

Subodh Public School

2017-18

IP Project



EXPERIMENTAL PHYSICS

Innovation in Education



TUSHAR JAIN

SCHOLARS 2A

SUBMITTED TO : - MR. Praveen Bansal

ACKNOWLEDGEMENTS

I am glad that I am able to made this project. But I would not have been able to made this project without the help of my school and my IP teacher Praveen sir from whom I learned Java and SQL also I am very thankful towards my friends as they gave me their precious suggestions on errors and bugs. Finally, I must say thnks to many online sources like StackOverFlow and YouTube channel ProgrammingKnowledge I cannot end this list without saying thanks to my parents



Certificate

This project is certified to be the bonafide work of
_____ of class _____ and school

in the academic year _____ .

PRAVEEN BANSAL

[Subject Teacher]

JAVA CODE

METHOD DECLARATION

1. Conection.java

In this I have declared methods for

- MySQL Connection
- Insert into database
- Select from Database

```
1. public class conection {
2.     public static String uid2;
3.     public static String statusno;
4.     public static Connection con = null;
5.     public static Connection sqlconnect() {
6.         try {
7.             Class.forName("java.sql.DriverManager");
8.             con = (Connection)
9.                 DriverManager.getConnection("jdbc:mysql://localhost:3306/experimen
10. t", "root", "manager");
11.             return con;
12.         } catch (Exception e) {}
13.         return null;
14.     }
15.     public static Connection coondb2() {
16.         try {
17.             Class.forName("java.sql.DriverManager");
18.             con = (Connection)
19.                 DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root"
20. , "manager");
21.             return con;
22.         } catch (Exception e) {}
23.         return null;
24.     }
25.     public static void exup(String s, String y) {
26.         try {
27.             con = (Connection) DriverManager.getConnection("jdbc:mysql://local
28. host:3306/" + y, "root", "manager");
29.             Statement stmt = (Statement) con.createStatement();
30.             stmt.executeUpdate(s);
31.         } catch (SQLException | HeadlessException e) {}
32.     }
33. }
```

```

30.
31.     public static String uid() {
32.         try {
33.             Statement stmt = (Statement) con.createStatement();
34.             String sql = "select max(statusno) from activestatus";
35.             ResultSet rs = stmt.executeQuery(sql);
36.             if (rs.next()) {
37.                 statusno = rs.getString("max(statusno)");
38.                 String sql2 = "select * from activestatus where statusno=" + s
tatusno;
39.                 ResultSet rs2 = stmt.executeQuery(sql2);
40.                 while (rs2.next()) {
41.                     uid2 = rs2.getString("uid");
42.                 }
43.             }
44.         } catch (SQLException | HeadlessException e) {}
45.         return uid2;
46.     }
47.     public static String statusno() {
48.         return statusno;
49.     }
50.     public static ResultSet rs(String s, String y) {
51.         ResultSet rst = null;
52.         try {
53.             con = (Connection) DriverManager.getConnection("jdbc:mysql://local
host:3306/" + y, "root", "manager");
54.         } catch (SQLException ex) {
55.             Logger.getLogger(conection.class.getName()).log(Level.SEVERE, null
, ex);
56.         }
57.         try {
58.             Statement stmt = (Statement) con.createStatement();
59.             String sql = s;
60.             rst = stmt.executeQuery(sql);
61.         } catch (SQLException e) {}
62.         return rst;     }}

```

2. [screenshot.java](#)

In this I have Declared Method for Capturing screenshot

```

1. public class screenshot {
2.     public static void screenshot(double imgcode) {
3.         try {
4.             Rectangle screenRect = null;
5.             screenRect = new Rectangle(Toolkit.getDefaultToolkit().getScreenSi
ze());
6.             BufferedImage capture = new Robot().createScreenCapture(screenRect
);
7.             ImageIO.write(capture, "png", new File(System.getProperty("user.ho
me") + "/EXPHYSICS/" + imgcode + ".png"));
8.         } catch (AWTException | HeadlessException e) {} catch (IOException ex)
{

