToolBar();
        filler1 = new javax.swing.Box.Filler(new java.awt.Dimension(0, 0), new
java.awt.Dimension(0, 0), new java.awt.Dimension(0, 0));
        newB = new javax.swing.JButton();
        filler2 = new javax.swing.Box.Filler(new java.awt.Dimension(0, 0), new
java.awt.Dimension(0, 0), new java.awt.Dimension(32767, 0));
        openB = new javax.swing.JButton();
        saveB = new javax.swing.JButton();
        cutB = new javax.swing.JButton();
        copyB = new javax.swing.JButton();
        pasteB = new javax.swing.JButton();
        compileB = new javax.swing.JButton();
        runB = new javax.swing.JButton();
        jSeparator1 = new javax.swing.JToolBar.Separator();
        fontSizeC = new javax.swing.JComboBox();
        jSeparator2 = new javax.swing.JToolBar.Separator();
        fontStyleC = new javax.swing.JComboBox();
        jSeparator3 = new javax.swing.JToolBar.Separator();
        langB = new javax.swing.JButton();
        hmapB = new javax.swing.JButton();
        jSeparator4 = new javax.swing.JToolBar.Separator();
        findT = new javax.swing.JTextField();
        jSeparator5 = new javax.swing.JToolBar.Separator();
        findPB = new javax.swing.JButton();
        findNB = new javax.swing.JButton();
        jSeparator8 = new javax.swing.JToolBar.Separator();
        showReplace = new javax.swing.JButton();
        jSeparator6 = new javax.swing.JToolBar.Separator();
        readUrlB = new javax.swing.JButton();
        pbar = new javax.swing.JProgressBar();
        jSeparator9 = new javax.swing.JToolBar.Separator();
        aboutB = new javax.swing.JButton();
        jSeparator10 = new javax.swing.JToolBar.Separator();
        showOutputB = new javax.swing.JButton();
        jSeparator7 = new javax.swing.JToolBar.Separator();
        closeB = new javax.swing.JButton();
        workArea = new javax.swing.JTabbedPane();
        bottomContainer = new javax.swing.JPanel();
        jToolBar2 = new javax.swing.JToolBar();
        jLabel2 = new javax.swing.JLabel();
        replaceWhatT = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        replaceWithT = new javax.swing.JTextField();
        replaceB = new javax.swing.JButton();
        replaceAllB = new javax.swing.JButton();
        output = new javax.swing.JToolBar();
        jTabbedPane1 = new javax.swing.JTabbedPane();
        jScrollPane2 = new javax.swing.JScrollPane();
        outputT = new javax.swing.JTextArea();
        linecurL = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("My Editor");
        setBounds(new java.awt.Rectangle(0, 0, 600, 0));
        setIconImage(new
ImageIcon(getClass().getResource("/img/logo.png")).getImage());
        setMinimumSize(new java.awt.Dimension(600, 450));

        topToolbar.setBorder(javax.swing.BorderFactory.createEtchedBorder(new
java.awt.Color(204, 204, 255), null));
        topToolbar.setFloatable(false);
        topToolbar.setRollover(true);
        topToolbar.add(filler1);

        newB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/newb.png"))); // NOI18N
        newB.setMnemonic('N');
        newB.setToolTipText("Create new file (Alt+N)");
        newB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2, 2));
        newB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        newB.setFocusable(false);
        newB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        newB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        newB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                newBActionPerformed(evt);
            }
        });
        topToolbar.add(newB);
        topToolbar.add(filler2);

        openB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/openb.png"))); // NOI18N
        openB.setMnemonic('O');
        openB.setToolTipText("Open file (Alt+O)");
        openB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        openB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        openB.setFocusable(false);
        openB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        openB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        openB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                openBActionPerformed(evt);
            }
        });
        topToolbar.add(openB);

        saveB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/saveb.png"))); // NOI18N
        saveB.setMnemonic('S');
        saveB.setToolTipText("Save file (Alt+S)");
        saveB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        saveB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        saveB.setFocusable(false);
        saveB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        saveB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        saveB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                saveBActionPerformed(evt);
            }
        });
        topToolbar.add(saveB);

        cutB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/cutb.png"))); // NOI18N
        cutB.setMnemonic('X');
        cutB.setToolTipText("Cut (Alt+X)");
        cutB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2, 2));
        cutB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        cutB.setFocusable(false);
        cutB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        cutB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        cutB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cutBActionPerformed(evt);
            }
        });
        topToolbar.add(cutB);

        copyB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/copyb.png"))); // NOI18N
        copyB.setMnemonic('C');
        copyB.setToolTipText("Copy (Alt+C)");
        copyB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        copyB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        copyB.setFocusable(false);
        copyB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        copyB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        copyB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                copyBActionPerformed(evt);
            }
        });
        topToolbar.add(copyB);

        pasteB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/pasteb.png"))); // NOI18N
        pasteB.setMnemonic('P');
        pasteB.setToolTipText("Paste (Alt+P)");
        pasteB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        pasteB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        pasteB.setFocusable(false);
        pasteB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        pasteB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        pasteB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                pasteBActionPerformed(evt);
            }
        });
        topToolbar.add(pasteB);

        compileB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/compileb.png"))); // NOI18N
        compileB.setToolTipText("Compile");
        compileB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        compileB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
```

```
        compileB.setFocusable(false);

compileB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        compileB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        compileB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                compileBActionPerformed(evt);
            }
        });
        topToolbar.add(compileB);

        runB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/runb.png"))); // NOI18N
        runB.setToolTipText("Run");
        runB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2, 2));
        runB.setFocusable(false);
        runB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        runB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        runB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                runBActionPerformed(evt);
            }
        });
        topToolbar.add(runB);
        topToolbar.add(jSeparator1);

        fontSizeC.setEditable(true);
        fontSizeC.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "3", "7", "12", "16", "24", "28", "32", "36" }));
        fontSizeC.setSelectedItem(12);
        fontSizeC.setToolTipText("Font Size");
        fontSizeC.setMaximumSize(new java.awt.Dimension(32767, 28));
        fontSizeC.setMinimumSize(new java.awt.Dimension(93, 28));
        fontSizeC.setPreferredSize(new java.awt.Dimension(98, 28));
        fontSizeC.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                fontSizeCActionPerformed(evt);
            }
        });
        topToolbar.add(fontSizeC);
        topToolbar.add(jSeparator2);

        fontStyleC.setModel(new        javax.swing.DefaultComboBoxModel(new
String[] { "Plain", "Bold", "Italic" }));
        fontStyleC.setToolTipText("Font Style");
        fontStyleC.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        fontStyleC.setFocusable(false);
        fontStyleC.setMaximumSize(new java.awt.Dimension(32767, 28));
        fontStyleC.setMinimumSize(new java.awt.Dimension(93, 28));
        fontStyleC.setPreferredSize(new java.awt.Dimension(98, 28));
        fontStyleC.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                fontStyleCActionPerformed(evt);
            }
        });
        topToolbar.add(fontStyleC);
        topToolbar.add(jSeparator3);

        langB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/abc.png"))); // NOI18N
        langB.setToolTipText("English");
        langB.setFocusable(false);
        langB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        langB.setMaximumSize(new java.awt.Dimension(29, 29));
        langB.setMinimumSize(new java.awt.Dimension(29, 29));
        langB.setPreferredSize(new java.awt.Dimension(29, 29));
        langB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                langBActionPerformed(evt);
            }
        });
        topToolbar.add(langB);

        hmapB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/map.png"))); // NOI18N
        hmapB.setToolTipText("Hindi Map");
        hmapB.setEnabled(false);
        hmapB.setFocusable(false);
        hmapB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        hmapB.setMaximumSize(new java.awt.Dimension(29, 29));
        hmapB.setMinimumSize(new java.awt.Dimension(29, 29));
        hmapB.setPreferredSize(new java.awt.Dimension(29, 29));
        hmapB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        hmapB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                hmapBActionPerformed(evt);
            }
        });
        topToolbar.add(hmapB);
        topToolbar.add(jSeparator4);

        findT.setColumns(25);
        findT.setFont(new java.awt.Font("Nyala", 0, 18)); // NOI18N
        findT.setToolTipText("Find Text");
        findT.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
        findT.setMaximumSize(new java.awt.Dimension(2147483647, 28));
        findT.setMinimumSize(new java.awt.Dimension(6, 28));
        findT.setPreferredSize(new java.awt.Dimension(331, 28));
        findT.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                findTKeyReleased(evt);
            }
        });
        topToolbar.add(findT);
        topToolbar.add(jSeparator5);

        findPB.setFont(new java.awt.Font("Segoe Script", 1, 24)); // NOI18N
        findPB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/fprevious.png"))); // NOI18N
        findPB.setToolTipText("Find Backward");
        findPB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2,    2,    2,
2));
        findPB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        findPB.setFocusable(false);
        findPB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        findPB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        findPB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                findPBActionPerformed(evt);
            }
        });
        topToolbar.add(findPB);

        findNB.setFont(new java.awt.Font("Segoe Script", 1, 24)); // NOI18N
        findNB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/fnext.png"))); // NOI18N
        findNB.setToolTipText("Find Forward");
        findNB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2,    2,    2,
2));
        findNB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        findNB.setFocusable(false);
        findNB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        findNB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        findNB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                findNBActionPerformed(evt);
            }
        });
        topToolbar.add(findNB);
        topToolbar.add(jSeparator8);

        showReplace.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/findreplaceb.png")));              //
NOI18N
        showReplace.setMnemonic('H');
        showReplace.setToolTipText("Replace (Alt+H)");
```

