


## Declaration of Authenticity

A critical aspect of any assignment is *authenticity*. Because you are completing much of the work for the assignments *unsupervised*, the examiner must be convinced that it is all your own work. For this reason, you must complete the *Declaration of Authenticity* provided in the study guide and have it counter-signed by your manager, mentor, or lecturer.

	<p>The declaration of authenticity is a legal document, and if found that you have made a false declaration, then not only will your results be declared null and void, but you could also have criminal charges brought against you. It is not worth taking the risk!</p>
---	--

Please complete the declaration of authenticity below for all assignments:

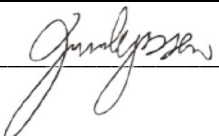


### DECLARATION OF AUTHENTICITY

I Jonathan Van Eyssen hereby  
(FULL NAME)

declare that the contents of this assignment is entirely my own work with the exception of the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	Date

Signature:  Date: 04/09/2024

# Advanced Java Semester 2

## JD522

### Formative 1

Jonathan Van Eyssen 20231438

#### Main Window Code

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.StandardOpenOption;
import java.nio.file.attribute.BasicFileAttributes;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.util.HashMap;
import javax.swing.DefaultComboBoxModel;
```

```

public class MainF extends javax.swing.JFrame {

    public final String url = "jdbc:sqlite:TaskStorageDB";

    ArrayList<TaskClass> TasksList = new ArrayList<TaskClass>(); //this arraylist is for tasks that are
    read from the Database

    ArrayList<TaskClass> ImportList = new ArrayList<TaskClass>(); //this list is for tasks read from the
    file

    HashMap<String, ArrayList<TaskClass>> categoryHashes = new HashMap<String,
    ArrayList<TaskClass>>();

    String messages;

    public void createDB() { //this makes the database and the tables

        File fileCheck = new File("taskstoragedb"); //checks if the database exists.

        if (!fileCheck.exists()) {

            try {

                Connection connect = DriverManager.getConnection(url);

                if (connect != null) {

                    DatabaseMetaData meta = connect.getMetaData(); //this attempts to get the meta data finds
                    no database and then creates it.

                    JOptionPane.showMessageDialog(rootPane, "Database Made");

                    String[] CreateTblS = new String[]{"CREATE TABLE TASKS"
                        + "(ID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,"
                        + "NAME VARCHAR(50),"
                        + "DESCRIPTION VARCHAR(200),"
                        + "STATUS VARCHAR (10),"
                        + "CATEGORY VARCHAR (10)); ",
                        "CREATE TABLE CATEGORIES"}

```

```

+ "(ID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,"
+ "CATEGORY_NAME VARCHAR(20));",

"INSERT INTO CATEGORIES (CATEGORY_NAME)"
+ "VALUES ('Personal'); ",

"INSERT INTO CATEGORIES (CATEGORY_NAME)"
+ "VALUES ('Work');",

"INSERT INTO CATEGORIES (CATEGORY_NAME)"
+ "VALUES ('Home');",

"INSERT INTO CATEGORIES (CATEGORY_NAME)"
+ "VALUES ('School');",

"INSERT INTO TASKS(NAME,DESCRIPTION,STATUS,CATEGORY)"
+ "VALUES ('App','Fix app bugs','Started','Work');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('App', 'Fix app bugs', 'Started', 'Work');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('Website Redesign', 'Update the website to a modern look and feel', 'In
Progress', 'Home');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('Marketing Campaign', 'Develop a new marketing campaign for the product
launch', 'Stalled', 'School');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('Client Meeting', 'Prepare for a meeting with a potential new client', 'Started',
'Work');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('Code Review', 'Review the code for the new feature', 'Completed', 'Work');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('Financial Report', 'Prepare the monthly financial report', 'Stalled', 'Home');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('Research Paper', 'Write a research paper on AI ethics', 'In Progress', 'School');",

"INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
+ "VALUES ('Team Building', 'Organize a team-building activity', 'Stalled', 'Work');",

```

```

        "INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
        + "VALUES ('Bug Fix', 'Fix the critical bug in the production environment', 'In Progress',
'Work');";

        "INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
        + "VALUES ('Content Creation', 'Write blog posts and social media content', 'Stalled',
'Personal');";

        "INSERT INTO TASKS (NAME, DESCRIPTION, STATUS, CATEGORY)"
        + "VALUES ('Server Migration', 'Migrate the server to a new data center', 'Started', 'Home');";

Statement CreateTable = connect.createStatement();

for (String tablecre : CreateTbls) {
    boolean res = CreateTable.execute(tablecre);
    if (!res) {
        messages = "The operations were successful \n";
    } else {
        JOptionPane.showMessageDialog(rootPane, "Tables NOT Made");
    }
}

JOptionPane.showMessageDialog(rootPane, messages);

CreateTable.close();
connect.close();
}
} catch (SQLException e) {
    JOptionPane.showMessageDialog(rootPane, e.getMessage());
}
}
}

```

```

public void DatafromDB() { //this collects the data from the database and displays it in the table

    File alive = new File("taskstoragedb");

    if (alive.exists()) {

        try {

            Connection connect = DriverManager.getConnection(url);

            String AllData = "SELECT * FROM TASKS";

            Statement collectData = connect.createStatement();

            ResultSet SetResult = collectData.executeQuery(AllData);


            DefaultTableModel ResultTable = (DefaultTableModel) ResTab.getModel();

            ResultTable.setRowCount(0);


            while (SetResult.next()) {

                TaskClass taskNew = new TaskClass(Integer.parseInt(SetResult.getString("ID")),
                SetResult.getString("Name"), SetResult.getString("Description"), SetResult.getString("Status"),
                SetResult.getString("Category"));

                TasksList.add(taskNew); //this adds tasks created and constructed to the arraylist from the
                database

                ResultTable.addRow(new Object[]{SetResult.getString("Name"),
                SetResult.getString("Description"), SetResult.getString("Status"), SetResult.getString("Category")});

            }


            SetResult.close();

            collectData.close();

            connect.close();


        } catch (SQLException e) {

            JOptionPane.showMessageDialog(rootPane, e.getMessage());

        }

    }

}

```

public void Searching(String CatName, String UserValue) { //this is the method used to find data in the database and returns it to the table in the main window

```
File alive = new File("taskstoragedb");

if (alive.exists()) {

    try {

        Connection connect = DriverManager.getConnection(url);

        String search = "SELECT * FROM TASKS WHERE " + CatName + " LIKE '" + UserValue + "%'";

        Statement SEARCHING = connect.createStatement();

        ResultSet SetResult = SEARCHING.executeQuery(search);

        DefaultTableModel ResultTable = (DefaultTableModel) ResTab.getModel();

        ResultTable.setRowCount(0);

        while (SetResult.next()) {

            ResultTable.addRow(new Object[]{SetResult.getString("Name"),
SetResult.getString("Description"), SetResult.getString("Status"), SetResult.getString("Category")});

        }

        SetResult.close();

        SEARCHING.close();

        connect.close();

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(rootPane, e.getMessage());

    }

}

}
```

public void sorting(String ColName) { //this allows the user to sort the data by column name.

```
File alive = new File("taskstoragedb");

if (alive.exists()) {
```

```

try {
    Connection connect = DriverManager.getConnection(url);

    String order = "SELECT * FROM TASKS ORDER BY " + ColName + " ASC";

    Statement ordering = connect.createStatement();

    ResultSet SetResult = ordering.executeQuery(order);

    DefaultTableModel ResultTable = (DefaultTableModel) ResTab.getModel();

    ResultTable.setRowCount(0);

    while (SetResult.next()) {

        ResultTable.addRow(new Object[]{SetResult.getString("Name"),
SetResult.getString("Description"), SetResult.getString("Status"), SetResult.getString("Category")});

    }

    SetResult.close();

    ordering.close();

    connect.close();

} catch (SQLException e) {

    JOptionPane.showMessageDialog(rootPane, e.getMessage());

}

}
}

```

```

public void popCategoryCombo() {

    File alive = new File("taskstoragedb");

    if (alive.exists()) {

        try {

            Connection connect = DriverManager.getConnection(url);

            String combopop = "SELECT * FROM CATEGORIES ";

            Statement popcombo = connect.createStatement();

            ResultSet SetResult = popcombo.executeQuery(combopop);

```



```
DefaultComboBoxModel comboMod = (DefaultComboBoxModel) specCatCombo.getModel();
```

```
specCatCombo.removeAllItems();
```

```
TaskClass task;
```

```
while (SetResult.next()) {
```

```
    specCatCombo.addItem(SetResult.getString("CATEGORY_NAME"));
```

```
}
```

```
SetResult.close();
```

```
popcombo.close();
```

```
connect.close();
```

```
} catch (SQLException e) {
```

```
    JOptionPane.showMessageDialog(rootPane, e.getMessage());
```

```
}
```

```
}
```

```
}
```

```
public void popHasMap() {
```

```
    categoryHashes.clear();
```

```
    File alive = new File("taskstoragedb");
```

```
    if (alive.exists()) {
```

```
        try {
```

```
            Connection connect = DriverManager.getConnection(url);
```

```
            String combopop = "SELECT * FROM Categories";
```

```
            Statement popcombo = connect.createStatement();
```

```
            ResultSet SetResult = popcombo.executeQuery(combopop);
```

```
            while (SetResult.next()) {
```

```

        categoryHashes.put(SetResult.getString("CATEGORY_NAME"), new ArrayList<TaskClass>());