```

```

9.         Logger.getLogger(Helicalspring.class.getName()).log(Level.SEVERE,
    null, ex);
10.    }}}}

```

3. simage.java

In this I have declared methods for resizing images

```

1. public class simage {
2.     public static Image scaledImage(Image img, int w, int h) {
3.         BufferedImage resizedImage = new BufferedImage(w, h, BufferedImage.TYP
        E_INT_ARGB);
4.         Graphics2D g2 = resizedImage.createGraphics();
5.         g2.setRenderingHint(RenderingHints.KEY_INTERPOLATION, RenderingHints.V
        ALUE_INTERPOLATION_BILINEAR);
6.         g2.drawImage(img, 0, 0, w, h, null);
7.         return resizedImage; }}

```

REFERENCES IN FRAMES

1. CONECTING TO MySQL

```

18  * @author Tushar Jain
19  */
20  public class Login extends javax.swing.JFrame {
21      Connection con;
22      /**
23       * Creates new form NewJFrame
24       */
25      public Login() {
26          initComponents(); con = conection.coondb();
27      }
28

```

Connection to MySQL Database is created by declaring a public connection which is specified in class `conection.java`

`con = conection.coondb();`

2. RESIZE IMAGES

In this a method **scaledImage** is declared which being specified in class `simage.java`

```
43  
44     @SuppressWarnings("unchecked")  
45     Generated Code  
372 private Image scaledImage(Image img ,int w,int h){  
373     return simage.scaledImage(img, w, h);}  
374
```

INSTALLATION[installation.java]



JAVA CODE[Install Button]:-

```
1. try {  
2.     Statement stmt = (Statement) con.createStatement();  
3.     jButton1.setText("INSTALLING");  
4.     conection.exup("create database experiment;", "");  
5.     jProgressBar1.setValue(8);  
6.     conection.exup("create table experiment.userdata(uid varchar(30) primary k  
    ey ,name varchar(70) not null,pass varchar(15) not null);", "");  
7.     jProgressBar1.setValue(24);  
8.     conection.exup("create table experiment.activestatus(uid VARCHAR(30),statu  
    sno integer primary key auto_increment" + ",foreign key(uid) references us  
    erdata(uid)on update cascade on delete cascade);", "");  
9.     jProgressBar1.setValue(32);  
10.    String sql5 = "create table experiment.cvexm(uid VARCHAR(30),eid var  
    char(1000), edate date" + ",focus integer,obdis decimal(10,3),imgdis decim
```

```

        al(10,3),mgficon decimal(10,3),img varchar(100) ," + "foreign key(uid) r
        eferences userdata(uid)on update cascade on delete cascade);";
11.      stmt.executeUpdate(sql5);
12.      jProgressBar1.setValue(40);
13.      String sql6 = "create table experiment.cavem(uid VARCHAR(30),eid var
        char(1000), edate date" + ",focus integer,obdis decimal(10,3),imgdis decim
        al(10,3),mgficon decimal(10,3),img varchar(100) ," + "foreign key(uid) r
        eferences userdata(uid)on update cascade on delete cascade);";
14.      stmt.executeUpdate(sql6);
15.      jProgressBar1.setValue(48);
16.      String sql7 = "create table experiment.cvexl(uid VARCHAR(30),eid var
        char(1000), edate date" + ",focus integer,obdis decimal(10,3),imgdis decim
        al(10,3),
17.      mgficon decimal(10, 3), img varchar(100), "+ "
18.      foreign key(uid) references userdata(uid) on update cascade
        on delete cascade);
19. ";
20. stmt.executeUpdate(sql7);
21. jProgressBar1.setValue(56);
22. String sql8 = "create table experiment.cavel(uid VARCHAR(30), eid varcha
        r(1000),edate date" + ",focus integer,obdis decimal(10,3),imgdis decimal(1
        0,3),mgficon decimal(10,3)" + ",img varchar(100) ,foreign key(uid) refer
        ences userdata(uid)on update cascade on delete cascade);";
23. stmt.executeUpdate(sql8);
24. jProgressBar1.setValue(64);
25. String sql9 = "create table experiment.mbridge(uid VARCHAR(30),eid varch
        ar(1000),edate date" + ",la integer(2),lb integer(2),rbresist integer,ures
        ist integer,rmeter decimal(30,5)" + ",rammeter decimal(30,5),vol decimal(1
        0,4),img varchar(100) ," + "foreign key(uid) references userdata(uid)on u
        pdate cascade on delete cascade );";
26. stmt.executeUpdate(sql9);
27. jProgressBar1.setValue(72);
28. String sql10 = "create table experiment.helicalspring(uid VARCHAR(30),ei
        d varchar(1000)" + ",edate date,wght integer(8),extension decimal(30,10),i
        mge varchar(100)" + ",foreign key(uid) references userdata(uid)on update c
        ascade on delete cascade );";
29. stmt.executeUpdate(sql10);
30. jProgressBar1.setValue(80);
31. String sql11 = "create table experiment.install(statusno integer(1));";
32. stmt.executeUpdate(sql11);
33. jProgressBar1.setValue(88);
34. String sql12 = "insert into experiment.install values(1);";
35. String sql13 = "insert into experiment.userdata values(1000,'Admin','adm
        in');";
36. stmt.executeUpdate(sql12);
37. stmt.executeUpdate(sql13);
38. jProgressBar1.setValue(100);
39. jButton1.setText("INSTALLED");
40. JOptionPane.showMessageDialog(this, "Database installed successfully");

41. File dir = new File(System.getProperty("user.home") + "/EXPHYSICS"
        );
42. dir.mkdir();
43. }