```
        showReplace.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2,
2, 2));
        showReplace.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        showReplace.setFocusable(false);

showReplace.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

showReplace.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        showReplace.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                showReplaceActionPerformed(evt);
            }
        });
        topToolbar.add(showReplace);
        topToolbar.add(jSeparator6);

        readUrlB.setFont(new java.awt.Font("Tahoma", 3, 12)); // NOI18N
        readUrlB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/urlreaderb.png"))); // NOI18N
        readUrlB.setMnemonic('U');
        readUrlB.setToolTipText("Read Url (Alt+U)");
        readUrlB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        readUrlB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        readUrlB.setFocusable(false);

readUrlB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        readUrlB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        readUrlB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                readUrlBActionPerformed(evt);
            }
        });
        topToolbar.add(readUrlB);

        pbar.setValue(100);
        pbar.setMaximumSize(new java.awt.Dimension(50, 22));
        pbar.setMinimumSize(new java.awt.Dimension(50, 22));
        pbar.setPreferredSize(new java.awt.Dimension(50, 22));
        pbar.setString("Reading");
        topToolbar.add(pbar);
        topToolbar.add(jSeparator9);

        aboutB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/aboutb.png"))); // NOI18N
        aboutB.setToolTipText("About");
        aboutB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
```

```
        aboutB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        aboutB.setFocusable(false);
        aboutB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        aboutB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        aboutB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                aboutBActionPerformed(evt);
            }
        });
        topToolbar.add(aboutB);
        topToolbar.add(jSeparator10);

        showOutputB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/showhide.png"))); // NOI18N
        showOutputB.setToolTipText("Show/Hide Output");
        showOutputB.setFocusable(false);

showOutputB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        showOutputB.setMaximumSize(new java.awt.Dimension(29, 29));
        showOutputB.setMinimumSize(new java.awt.Dimension(29, 29));
        showOutputB.setPreferredSize(new java.awt.Dimension(29, 29));

showOutputB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        showOutputB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                showOutputBActionPerformed(evt);
            }
        });
        topToolbar.add(showOutputB);
        topToolbar.add(jSeparator7);

        closeB.setFont(new java.awt.Font("Ebrima", 1, 22)); // NOI18N
        closeB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/closeb.png"))); // NOI18N
        closeB.setMnemonic('E');
        closeB.setToolTipText("Close (Alt+E)");
        closeB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        closeB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        closeB.setFocusable(false);
        closeB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        closeB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        closeB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                closeBActionPerformed(evt);
            }
        });
        topToolbar.add(closeB);
        getContentPane().add(topToolbar, java.awt.BorderLayout.NORTH);

        workArea.setCursor(new
java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        workArea.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                workAreaMouseClicked(evt);
            }
        });
        getContentPane().add(workArea, java.awt.BorderLayout.CENTER);

        bottomContainer.setLayout(new java.awt.BorderLayout());

        jToolBar2.setBorder(javax.swing.BorderFactory.createEtchedBorder(new
java.awt.Color(204, 204, 255), null));
        jToolBar2.setFloatable(false);
        jToolBar2.setRollover(true);
        jToolBar2.setPreferredSize(new java.awt.Dimension(227, 23));

        jLabel2.setText(" What: ");
        jToolBar2.add(jLabel2);

        replaceWhatT.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        replaceWhatT.setMinimumSize(new java.awt.Dimension(6, 26));
        replaceWhatT.setPreferredSize(new java.awt.Dimension(6, 10));
        jToolBar2.add(replaceWhatT);

        jLabel1.setText(" Replace with: ");
        jToolBar2.add(jLabel1);

        replaceWithT.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        replaceWithT.setMinimumSize(new java.awt.Dimension(6, 26));
        jToolBar2.add(replaceWithT);

        replaceB.setText("Replace");
        replaceB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        replaceB.setFocusable(false);

replaceB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        replaceB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        replaceB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                replaceBActionPerformed(evt);
            }
        });
        jToolBar2.add(replaceB);

        replaceAllB.setText("Replace All");
```

```
            replaceAllB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
            replaceAllB.setFocusable(false);

replaceAllB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

replaceAllB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
            replaceAllB.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    replaceAllBActionPerformed(evt);
                }
            });
            jToolBar2.add(replaceAllB);

            bottomContainer.add(jToolBar2, java.awt.BorderLayout.NORTH);
            jToolBar2.setVisible(false);

            output.setFloatable(false);
            output.setRollover(true);
            output.setToolTipText("Output");
            output.setMaximumSize(new java.awt.Dimension(32780, 100));
            output.setName("Output"); // NOI18N
            output.setPreferredSize(new java.awt.Dimension(113, 100));

            outputT.setColumns(20);
            outputT.setRows(5);
            jScrollPane2.setViewportView(outputT);

            jTabbedPane1.addTab("Output", jScrollPane2);

            output.add(jTabbedPane1);

            bottomContainer.add(output, java.awt.BorderLayout.CENTER);

            linecurL.setText("Total Line: 50; Current Line: 25; Character: 10;");
            bottomContainer.add(linecurL, java.awt.BorderLayout.PAGE_END);

            getContentPane().add(bottomContainer,
java.awt.BorderLayout.PAGE_END);

            pack();
        }// </editor-fold>//GEN-END:initComponents

private void openBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_openBActionPerformed
        JFileChooser jfc = new JFileChooser();
        int response = jfc.showOpenDialog(this);
        if (response == JFileChooser.APPROVE_OPTION) {
            try {
                file = jfc.getSelectedFile();//future check
                FileInputStream fis = new FileInputStream(file);
                byte b[] = new byte[fis.available()];
                fis.read(b);
                String str = new String(b);
                area = new Area();
                area.setFont("Aerial", fontStyleC.getSelectedIndex(),
Integer.parseInt(String.valueOf(fontSizeC.getSelectedItem())));
                area.getArea().setText(str);
                String name = file.toString();
                addTab(name, area);
                area.setEdited(false);
                area.setSaved(true);
            } catch (Exception e) {
                JOptionPane.showMessageDialog(this, "File Cannot Open");
            }
        }
}//GEN-LAST:event_openBActionPerformed

private void newBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_newBActionPerformed
        area = new Area();
        area.setFont("Aerial", fontStyleC.getSelectedIndex(),
Integer.parseInt(String.valueOf(fontSizeC.getSelectedItem())));
        addTab("Untitled", area);
}//GEN-LAST:event_newBActionPerformed

private void aboutBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_aboutBActionPerformed
        visible = !visible;
        aboutWin.setLocation(-aboutWin.getWidth() / 2 + this.getWidth() / 2, this.getY()
- aboutWin.getHeight() / 2 + this.getHeight() / 2);
        aboutWin.setVisible(visible);
}//GEN-LAST:event_aboutBActionPerformed

private void saveBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_saveBActionPerformed
        save();
}//GEN-LAST:event_saveBActionPerformed

private void closeBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_closeBActionPerformed
        area = (Area) workArea.getSelectedComponent();
        if (area != null) {
            if (area.isEdited()) {
                int response = JOptionPane.showConfirmDialog(rootPane, "Do you
want to Save Changes");
                if (JOptionPane.YES_OPTION == response) {
                    save();
                    workArea.removeTabAt(workArea.getSelectedIndex());
                } else if (JOptionPane.NO_OPTION == response) {
                    workArea.removeTabAt(workArea.getSelectedIndex());
                }
            } else {
                workArea.removeTabAt(workArea.getSelectedIndex());
            }
        } else {
            JOptionPane.showMessageDialog(rootPane, "There is no any opened
document!");
        }
}//GEN-LAST:event_closeBActionPerformed

private void cutBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_cutBActionPerformed
        area = (Area) workArea.getSelectedComponent();
        area.getArea().cut();
}//GEN-LAST:event_cutBActionPerformed

private void copyBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_copyBActionPerformed
        area = (Area) workArea.getSelectedComponent();
        area.getArea().copy();
}//GEN-LAST:event_copyBActionPerformed

private void pasteBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_pasteBActionPerformed
        area = (Area) workArea.getSelectedComponent();
        area.getArea().paste();
}//GEN-LAST:event_pasteBActionPerformed

