Koneru Lakshmaiah Education Foundation

(Deemed to be University)

FRESHMAN ENGINEERING DEPARTMENT

A Project Based Lab Report

On

Feedback Survey On Electronics

SUBMITTED BY:

180030245	**M. SRINIVAS**
180030497	**D. HEMANTH**
180030537	**TEJASWI REDDY**

UNDER THE GUIDANCE OF

Mr. B. VENKATESWARULU

Assistant professor

KL UNIVERSITY

Green fields, Vaddeswaram – 522 502

Guntur Dt., AP, India.

}

DEPARTMENT OF BASIC ENGINEERING SCIENCES-1

CERTIFICATE

This is to certify that the project based laboratory report entitled submitted by Ms. TEJASWI REDDY.K bearing RegdNo. **180030537** to the **Department of Basic Engineering Sciences-1, KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in "**OOPS SKILLING**" course in I B Tech II Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2019 – 2 020.

PROJECT SUPERVISOR

B. VENKATESWARULU

HEAD OF THE DEPARTMENT

Dr. L. SRIDHARA RAO

ACKNOWLEDGEMENTS

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our principal **Prof Dr. N. Venkataram** for his administration towards our academic growth.

I express sincere gratitude to HOD-BES-1**Dr. L. Sridhar Rao** for his leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **B.VENKATESWARULU** for his/her novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Reg. No.	Student Name
180030537	**TEJASWI REDDY.K**

}

ABSTRACT

This project is to implement a GUI based program for conducting a survey. This survey was conducted on three categories, i.e., Air Conditioner, Mobile Phones and Laptops. In this survey, the user need to select any of the three categories. And each category should consist of five different models. And the user should be given an option to select whether the model is good or bad. Later rank the different products accordingly after selection of options by five users.

Here, in this project we have done system design, source coding, and program testing and added many more features that facilitate the user to do best selections. Here we used the concepts of Graphical User Interface (GUI) by using packages like the swing package and awt packages to solve the project. By doing this project we have gained the knowledge regarding event delegation also.

INDEX

}

INTRODUCTION

The project is to implement GUI and prepare a java code on our general life application. The main goal of this project is to conduct a survey in selection of electronics and take its feedback. Here the user need to first select electronic device to be, AC, Mobile Phone or Laptop. Then go for further selections of model of device. Later take the feedback to be either good or bad. This project is based on Graphical User Interface(GUI) and Event Delegation Model. Here we used the swing package to develop GUI rich client interface. The reason for using swing instead of awt package is its independent nature towards platform. We used the awt package to perform the event delegation method. This event delegation model is used to perform events by make selections with buttons by using Listener interface in the program. We perform various operations by using some of the pre-defined methods like setSize(), setBounds(), setButtons(), etc and other methods to perform event delegation to Windows are windowOpening(), windowListener(), windowClosing(), etc. Here we have even used the ActionListener() and RadioButton() for handling the event.

AIM

This project is to implement a GUI based program for conducting a survey. This survey was conducted on three categories, i.e., Air Conditioner, Mobile Phones and Laptops. In this survey, the user need to select any of the three categories. And each category should consist of five different models. And the user should be given an option to select whether the model is good or bad. Later rank the different products accordingly after selection of options by five users.

ADVANTAGES

- A major advantage of GUIs is that they make computer operation more intuitive, and thus easier to learn and use.
- GUIs generally provide users with immediate, visual feedback about the effect of each action.
- GUI allows multiple programs and/or instances to be displayed simultaneously.
- Users do not need to know any programming languages.
- Java frameworks give security and support to the project.

}

DISADVANTAGES

- It uses more computer memory as the aim is to make it for user friendly and not resource optimized. As a result it can be slow on older machines.
- GUI becomes more complex if user needs to communicate with the computer directly.
- Certain tasks may take long due to many menus to select the desired choice.
- Hidden commands need to be searched using Help file.
- GUI based applications require more RAM in order to run.
- It uses more processing power compare to other interface types.