    }

    combopop = "SELECT * FROM TASKS";
    SetResult = popcombo.executeQuery(combopop);

    while (SetResult.next()) {

        categoryHashes.get(SetResult.getString("Category")).add(new
TaskClass(SetResult.getString("Name"), SetResult.getString("Description"),
SetResult.getString("Status"), SetResult.getString("Category")));

    }

    popcombo.close();
    SetResult.close();

} catch (SQLException e) {

    JOptionPane.showMessageDialog(rootPane, e.getMessage());

}

}

}

public void hasMapTablePop(String category) {

    DefaultTableModel ResultTable = (DefaultTableModel) ResTab.getModel();
    ResultTable.setRowCount(0);

    for (TaskClass task : categoryHashes.get(category)) {

```

```

        ResultTable.addRow(new Object[]{task.getName(), task.getDescription(), task.getStatus(),
task.getCategory()});

    }

}

```

```

public MainF() {

    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">

```

```

private void initComponents() {

```

```

    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    SearchTxt = new javax.swing.JTextField();
    jPanel2 = new javax.swing.JPanel();
    BtnImport = new javax.swing.JButton();
    jLabel3 = new javax.swing.JLabel();
    serbyCom = new javax.swing.JComboBox<>();
    jLabel5 = new javax.swing.JLabel();
    SerBtn = new javax.swing.JButton();
    jPanel3 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    ResTab = new javax.swing.JTable();

```

```
sortByCom = new javax.swing.JComboBox<>();
jLabel4 = new javax.swing.JLabel();
specCatCombo = new javax.swing.JComboBox<>();
catLab = new javax.swing.JLabel();
SortBtn = new javax.swing.JButton();
jPanel4 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
jPanel5 = new javax.swing.JPanel();
TxtBtn = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
jPanel6 = new javax.swing.JPanel();
CSVEpoBtn = new javax.swing.JButton();
jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
PropLab = new javax.swing.JLabel();
jPanel7 = new javax.swing.JPanel();
sec3Lab1 = new javax.swing.JLabel();
jPanel8 = new javax.swing.JPanel();
jLabel11 = new javax.swing.JLabel();
AddBtn = new javax.swing.JButton();
jPanel9 = new javax.swing.JPanel();
jLabel12 = new javax.swing.JLabel();
ChgBtn = new javax.swing.JButton();
jPanel10 = new javax.swing.JPanel();
jLabel13 = new javax.swing.JLabel();
DelBtn = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Task Manager");
```

```

setLocation(new java.awt.Point(0, 0));

setMaximumSize(new java.awt.Dimension(1086, 826));

setMinimumSize(new java.awt.Dimension(1086, 826));

setResizable(false);

addWindowListener(new java.awt.event.WindowAdapter() {

    public void windowActivated(java.awt.event.WindowEvent evt) {

        formWindowActivated(evt);

    }

    public void windowOpened(java.awt.event.WindowEvent evt) {

        formWindowOpened(evt);

    }

});

jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel2.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 24)); // NOI18N
jLabel2.setText("Search:");

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

BtnImport.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
BtnImport.setText("Import");
BtnImport.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BtnImportActionPerformed(evt);

    }

});

jLabel3.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

```

```

jLabel3.setText("Import from a file");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(15, 15, 15)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(71, 71, 71)
                .addComponent(BtnImport)
                .addGap(84, 84, 84)
            )
        )
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE, 30, Short.MAX_VALUE)
            .addGap(18, 18, 18)
            .addComponent(BtnImport)
            .addGap(14, 14, 14)
        )
);

serbyCom.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Task Name",
"Description ", "Status", "Category" }));

serbyCom.setSelectedIndex(-1);

```

```

jLabel5.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 24)); // NOI18N
jLabel5.setText("Search by:");

SerBtn.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
SerBtn.setText("Search");
SerBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SerBtnActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(19, 19, 19)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel2)
                    .addGap(48, 48, 48)
                    .addComponent(SearchTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 612,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel5)
                    .addGap(18, 18, 18)
                    .addComponent(serbyCom, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addComponent(SerBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26,
Short.MAX_VALUE)

        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(27, 27, 27)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(SearchTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(SerBtn, javax.swing.GroupLayout.DEFAULT_SIZE, 29, Short.MAX_VALUE)

.addGap(1, 1, 1))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(serbyCom, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel5)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap()

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```



```
.addContainerGap()
```

```
);
```

```
jPanel3.setBorder(javax.swing.BorderFactory.createEtchedBorder());
```

```
ResTab.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {
```

```
    },
```

```
    new String [] {
```

```
        "Name", "Description", "Status", "Category"
```

```
    }  
));
```

```
jScrollPane1.setViewportViewView(ResTab);
```

```
sortbyCom.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Task Name",  
"Description", "Status", "Category" }));
```

```
sortbyCom.setSelectedIndex(-1);
```

```
sortbyCom.addItemListener(new java.awt.event.ItemListener() {
```

```
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
```

```
        sortbyComItemStateChanged(evt);
```

```
    }  
});
```

```
sortbyCom.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        sortbyComActionPerformed(evt);
```

```
    }  
});
```

```
jLabel4.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 24)); // NOI18N
```

```
jLabel4.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 24)); // NOI18N
```

```

jLabel4.setText("Sort by:");

specCatCombo.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        specCatComboItemStateChanged(evt);
    }
});

specCatCombo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        specCatComboActionPerformed(evt);
    }
});

catLab.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
catLab.setText("Category");

SortBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
SortBtn.setText("Sort");
SortBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SortBtnActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(18, 18, 18)

```

```

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addComponent(jLabel4)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(sortbyCom, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(64, 64, 64)
        .addComponent(catLab)
        .addGap(18, 18, 18)
        .addComponent(specCatCombo, javax.swing.GroupLayout.PREFERRED_SIZE, 87,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(32, 32, 32)
        .addComponent(SortBtn))
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 985,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(17, Short.MAX_VALUE))
);

jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(16, 16, 16)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(sortbyCom, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel4)
    .addComponent(specCatCombo)
    .addComponent(catLab)
    .addComponent(SortBtn))

```

```

.addContainerGap(16, Short.MAX_VALUE))
);

jPanel4.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel6.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 24)); // NOI18N
jLabel6.setText("Export:");

jSeparator1.setOrientation(javax.swing.SwingConstants.VERTICAL);

jPanel5.setBorder(javax.swing.BorderFactory.createEtchedBorder());

TxtBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
TxtBtn.setText("Export");
TxtBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TxtBtnActionPerformed(evt);
    }
});

jLabel7.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel7.setText("Export to .txt");

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
jPanel5.setLayout(jPanel5Layout);
jPanel5Layout.setHorizontalGroup(
    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel5Layout.createSequentialGroup()
            .addGap(45, 45, 45)
            .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jLabel7)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup()

        .addComponent(TxtBtn)

        .addGap(11, 11, 11)))

        .addContainerGap(45, Short.MAX_VALUE))

);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup()

        .addGap(24, 24, 24)

        .addComponent(jLabel7)

        .addGap(18, 18, 18)

        .addComponent(TxtBtn)

        .addContainerGap(19, Short.MAX_VALUE))

);

jPanel6.setBorder(javax.swing.BorderFactory.createEtchedBorder());

CSVEpoBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
CSVEpoBtn.setText("Export");
CSVEpoBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CSVEpoBtnActionPerformed(evt);
    }
});

jLabel8.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel8.setText("Export to .csv");

```

```

javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
jPanel6.setLayout(jPanel6Layout);
jPanel6Layout.setHorizontalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel6Layout.createSequentialGroup()
            .addGap(41, 41, 41)
            .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel8)
                .addGroup(jPanel6Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(CSVEpoBtn)))
            .addGap(39, Short.MAX_VALUE))
    );
jPanel6Layout.setVerticalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel6Layout.createSequentialGroup()
            .addGap(23, 23, 23)
            .addComponent(jLabel8)
            .addGap(18, 18, 18)
            .addComponent(CSVEpoBtn)
            .addGap(20, Short.MAX_VALUE))
    );

jLabel9.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 24)); // NOI18N
jLabel9.setText("File Properties:");

PropLab.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N

```

```

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 77,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(31, 31, 31)
            .addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(44, 44, 44)
            .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel9)
                .addComponent(PropLab))
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel4Layout.createSequentialGroup()
                    .addGap(15, 15, 15)
                    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel4Layout.createSequentialGroup()
                            .addGap(15, 15, 15)
                            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 77,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(31, 31, 31)
                            .addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(44, 44, 44)
                            .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(jLabel9)
                            .addComponent(PropLab))
                        .addGroup(jPanel4Layout.createSequentialGroup()
                            .addGap(15, 15, 15)
                            .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 77,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(jPanel7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(31, 31, 31)
                            .addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(44, 44, 44)
                            .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(jLabel8)
                            .addComponent(PropLab))
                    ))
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