private void findNBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_findNBActionPerformed
        find(findT.getText(), pos, 1);
}//GEN-LAST:event_findNBActionPerformed

private void findPBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_findPBActionPerformed
        find(findT.getText(), pos, -1);
}//GEN-LAST:event_findPBActionPerformed

private void workAreaMouseClicked(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_workAreaMouseClicked
        area = (Area) workArea.getSelectedComponent();
        String size = String.valueOf(area.getFont().getSize());
        fontStyleC.setSelectedIndex(area.getFont().getStyle());
        fontSizeC.setSelectedItem(size);
}//GEN-LAST:event_workAreaMouseClicked

private void findTKeyReleased(java.awt.event.KeyEvent evt)
{//GEN-FIRST:event_findTKeyReleased
        if (area.isHindi()) {
            area.getArea().getHighlighter().removeAllHighlights();
            findAll(Translator.toHindi(findT.getText()), 0);
        } else {
            area.getArea().getHighlighter().removeAllHighlights();
            findAll(findT.getText(), 0);
        }
    }//GEN-LAST:event_findTKeyReleased

    private void readUrlBActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_readUrlBActionPerformed
        new Thread() {
            @Override
            public void run() {
                try {
                    String path = JOptionPane.showInputDialog("Enter Url");
                    if (!path.startsWith("http://")) {
                        path = "http://" + path;
                    }
                    new Thread() {
                        @Override
                        public void run() {
                            int i = 0;
                            pbar.setStringPainted(true);
                            while (!taskDone) {
                                if (i == 100) {
                                    i = 0;
                                }
                                pbar.setValue(i);
                                i += 5;
                                try {
                                    Thread.sleep(50);
                                } catch (InterruptedException e) {
                                }
                            }
                            pbar.setValue(100);
                            pbar.setStringPainted(false);
                            taskDone = false;
                        }
                    }.start();
                    URL url = new URL(path);
                    BufferedReader br = new BufferedReader(new
    InputStreamReader(url.openStream()));
                    String read = null;
                    String content = "";
                    while ((read = br.readLine()) != null) {
                        content = content + read + "\n";
                    }
                    br.close();
                    area = new Area();
                    area.getArea().setText(content);
                    workArea.addTab(url.toString(), area);
                    workArea.setSelectedIndex(workArea.getTabCount() - 1);
                    taskDone = true;
                } catch (Exception ex) {
                    taskDone = true;
                    JOptionPane.showMessageDialog(null, ex);
                }
            }
        }.start();
    }//GEN-LAST:event_readUrlBActionPerformed

    private void showReplaceActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_showReplaceActionPerformed
        visible = !visible;
        jToolBar2.setVisible(visible);
        replaceWhatT.setText(findT.getText());
    }//GEN-LAST:event_showReplaceActionPerformed

    private void replaceBActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_replaceBActionPerformed
        String string = area.getArea().getText();
        string = string.replaceFirst(replaceWhatT.getText(), replaceWithT.getText());
        area.getArea().setText(string);
    }//GEN-LAST:event_replaceBActionPerformed

    private void replaceAllBActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_replaceAllBActionPerformed
        String string = area.getArea().getText();
        string = string.replaceAll(replaceWhatT.getText(), replaceWithT.getText());
        area.getArea().setText(string);
    }//GEN-LAST:event_replaceAllBActionPerformed

    private void fontSizeCActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_fontSizeCActionPerformed
        try {
            int size = Integer.parseInt(String.valueOf(fontSizeC.getSelectedItem()));
            int style = fontStyleC.getSelectedIndex();
            area.setFont("Aerial", style, size);
        } catch (Exception e) {
            fontSizeC.setSelectedIndex(2);
        }
    }//GEN-LAST:event_fontSizeCActionPerformed

    private void fontStyleCActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_fontStyleCActionPerformed
        int size = Integer.parseInt((String.valueOf(fontSizeC.getSelectedItem())));
        int style = fontStyleC.getSelectedIndex();
        area.setFont("Aerial", style, size);

    }//GEN-LAST:event_fontStyleCActionPerformed
```

```
    private void compileBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_compileBActionPerformed
        try {
            save();
            if (area.isSaved()) {

if(workArea.getTitleAt(workArea.getSelectedIndex()).lastIndexOf(".java")>0){
                Runtime runtime = Runtime.getRuntime();
                runtime.exec("javac " +
workArea.getTitleAt(workArea.getSelectedIndex()));
                outputT.setText("Compiled Successfully");
                }else{
                    outputT.setText("File Should be of .java extension");
                }
            } else {
                outputT.setText("Cannot compile unsaved file");
            }
        } catch (Exception ex) {
            outputT.setText("Error in Compilation! " + ex);

        }
    }//GEN-LAST:event_compileBActionPerformed

    private void runBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_runBActionPerformed
        save();
        new Thread() {
            @Override
            public void run() {
                try {
                    Runtime rt = Runtime.getRuntime();
                    String location =
workArea.getTitleAt(workArea.getSelectedIndex());
                    String fl = location.substring(location.lastIndexOf("\\") + 1,
location.lastIndexOf("."));
                    location = location.substring(0, location.lastIndexOf("\\"));
                    Process pr = rt.exec("java -classpath \"" + location + "\" " +
fl);
                    BufferedReader br = new BufferedReader(new
InputStreamReader(pr.getInputStream()));
                    String line;
                    outputT.setText("");
                    while ((line = br.readLine()) != null) {
                        outputT.setText(outputT.getText() + line + "\n");
                    }

                } catch (IOException ex) {
                    outputT.setText(ex.toString());
                }
            }
        }.start();

    }//GEN-LAST:event_runBActionPerformed

    private    void    langBActionPerformed(java.awt.event.ActionEvent    evt)
{//GEN-FIRST:event_langBActionPerformed

        if (area.isHindi()) {
            langB.setToolTipText("English");
            area.setHindi(false);
            hmapB.setEnabled(false);
            langB.setIcon(new
ImageIcon(getClass().getResource("/img/abc.png")));
        } else {
            langB.setToolTipText("Hindi");
            area.setHindi(true);
            hmapB.setEnabled(true);
            langB.setIcon(new ImageIcon(getClass().getResource("/img/a.png")));
        }
    }//GEN-LAST:event_langBActionPerformed

    private    void    hmapBActionPerformed(java.awt.event.ActionEvent    evt)
{//GEN-FIRST:event_hmapBActionPerformed
        if (hMap == null) {
            hMap = new HindiMap();
        }
        if (hMap.isVisible()) {
            hMap.setVisible(false);
        } else {
            hMap.setVisible(true);
        }
    }//GEN-LAST:event_hmapBActionPerformed

    private   void   showOutputBActionPerformed(java.awt.event.ActionEvent   evt)
{//GEN-FIRST:event_showOutputBActionPerformed
        if (output.isVisible()) {
            output.setVisible(false);
        } else {
            output.setVisible(true);
        }
    }//GEN-LAST:event_showOutputBActionPerformed

    public void find(String search, int pos, int d) {
        area.getArea().getHighlighter().removeAllHighlights();
        if (d == 1) {
            pos   =   area.getArea().getText().replaceAll("\n",   "").indexOf(search,
pos);
        }//replaceAll to remove new lines character
        if (d == -1) {
            pos = area.getArea().getText().replaceAll("\n", "").lastIndexOf(search,
pos);
        }
        if (pos >= 0) {
            try {
                Highlighter h = area.getArea().getHighlighter();
                h.addHighlight(pos, pos + search.length(), new
DefaultHighlighter.DefaultHighlightPainter(Color.GREEN));
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(area.getArea(), ex);
            }
            if (d == -1) {
                this.pos = pos - search.length();
            }
            if (d == 1) {
                this.pos = pos + search.length();
            }
        }
    }

    public void findAll(String search, int pos) {
        pos = area.getArea().getText().replaceAll("\n", "").indexOf(search,
pos);//replaceAll to remove new lines character
        if (pos >= 0 && search.length() > 0) {
            try {
                Highlighter h = area.getArea().getHighlighter();
                h.addHighlight(pos, pos + search.length(), new
DefaultHighlighter.DefaultHighlightPainter(Color.GREEN));
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(area.getArea(), ex);
            }
            pos += search.length();
            if (pos < area.getArea().getText().length()) {
                findAll(search, pos);
            }//recursion
        }
    }

    public void save() {

        JFileChooser jfc = new JFileChooser();
        area = (Area) workArea.getSelectedComponent();
        if (!area.isSaved()) {
            int response = jfc.showSaveDialog(this);
            if (response == JFileChooser.APPROVE_OPTION) {
                file = jfc.getSelectedFile();
                if (file.exists()) {
                    response = JOptionPane.showConfirmDialog(this, "File
allready exists, Dou you want to overrite");
                    if (response == JOptionPane.YES_OPTION) {
```

```
                    write(area);
                } else if (response == JOptionPane.NO_OPTION) {
                    save();