FUTURE ENHANCEMENT

Windows technology is continuing to advance. Transparent windows are already used to some extent in higher-end graphics programs, where they are referred to as *layers*. However, they are still not in general use, such as for text editing and word processing programs. Features of transparent windows could include the ability of the user to easily (1) adjust the degree of transparency and (2) set windows so that they only become transparent or opaque under certain conditions. For example, windows could automatically become semi-transparent when they are *inactive* (i.e., when another window or application is currently being used). A useful example of the opposite situation, i.e., a semi-transparent active window, would be typing text into such a window while reading the contents of an opaque window behind it. One more useful, function for windows would be the ability for users to easily merge them so that two images or columns of text can be joined just by moving the windows to the appropriate locations and issuing simple mouse or keyboard commands. Likewise, it would be useful if users had simple techniques for cutting windows (and their contents) into multiple windows.

SOFTWARE AND HARDWARE DETIALS

SOFTWARE REQUIREMENTS:

The major software requirements of the project are as follows:

Language : java

Operating system**:** WINDOWS 10

Compiler : java compiler

HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are as follows:

RAM : 8bytes

Processor : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz

}

IMPLEMENTATION

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
class MyProject extends JFrame {
public static void main(String args[]){
JFrame jf=new JFrame("TeamProject");
JPanel p=new JPanel();
JLabel l=new JLabel("AC");
  l.setBounds(10,10,100,30);
  JLabel l1=new JLabel("Mobile");
  l1.setBounds(10,50,100,30);
  JLabel l2=new JLabel("Laptop");
  l2.setBounds(10,90,100,30);
JButton jb=new JButton("AC");
jb.setBounds(110,10,100,30);
JButton j1=new JButton("Mobile");
j1.setBounds(110,50,100,30);
JButton j2=new JButton("Laptop");
j2.setBounds(110,90,100,30);
JRadioButton jr=new JRadioButton("Samsung AC");
jr.setBounds(10,10,20,20);
JRadioButton jr1=new JRadioButton("LG AC");
jr1.setBounds(10,50,20,20);
```

```
JRadioButton jr2=new JRadioButton("Panasonic AC");
jr2.setBounds(10,100,20,20);
JRadioButton jr3=new JRadioButton("Haier AC");
jr3.setBounds(10,150,20,20);
JRadioButton jr4=new JRadioButton("Sony AC");
jr4.setBounds(10,200,20,20);
JRadioButton s=new JRadioButton("Product is good");
s.setBounds(10,250,20,20);
JRadioButton s1=new JRadioButton("Product is bad");
s1.setBounds(10,300,20,20);
JRadioButton s2=new JRadioButton("Excellent");
s2.setBounds(10,350,20,20);
JRadioButton s3=new JRadioButton("Better");
s3.setBounds(10,400,20,20);
JRadioButton s4=new JRadioButton("worst");
s4.setBounds(10,450,20,20);
JRadioButton a=new JRadioButton("Samsung Mobile");
a.setBounds(10,10,20,20);
JRadioButton b=new JRadioButton("Iphone Mobile");
b.setBounds(10,50,20,20);
JRadioButton c=new JRadioButton("Redmi Mobile");
c.setBounds(10,100,20,20);
JRadioButton d=new JRadioButton("OnePlus Mobile");
d.setBounds(10,150,20,20);
JRadioButton e=new JRadioButton("Vivo Mobile");
e.setBounds(10,200,20,20);
JRadioButton g=new JRadioButton("Product is good");
g.setBounds(10,250,20,20);
```

```
}

JRadioButton h=new JRadioButton("Product is bad");

h.setBounds(10,300,20,20);

JRadioButton j=new JRadioButton("Excellent");

j.setBounds(10,350,20,20);

JRadioButton k=new JRadioButton("Better");

k.setBounds(10,400,20,20);

JRadioButton m=new JRadioButton("worst");

m.setBounds(10,450,20,20);

JRadioButton n=new JRadioButton("Lenovo Laptop");

n.setBounds(10,10,20,20);

JRadioButton o=new JRadioButton("MacBook Laptop");

o.setBounds(10,50,20,20);

JRadioButton p1=new JRadioButton("Dell Laptop");

p1.setBounds(10,100,20,20);

JRadioButton q=new JRadioButton("Hp Laptop");

q.setBounds(10,150,20,20);