```

```

        .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addContainerGap(23, Short.MAX_VALUE))

    .addGroup(jPanel4Layout.createSequentialGroup())

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel4Layout.createSequentialGroup())

        .addComponent(jLabel9)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(PropLab))

        .addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 107,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(0, 0, Short.MAX_VALUE)))

);

jPanel7.setBorder(javax.swing.BorderFactory.createEtchedBorder());

sec3Lab1.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 24)); // NOI18N
sec3Lab1.setText("Add Change Delete:");

jPanel8.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel11.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel11.setText("Add a Task");
jLabel11.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

```



```

AddBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
AddBtn.setText("Add");
AddBtn.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
AddBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        AddBtnActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel8Layout = new javax.swing.GroupLayout(jPanel8);
jPanel8.setLayout(jPanel8Layout);
jPanel8Layout.setHorizontalGroup(
    jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel8Layout.createSequentialGroup()
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(jPanel8Layout.createSequentialGroup()
            .addGap(44, 44, 44)
            .addComponent(AddBtn)
            .addGap(48, Short.MAX_VALUE))
);
jPanel8Layout.setVerticalGroup(
    jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel8Layout.createSequentialGroup()
            .addGap(23, 23, 23)
            .addComponent(jLabel11)
            .addGap(18, 18, 18)

```

```

        .addComponent(AddBtn)

        .addContainerGap(23, Short.MAX_VALUE))

);

jPanel9.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel12.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel12.setText("Change a task");

ChgBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
ChgBtn.setText("Change");
ChgBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ChgBtnActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel9Layout = new javax.swing.GroupLayout(jPanel9);
jPanel9.setLayout(jPanel9Layout);
jPanel9Layout.setHorizontalGroup(
    jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel9Layout.createSequentialGroup()
            .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel9Layout.createSequentialGroup()
                    .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel9Layout.createSequentialGroup()
                            .addGap(38, 38, 38)
                            .addComponent(jLabel12))
                        .addGroup(jPanel9Layout.createSequentialGroup()
                            .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(jPanel9Layout.createSequentialGroup()
                                    .addGap(46, 46, 46)
                                    .addComponent(ChgBtn)))
                                .addGroup(jPanel9Layout.createSequentialGroup()
                                    .addContainerGap(34, Short.MAX_VALUE))

```

```

);

jPanel9Layout.setVerticalGroup(

    jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel9Layout.createSequentialGroup()

            .addGap(22, 22, 22)

            .addComponent(jLabel12)

            .addGap(18, 18, 18)

            .addComponent(ChgBtn)

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

);

jPanel10.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel13.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel13.setText("Delete a task");

DelBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
DelBtn.setText("Delete");
DelBtn.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        DelBtnActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel10Layout = new javax.swing.GroupLayout(jPanel10);
jPanel10.setLayout(jPanel10Layout);
jPanel10Layout.setHorizontalGroup(

    jPanel10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel10Layout.createSequentialGroup()

            .addContainerGap(44, Short.MAX_VALUE)

```

```

.addGroup(jPanel10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel10Layout.createSequentialGroup()
    .addComponent(DelBtn)
    .addGap(53, 53, 53))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel10Layout.createSequentialGroup()
    .addComponent(jLabel13)
    .addGap(41, 41, 41))))
);

jPanel10Layout.setVerticalGroup(
jPanel10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel10Layout.createSequentialGroup()
    .addGap(23, 23, 23)
    .addComponent(jLabel13)
    .addGap(18, 18, 18)
    .addComponent(DelBtn)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

javax.swing.GroupLayout jPanel7Layout = new javax.swing.GroupLayout(jPanel7);
jPanel7.setLayout(jPanel7Layout);
jPanel7Layout.setHorizontalGroup(
jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel7Layout.createSequentialGroup()
    .addGap(19, 19, 19)
    .addComponent(sec3Lab1)
    .addGap(69, 69, 69)
    .addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addGap(60, 60, 60)

        .addComponent(jPanel9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(60, 60, 60)

        .addComponent(jPanel10, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

jPanel7Layout.setVerticalGroup(

jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel7Layout.createSequentialGroup())

.addGap(18, 18, 18)

.addGroup(jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jPanel9, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jPanel10, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(sec3Lab1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addContainerGap(21, Short.MAX_VALUE))

);

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

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addGap(40, 40, 40)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

```

```

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jPanel7, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addContainerGap(22, Short.MAX_VALUE))

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGap(21, 21, 21)

            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addComponent(jPanel7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addContainerGap(12, Short.MAX_VALUE))

        );

pack();
} // </editor-fold>

```

```

private void formWindowOpened(java.awt.event.WindowEvent evt) {

```

```

PropLab.setText("No File Selected!");

sec3Lab1.setText("<HTML> Add <BR> Change <BR> Delete </HTML>");

createDB();

ImageIcon icon = new ImageIcon("src/FA2.png");//these two lines are used to change the icon on
the window.

this.setIconImage(icon.getImage());

popHasMap();
}

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

    AddF add = new AddF();

    add.setVisible(true);

}

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

    ChangeF change = new ChangeF();

    change.setVisible(true);

}

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

    DelF deletf = new DelF();

    deletf.setVisible(true);

}

private void formWindowActivated(java.awt.event.WindowEvent evt) {

    TasksList.clear();

    DatafromDB();

    specCatCombo.setVisible(false);

    catLab.setVisible(false);

    SortBtn.setVisible(false);

```

```
}
```

```
private void SerBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if (SearchTxt.getText().isBlank() || serbyCom.getSelectedIndex() == -1) { //error handling to ensure  
that no null values are passed.
```

```
        JOptionPane.showMessageDialog(rootPane, "Search or search by cannot be empty!");
```

```
    } else {
```

```
        if (serbyCom.getSelectedItem() == "Task Name") { //sets the column names and passes the user  
input to the method
```

```
            Searching("Name", SearchTxt.getText());
```

```
        } else if (serbyCom.getSelectedItem() == "Description") {
```

```
            Searching("Description", SearchTxt.getText());
```

```
        } else if (serbyCom.getSelectedItem() == "Status") {
```

```
            Searching("Status", SearchTxt.getText());
```

```
        } else if (serbyCom.getSelectedItem() == "Category") {
```

```
            Searching("Category", SearchTxt.getText());
```

```
        }
```

```
    }
```

```
    serbyCom.setSelectedIndex(-1); //resets the input fields in the window
```

```
    SearchTxt.setText("");
```

```
    sortbyCom.setSelectedIndex(-1);
```

```
}
```

```
private void sortbyComActionPerformed(java.awt.event.ActionEvent evt) {
```

```
}
```

```
private void TxtBtnActionPerformed(java.awt.event.ActionEvent evt) {
```



```

//this uses IO to write the data in the table to a text file.

try {
    FileOutputStream filestream = new FileOutputStream("TextFile.txt");
    for (int i = 0; i < TasksList.size(); i++) {
        TaskClass task = TasksList.get(i);

        String objectString = task.getName() + ";" + task.getDescription() + ";" + task.getStatus() + ";" +
task.getCategory();

        if (i < TasksList.size() - 1) {
            objectString += ",\n";
        }

        byte[] StringObject = objectString.getBytes();
        filestream.write(StringObject);
    }

    JOptionPane.showMessageDialog(rootPane, "File Made!");
} catch (Exception e) {
    JOptionPane.showMessageDialog(rootPane, e.getMessage());
}
}

```

```

private void CSVEpoBtnActionPerformed(java.awt.event.ActionEvent evt) {
    //this uses NIO to write the data from the table into a CSV file.

    Path filePath = Path.of("CSVFile.csv");

    try (var writer = Files.newBufferedWriter(filePath, StandardCharsets.UTF_8,
        StandardOpenOption.CREATE, StandardOpenOption.TRUNCATE_EXISTING)) {
        writer.write("Name,Description,Status,Category\n");

        for (TaskClass task : TasksList) {
            writer.write(task.toString());
            writer.newLine();
        }

        JOptionPane.showMessageDialog(rootPane, "File Made!");
    }
}

```

```

    } catch (Exception e) {
        JOptionPane.showMessageDialog(rootPane, e.getMessage());
    }
}

```

```

private void BtnImportActionPerformed(java.awt.event.ActionEvent evt) {
    //this uses IO to read the tasks from the text file created earlier.
    try {
        FileInputStream inputStream = new FileInputStream("TextFile.txt");

        int ch;

        String ReadRows = "";
        while ((ch = inputStream.read()) != -1) {
            ReadRows = ReadRows + (char) ch;
        }

        DefaultTableModel ResultTable = (DefaultTableModel) ResTab.getModel();
        ResultTable.setRowCount(0);
        ResultTable.setRowCount(0);

        String[] CollectRows = ReadRows.split(",");

        for (int i = 0; i <= CollectRows.length - 1; i += 4) {
            TaskClass task = new TaskClass(CollectRows[i], CollectRows[i + 1], CollectRows[i + 2],
CollectRows[i + 3]);

            ImportList.add(task);
        }

        for (TaskClass task : ImportList) {
            ResultTable.addRow(new Object[]{task.getName(), task.getDescription(), task.getStatus(),
task.getCategory()});