```



```

44. catch (SQLException e) {
45.     JOptionPane.showMessageDialog(this, "Unable to install");
46.     jButton1.setText("INSTALL");
47.     conection.exup("drop database experiment", "");
48.     JOptionPane.showMessageDialog(this, e.getMessage());
49. }
50. try {
51.     int statusno = 0;
52.     ResultSet rs = conection.rs("select * from experiment.install;", "
");
53.     if (rs.next()) {
54.         statusno = Integer.parseInt(rs.getString("statusno"));
55.         if (statusno == 1) {
56.             new Login().setVisible(true);
57.             this.dispose();
58.         }
59.     }
60. }
61. catch (Exception e) {
62.     jButton1.setText("INSTALL");
63.     jProgressBar1.setValue(0);
64. }

```

JAVA CODE[Cancel]: -

```

1. System.exit(0);

```

JAVA CODE [Window focus Gained]: -

```

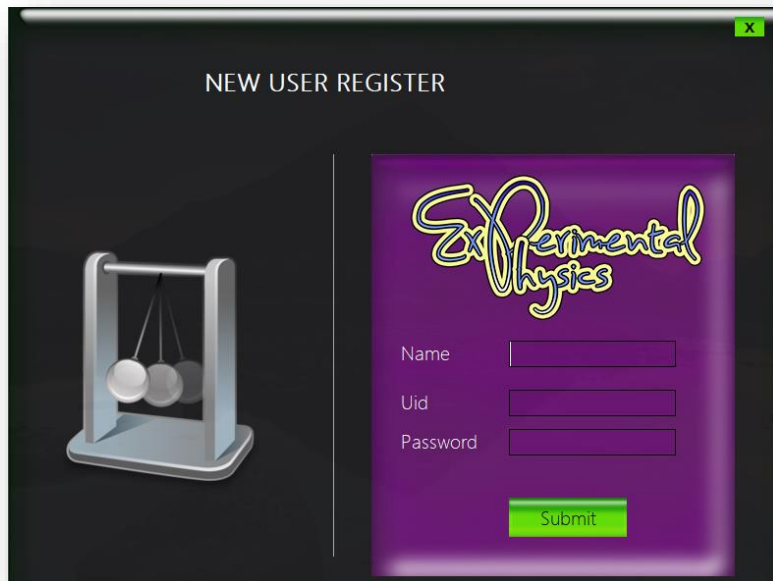
1. try {
2.     int statusno = 0;
3.     ResultSet rs = conection.rs("select * from experiment.install;", "");
4.     if (rs.next()) {
5.         statusno = Integer.parseInt(rs.getString("statusno"));
6.         if (statusno == 1) {
7.             new Login().setVisible(true);
8.             this.dispose();
9.         }
10.    }
11. }
12. catch (Exception e) {
13.     jButton1.setText("INSTALL");
14.     jProgressBar1.setValue(0);
15. }

