**By :- Abhishek Gupta (501824)**

**“Java Project”**

**\***

**“File Comipler”**

**Introduction :-**

This Application Helps Group of People to Upload Files and Folders and Automate the task of Making Collection ,Searching and sorting . It Allows user to upload a document which is then uploaded to a MEGA FOLDER in DROPBOX (by using dropbox API).

The Main Feature which make it differ out from Manually Uploading is that even an illiterate person or non IT Professional can use this application with ease without worring about accidentally using wrong URL or Don’t know the proper procedure to upload files.

**Design :-**

The Main Purpose which Lead to Designing of this application is to AUTOMATE the task of Humans of manually sorting , searching and assembling files given as input and also to remove the possibility that once given access to folder the Individual can Delete the Folder and Files of other Individual which can lead to Serious Problem.

**Features :-**

* Cross Platform Application.
* Reduces Man Labour.
* Reduces Chances Of Errors and Wrong Format.
* Much Safer than Normal Dropbox File Access that is the members are not allowed to delete file uploaded by others.

**Code :-**

**Main\_Interface.java ==>**

import com.dropbox.core.DbxException;

import com.dropbox.core.DbxRequestConfig;

import com.dropbox.core.v2.DbxClientV2;

import com.dropbox.core.v2.files.FileMetadata;

import fileuploader.Interface;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStream;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

public class test extends javax.swing.JFrame {

private static final String ACCESS\_TOKEN = "ixasUYYT0iAAAAAAAAAADtVB32RQvi6wJLlU4pjks\_xRRZMkjLqGnyfhR65T\_H61";

public test() {

initComponents();

}

private void attachActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser chooser = new JFileChooser();

chooser.showOpenDialog(null);

File f = chooser.getSelectedFile();

String filepath = f.getAbsolutePath();

path.setText(filepath);

}

private void submitActionPerformed(java.awt.event.ActionEvent evt) {

try {

int flag =0;

Path s1 = Paths.get(path.getText().toString());

String s = s1.toString();

System.out.println(s);

// String s = path.getText().toString();

File filepath;

filepath = new File(s);

if (path.getText().toString() != null && filepath.exists() ) {

DbxRequestConfig config = DbxRequestConfig.newBuilder("dropbox/JavaProject").build();

DbxClientV2 client = new DbxClientV2(config,ACCESS\_TOKEN);

// Uploading

String platform = System.getProperty("os.name");

if (platform.startsWith("Linux")) {

for(int i=0;i<s.length();i++) {

if (s.charAt(i)=='/')

flag = i;

}

}

else if (platform.startsWith("Windows"))

{

for(int i=0;i<s.length();i++) {

if (s.charAt(i)=='\\')

flag = i;

}

}

else {

JOptionPane.showMessageDialog(null,"System Not Supported");

}

String file = s.substring(flag+1);

try (InputStream in = new FileInputStream(filepath)) {

FileMetadata metadata = client.files().uploadBuilder("/" + file)

.uploadAndFinish(in);

java.awt.EventQueue.invokeLater(() -> {

new Interface().setVisible(true);

});

JOptionPane.showMessageDialog(null,"File Uploded Successfully");

System.exit(0);

} catch (FileNotFoundException ex) {

Logger.getLogger(Interface.class.getName()).log(Level.SEVERE, null, ex);

} catch (DbxException ex) {

Logger.getLogger(Interface.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(Interface.class.getName()).log(Level.SEVERE, null, ex);

}

}

else {

JOptionPane.showMessageDialog(null,"Please see The help menu for help, if issuse not resolved then contact the maker");

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null,e);

System.exit(0);

}

}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {

this.setVisible(false);

new help().setVisible(true);

}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {

this.setVisible(false);

new help().setVisible(true);

}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

this.setVisible(false);

new About().setVisible(true);

}

private void pathActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(test.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(test.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(test.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(test.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new test().setVisible(true);

}

});

}

**Help.java Code :-**

public class help extends javax.swing.JFrame {

public help() {

initComponents();

}

private void gobackActionPerformed(java.awt.event.ActionEvent evt) {

this.setVisible(false);

new test().setVisible(true);

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new help().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JLabel background;

private javax.swing.JButton goback;

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;