                }
            } else if (!file.exists()) {
                write(area);
            }
            area.setSaved(true);
        }

    } else {
        write(area);
    }


}

private void write(Area area) {
    try {
        String str = area.getArea().getText();
        pw = new PrintWriter(new FileOutputStream(file));
        pw.write(str);
        pw.flush();
        String name = file.toString();
        //name = name.substring(name.lastIndexOf("\\") + 1, name.length());
        workArea.setTitleAt(workArea.getSelectedIndex(), name);
        area.setEdited(false);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Cannot Save");
    }
}

private void addTab(String title, Area area) {
    workArea.addTab(title, area);
    workArea.setSelectedIndex(workArea.getTabCount() - 1);
}

boolean taskDone = false;
boolean running = true;
private HindiMap hMap;
private PrintWriter pw;
private File file;
private Area area;
private int pos;
private boolean visible;
private About aboutWin = new About();
boolean closeAll = false;
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton aboutB;
private javax.swing.JPanel bottomContainer;
private javax.swing.JButton closeB;
private javax.swing.JButton compileB;
private javax.swing.JButton copyB;
private javax.swing.JButton cutB;
private javax.swing.Box.Filler filler1;
private javax.swing.Box.Filler filler2;
private javax.swing.JButton findNB;
private javax.swing.JButton findPB;
private javax.swing.JTextField findT;
private javax.swing.JComboBox fontSizeC;
private javax.swing.JComboBox fontStyleC;
private javax.swing.JButton hmapB;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JToolBar.Separator jSeparator1;
private javax.swing.JToolBar.Separator jSeparator10;
private javax.swing.JToolBar.Separator jSeparator2;
private javax.swing.JToolBar.Separator jSeparator3;
private javax.swing.JToolBar.Separator jSeparator4;
private javax.swing.JToolBar.Separator jSeparator5;
private javax.swing.JToolBar.Separator jSeparator6;
private javax.swing.JToolBar.Separator jSeparator7;
private javax.swing.JToolBar.Separator jSeparator8;
private javax.swing.JToolBar.Separator jSeparator9;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JToolBar jToolBar2;
private javax.swing.JButton langB;
private javax.swing.JLabel linecurL;
private javax.swing.JButton newB;
private javax.swing.JButton openB;
private javax.swing.JToolBar output;
private javax.swing.JTextArea outputT;
private javax.swing.JButton pasteB;
private javax.swing.JProgressBar pbar;
private javax.swing.JButton readUrlB;
private javax.swing.JButton replaceAllB;
private javax.swing.JButton replaceB;
private javax.swing.JTextField replaceWhatT;
private javax.swing.JTextField replaceWithT;
private javax.swing.JButton runB;
private javax.swing.JButton saveB;
private javax.swing.JButton showOutputB;
private javax.swing.JButton showReplace;
private javax.swing.JToolBar topToolbar;
private javax.swing.JTabbedPane workArea;
// End of variables declaration//GEN-END:variables

@Override
```

```
	public void windowOpened(WindowEvent e) {
	}

	@Override
	public void windowClosing(WindowEvent e) {
		closeAll = true;
		running = false;
	}