```

NEW USER LOGIN[newuserlogin.java]

JAVA CODE[Submit Button]: -

```
1. String uid = jTextField3.getText();
2. String Name = jTextField2.getText();
3. String pass = new String(jPasswordField1.getPassword());
4. if (found == 0) {
5.     conection.exup("INSERT INTO userdata VALUES('" + uid + "','" + Name
6.         + "','" + pass + "')";, "experiment");
7.     jTextField1.setText("");
8.     jTextField2.setText("");
9.     JOptionPane.showMessageDialog(this, "You have been successfully registere
10.         d");
11.     this.dispose();
12.     new Login().setVisible(true);
13. }
14. else if (found > 0) JOptionPane.showMessageDialog(this, "Someone else is
15.     having the same UID");
```



JAVA CODE[Uid Text Field Caret Update]: -

```
1. found = 0;
2. try {
3.     String uid = jTextField3.getText();
4.     ResultSet rst = conection.rs("select* from userdata where uid='" + ui
5.         d + "'", "experiment");
6.     while (rst.next()) {
```

```

6.      String uid2 = rst.getString("uid");
7.      if (uid.equals(uid2)) {
8.          found++;
9.      }
10.     }
11.     if (found == 0) {
12.         jLabel12.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/right.png")));
13.     }
14.     else if (found > 0) {
15.         jLabel12.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/cancel.png")));
16.     }
17. }
18. catch (SQLException | HeadlessException e) {}

```

JAVA CODE[name Text Field Caret Update]: -

```

1. String name = jTextField2.getText();
2. if (name.isEmpty()) {
3.     jLabel13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/cancel.png")));
4. }
5. else if (!name.isEmpty()) jLabel13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/right.png")));

```

JAVA CODE [X Label Mouse Clicked]: -

```

1. System.exit(0);

```

LOGIN [login.java]

JAVA CODE[Window focus Gained]: -

```

1. try {
2.     int statusno = 0;
3.     Statement stmt = con.createStatement();
4.     String sql = "select * from experiment.install;";
5.     ResultSet rs = stmt.executeQuery(sql);
6.     if (rs.next()) {

```

```

7.         statusno = Integer.parseInt(rs.getString("statusno"));
8.     }
9. }
10. catch (Exception e) {
11.     new installation().setVisible(true);
12.     this.dispose();
13. }

```



JAVA CODE[Login Button]: -

```

1. String uid = jTextField1.getText();
2. String pwd = new String(jPasswordField1.getPassword());
3. try {
4.     ResultSet rs = conection.rs("select * from userdata where uid= '" + uid +
        "'", "experiment");
5.     if (rs.next()) {
6.         String uidreceved = rs.getString("uid");
7.         String pass = rs.getString("pass");
8.         String name = rs.getString("name");
9.         if (uid.equals(uidreceved) && pwd.equals(pass)) {
10.            JOptionPane.showMessageDialog(this, "welcome " + name);
11.            conection.exup("insert into activestatus values ('" + uidreceved +
                "',null);", "experiment");
12.            new Mainmenu().setVisible(true);
13.            this.dispose();
14.        } else JOptionPane.showMessageDialog(this, "invalid user");
15.    } else JOptionPane.showMessageDialog(this, "invalid user");