```

```

    }

    Path pathToFile = Path.of("TextFile.txt");

    BasicFileAttributes filestuff = Files.readAttributes(pathToFile, BasicFileAttributes.class);

    PropLab.setText("<HTML>File Size: " + filestuff.size() + "<BR>File Creation Date: " +
filestuff.creationTime() + "<BR>Last Modified: " + filestuff.lastModifiedTime() + "</HTML>");

    } catch (IOException e) {

        JOptionPane.showMessageDialog(rootPane, e.getMessage());

    } catch (Exception e) {

        JOptionPane.showMessageDialog(rootPane, e.getMessage());

    }

}

```

```

private void sortByComItemStateChanged(java.awt.event.ItemEvent evt) {

```

```

    if (sortByCom.getSelectedItem() == "Category") {

        specCatCombo.setVisible(true);

        catLab.setVisible(true);

        SortBtn.setVisible(true);

        popCategoryCombo();

    } else if (sortByCom.getSelectedItem() == "Task Name") {

        sorting("Name"); //sets the column name

    } else if (sortByCom.getSelectedItem() == "Description") {

        sorting(sortbyCom.getSelectedItem().toString());

    } else if (sortByCom.getSelectedItem() == "Status") {

        sorting(sortbyCom.getSelectedItem().toString());

    }

}

```

```

private void specCatComboItemStateChanged(java.awt.event.ItemEvent evt) {
    if (specCatCombo.getSelectedItem() != null) {
        //popHasMap(specCatCombo.getSelectedItem().toString());

    }

}

```

```

private void specCatComboActionPerformed(java.awt.event.ActionEvent evt) {
    if (specCatCombo.getSelectedItem() != null) {

    }

}

```

```

private void SortBtnActionPerformed(java.awt.event.ActionEvent evt) {
    popHasMap();
    hasMapTablePop(specCatCombo.getSelectedItem().toString());
}

```

```

/**
 * @param args the command line arguments
 */
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(MainF.class.getName()).log(java.util.logging.Level.SEVERE,
        null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(MainF.class.getName()).log(java.util.logging.Level.SEVERE,
        null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(MainF.class.getName()).log(java.util.logging.Level.SEVERE,
        null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(MainF.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 MainF().setVisible(true);
    }
});
}

```

```
// Variables declaration - do not modify

private javax.swing.JButton AddBtn;
private javax.swing.JButton BtnImport;
private javax.swing.JButton CSVEpoBtn;
private javax.swing.JButton ChgBtn;
private javax.swing.JButton DelBtn;
private javax.swing.JLabel PropLab;
private javax.swing.JTable ResTab;
private javax.swing.JTextField SearchTxt;
private javax.swing.JButton SerBtn;
private javax.swing.JButton SortBtn;
private javax.swing.JButton TxtBtn;
private javax.swing.JLabel catLab;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
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 jPanel1;
private javax.swing.JPanel jPanel10;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
```

```
private javax.swing.JPanel jPanel6;  
private javax.swing.JPanel jPanel7;  
private javax.swing.JPanel jPanel8;  
private javax.swing.JPanel jPanel9;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JSeparator jSeparator1;  
private javax.swing.JLabel sec3Lab1;  
private javax.swing.JComboBox<String> serbyCom;  
private javax.swing.JComboBox<String> sortbyCom;  
private javax.swing.JComboBox<String> specCatCombo;  
// End of variables declaration  
}  
  
//VF/FH
```

## ADD Task Code

```
import java.io.File;

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.DefaultComboBoxModel;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;


public class AddF extends javax.swing.JFrame {


    public final String url = "jdbc:sqlite:TaskStorageDB";


    public AddF() {

        initComponents();

    }


    public void addTask() { //gets the user input from each input field and makes the query and then runs
it and the input fields are reset after the data is added.

        File alive = new File("taskstoragedb");

        if (alive.exists()) {

            try {

                Connection connect = DriverManager.getConnection(url);

                String addquery = "INSERT INTO TASKS (NAME,DESCRIPTION,STATUS,CATEGORY)"

                    + "Values("

                    + NameTxt.getText()
```



```

+ "" + DescTxt.getText()
+ "" + statComb.getSelectedItemId()
+ "" + CatComb.getSelectedItemId()
+ ""));

```

```

Statement addTask = connect.createStatement();
addTask.execute(addquery);
JOptionPane.showMessageDialog(rootPane, "Task added!");
NameTxt.setText("");
DescTxt.setText("");
statComb.setSelectedIndex(-1);
CatComb.setSelectedIndex(-1);

```

```

addTask.close();
connect.close();
} catch (SQLException e) {
    JOptionPane.showMessageDialog(rootPane, e.getMessage());
}
}
}

```

```

public void populateCategoryCombo() {
    File alive = new File("taskstoragedb");
    if (alive.exists()) {
        try {
            Connection connect = DriverManager.getConnection(url);
            String combopop = "SELECT * FROM CATEGORIES ";
            Statement popcombo = connect.createStatement();
            ResultSet SetResult = popcombo.executeQuery(combopop);
            DefaultComboBoxModel comboMod = (DefaultComboBoxModel) CatComb.getModel();

```

```

CatComb.removeAllItems();

TaskClass task;

while (SetResult.next()) {
    CatComb.addItem(SetResult.getString("CATEGORY_NAME"));

}

SetResult.close();

popcombo.close();

connect.close();

} catch (SQLException e) {
    JOptionPane.showMessageDialog(rootPane, e.getMessage());
}
}
}

public void addCustomCategory() {
    File alive = new File("taskstoragedb");
    if (alive.exists()) {
        try {
            Connection connect = DriverManager.getConnection(url);

            String create = "INSERT INTO CATEGORIES (CATEGORY_NAME)"
                + "VALUES ('" + NewNameTxt.getText() + "');"

            Statement addcategory = connect.createStatement();
            addcategory.execute(create);

            JOptionPane.showMessageDialog(rootPane, "Category Added.");

            connect.close();

            addcategory.close();

```

```

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(rootPane, e.getMessage());

    }

}

```

```

/**

```

\* 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">

```

```

private void initComponents() {

```

```

    jLabel1 = new javax.swing.JLabel();

```

```

    SubNewTask = new javax.swing.JButton();

```

```

    jLabel3 = new javax.swing.JLabel();

```

```

    jLabel4 = new javax.swing.JLabel();

```

```

    jLabel5 = new javax.swing.JLabel();

```

```

    jLabel6 = new javax.swing.JLabel();

```

```

    NameTxt = new javax.swing.JTextField();

```

```

    DescTxt = new javax.swing.JTextField();

```

```

    statComb = new javax.swing.JComboBox<>();

```

```

    CatComb = new javax.swing.JComboBox<>();

```

```

    jPanel1 = new javax.swing.JPanel();

```

```

    jLabel2 = new javax.swing.JLabel();

```

```

    CreCusCatBtn = new javax.swing.JButton();

```

```

    NewNameTxt = new javax.swing.JTextField();

```

```

    jLabel7 = new javax.swing.JLabel();

```

```

setTitle("Add Task");

setLocation(new java.awt.Point(750, 150));

setMaximumSize(new java.awt.Dimension(477, 350));

setMinimumSize(new java.awt.Dimension(477, 350));

setResizable(false);

addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowActivated(java.awt.event.WindowEvent evt) {
        formWindowActivated(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 36)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Add A Task");

SubNewTask.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
SubNewTask.setText("Submit");
SubNewTask.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SubNewTaskActionPerformed(evt);
    }
});

jLabel3.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel3.setText("Task Name");

jLabel4.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
jLabel4.setText("Description");

jLabel5.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N

```

```
jLabel5.setText("Status");
```

```
jLabel6.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
```

```
jLabel6.setText("Category");
```

```
statComb.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Started", "In  
Progress", "Stalled", "Completed" }));
```

```
jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
```

```
jLabel2.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel2.setText("Create a Custom Category:");
```

```
CreCusCatBtn.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
```

```
CreCusCatBtn.setText("Create");
```

```
CreCusCatBtn.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        CreCusCatBtnActionPerformed(evt);
```

```
    }
```

```
});
```

```
jLabel7.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel7.setText("Name");
```

```
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
```

```
jPanel1.setLayout(jPanel1Layout);
```

```
jPanel1Layout.setHorizontalGroup(
```

```
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(jPanel1Layout.createSequentialGroup()
```

```
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addGroup(jPanel1Layout.createSequentialGroup())
            .addGap(15, 15, 15)
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
            .addContainerGap(32, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(CreCusCatBtn, javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
                .addComponent(jLabel7)
                .addGap(18, 18, 18)
                .addComponent(NewNameTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 308,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(9, 9, 9))))
        .addContainerGap())
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())
            .addContainerGap()
            .addComponent(jLabel2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 15,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(NewNameTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel7))

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(CreCusCatBtn)

        .addContainerGap()

    );