	@Override
	public void windowClosed(WindowEvent e) {
		System.exit(0);
	}

	@Override
	public void windowIconified(WindowEvent e) {
	}

	@Override
	public void windowDeiconified(WindowEvent e) {
	}

	@Override
	public void windowActivated(WindowEvent e) {
	}

	@Override
	public void windowDeactivated(WindowEvent e) {
	}
}
```

HindiMap.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * @author Harshit Agarwal
 *
 * Created on Aug 12, 2012, 9:16:46 PM
 */
package jeditor;
```

```
import java.awt.Color;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.URL;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.text.DefaultHighlighter;
import javax.swing.text.Highlighter;

/**
 *
 * @author Rajesh
 */
public class Editor extends javax.swing.JFrame implements WindowListener {

    /**
     * Creates new form Editor
     */
    public Editor() {
        setIconImage(new
ImageIcon(getClass().getResource("/img/logo.png")).getImage());
        initComponents();
        area = new Area();
        addTab("Untitled", area);
        addWindowListener(this);
        output.setVisible(false);
        startCounter();
    }

    private void startCounter(){
        new Thread(){
            @Override
            public void run(){
                String text="";
                int lastp = 0;
                while(running){
                    if(area!=null){
                        try{
                            text = "Total Line: "+
area.getArea().getText().split("\n").length+" | ";
                            text += "Line: "+area.getArea().getText(0,
area.getArea().getCaretPosition()).split("\n").length+" | ";
```

```
                                        if (area.getArea().getText().lastIndexOf("\n",
area.getArea().getCaretPosition()) > 0) {
                                            lastp = area.getArea().getText(0,
area.getArea().getCaretPosition()).lastIndexOf("\n");
                                            text += "Character: " +
(area.getArea().getCaretPosition() - lastp);
                                        } else {
                                            text += "Character: " +
(area.getArea().getCaretPosition()+1);
                                        }
                                    linecurL.setText(text);
                                    Thread.sleep(10);
                                    }catch(Exception e){}
                                }
                            }
                        }
                    }.start();
                }
                /**
                 * This method is called from within the constructor to initialize the form.
                 * WARNING: Do NOT modify this code. The content of this method is always
                 * regenerated by the Form Editor.
                 */
                @SuppressWarnings("unchecked")
                // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
                private void initComponents() {

                    topToolbar = new javax.swing.JToolBar();
                    filler1 = new javax.swing.Box.Filler(new java.awt.Dimension(0, 0), new
java.awt.Dimension(0, 0), new java.awt.Dimension(0, 0));
                    newB = new javax.swing.JButton();
                    filler2 = new javax.swing.Box.Filler(new java.awt.Dimension(0, 0), new
java.awt.Dimension(0, 0), new java.awt.Dimension(32767, 0));
                    openB = new javax.swing.JButton();
                    saveB = new javax.swing.JButton();
                    cutB = new javax.swing.JButton();
                    copyB = new javax.swing.JButton();
                    pasteB = new javax.swing.JButton();
                    compileB = new javax.swing.JButton();
                    runB = new javax.swing.JButton();
                    jSeparator1 = new javax.swing.JToolBar.Separator();
                    fontSizeC = new javax.swing.JComboBox();
                    jSeparator2 = new javax.swing.JToolBar.Separator();
                    fontStyleC = new javax.swing.JComboBox();
                    jSeparator3 = new javax.swing.JToolBar.Separator();
                    langB = new javax.swing.JButton();
                    hmapB = new javax.swing.JButton();
                    jSeparator4 = new javax.swing.JToolBar.Separator();
                    findT = new javax.swing.JTextField();
```

```
        jSeparator5 = new javax.swing.JToolBar.Separator();
        findPB = new javax.swing.JButton();
        findNB = new javax.swing.JButton();
        jSeparator8 = new javax.swing.JToolBar.Separator();
        showReplace = new javax.swing.JButton();
        jSeparator6 = new javax.swing.JToolBar.Separator();
        readUrlB = new javax.swing.JButton();
        pbar = new javax.swing.JProgressBar();
        jSeparator9 = new javax.swing.JToolBar.Separator();
        aboutB = new javax.swing.JButton();
        jSeparator10 = new javax.swing.JToolBar.Separator();
        showOutputB = new javax.swing.JButton();
        jSeparator7 = new javax.swing.JToolBar.Separator();
        closeB = new javax.swing.JButton();
        workArea = new javax.swing.JTabbedPane();
        bottomContainer = new javax.swing.JPanel();
        jToolBar2 = new javax.swing.JToolBar();
        jLabel2 = new javax.swing.JLabel();
        replaceWhatT = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        replaceWithT = new javax.swing.JTextField();
        replaceB = new javax.swing.JButton();
        replaceAllB = new javax.swing.JButton();
        output = new javax.swing.JToolBar();
        jTabbedPane1 = new javax.swing.JTabbedPane();
        jScrollPane2 = new javax.swing.JScrollPane();
        outputT = new javax.swing.JTextArea();
        linecurL = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("My Editor");
        setBounds(new java.awt.Rectangle(0, 0, 600, 0));
        setIconImage(new
ImageIcon(getClass().getResource("/img/logo.png")).getImage());
        setMinimumSize(new java.awt.Dimension(600, 450));

        topToolbar.setBorder(javax.swing.BorderFactory.createEtchedBorder(new
java.awt.Color(204, 204, 255), null));
        topToolbar.setFloatable(false);
        topToolbar.setRollover(true);
        topToolbar.add(filler1);

        newB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/newb.png"))); // NOI18N
        newB.setMnemonic('N');
        newB.setToolTipText("Create new file (Alt+N)");
        newB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2, 2));
        newB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        newB.setFocusable(false);
        newB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        newB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        newB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                newBActionPerformed(evt);
            }
        });
        topToolbar.add(newB);
        topToolbar.add(filler2);

        openB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/openb.png"))); // NOI18N
        openB.setMnemonic('O');
        openB.setToolTipText("Open file (Alt+O)");
        openB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2,    2,    2,
2));
        openB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        openB.setFocusable(false);
        openB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        openB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        openB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                openBActionPerformed(evt);
            }
        });
        topToolbar.add(openB);

        saveB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/saveb.png"))); // NOI18N
        saveB.setMnemonic('S');
        saveB.setToolTipText("Save file (Alt+S)");
        saveB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2,    2,    2,
2));
        saveB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        saveB.setFocusable(false);
        saveB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        saveB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        saveB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                saveBActionPerformed(evt);
            }
        });
        topToolbar.add(saveB);

        cutB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/cutb.png"))); // NOI18N
        cutB.setMnemonic('X');
        cutB.setToolTipText("Cut (Alt+X)");
        cutB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2, 2));
        cutB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        cutB.setFocusable(false);
        cutB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        cutB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        cutB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cutBActionPerformed(evt);
            }
        });
        topToolbar.add(cutB);

        copyB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/copyb.png"))); // NOI18N
        copyB.setMnemonic('C');
        copyB.setToolTipText("Copy (Alt+C)");
        copyB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        copyB.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        copyB.setFocusable(false);
        copyB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        copyB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        copyB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                copyBActionPerformed(evt);
            }
        });
        topToolbar.add(copyB);

        pasteB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/pasteb.png"))); // NOI18N
        pasteB.setMnemonic('P');
        pasteB.setToolTipText("Paste (Alt+P)");
        pasteB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        pasteB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        pasteB.setFocusable(false);
        pasteB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        pasteB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        pasteB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                pasteBActionPerformed(evt);
            }
        });
        topToolbar.add(pasteB);

        compileB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/compileb.png"))); // NOI18N
        compileB.setToolTipText("Compile");
        compileB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
```

```
        compileB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        compileB.setFocusable(false);

compileB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        compileB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        compileB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                compileBActionPerformed(evt);
            }
        });
        topToolbar.add(compileB);

        runB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/runb.png"))); // NOI18N
        runB.setToolTipText("Run");
        runB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2, 2));
        runB.setFocusable(false);
        runB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        runB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        runB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                runBActionPerformed(evt);
            }
        });
        topToolbar.add(runB);
        topToolbar.add(jSeparator1);

        fontSizeC.setEditable(true);
        fontSizeC.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "3", "7", "12", "16", "24", "28", "32", "36" }));
        fontSizeC.setSelectedItem(12);
        fontSizeC.setToolTipText("Font Size");
        fontSizeC.setMaximumSize(new java.awt.Dimension(32767, 28));
        fontSizeC.setMinimumSize(new java.awt.Dimension(93, 28));
        fontSizeC.setPreferredSize(new java.awt.Dimension(98, 28));
        fontSizeC.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                fontSizeCActionPerformed(evt);
            }
        });
        topToolbar.add(fontSizeC);
        topToolbar.add(jSeparator2);

        fontStyleC.setModel(new        javax.swing.DefaultComboBoxModel(new
String[] { "Plain", "Bold", "Italic" }));
        fontStyleC.setToolTipText("Font Style");
        fontStyleC.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        fontStyleC.setFocusable(false);
        fontStyleC.setMaximumSize(new java.awt.Dimension(32767, 28));
        fontStyleC.setMinimumSize(new java.awt.Dimension(93, 28));
        fontStyleC.setPreferredSize(new java.awt.Dimension(98, 28));
        fontStyleC.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                fontStyleCActionPerformed(evt);
            }
        });
        topToolbar.add(fontStyleC);
        topToolbar.add(jSeparator3);

        langB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/abc.png"))); // NOI18N
        langB.setToolTipText("English");
        langB.setFocusable(false);
        langB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        langB.setMaximumSize(new java.awt.Dimension(29, 29));
        langB.setMinimumSize(new java.awt.Dimension(29, 29));
        langB.setPreferredSize(new java.awt.Dimension(29, 29));
        langB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                langBActionPerformed(evt);
            }
        });
        topToolbar.add(langB);

        hmapB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/map.png"))); // NOI18N
        hmapB.setToolTipText("Hindi Map");
        hmapB.setEnabled(false);
        hmapB.setFocusable(false);
        hmapB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        hmapB.setMaximumSize(new java.awt.Dimension(29, 29));