```

```
16. } catch (SQLException | HeadlessException e) {}
```

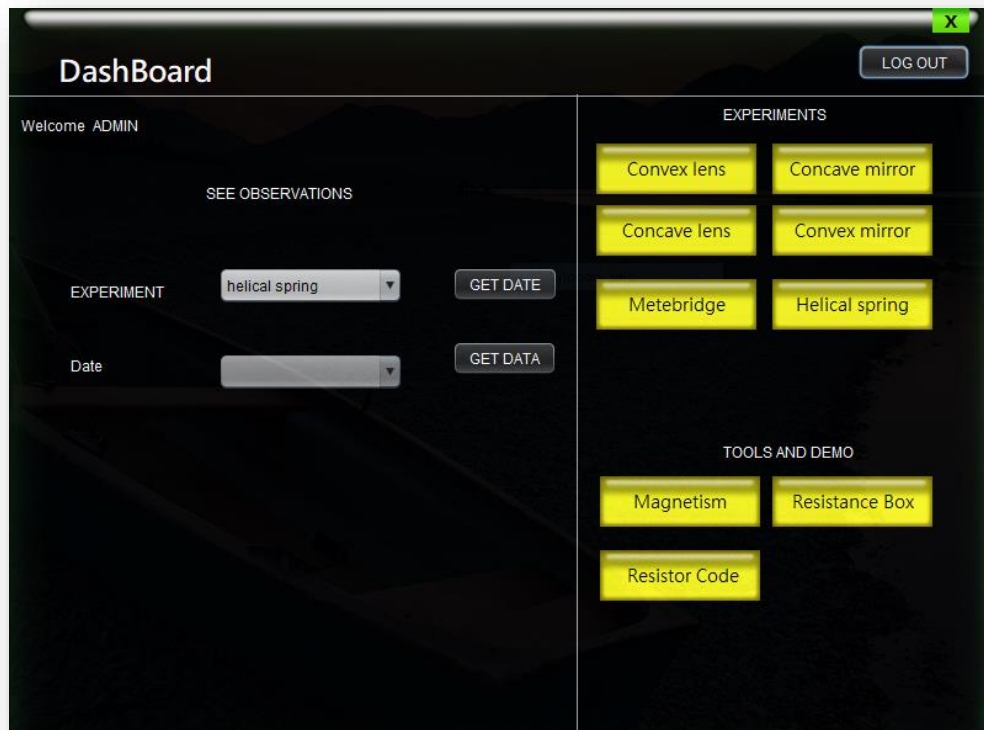
[JAVA CODE\[Window focus Gained\]: -](#)

```
1. new newuserlogin().setVisible(true);  
2. this.dispose();
```

[JAVA CODE\[X Label Mouse Clicked\]: -](#)

```
1. System.exit(0);
```

Dashboard[Mainmenu.java]



[JAVA CODE\[Get Date Button\]: -](#)

```
1. jComboBox1.removeAllItems();  
2. int m = jComboBox3.getSelectedIndex();  
3. String table = null;  
4. switch (m) {  
5.     case 0:  
6.         table = "helicalspring";  
7.         break;  
8.     case 1:
```

```

9.         table    = "cvexl";
10.        break;
11.    case    2:
12.        table    = "cavel";
13.        break;
14.    case    3:
15.        table    = "cvexm";
16.        break;
17.    case    4:
18.        table    = "cavem";
19.        break;
20.    case    5:
21.        table    = "mbridge";
22.        break;
23. }
24. try    {
25.     ResultSet rs4 = conection.rs(" select distinct edate from " + table
        + " where uid = '" + conection.uid() + "'", "experiment");

26.     while    (rs4.next()) { String    date = rs4.getString("edate");

27.         jComboBox1.addItem(date);
28.     }
29. }
30. catch    (SQLException|HeadlessException    e)    {
31.     System.out.println(e.getMessage());
32. }