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

    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(layout.createSequentialGroup()
                                    .addGap(151, 151, 151)
                                    .addComponent(jLabel1))
                                .addGroup(layout.createSequentialGroup()
                                    .addGroup(layout.createSequentialGroup()
                                        .addGap(69, 69, 69)
                                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                            .addComponent(jLabel3)
                                            .addComponent(jLabel4)
                                            .addComponent(jLabel5)
                                            .addComponent(jLabel6))
                                        .addGap(35, 35, 35)
                                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                            .addComponent(SubNewTask, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
                                                javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
                                                false)
                                                    .addComponent(NameTxt)
                                                    .addComponent(DescTxt)
                                                    .addComponent(statComb, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                        Short.MAX_VALUE)
                                                    )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    );

```

```

        .addComponent(CatComb, javax.swing.GroupLayout.PREFERRED_SIZE, 212,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(layout.createSequentialGroup())

        .addGap(35, 35, 35)

        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addContainerGap(19, Short.MAX_VALUE)

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGap(26, 26, 26)

        .addComponent(jLabel1)

        .addGap(26, 26, 26)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel3)

            .addComponent(NameTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(24, 24, 24)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel4)

            .addComponent(DescTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(24, 24, 24)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel5)

            .addComponent(statComb, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(24, 24, 24)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel6)

```



```

        .addComponent(CatComb, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 10,
Short.MAX_VALUE)

        .addComponent(SubNewTask, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(15, 15, 15))

    );

    pack();
} // </editor-fold>

private void SubNewTaskActionPerformed(java.awt.event.ActionEvent evt) {
    addTask();
}

private void formWindowActivated(java.awt.event.WindowEvent evt) {
    ImageIcon icon = new ImageIcon("src/FA2.png");//used to set the icon for the window
    this.setIconImage(icon.getImage());
    populateCategoryCombo();

}

private void CreCusCatBtnActionPerformed(java.awt.event.ActionEvent evt) {
    addCustomCategory();
}

/**

```

```

* @param args the command line arguments
*/
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(AddF.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(AddF.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(AddF.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(AddF.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 AddF().setVisible(true);

    }

});

}

// Variables declaration - do not modify

private javax.swing.JComboBox<String> CatComb;

private javax.swing.JButton CreCusCatBtn;

private javax.swing.JTextField DescTxt;

private javax.swing.JTextField NameTxt;

private javax.swing.JTextField NewNameTxt;

private javax.swing.JButton SubNewTask;

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.JPanel jPanel1;

private javax.swing.JComboBox<String> statComb;

// End of variables declaration

}

//VH/FH

```

## Change A Task Code

```
import java.io.File;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import javax.swing.DefaultComboBoxModel;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author jonva
 */
public class ChangeF extends javax.swing.JFrame {

    public final String url = "jdbc:sqlite:TaskStorageDB";
    ArrayList<TaskClass> TaskList = new ArrayList<TaskClass>();
```

```

public void Searching(String CatName, String UserValue) {
    File alive = new File("taskstoragedb");
    if (alive.exists()) {
        try {
            Connection connect = DriverManager.getConnection(url);

            String search = "SELECT * FROM TASKS WHERE " + CatName + " LIKE '" + UserValue + "%'";

            Statement SEARCHING = connect.createStatement();

            ResultSet SetResult = SEARCHING.executeQuery(search);

            DefaultComboBoxModel comboMod = (DefaultComboBoxModel) serresultcmb.getModel();

            int itemnum = 0;

            while (SetResult.next()) {

                //here the returned data is used to make an object of the taskclass to store the data and is
                inserted into the combobox for the user to see.

                TaskClass tasks = new TaskClass(SetResult.getInt("ID"), SetResult.getString("Name"),
                SetResult.getString("Description"), SetResult.getString("Status"), SetResult.getString("Category"));

                TaskList.add(tasks);

                comboMod.insertElementAt(tasks, itemnum);

                itemnum++;
            }

            SetResult.close();

            SEARCHING.close();

            connect.close();
        } catch (SQLException e) {

            JOptionPane.showMessageDialog(rootPane, e.getMessage());
        }
    }
}

```

public void populateUpdate() { //this is used to enable to unenable the specific inputs fields depending on which input field is selected from the combobox

```
    if (UpSelCom.getSelectedItemId() == "Task Name") {
        updateTxt.setEnabled(true);
        updateCombo.setEnabled(false);
        String newName = updateTxt.getText();
    } else if (UpSelCom.getSelectedItemId() == "Description") {
        updateTxt.setEnabled(true);
        updateCombo.setEnabled(false);
        String newDesc = updateTxt.getText();
    } else if (UpSelCom.getSelectedItemId() == "Status") {
        updateCombo.setEnabled(true);
        updateTxt.setEnabled(false);
        updateCombo.removeAllItems();
        updateCombo.addItem("Started");
        updateCombo.addItem("In Progress");
        updateCombo.addItem("Stalled");
        updateCombo.addItem("Completed");
    } else if (UpSelCom.getSelectedItemId() == "Category") {
        updateCombo.setEnabled(true);
        updateTxt.setEnabled(false);
        updateCombo.removeAllItems();
        populateCategoryCombo();
    }
}
```

```
public void populateCategoryCombo() {
    File alive = new File("taskstoragedb");
    if (alive.exists()) {
        try {
```

```
Connection connect = DriverManager.getConnection(url);

String combopop = "SELECT * FROM CATEGORIES ";

Statement popcombo = connect.createStatement();

ResultSet SetResult = popcombo.executeQuery(combopop);

DefaultComboBoxModel comboMod = (DefaultComboBoxModel) updateCombo.getModel();

updateCombo.removeAllItems();


TaskClass task;

while (SetResult.next()) {

    updateCombo.addItem(SetResult.getString("CATEGORY_NAME"));

}

connect.close();

popcombo.close();

SetResult.close();

} catch (SQLException e) {

    JOptionPane.showMessageDialog(rootPane, e.getMessage());

}

}

}

}

public ChangeF() {

    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">

private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    updateBtn = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    serTxt = new javax.swing.JTextField();
    serComBx = new javax.swing.JComboBox<>();
    serresultcmb = new javax.swing.JComboBox<>();
    jLabel3 = new javax.swing.JLabel();
    serBtn = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();
    UpSelCom = new javax.swing.JComboBox<>();
    jLabel9 = new javax.swing.JLabel();
    updateTxt = new javax.swing.JTextField();
    updateCombo = new javax.swing.JComboBox<>();

    setTitle("Change a Task");
    setLocation(new java.awt.Point(750, 150));
    setMaximumSize(new java.awt.Dimension(477, 412));
    setMinimumSize(new java.awt.Dimension(477, 412));
    setResizable(false);
    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowActivated(java.awt.event.WindowEvent evt) {
            formWindowActivated(evt);
        }
    });

    jLabel1.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 36)); // NOI18N

```



```
jLabel1.setText("Change a Task");
```

```
updateBtn.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
```

```
updateBtn.setText("Update");
```

```
updateBtn.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        updateBtnActionPerformed(evt);
```

```
    }
```

```
});
```

```
jLabel2.setFont(new java.awt.Font("Franklin Gothic Demi Cond", 0, 18)); // NOI18N
```

```
jLabel2.setText("Search");
```

```
jLabel2.setToolTipText("");
```

```
serComBx.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
```

```
serComBx.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Task Name",  
"Description", "Status", "Category" }));
```

```
serComBx.setSelectedIndex(-1);
```

```
jLabel3.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel3.setText("Select the task to change");
```

```
serBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
```

```
serBtn.setText("Search");
```

```
serBtn.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        serBtnActionPerformed(evt);
```

```
    }
```

```
});
```

```
jLabel4.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel4.setText("Search By:");
```

```
UpSelCom.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Task Name",  
"Description", "Status", "Category" }));
```

```
UpSelCom.setSelectedIndex(-1);
```

```
UpSelCom.addItemListener(new java.awt.event.ItemListener() {
```

```
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
```

```
        UpSelComItemStateChanged(evt);
```

```
    }
```

```
});
```

```
jLabel9.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel9.setText("Select what to change:");
```

```
updateTxt.setEnabled(false);
```

```
updateCombo.setEnabled(false);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(
```

```
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()
```

```
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                .addGroup(layout.createSequentialGroup()
```

```
                    .addGap(117, 117, 117)
```

```
                    .addComponent(jLabel1))
```

```
                .addGroup(layout.createSequentialGroup()
```

```
                    .addGap(195, 195, 195)
```

```

        .addComponent(updateBtn))
    .addGroup(layout.createSequentialGroup())
        .addGap(46, 46, 46)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
            .addComponent(UpSelCom, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel3)
            .addComponent(serresultcmb, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel2)
                    .addComponent(jLabel4))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(serComBx, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(serBtn))
                    .addComponent(serTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 297,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addComponent(jLabel9)
                    .addComponent(updateTxt)
                    .addComponent(updateCombo, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
        .addContainerGap(31, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(

```

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGap(29, 29, 29)
.addComponent(jLabel1)
.addGap(23, 23, 23)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel2)
.addComponent(serTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(serComBx, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(serBtn)
.addComponent(jLabel4))
.addGap(16, 16, 16)
.addComponent(jLabel3)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(serresultcmb, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(23, 23, 23)
.addComponent(jLabel9)
.addGap(18, 18, 18)
.addComponent(UpSelCom, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(18, 18, 18)
.addComponent(updateTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(18, 18, 18)
.addComponent(updateCombo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 68,
Short.MAX_VALUE)

        .addComponent(updateBtn)

        .addGap(29, 29, 29))

);

pack();
} // </editor-fold>