JRadioButton r=new JRadioButton("Asus Laptop");

r.setBounds(10,200,20,20);

JRadioButton z=new JRadioButton("Product is good");

z.setBounds(10,250,20,20);

JRadioButton t=new JRadioButton("Product is bad");

t.setBounds(10,300,20,20);

JRadioButton u=new JRadioButton("Excellent");

u.setBounds(10,350,20,20);

JRadioButton v=new JRadioButton("Better");

v.setBounds(10,400,20,20);

JRadioButton w=new JRadioButton("worst");

w.setBounds(10,450,20,20);
```

```
p.add(l);
p.add(l1);
p.add(l2);
p.add(jb);
p.add(j1);
p.add(j2);
jf.add(p);
jf.setSize(300,300);
p.setLayout(null);
jf.setVisible(true);
jb.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent ae) {
JFrame jf=new JFrame("AC");
JPanel p=new JPanel();
p.add(jr);
p.add(jr1);
p.add(jr2);
p.add(jr3);
p.add(jr4);
p.add(s);
p.add(s1);
p.add(s2);
p.add(s3);
p.add(s4);
jf.add(p);
jf.setSize(300,400);
p.setLayout(new FlowLayout());
jf.setVisible(true);
```

```
}

}

});

j1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ae)

{

JFrame jf=new JFrame("Mobile");

JPanel p=new JPanel();

p.add(a);

p.add(b);

p.add(c);

p.add(d);

p.add(e);

p.add(g);

p.add(h);

p.add(j);

p.add(k);

p.add(m);

jf.add(p);

jf.setSize(300,400);

p.setLayout(new FlowLayout());

jf.setVisible(true);

}

});

j2.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ae)

{
```

```
JFrame jf=new JFrame("Laptop");
JPanel p=new JPanel();
p.add(n);
p.add(o);
p.add(p1);
p.add(q);
p.add(r);
p.add(z);
p.add(t);
p.add(u);
p.add(v);
p.add(w);
jf.add(p);
jf.setSize(400,400);
p.setLayout(new FlowLayout());
jf.setVisible(true);
}
});
jr.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(jr.isSelected())
{
jr1.setSelected(false);
jr2.setSelected(false);
jr3.setSelected(false);
jr4.setSelected(false);
```

```
}

}

}

});

jr1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ae)

{

if(jr1.isSelected())

{

jr.setSelected(false);

jr2.setSelected(false);

jr3.setSelected(false);

jr4.setSelected(false);

}

}

});

jr2.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ae)

{

if(jr2.isSelected())

{

jr.setSelected(false);

jr1.setSelected(false);

jr3.setSelected(false);

jr4.setSelected(false);

}

}
```

```
});
jr3.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(jr3.isSelected())
{
jr.setSelected(false);
jr2.setSelected(false);
jr1.setSelected(false);
jr4.setSelected(false);
}
}
});
jr4.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(jr4.isSelected())
{
jr.setSelected(false);
jr2.setSelected(false);
jr3.setSelected(false);
jr1.setSelected(false);
}
}
});
s.addActionListener(new ActionListener()
```

```
}
{
public void actionPerformed(ActionEvent ae)
{
if(s.isSelected())
{
s1.setSelected(false);
}
}
});
s1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(s1.isSelected())
{
s.setSelected(false);
}
}
});
s2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(s2.isSelected())
{
s3.setSelected(false);
s4.setSelected(false);
}
```

```
}
});
s3.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(s3.isSelected())
{
s2.setSelected(false);
s4.setSelected(false);
}
}
});
s4.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(s4.isSelected())
{
s2.setSelected(false);
s3.setSelected(false);
}
}
});
a.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
```

```
}
if(a.isSelected())
{
b.setSelected(false);
c.setSelected(false);
d.setSelected(false);
e.setSelected(false);
}
}
});
b.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(b.isSelected())
{
a.setSelected(false);
c.setSelected(false);
d.setSelected(false);
e.setSelected(false);
}
}
});
c.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(c.isSelected())
{
```

```
a.setSelected(false);
b.setSelected(false);
d.setSelected(false);
e.setSelected(false);
}
}
});
d.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(d.isSelected())
{
a.setSelected(false);
b.setSelected(false);
c.setSelected(false);
e.setSelected(false);
}
}
});
e.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(e.isSelected())
{
a.setSelected(false);
b.setSelected(false);
```