```

JAVA CODE[Log Out Button]: -

```

1. conection.exup("delete from activestatus", "experiment");
2. new Login().setVisible(true);
3. this.dispose();

```

JAVA CODE[Convex lens]

```

1. new convxlens().setVisible(true);
2. this.dispose();

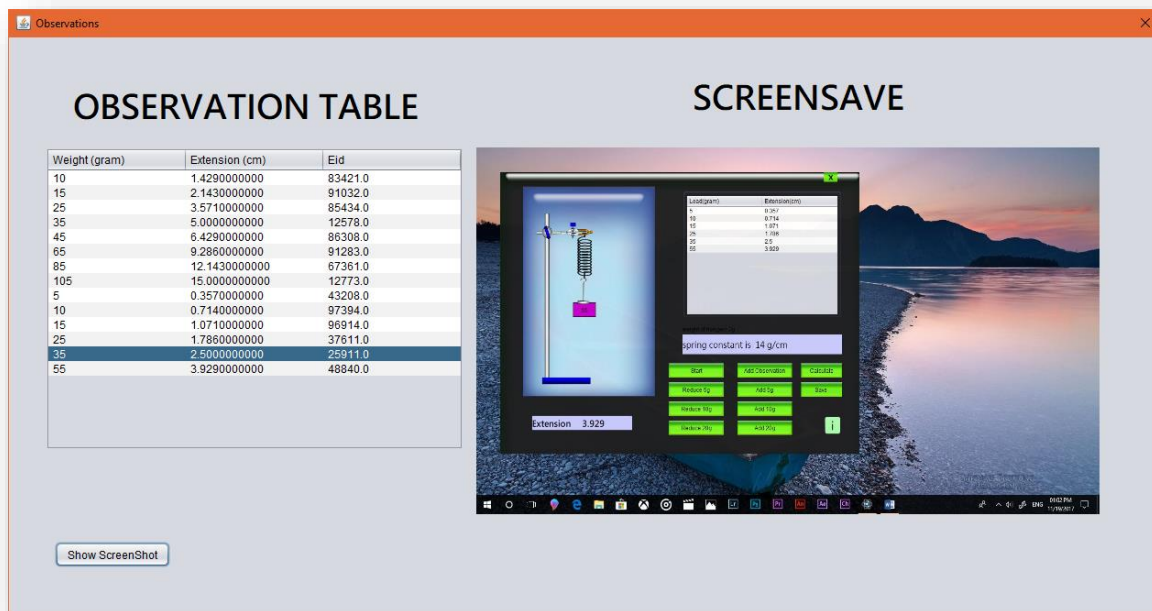
```

CODE[Concave lens]

```

1. new Concavelens().setVisible(true);
2. this.dispose();

```



JAVA CODE[Concave mirror]

1. `new Concavemirror().setVisible(true);`
2. `this.dispose();`

JAVA CODE[Convex mirror]

1. `new convexmirror().setVisible(true);`
2. `this.dispose();`

JAVA CODE[meterbridge]

1. `new mbridge().setVisible(true);`

JAVA CODE[Helical spring]

1. `new Helicalspring().setVisible(true);`
2. `this.dispose`

JAVA CODE[magnetism]

1. `new MFWIREFINITE().setVisible(true);`

JAVA CODE[resistance box]

```
1. new resistancebox().setVisible(true);
```

JAVA CODE[resistor code]

```
1. new Rsistorcode().setVisible(true);  
2. this.dispose();
```