```

```

private void serBtnActionPerformed(java.awt.event.ActionEvent evt) {
    if (serTxt.getText().isBlank() || serComBx.getSelectedIndex() == -1) {
        JOptionPane.showMessageDialog(rootPane, "Search or search by cannot be empty!");
    } else {
        serresultcmb.removeAllItems();
        if (serComBx.getSelectedItem() == "Task Name") {
            Searching("Name", serTxt.getText());
        } else if (serComBx.getSelectedItem() == "Description") {
            Searching("Description", serTxt.getText());
        } else if (serComBx.getSelectedItem() == "Status") {
            Searching("Status", serTxt.getText());
        } else if (serComBx.getSelectedItem() == "Category") {
            Searching("Category", serTxt.getText());
        }
    }
    serTxt.setText("");
    serComBx.setSelectedIndex(-1);
}

```

```

private void UpSelComItemStateChanged(java.awt.event.ItemEvent evt) {
    populateUpdate();
}

```

```
}
```

```
private void updateBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    String LatestValue;  
  
    TaskClass task = (TaskClass) serresultcmb.getSelectedItemAt()  
    int spectaskID = task.getID();  
  
    String columnName = UpSelCom.getSelectedItemAt().toString();  
    if (columnName == "Task Name" || columnName == "Description") {  
        LatestValue = updateTxt.getText();  
    } else {  
        LatestValue = updateCombo.getSelectedItemAt().toString();  
    }  
  
    File alive = new File("taskstoragedb");  
    if (alive.exists()) {  
        try {  
            Connection connect = DriverManager.getConnection(url);  
  
            String updateSTMT = "UPDATE TASKS SET " + columnName + " = " + LatestValue + " WHERE ID  
= " + spectaskID;  
  
            Statement updateDB = connect.createStatement();  
            updateDB.execute(updateSTMT);  
            JOptionPane.showMessageDialog(rootPane, "Changes Made!");  
  
            updateDB.close();  
            connect.close();  
        } catch (SQLException e) {  
            JOptionPane.showMessageDialog(rootPane, e.getMessage());  
        }  
        serresultcmb.setSelectedIndex(-1);  
        updateTxt.setText("");  
    }  
}
```

```

        updateCombo.setSelectedIndex(-1);
        UpSelCom.setSelectedIndex(-1);
    }
}

```

```

private void formWindowActivated(java.awt.event.WindowEvent evt) {
    ImageIcon icon = new ImageIcon("src/FA2.png");
    this.setIconImage(icon.getImage());
    populateCategoryCombo();
}

```

```

/**
 * @param args the command line arguments
 */
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(ChangeF.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (InstantiationException ex) {

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

    } catch (IllegalAccessException ex) {

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

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(ChangeF.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 ChangeF().setVisible(true);

    }

});

}

```

```

// Variables declaration - do not modify

private javax.swing.JComboBox<String> UpSelCom;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel9;

private javax.swing.JButton serBtn;

private javax.swing.JComboBox<String> serComBx;

```



```
private javax.swing.JTextField serTxt;  
private javax.swing.JComboBox<String> serresultcmb;  
private javax.swing.JButton updateBtn;  
private javax.swing.JComboBox<String> updateCombo;  
private javax.swing.JTextField updateTxt;  
  
// End of variables declaration  
  
}  
  
//VH/FH
```

## Delete A Task Code

```
import java.util.ArrayList;

import javax.swing.JOptionPane;

import java.io.File;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.DefaultComboBoxModel;

import javax.swing.ImageIcon;


public class DelF extends javax.swing.JFrame {


    public final String url = "jdbc:sqlite:TaskStorageDB";

    ArrayList<TaskClass> TaskList = new ArrayList<TaskClass>();


    public void Searching(String CatName, String UserValue) {

        File alive = new File("taskstoragedb");

        if (alive.exists()) {

            try {

                Connection connect = DriverManager.getConnection(url);

                String search = "SELECT * FROM TASKS WHERE " + CatName + " LIKE '" + UserValue + "%'";

                Statement SEARCHING = connect.createStatement();

                ResultSet SetResult = SEARCHING.executeQuery(search);

                DefaultComboBoxModel comboMod = (DefaultComboBoxModel) ResultCombo.getModel();

                int itemnum = 0;


                while (SetResult.next()) {
```

```

        TaskClass tasks = new TaskClass(SetResult.getInt("ID"), SetResult.getString("Name"),
SetResult.getString("Description"), SetResult.getString("Status"), SetResult.getString("Category"));

        TaskList.add(tasks);

        comboMod.insertElementAt(tasks, itemnum);

        itemnum++;

    }

```

```

        SetResult.close();

        SEARCHING.close();

        connect.close();

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(rootPane, e.getMessage());

    }

}

}

```

```

public DelF() {

    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">

```

```

private void initComponents() {

```

```

    jLabel1 = new javax.swing.JLabel();

```

```

DelBtn = new javax.swing.JButton();

jLabel2 = new javax.swing.JLabel();

SearchTxt = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

colSelComb = new javax.swing.JComboBox<>();

SearchBtn = new javax.swing.JButton();

ResultCombo = new javax.swing.JComboBox<>();

jLabel4 = new javax.swing.JLabel();


setTitle("Delete a Task");

setLocation(new java.awt.Point(750, 150));

setMaximumSize(new java.awt.Dimension(477, 412));

setMinimumSize(new java.awt.Dimension(477, 412));

setResizable(false);

addWindowListener(new java.awt.event.WindowAdapter() {

    public void windowActivated(java.awt.event.WindowEvent evt) {

        formWindowActivated(evt);

    }

});


jLabel1.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 36)); // NOI18N

jLabel1.setText("Delete a Task");


DelBtn.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N

DelBtn.setText("Delete");

DelBtn.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        DelBtnActionPerformed(evt);

    }

});

```

```
jLabel2.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel2.setText("Search:");
```

```
jLabel3.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel3.setText("Search by:");
```

```
colSelComb.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Task Name",  
"Description", "Status", "Category" }));
```

```
colSelComb.setSelectedIndex(-1);
```

```
SearchBtn.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
```

```
SearchBtn.setText("Search");
```

```
SearchBtn.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        SearchBtnActionPerformed(evt);
```

```
    }
```

```
});
```

```
jLabel4.setFont(new java.awt.Font("Franklin Gothic Demi", 0, 18)); // NOI18N
```

```
jLabel4.setText("Select the task to delete:");
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(
```

```
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()
```

```
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                .addGroup(layout.createSequentialGroup()
```

```
                    .addGap(190, 190, 190)
```

```

        .addComponent(DelBtn))
    .addGroup(layout.createSequentialGroup())
        .addGap(125, 125, 125)
        .addComponent(jLabel1))
    .addGroup(layout.createSequentialGroup())
        .addGap(46, 46, 46)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel4)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)

                .addGroup(layout.createSequentialGroup())
                    .addComponent(jLabel3)
                    .addGap(18, 18, 18)
                    .addComponent(colSelComb, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(SearchBtn))
                .addGroup(layout.createSequentialGroup())
                    .addComponent(jLabel2)
                    .addGap(18, 18, 18)
                    .addComponent(SearchTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 322,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(ResultCombo, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))))
        .addContainerGap(31, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addGap(24, 24, 24)

```

```

        .addComponent(jLabel1)

        .addGap(4, 4, 4)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(SearchTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(24, 24, 24)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel3)

            .addComponent(colSelComb, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(SearchBtn))

        .addGap(19, 19, 19)

        .addComponent(jLabel4)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(ResultCombo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 86,
Short.MAX_VALUE)

        .addComponent(DelBtn)

        .addGap(27, 27, 27))

    );

```

```

    pack();

} // </editor-fold>

```

```

private void SearchBtnActionPerformed(java.awt.event.ActionEvent evt) {
    if (SearchTxt.getText().isBlank() || colSelComb.getSelectedIndex() == -1) {
        JOptionPane.showMessageDialog(rootPane, "Search or search by cannot be empty!");
    } else {
        ResultCombo.removeAllItems();
    }
}

```

```

if (colSelComb.getSelectedItem() == "Task Name") {
    Searching("Name", SearchTxt.getText());
} else if (colSelComb.getSelectedItem() == "Description") {
    Searching("Description", SearchTxt.getText());
} else if (colSelComb.getSelectedItem() == "Status") {
    Searching("Status", SearchTxt.getText());
} else if (colSelComb.getSelectedItem() == "Category") {
    Searching("Category", SearchTxt.getText());
}
}

SearchTxt.setText("");
colSelComb.setSelectedIndex(-1);
}

private void DelBtnActionPerformed(java.awt.event.ActionEvent evt) {
    TaskClass task = (TaskClass) ResultCombo.getSelectedItem();
    int spectaskID = task.getID();
    File alive = new File("taskstoragedb");
    if (alive.exists()){
        try {
            Connection connect = DriverManager.getConnection(url);
            String DelRow = "Delete From Tasks Where ID = " + spectaskID;
            Statement DelStmt = connect.createStatement();
            DelStmt.execute(DelRow);
            JOptionPane.showMessageDialog(rootPane, "Task Deleted!");

            DelStmt.close();
            connect.close();

            ResultCombo.setSelectedIndex(-1);
        }
    }
}