        hmapB.setMinimumSize(new java.awt.Dimension(29, 29));
        hmapB.setPreferredSize(new java.awt.Dimension(29, 29));
        hmapB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        hmapB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                hmapBActionPerformed(evt);
            }
        });
        topToolbar.add(hmapB);
        topToolbar.add(jSeparator4);

        findT.setColumns(25);
        findT.setFont(new java.awt.Font("Nyala", 0, 18)); // NOI18N
        findT.setToolTipText("Find Text");
        findT.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
        findT.setMaximumSize(new java.awt.Dimension(2147483647, 28));
        findT.setMinimumSize(new java.awt.Dimension(6, 28));
        findT.setPreferredSize(new java.awt.Dimension(331, 28));
        findT.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                findTKeyReleased(evt);
            }
        });
        topToolbar.add(findT);
        topToolbar.add(jSeparator5);

        findPB.setFont(new java.awt.Font("Segoe Script", 1, 24)); // NOI18N
        findPB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/fprevious.png"))); // NOI18N
        findPB.setToolTipText("Find Backward");
        findPB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        findPB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        findPB.setFocusable(false);
        findPB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        findPB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        findPB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                findPBActionPerformed(evt);
            }
        });
        topToolbar.add(findPB);

        findNB.setFont(new java.awt.Font("Segoe Script", 1, 24)); // NOI18N
        findNB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/fnext.png"))); // NOI18N
        findNB.setToolTipText("Find Forward");
        findNB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2, 2,
2));
        findNB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        findNB.setFocusable(false);
        findNB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        findNB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        findNB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                findNBActionPerformed(evt);
            }
        });
        topToolbar.add(findNB);
        topToolbar.add(jSeparator8);

        showReplace.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/findreplaceb.png")));            //
NOI18N
        showReplace.setMnemonic('H');
        showReplace.setToolTipText("Replace (Alt+H)");
        showReplace.setBorder(javax.swing.BorderFactory.createEmptyBorder(2, 2,
2, 2));
        showReplace.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        showReplace.setFocusable(false);

showReplace.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

showReplace.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        showReplace.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                showReplaceActionPerformed(evt);
            }
        });
        topToolbar.add(showReplace);
        topToolbar.add(jSeparator6);

        readUrlB.setFont(new java.awt.Font("Tahoma", 3, 12)); // NOI18N
        readUrlB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/urlreaderb.png"))); // NOI18N
        readUrlB.setMnemonic('U');
        readUrlB.setToolTipText("Read Url (Alt+U)");
        readUrlB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2,  2,  2,
2));
        readUrlB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        readUrlB.setFocusable(false);

readUrlB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        readUrlB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        readUrlB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                readUrlBActionPerformed(evt);
            }
        });
        topToolbar.add(readUrlB);

        pbar.setValue(100);
        pbar.setMaximumSize(new java.awt.Dimension(50, 22));
        pbar.setMinimumSize(new java.awt.Dimension(50, 22));
        pbar.setPreferredSize(new java.awt.Dimension(50, 22));
        pbar.setString("Reading");
        topToolbar.add(pbar);
        topToolbar.add(jSeparator9);

        aboutB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/aboutb.png"))); // NOI18N
        aboutB.setToolTipText("About");
        aboutB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2,   2,   2,
2));
        aboutB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        aboutB.setFocusable(false);
        aboutB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        aboutB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        aboutB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                aboutBActionPerformed(evt);
            }
        });
        topToolbar.add(aboutB);
        topToolbar.add(jSeparator10);

        showOutputB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/showhide.png"))); // NOI18N
        showOutputB.setToolTipText("Show/Hide Output");
        showOutputB.setFocusable(false);

showOutputB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        showOutputB.setMaximumSize(new java.awt.Dimension(29, 29));
        showOutputB.setMinimumSize(new java.awt.Dimension(29, 29));
        showOutputB.setPreferredSize(new java.awt.Dimension(29, 29));

showOutputB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        showOutputB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                showOutputBActionPerformed(evt);
            }
        });
        topToolbar.add(showOutputB);
        topToolbar.add(jSeparator7);

        closeB.setFont(new java.awt.Font("Ebrima", 1, 22)); // NOI18N
        closeB.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/closeb.png"))); // NOI18N
        closeB.setMnemonic('E');
        closeB.setToolTipText("Close (Alt+E)");
        closeB.setBorder(javax.swing.BorderFactory.createEmptyBorder(2,   2,   2,
2));
        closeB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        closeB.setFocusable(false);
        closeB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        closeB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        closeB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                closeBActionPerformed(evt);
            }
        });
        topToolbar.add(closeB);

        getContentPane().add(topToolbar, java.awt.BorderLayout.NORTH);

        workArea.setCursor(new
java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        workArea.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                workAreaMouseClicked(evt);
            }
        });
        getContentPane().add(workArea, java.awt.BorderLayout.CENTER);

        bottomContainer.setLayout(new java.awt.BorderLayout());

        jToolBar2.setBorder(javax.swing.BorderFactory.createEtchedBorder(new
java.awt.Color(204, 204, 255), null));
        jToolBar2.setFloatable(false);
        jToolBar2.setRollover(true);
        jToolBar2.setPreferredSize(new java.awt.Dimension(227, 23));

        jLabel2.setText(" What: ");
        jToolBar2.add(jLabel2);

        replaceWhatT.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        replaceWhatT.setMinimumSize(new java.awt.Dimension(6, 26));
        replaceWhatT.setPreferredSize(new java.awt.Dimension(6, 10));
        jToolBar2.add(replaceWhatT);

        jLabel1.setText(" Replace with: ");
        jToolBar2.add(jLabel1);

        replaceWithT.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        replaceWithT.setMinimumSize(new java.awt.Dimension(6, 26));
        jToolBar2.add(replaceWithT);

        replaceB.setText("Replace");
        replaceB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        replaceB.setFocusable(false);

replaceB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        replaceB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        replaceB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                replaceBActionPerformed(evt);
            }
        });
        jToolBar2.add(replaceB);
```

```
            replaceAllB.setText("Replace All");
            replaceAllB.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
            replaceAllB.setFocusable(false);

replaceAllB.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

replaceAllB.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
            replaceAllB.addActionListener(new java.awt.event.ActionListener() {
                  public void actionPerformed(java.awt.event.ActionEvent evt) {
                        replaceAllBActionPerformed(evt);
                  }
            });
            jToolBar2.add(replaceAllB);

            bottomContainer.add(jToolBar2, java.awt.BorderLayout.NORTH);
            jToolBar2.setVisible(false);

            output.setFloatable(false);
            output.setRollover(true);
            output.setToolTipText("Output");
            output.setMaximumSize(new java.awt.Dimension(32780, 100));
            output.setName("Output"); // NOI18N
            output.setPreferredSize(new java.awt.Dimension(113, 100));

            outputT.setColumns(20);
            outputT.setRows(5);
            jScrollPane2.setViewportView(outputT);

            jTabbedPane1.addTab("Output", jScrollPane2);

            output.add(jTabbedPane1);

            bottomContainer.add(output, java.awt.BorderLayout.CENTER);

            linecurL.setText("Total Line: 50; Current Line: 25; Character: 10;");
            bottomContainer.add(linecurL, java.awt.BorderLayout.PAGE_END);

            getContentPane().add(bottomContainer,
java.awt.BorderLayout.PAGE_END);

            pack();
      }// </editor-fold>//GEN-END:initComponents

private void openBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_openBActionPerformed
      JFileChooser jfc = new JFileChooser();
      int response = jfc.showOpenDialog(this);
      if (response == JFileChooser.APPROVE_OPTION) {
            try {
                  file = jfc.getSelectedFile();//future check
                  FileInputStream fis = new FileInputStream(file);
                  byte b[] = new byte[fis.available()];
                  fis.read(b);
                  String str = new String(b);
                  area = new Area();
                  area.setFont("Aerial", fontStyleC.getSelectedIndex(),
Integer.parseInt(String.valueOf(fontSizeC.getSelectedItem())));
                  area.getArea().setText(str);
                  String name = file.toString();
                  addTab(name, area);
                  area.setEdited(false);
                  area.setSaved(true);
            } catch (Exception e) {
                  JOptionPane.showMessageDialog(this, "File Cannot Open");
            }
      }
}//GEN-LAST:event_openBActionPerformed

private void newBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_newBActionPerformed
      area = new Area();
      area.setFont("Aerial", fontStyleC.getSelectedIndex(),
Integer.parseInt(String.valueOf(fontSizeC.getSelectedItem())));
      addTab("Untitled", area);
}//GEN-LAST:event_newBActionPerformed

private void aboutBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_aboutBActionPerformed
      visible = !visible;
      aboutWin.setLocation(-aboutWin.getWidth() / 2 + this.getWidth() / 2, this.getY()
- aboutWin.getHeight() / 2 + this.getHeight() / 2);
      aboutWin.setVisible(visible);
}//GEN-LAST:event_aboutBActionPerformed

private void saveBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_saveBActionPerformed
      save();
}//GEN-LAST:event_saveBActionPerformed

private void closeBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_closeBActionPerformed
      area = (Area) workArea.getSelectedComponent();
      if (area != null) {
            if (area.isEdited()) {
                  int response = JOptionPane.showConfirmDialog(rootPane, "Do you
want to Save Changes");
                  if (JOptionPane.YES_OPTION == response) {
                        save();
                workArea.removeTabAt(workArea.getSelectedIndex());
            } else if (JOptionPane.NO_OPTION == response) {
                workArea.removeTabAt(workArea.getSelectedIndex());
            }
        } else {
            workArea.removeTabAt(workArea.getSelectedIndex());
        }
    } else {
        JOptionPane.showMessageDialog(rootPane, "There is no any opened
document!");
    }
}//GEN-LAST:event_closeBActionPerformed

private void cutBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_cutBActionPerformed
    area = (Area) workArea.getSelectedComponent();
    area.getArea().cut();
}//GEN-LAST:event_cutBActionPerformed

private void copyBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_copyBActionPerformed
    area = (Area) workArea.getSelectedComponent();
    area.getArea().copy();
}//GEN-LAST:event_copyBActionPerformed

private void pasteBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_pasteBActionPerformed
    area = (Area) workArea.getSelectedComponent();
    area.getArea().paste();
}//GEN-LAST:event_pasteBActionPerformed

private void findNBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_findNBActionPerformed
    find(findT.getText(), pos, 1);
}//GEN-LAST:event_findNBActionPerformed

private void findPBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_findPBActionPerformed
    find(findT.getText(), pos, -1);
}//GEN-LAST:event_findPBActionPerformed