JAVA CODE[Get Data Button]:-

```
1. jLabel14.setIcon(null);  
2. DefaultTableModel model = (DefaultTableModel) jTable2.getModel();  
3. jTable2.removeAll();  
4. model.setColumnCount(0);  
5. model.setRowCount(0);  
6. int m = jComboBox3.getSelectedIndex();  
7. String table = null;  
8. switch (m) {  
9.     case 0:  
10.         table = "helicalspring";  
11.         break;  
12.     case 1:  
13.         table = "cvexl";  
14.         break;  
15.     case 2:  
16.         table = "cavel";  
17.         break;  
18.     case 3:  
19.         table = "cvexm";  
20.         break;  
21.     case 4:  
22.         table = "cavem";  
23.         break;  
24.     case 5:  
25.         table = "mbridge";  
26.         break;  
27. }  
28. switch (m) {  
29.     case 0:  
30.         table = "helicalspring";  
31.         model.addColumn("Weight (gram)");  
32.         model.addColumn("Extension (cm)");  
33.         model.addColumn("Eid");  
34.         break;  
35.     case 1:  
36.     case 2:  
37.     case 3:  
38.     case 4:  
39.         model.addColumn("Focus");  
40.         model.addColumn("Object Distance(cm)");
```



```

94.             break;
95.         case 5:
96.             while (rs4.next()) {
97.                 String focus = rs4.getString("la");
98.                 String od = rs4.getString("lb");
99.                 String id = rs4.getString("rbresist");
100.                String mg = rs4.getString("uresist");
101.                String volt = rs4.getString("vol");
102.                String mwire = rs4.getString("rmeter");
103.                String ampere = rs4.getString("rammeter");
104.                String eid = rs4.getString("eid");
105.                model.addRow(new Object[] {
106.                    focus, od, id, mwire, ampere, volt, mg, eid
107.                });
108.            }
109.            break;
110.        }
111.    }
112. }
113. } catch (SQLException | HeadlessException e) {
114.     System.out.println(e.getMessage());
115. }
116. jDialog1.setVisible(true);

```

JAVA CODE[Show Screenshot]:-

```

1. DefaultTableModel model = (DefaultTableModel) jTable2.getModel();
2. System.out.println(jTable2.getValueAt(jTable2.getSelectedRow(), 2));
3. int n = 0;
4. int m = jComboBox3.getSelectedIndex();
5. String table = null;
6. switch (m) {
7.     case 0:
8.         table = "helicalspring";
9.         n = 2;
10.        break;
11.     case 1:
12.         table = "cvexl";
13.         n = 4;
14.        break;
15.     case 2:
16.         table = "cavel";
17.         n = 4;
18.        break;
19.     case 3:
20.         table = "cvexm";
21.         n = 4;
22.        break;
23.     case 4:
24.         table = "cavem";
25.         n = 4;
26.        break;
27.     case 5:

```

```

28.         table = "mbridge";
29.         n = 7;
30.         break;
31.     }
32.     try {
33.         ResultSet rs = conection.rs("select imge from experiment." + table +
            " where eid='" + jTable2.getValueAt(jTable2.getSelectedRow(), n) + "'",
            "experiment");
34.         if (rs.next()) {
35.             BufferedImage pic = null;
36.             try {
37.                 pic = ImageIO.read(new File(System.getProperty("user.home") +
                    "/EXPHYSICS/" + rs.getString("imge") + ".png"));
38.                 ImageIcon icon = new ImageIcon(scaledImage(pic, jLabel14.getW
                    idth(), jLabel14.getHeight()));
39.                 jLabel14.setIcon(icon);
40.             }
41.             catch (IOException e) {
42.                 JOptionPane.showMessageDialog(this, e.getMessage());
43.             }
44.         }
45.     }
46.     catch (SQLException e) {}

```

Helical Spring[helicalspring.java]