```
}
c.setSelected(false);
d.setSelected(false);
}
}
});
g.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(g.isSelected())
{
h.setSelected(false);
}
}
});
h.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(h.isSelected())
{
g.setSelected(false);
}
}
});
j.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
```

```
{
if(j.isSelected())
{
k.setSelected(false);
m.setSelected(false);
}
}
});
k.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(k.isSelected())
{
j.setSelected(false);
m.setSelected(false);
}
}
});
m.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(m.isSelected())
{
j.setSelected(false);
k.setSelected(false);
}
```

```
}

}

});

n.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ae)

{

if(n.isSelected())

{

o.setSelected(false);

p1.setSelected(false);

q.setSelected(false);

r.setSelected(false);

}

}

});

o.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ae)

{

if(o.isSelected())

{

n.setSelected(false);

p1.setSelected(false);

q.setSelected(false);

r.setSelected(false);

}

}

});
```

```
p1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(p1.isSelected())
{
n.setSelected(false);
o.setSelected(false);
q.setSelected(false);
r.setSelected(false);
}
}
});
q.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(q.isSelected())
{
n.setSelected(false);
o.setSelected(false);
p1.setSelected(false);
r.setSelected(false);
}
}
});
r.addActionListener(new ActionListener()
{
```

```
}
public void actionPerformed(ActionEvent ae)
{
if(r.isSelected())
{
n.setSelected(false);
o.setSelected(false);
p1.setSelected(false);
q.setSelected(false);
}
}
});
z.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(z.isSelected())
{
t.setSelected(false);
}
}
});
t.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(t.isSelected())
{
z.setSelected(false);
```

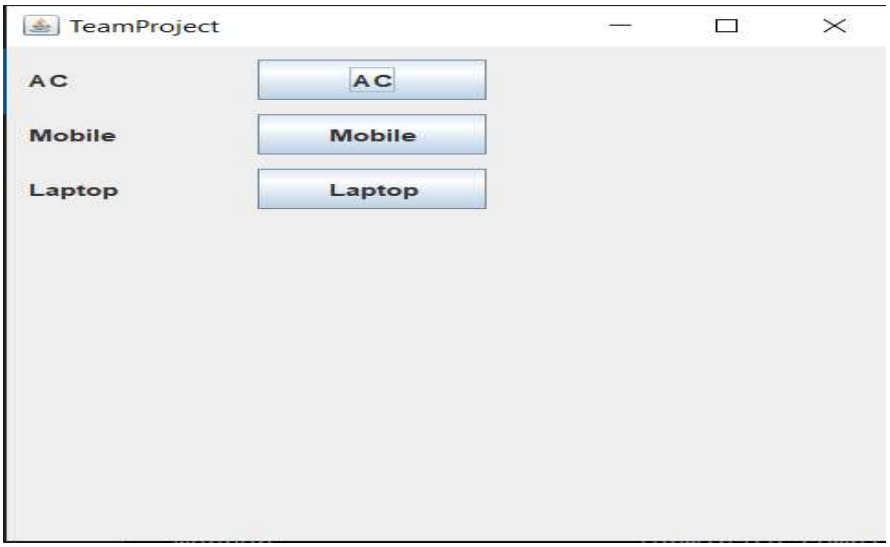
```
}
}
});
u.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(u.isSelected())
{
v.setSelected(false);
w.setSelected(false);
}
}
});
v.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(v.isSelected())
{
u.setSelected(false);
w.setSelected(false);
}
}
});
w.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
```

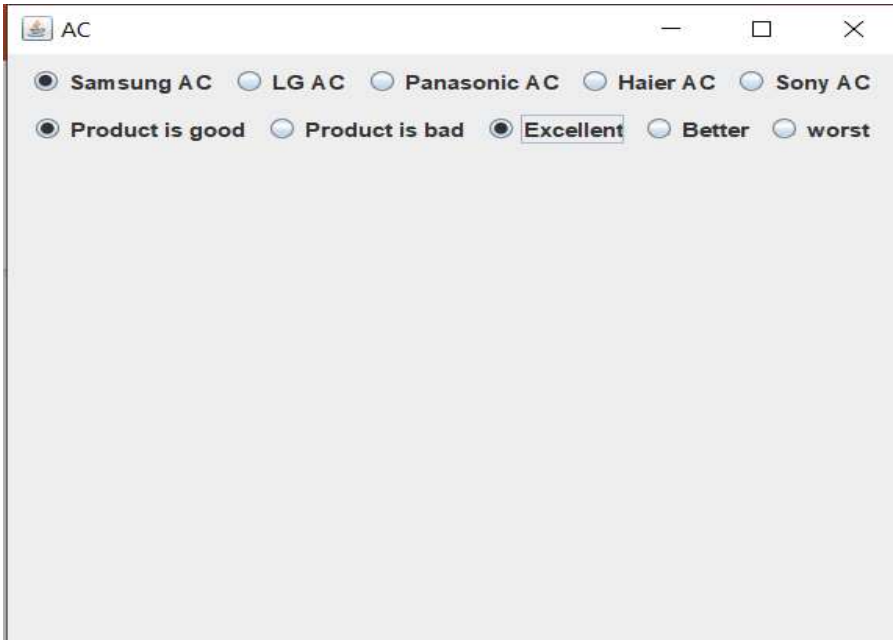
```
}