private void workAreaMouseClicked(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_workAreaMouseClicked
    area = (Area) workArea.getSelectedComponent();
    String size = String.valueOf(area.getFont().getSize());
    fontStyleC.setSelectedIndex(area.getFont().getStyle());
    fontSizeC.setSelectedItem(size);
}//GEN-LAST:event_workAreaMouseClicked
```

```
private void findTKeyReleased(java.awt.event.KeyEvent evt)
{//GEN-FIRST:event_findTKeyReleased
    if (area.isHindi()) {
        area.getArea().getHighlighter().removeAllHighlights();
        findAll(Translator.toHindi(findT.getText()), 0);
    } else {
        area.getArea().getHighlighter().removeAllHighlights();
        findAll(findT.getText(), 0);
    }
}//GEN-LAST:event_findTKeyReleased

private void readUrlBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_readUrlBActionPerformed
    new Thread() {
        @Override
        public void run() {
            try {
                String path = JOptionPane.showInputDialog("Enter Url");
                if (!path.startsWith("http://")) {
                    path = "http://" + path;
                }
                new Thread() {
                    @Override
                    public void run() {
                        int i = 0;
                        pbar.setStringPainted(true);
                        while (!taskDone) {
                            if (i == 100) {
                                i = 0;
                            }
                            pbar.setValue(i);
                            i += 5;
                            try {
                                Thread.sleep(50);
                            } catch (InterruptedException e) {
                            }
                        }
                        pbar.setValue(100);
                        pbar.setStringPainted(false);
                        taskDone = false;
                    }
                }.start();
                URL url = new URL(path);
                BufferedReader br = new BufferedReader(new
InputStreamReader(url.openStream()));
                String read = null;
                String content = "";
                while ((read = br.readLine()) != null) {
                    content = content + read + "\n";
                }
                br.close();
                area = new Area();
                area.getArea().setText(content);
                workArea.addTab(url.toString(), area);
                workArea.setSelectedIndex(workArea.getTabCount() - 1);
                taskDone = true;
            } catch (Exception ex) {
                taskDone = true;
                JOptionPane.showMessageDialog(null, ex);
            }
        }
    }.start();
}//GEN-LAST:event_readUrlBActionPerformed

private void showReplaceActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_showReplaceActionPerformed
    visible = !visible;
    jToolBar2.setVisible(visible);
    replaceWhatT.setText(findT.getText());
}//GEN-LAST:event_showReplaceActionPerformed

private void replaceBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_replaceBActionPerformed
    String string = area.getArea().getText();
    string = string.replaceFirst(replaceWhatT.getText(), replaceWithT.getText());
    area.getArea().setText(string);
}//GEN-LAST:event_replaceBActionPerformed

private void replaceAllBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_replaceAllBActionPerformed
    String string = area.getArea().getText();
    string = string.replaceAll(replaceWhatT.getText(), replaceWithT.getText());
    area.getArea().setText(string);
}//GEN-LAST:event_replaceAllBActionPerformed

private void fontSizeCActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_fontSizeCActionPerformed
    try {
        int size = Integer.parseInt(String.valueOf(fontSizeC.getSelectedItem()));
        int style = fontStyleC.getSelectedIndex();
        area.setFont("Aerial", style, size);
    } catch (Exception e) {
        fontSizeC.setSelectedIndex(2);
    }
}//GEN-LAST:event_fontSizeCActionPerformed

private void fontStyleCActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_fontStyleCActionPerformed
    int size = Integer.parseInt((String.valueOf(fontSizeC.getSelectedItem())));
    int style = fontStyleC.getSelectedIndex();
    area.setFont("Aerial", style, size);

}//GEN-LAST:event_fontStyleCActionPerformed

    private void compileBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_compileBActionPerformed
        try {
            save();
            if (area.isSaved()) {

if(workArea.getTitleAt(workArea.getSelectedIndex()).lastIndexOf(".java")>0){
                Runtime runtime = Runtime.getRuntime();
                runtime.exec("javac " +
workArea.getTitleAt(workArea.getSelectedIndex()));
                outputT.setText("Compiled Successfully");
                }else{
                    outputT.setText("File Should be of .java extension");
                }
            } else {
                outputT.setText("Cannot compile unsaved file");
            }
        } catch (Exception ex) {
            outputT.setText("Error in Compilation! " + ex);

        }
    }//GEN-LAST:event_compileBActionPerformed

    private void runBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_runBActionPerformed
        save();
        new Thread() {
            @Override
            public void run() {
                try {
                    Runtime rt = Runtime.getRuntime();
                    String location =
workArea.getTitleAt(workArea.getSelectedIndex());
                    String fl = location.substring(location.lastIndexOf("\\") + 1,
location.lastIndexOf("."));
                    location = location.substring(0, location.lastIndexOf("\\"));
                    Process pr = rt.exec("java -classpath \"" + location + "\" " +
fl);
                    BufferedReader br = new BufferedReader(new
InputStreamReader(pr.getInputStream()));
                    String line;
                    outputT.setText("");
                    while ((line = br.readLine()) != null) {
                        outputT.setText(outputT.getText() + line + "\n");
                    }
                } catch (IOException ex) {
                    outputT.setText(ex.toString());
                }
            }
        }.start();

    }//GEN-LAST:event_runBActionPerformed

    private void langBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_langBActionPerformed

        if (area.isHindi()) {
            langB.setToolTipText("English");
            area.setHindi(false);
            hmapB.setEnabled(false);
            langB.setIcon(new
ImageIcon(getClass().getResource("/img/abc.png")));
        } else {
            langB.setToolTipText("Hindi");
            area.setHindi(true);
            hmapB.setEnabled(true);
            langB.setIcon(new ImageIcon(getClass().getResource("/img/a.png")));
        }
    }//GEN-LAST:event_langBActionPerformed

    private void hmapBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_hmapBActionPerformed
        if (hMap == null) {
            hMap = new HindiMap();
        }
        if (hMap.isVisible()) {
            hMap.setVisible(false);
        } else {
            hMap.setVisible(true);
        }
    }//GEN-LAST:event_hmapBActionPerformed

    private void showOutputBActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_showOutputBActionPerformed
        if (output.isVisible()) {
            output.setVisible(false);
        } else {
            output.setVisible(true);
        }
    }//GEN-LAST:event_showOutputBActionPerformed

    public void find(String search, int pos, int d) {
        area.getArea().getHighlighter().removeAllHighlights();
        if (d == 1) {
            pos = area.getArea().getText().replaceAll("\n", "").indexOf(search,
pos);
        }//replaceAll to remove new lines character
        if (d == -1) {
            pos = area.getArea().getText().replaceAll("\n", "").lastIndexOf(search,
pos);
        }
        if (pos >= 0) {
            try {
                Highlighter h = area.getArea().getHighlighter();
                h.addHighlight(pos, pos + search.length(), new
DefaultHighlighter.DefaultHighlightPainter(Color.GREEN));
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(area.getArea(), ex);
            }
            if (d == -1) {
                this.pos = pos - search.length();
            }
            if (d == 1) {
                this.pos = pos + search.length();
            }
        }
    }

    public void findAll(String search, int pos) {
        pos = area.getArea().getText().replaceAll("\n", "").indexOf(search,
pos);//replaceAll to remove new lines character
        if (pos >= 0 && search.length() > 0) {
            try {
                Highlighter h = area.getArea().getHighlighter();
                h.addHighlight(pos, pos + search.length(), new
DefaultHighlighter.DefaultHighlightPainter(Color.GREEN));
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(area.getArea(), ex);
            }
            pos += search.length();
            if (pos < area.getArea().getText().length()) {
                findAll(search, pos);
            }//recursion
        }
    }

    public void save() {

        JFileChooser jfc = new JFileChooser();
        area = (Area) workArea.getSelectedComponent();
        if (!area.isSaved()) {
            int response = jfc.showSaveDialog(this);
            if (response == JFileChooser.APPROVE_OPTION) {
                file = jfc.getSelectedFile();
                if (file.exists()) {
                    response = JOptionPane.showConfirmDialog(this, "File allready exists, Dou you want to overrite");
                    if (response == JOptionPane.YES_OPTION) {
                        write(area);
                    } else if (response == JOptionPane.NO_OPTION) {
                        save();

                    }
                } else if (!file.exists()) {
                    write(area);
                }
                area.setSaved(true);
            }

        } else {
            write(area);
        }


    }

    private void write(Area area) {
        try {
            String str = area.getArea().getText();
            pw = new PrintWriter(new FileOutputStream(file));
            pw.write(str);
            pw.flush();
            String name = file.toString();
            //name = name.substring(name.lastIndexOf("\\") + 1, name.length());
            workArea.setTitleAt(workArea.getSelectedIndex(), name);
            area.setEdited(false);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Cannot Save");
        }
    }

    private void addTab(String title, Area area) {
        workArea.addTab(title, area);
        workArea.setSelectedIndex(workArea.getTabCount() - 1);
    }

    boolean taskDone = false;
    boolean running = true;
    private HindiMap hMap;
    private PrintWriter pw;
    private File file;
    private Area area;
    private int pos;
    private boolean visible;
```

```
private About aboutWin = new About();
boolean closeAll = false;
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton aboutB;
private javax.swing.JPanel bottomContainer;
private javax.swing.JButton closeB;
private javax.swing.JButton compileB;
private javax.swing.JButton copyB;
private javax.swing.JButton cutB;
private javax.swing.Box.Filler filler1;
private javax.swing.Box.Filler filler2;
private javax.swing.JButton findNB;
private javax.swing.JButton findPB;
private javax.swing.JTextField findT;
private javax.swing.JComboBox fontSizeC;
private javax.swing.JComboBox fontStyleC;
private javax.swing.JButton hmapB;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JToolBar.Separator jSeparator1;
private javax.swing.JToolBar.Separator jSeparator10;
private javax.swing.JToolBar.Separator jSeparator2;
private javax.swing.JToolBar.Separator jSeparator3;
private javax.swing.JToolBar.Separator jSeparator4;
private javax.swing.JToolBar.Separator jSeparator5;
private javax.swing.JToolBar.Separator jSeparator6;
private javax.swing.JToolBar.Separator jSeparator7;
private javax.swing.JToolBar.Separator jSeparator8;
private javax.swing.JToolBar.Separator jSeparator9;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JToolBar jToolBar2;
private javax.swing.JButton langB;
private javax.swing.JLabel linecurL;
private javax.swing.JButton newB;
private javax.swing.JButton openB;
private javax.swing.JToolBar output;
private javax.swing.JTextArea outputT;
private javax.swing.JButton pasteB;
private javax.swing.JProgressBar pbar;
private javax.swing.JButton readUrlB;
private javax.swing.JButton replaceAllB;
private javax.swing.JButton replaceB;
private javax.swing.JTextField replaceWhatT;
private javax.swing.JTextField replaceWithT;
private javax.swing.JButton runB;
private javax.swing.JButton saveB;
private javax.swing.JButton showOutputB;
private javax.swing.JButton showReplace;
private javax.swing.JToolBar topToolbar;
```

```
    private javax.swing.JTabbedPane workArea;
    // End of variables declaration//GEN-END:variables

    @Override
    public void windowOpened(WindowEvent e) {
    }

    @Override
    public void windowClosing(WindowEvent e) {
        closeAll = true;
        running = false;
    }

    @Override
    public void windowClosed(WindowEvent e) {
        System.exit(0);
    }

    @Override
    public void windowIconified(WindowEvent e) {
    }

    @Override
    public void windowDeiconified(WindowEvent e) {
    }

    @Override
    public void windowActivated(WindowEvent e) {
    }

    @Override
    public void windowDeactivated(WindowEvent e) {
    }
}
```