// End of variables declaration

}

**About.java Code :-**

public class About extends javax.swing.JFrame {

public About() {

initComponents();

}

private void gobackActionPerformed(java.awt.event.ActionEvent evt) {

this.setVisible(false);

new test().setVisible(true);

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new About().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JLabel background;

private javax.swing.JButton goback;

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;

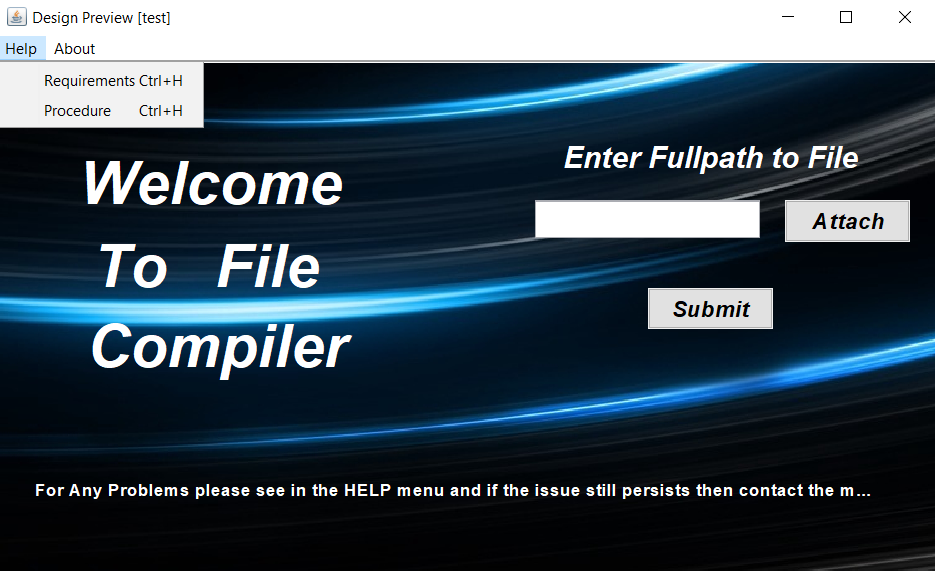
// End of variables declaration

}

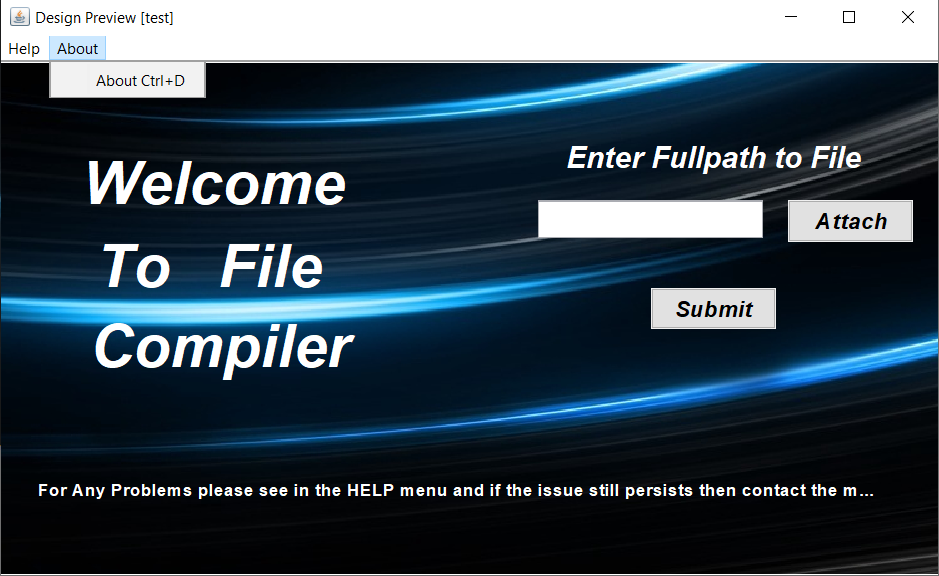
**Output :-**

**Main Interface (test.java) :-**

**Help Menu ==>**

****

**About Menu ==>**

****

**HELP PAGE (Help.java) :-**

****

**ABOUT MENU (About.java) :-**

****

**References :-**

* **Stack OverFlow**
* **Dropbox Community**