{

if(w.isSelected())

{

u.setSelected(false);

v.setSelected(false);}

}

});

jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

}
```

INTEGRATION AND SYSTEM TESTING

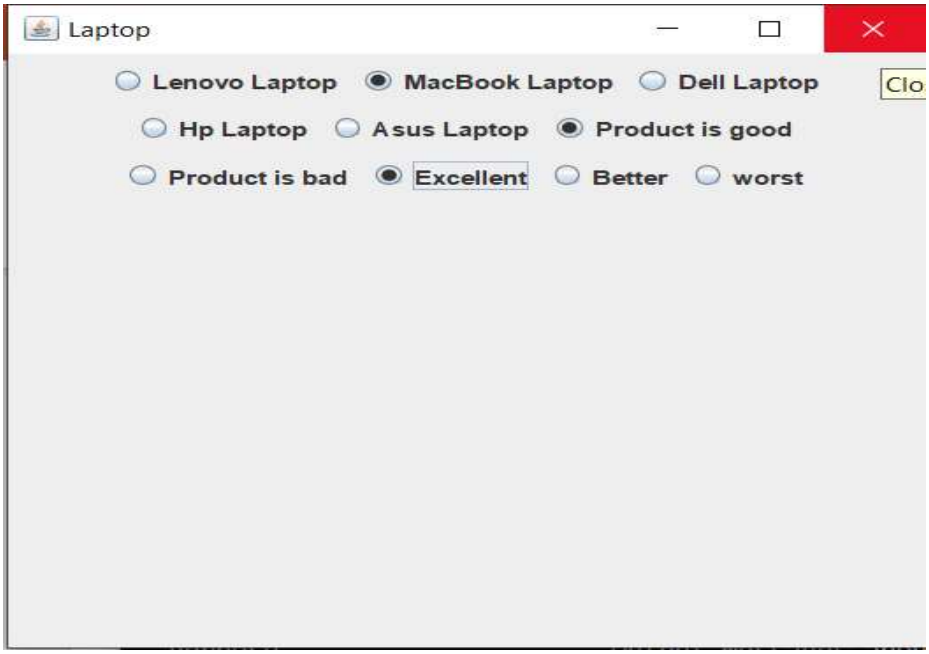
OUTPUTS

Screen Shots:





}


Laptop
Lenovo Laptop
MacBook Laptop
Dell Laptop
Clo
Hp Laptop
Asus Laptop
Product is good
Product is bad
Excellent
Better
worst


Mobile
Samsung Mobile
Iphone Mobile
Redmi Mobile
OnePlus Mobile
Vivo Mobile
Product is good
Product is bad
Excellent
Better
worst

CONCLUSION

This project of preparing a code regarding the survey taken, helps us to understand about the usage of Graphical User Interface(GUI) and Event Delegation in an easy manner.