Main.java

```
package jeditor;

import java.awt.EventQueue;

/**
 *
 * @author Harshit Agarwal
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {

            @Override
            public void run() {
                new Editor().setVisible(true);
            }
        });
    }
}
```

Translator.java

```
package jeditor;

import java.io.UnsupportedEncodingException;

/**
 *
 * @author Harshit Agarwal
 */
public class Translator {

    public static String toEnglish(String hindi) {
        String english = hindi;
        if (english != null) {

            english = english.replaceAll("\u0915\u094D\u0937","kSh");
            english = english.replaceAll("\u091C\u094D\u091E\u093E","GYa");
            english = english.replaceAll("\u091C\u094D\u091E","GY");

            english = english.replaceAll("\u0902",".n");
            english = english.replaceAll("\u0901",".N");
            english = english.replaceAll("\u0903",".h");

            english = english.replaceAll("\u0916", "kh");
            english = english.replaceAll("\u0915", "k");
            english = english.replaceAll("\u0918", "gh");
            english = english.replaceAll("\u0917", "g");
            english = english.replaceAll("\u0919", "~N");
            english = english.replaceAll("\u091B", "chh");
            english = english.replaceAll("\u091A", "ch");
            english = english.replaceAll("\u091D", "jh");
            english = english.replaceAll("\u091C", "j");
            english = english.replaceAll("\u091E", "~n");
            english = english.replaceAll("\u0920", "Th");
            english = english.replaceAll("\u091F", "T");
            english = english.replaceAll("\u0922", "Dh");
            english = english.replaceAll("\u0921", "D");
            english = english.replaceAll("\u0923", "N");
            english = english.replaceAll("\u0925", "th");
            english = english.replaceAll("\u0924", "t");
            english = english.replaceAll("\u0927", "dh");
            english = english.replaceAll("\u0926", "d");
            english = english.replaceAll("\u0928", "n");
            english = english.replaceAll("\u092B", "ph");
            english = english.replaceAll("\u092A", "p");
            english = english.replaceAll("\u092D", "bh");
            english = english.replaceAll("\u092C", "b");
            english = english.replaceAll("\u092E", "m");
            english = english.replaceAll("\u092F", "y");
            english = english.replaceAll("\u0930", "r");
            english = english.replaceAll("\u0932", "l");
            english = english.replaceAll("\u0923", "L");
            english = english.replaceAll("\u0935", "v");
            english = english.replaceAll("\u0935", "w");
            english = english.replaceAll("\u0937", "shh");
            english = english.replaceAll("\u0936", "sh");
            english = english.replaceAll("\u0938", "s");
            english = english.replaceAll("\u0939", "h");

            english = english.replaceAll("\u094D", "`");
            english = english.replaceAll("\u0906", "AA");
            english = english.replaceAll("\u0948", "ai");
            english = english.replaceAll("\u0910", "AI");
            english = english.replaceAll("\u093F", "i");
            english = english.replaceAll("\u0907", "I");
            english = english.replaceAll("\u0940", "ee");
            english = english.replaceAll("\u0908", "EE");
            english = english.replaceAll("\u0941", "u");
            english = english.replaceAll("\u0909", "U");
            english = english.replaceAll("\u0942", "oo");
            english = english.replaceAll("\u090A", "OO");
            english = english.replaceAll("\u090B", "RRi");
            english = english.replaceAll("\u090C", "LLi");
            english = english.replaceAll("\u0947", "e");
            english = english.replaceAll("\u090F", "E");
```

```
        english = english.replaceAll("\u094B", "o");
        english = english.replaceAll("\u0913", "O");
        english = english.replaceAll("\u094C", "au");
        english = english.replaceAll("\u0914", "AU");
        english = english.replaceAll("\u0905\u0902", "aM");
        english = english.replaceAll("\u0905\u0903", "aH");
        english = english.replaceAll("\u093E", "a");
        english = english.replaceAll("\u0905", "A");
    }
    return english;
}

public static String toHindi(String english) {
    String hindi = toEnglish(english);

    if (hindi != null) {

        hindi = hindi.replaceAll("kSha","\u0915\u094D\u0937\u093E");
        hindi = hindi.replaceAll("kSh","\u0915\u094D\u0937");
        hindi = hindi.replaceAll("GYa","\u091C\u094D\u091E\u093E");
        hindi = hindi.replaceAll("GY","\u091C\u094D\u091E");

        hindi = hindi.replaceAll("kh", "\u0916");
        hindi = hindi.replaceAll("k", "\u0915");
        hindi = hindi.replaceAll("gh", "\u0918\u094D");
        hindi = hindi.replaceAll("g", "\u0917");
        hindi = hindi.replaceAll("~N", "\u0919");
        hindi = hindi.replaceAll("chh", "\u091B");
        hindi = hindi.replaceAll("ch", "\u091A");
        hindi = hindi.replaceAll("jh", "\u091D");
        hindi = hindi.replaceAll("j", "\u091C");
        hindi = hindi.replaceAll("~n", "\u091E");
        hindi = hindi.replaceAll("Th", "\u0920");
        hindi = hindi.replaceAll("T", "\u091F");
        hindi = hindi.replaceAll("Dh", "\u0922");
        hindi = hindi.replaceAll("D", "\u0921");
        hindi = hindi.replaceAll("th", "\u0925");
        hindi = hindi.replaceAll("t", "\u0924");
        hindi = hindi.replaceAll("dh", "\u0927");
        hindi = hindi.replaceAll("d", "\u0926");
        //hindi = hindi.replaceAll("n", "\u0928");
        hindi = hindi.replaceAll("ph", "\u092B");
        hindi = hindi.replaceAll("f", "\u092B");
        hindi = hindi.replaceAll("p", "\u092A");
        hindi = hindi.replaceAll("bh", "\u092D");
        hindi = hindi.replaceAll("b", "\u092C");
        hindi = hindi.replaceAll("m", "\u092E");
        hindi = hindi.replaceAll("y", "\u092F");
        hindi = hindi.replaceAll("r", "\u0930");
        //hindi =    hindi.replaceAll("RRA", "\u0931");
    hindi = hindi.replaceAll("l", "\u0932");
    hindi = hindi.replaceAll("L", "\u0933");
    //hindi =    hindi.replaceAll("LLLA", "\u0934");
    hindi = hindi.replaceAll("v", "\u0935");
    hindi = hindi.replaceAll("w", "\u0935");
    hindi = hindi.replaceAll("shh", "\u0937");
    hindi = hindi.replaceAll("sh", "\u0936");
    hindi = hindi.replaceAll("s", "\u0938");

    hindi = hindi.replaceAll("\\.n", "\u0902");
    hindi = hindi.replaceAll("\\.N", "\u0901");
    hindi = hindi.replaceAll("\\.h", "\u0903");

    hindi = hindi.replaceAll("n", "\u0928");
    hindi = hindi.replaceAll("N", "\u0923");
    hindi = hindi.replaceAll("h", "\u0939");

    hindi = hindi.replaceAll("`", "\u094D");
    hindi = hindi.replaceAll("AA", "\u0906");
    hindi = hindi.replaceAll("ai", "\u0948");
    hindi = hindi.replaceAll("AI", "\u0910");
    hindi = hindi.replaceAll("ee", "\u0940");
    hindi = hindi.replaceAll("EE", "\u0908");
    hindi = hindi.replaceAll("oo", "\u0942");
    hindi = hindi.replaceAll("OO", "\u090A");
    hindi = hindi.replaceAll("RRi", "\u090B");
    hindi = hindi.replaceAll("Ri", "\u0943");
    hindi = hindi.replaceAll("LLi", "\u090C");
    hindi = hindi.replaceAll("i", "\u093F");
    hindi = hindi.replaceAll("I", "\u0907");
    hindi = hindi.replaceAll("e", "\u0947");
    hindi = hindi.replaceAll("E", "\u090F");
    hindi = hindi.replaceAll("au", "\u094C");
    hindi = hindi.replaceAll("AU", "\u0914");
    hindi = hindi.replaceAll("ou", "\u094C");
    hindi = hindi.replaceAll("OU", "\u0914");
    hindi = hindi.replaceAll("u", "\u0941");
    hindi = hindi.replaceAll("U", "\u0909");
    hindi = hindi.replaceAll("o", "\u094B");
    hindi = hindi.replaceAll("O", "\u0913");
    hindi = hindi.replaceAll("aM", "\u0905\u0902");
    hindi = hindi.replaceAll("aH", "\u0905\u0903");
    hindi = hindi.replaceAll("a", "\u093E");
    hindi = hindi.replaceAll("A", "\u0905");
}
try {
    return new String(hindi.getBytes("UTF-8"));
} catch (UnsupportedEncodingException ex) {
    return null;
} catch (NullPointerException ex) {
```

```
            return null;
        }
    }
}
```

THANK YOU !