```



```

    } catch (SQLException e) {
        JOptionPane.showMessageDialog(rootPane, e.getMessage());
    }
}

private void formWindowActivated(java.awt.event.WindowEvent evt) {

    ImageIcon icon = new ImageIcon("src/FA2.png");
    this.setIconImage(icon.getImage());
}

/**
 * @param args the command line arguments
 */
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(DelF.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (InstantiationException ex) {

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

    } catch (IllegalAccessException ex) {

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

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(DelF.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 DelF().setVisible(true);

    }

});

}

```

```

// Variables declaration - do not modify

private javax.swing.JButton DelBtn;

private javax.swing.JComboBox<String> ResultCombo;

private javax.swing.JButton SearchBtn;

private javax.swing.JTextField SearchTxt;

private javax.swing.JComboBox<String> colSelComb;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

```

```
private javax.swing.JLabel jLabel4;  
// End of variables declaration  
}
```

```
//VH/FH
```

## Task Class File

```
import java.io.Serializable;
```

```
public class TaskClass implements Serializable{
```

```
    private int ID;
```

```
    private String Name;
```

```
    private String Description;
```

```
    private String Status;
```

```
    private String Category;
```

```
    public TaskClass(int id, String name, String description, String status, String category) { //constructor  
for objects being retrived from the Database
```

```
        this.ID = id;
```

```
        this.Name = name;
```

```
        this.Description = description;
```

```
        this.Status = status;
```

```
        this.Category = category;
```

```
    }
```

```
    public TaskClass(String name,String description, String status, String category) { //constructor for the  
objects being written to the text file.
```

```
        this.Name = name;
```

```
        this.Description = description;
```

```
        this.Status = status;
```

```
        this.Category = category;
```

```
    }
```

```
public TaskClass(String category) { //constructor for the category objects in combo boxes.  
    this.Category = category;  
}  
  
public void setID(int id) {  
    this.ID = id;  
}  
  
public int getID() {  
    return this.ID;  
}  
  
public void setName(String name) {  
    this.Name = name;  
}  
  
public String getName() {  
    return this.Name;  
}  
  
public void setDescription(String desc) {  
    this.Description = desc;  
}  
  
public String getDescription() {  
    return this.Description;  
}  
  
public void setStatus(String stat) {
```

```

        this.Status = stat;
    }

    public String getStatus() {
        return this.Status;
    }

    public void setCategory(String cat) {
        this.Category = cat;
    }

    public String getCategory() {
        return this.Category;
    }

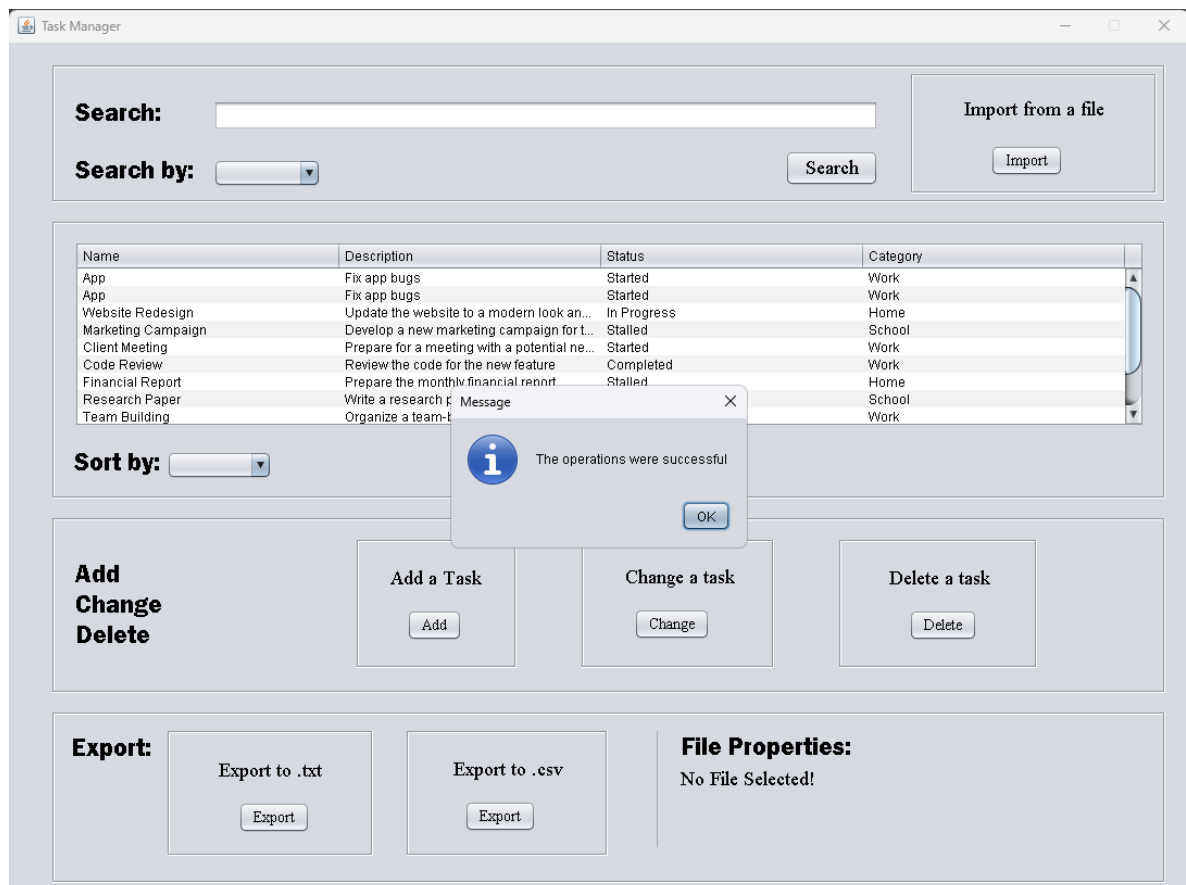
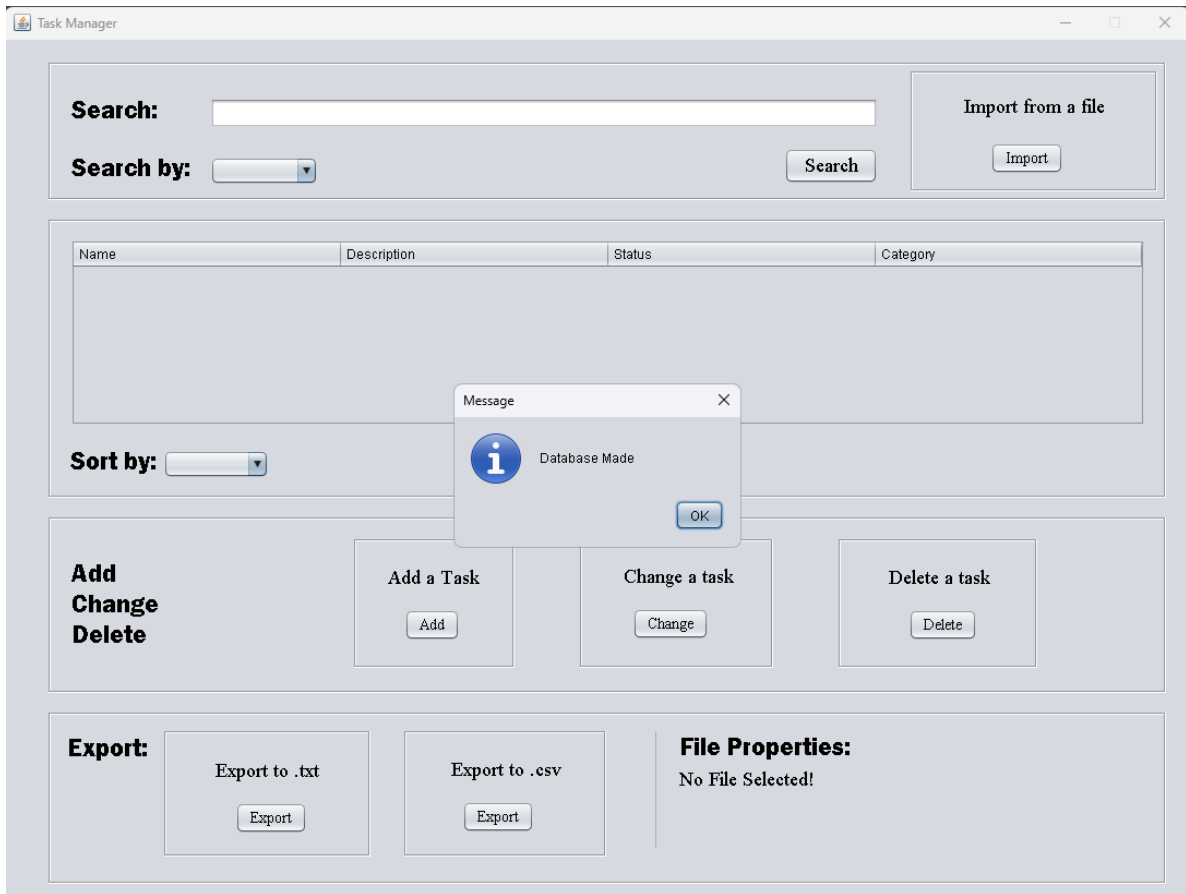
    @Override
    public String toString() { //used to display tasks in the task selection combobox for update and delete
    statments.
        return String.format("%s, %s, %s, %s", Name, Description, Status, Category);

    }
}

//VH/FH

```

## Screenshots of Output



Task Manager

Search:

App

Search by:

Task Name

Search

Import from a file

Import

Name	Description	Status	Category
App	Fix app bugs	Started	Work
App	Fix app bugs	Started	Work

Sort by:

Add  
Change  
Delete

Add a Task

Add

Change a task

Change

Delete a task

Delete

Export:

Export to .txt

Export

Export to .csv

Export

File Properties:

No File Selected!

Task Manager

Search:

Search by:

Search

Import from a file

Import

Name	Description	Status	Category
App	Fix app bugs	Started	Work
App	Fix app bugs	Started	Work

Sort by:

Add  
Change  
Delete

Add a Task

Add

Change a task

Change

Delete a task

Delete

Export:

Export to .txt

Export

Export to .csv

Export

File Properties:

No File Selected!

Message

Search or search by cannot be empty!

OK

79





Add Task

## Add A Task

Task Name

Description

Status

Category

**Create a Custom Category:**

Name

Create

Submit

Message

**i** Task added!

OK

Add Task

## Add A Task

Task Name

Description

Status

Category

**Create a Custom Category:**

Name

Create

Submit

Message

**i** Category Added.

OK

Add Task

## Add A Task

Task Name

Description

Status

Category

**Create a Custom**

Name

Create

Submit

Personal

Work

Home

School

Travel

Change a Task

## Change a Task

**Search**

**Search By:**

**Select the task to change**

App, Fix app bugs, Started, Work

App, Fix app bugs, Started, Work

Personal

Change a Task

## Change a Task

**Search**

**Search By:**

**Select the task to change**

App, Fix a

**Select v**

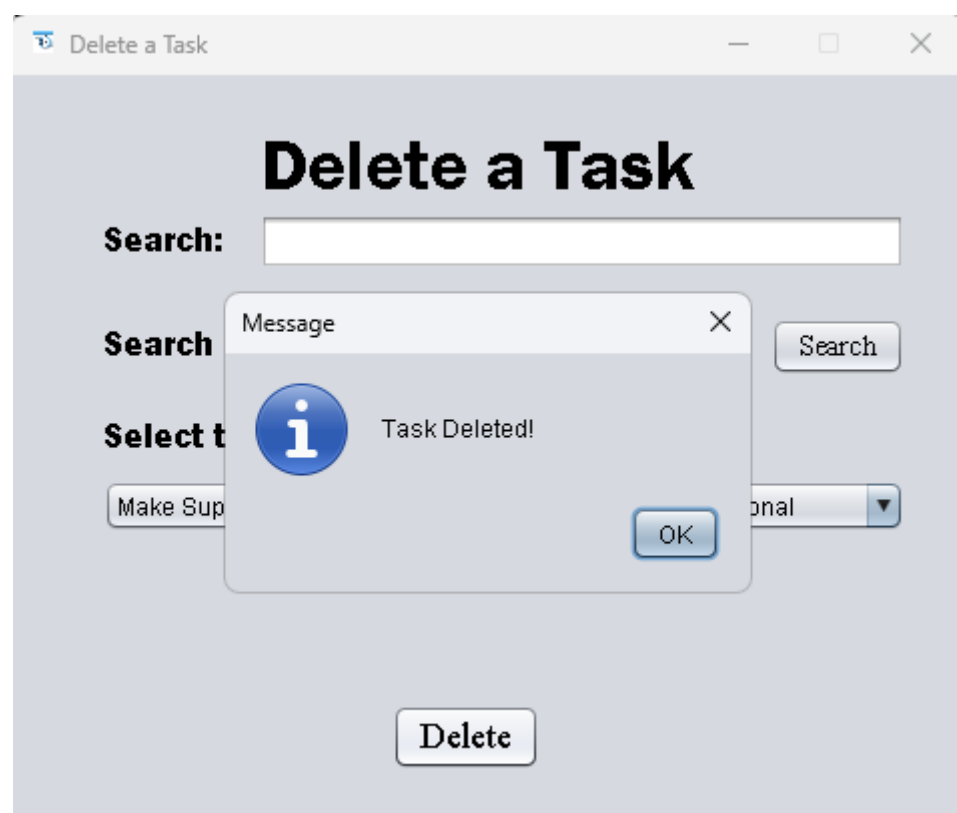
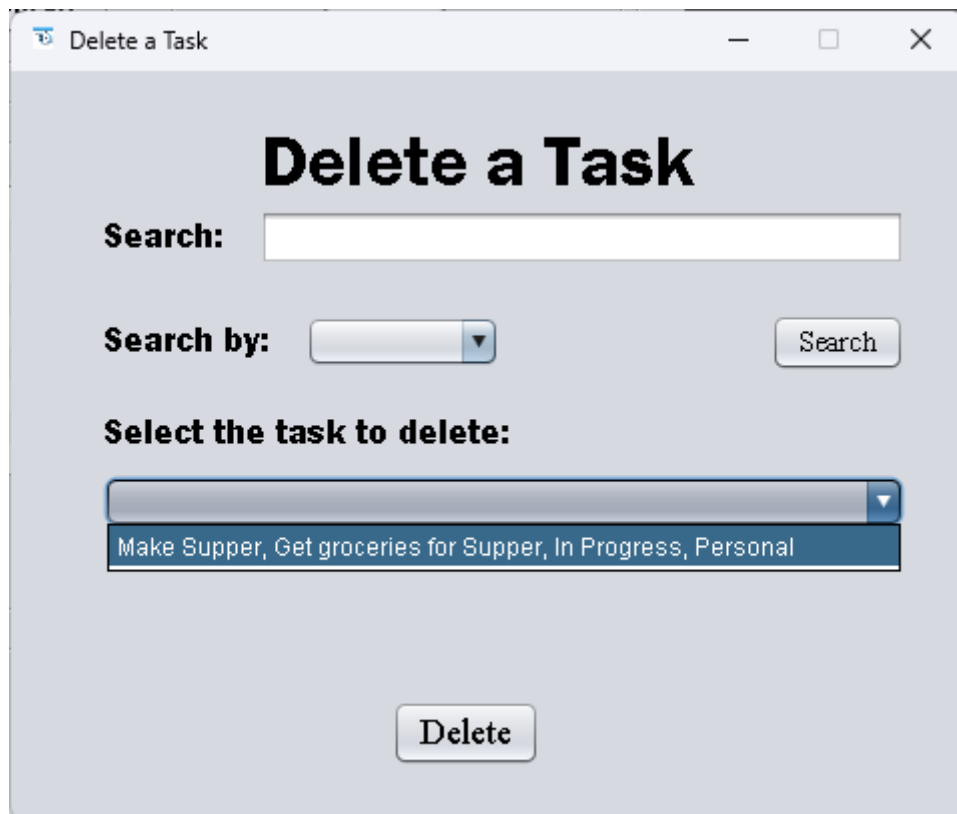
Description

Fix all the bugs

Personal

Message

 Changes Made!



Task Manager

Search:

Search by:

Search

Import from a file

Import

Name	Description	Status	Category
App	Fix all the bugs	Started	Work
App	Fix app bugs	Started	Work
Website Redesign	Update the website to a modern look an...	In Progress	Home
Marketing Campaign	Develop a new marketing campaign for t...	Stalled	School
Client Meeting	Prepare for a meeting with a potential ne...	Started	Work
Code Review	Review the code for the new feature	Completed	Work
Financial Report	Prepare the monthly financial report	Stalled	Home
Research Paper	Write a research paper on AI ethics	In Progress	School
Team Building	Organize a team-building activity	Stalled	Work

Sort by: 

Category

Add  
Change  
Delete

Add a Task

Add

Change a task

Change

Delete a task

Delete

Export:

Export to .txt

Export

Export to .csv

Export

File Properties:

File Size: 804

File Creation Date: 2024-09-04T15:45:22.6746007Z

Last Modified: 2024-09-04T15:45:22.6876078Z

Task Manager

Search:

Search by:

Search

Import from a file

Import

Name	Description	Status	Category
App	Fix all the bugs	Started	Work
App	Fix app bugs	Started	Work
Website Redesign	Update the website to a modern look an...	In Progress	Home
Marketing Campaign	Develop a new marketing campaign for t...	Stalled	School
Client Meeting	Prepare for a meeting with a potential ne...	Started	Work
Code Review	Review the code for the new feature	Completed	Work
Financial Report	Prepare the monthly financial report	Stalled	Home
Research Paper	Write a research pi	Message	School
Team Building	Organize a team-bi		Work

Sort by: 

Category

Add  
Change  
Delete

Add a Task

Add

Change a task

Change

Delete a task

Delete

Export:

Export to .txt

Export

Export to .csv

Export

File Properties:

No File Selected!

File Made!

OK