JAVA CODE[Start Button]: -

```

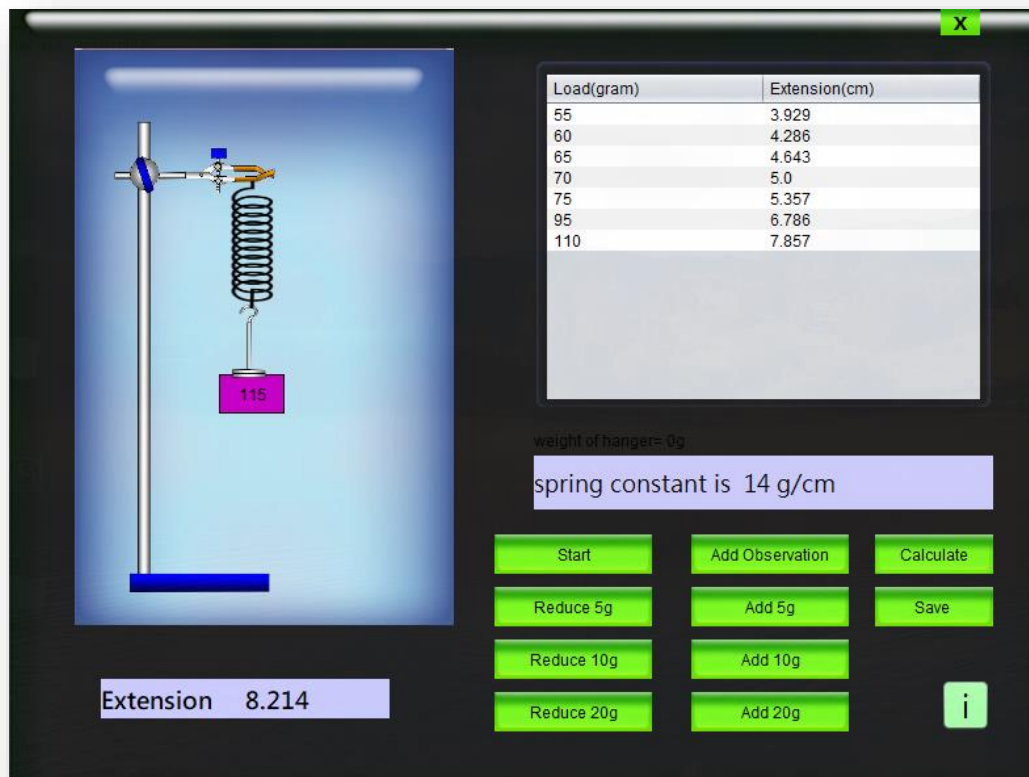
1. int k = (int)(Math.round(Math.random() * 100));
2. while (k > 15 || k == 0 || k < 7) k = (int)(Math.round(Math.random() * 100));
3. jLabel4.setText("" + k);
4. weightlabel.setText("0");
5. jTable1.removeAll();
6. jButton1.setEnabled(true);
7. jButton2.setEnabled(true);
8. jButton3.setEnabled(true);
9. jButton4.setEnabled(true);
10. jButton5.setEnabled(true);
11. jButton6.setEnabled(true);
12. jButton9.setEnabled(true);
13. jButton8.setEnabled(true);
14. BufferedImage pic = null;
15. try {
16.     pic = ImageIO.read(getClass().getResource("/data/SPRING.PNG"));
17.     ImageIcon icon = new ImageIcon(simage.scaledImage(pic, jLabel2.getWidth(),
        jLabel2.getHeight()));
18.     jLabel2.setIcon(icon);
19. }
20. catch (IOException e) {

```

```

21. JOptionPane.showMessageDialog(this, e.getMessage());
22. }
23. int y = 90;
24. jLabel3.setLocation(jLabel3.getX(), 180);
25. weightlabel.setLocation(weightlabel.getX(), 240); jLabel2.setBounds(jLabel2.
26. getX(), jLabel2.getY(), jLabel2.getWidth(), y);

```



JAVA CODE[WeightLabel PropertyChanged]:-

```

1. if (Integer.parseInt(weightlabel.getText()) < 0) weightlabel.setText("0");
2. BufferedImage pic = null;
3. try {
4.     pic = ImageIO.read(getClass().getResource("/data/SPRING.PNG"));
5.     ImageIcon icon = new ImageIcon(simage.scaledImage(pic, jLabel2.getWidth(),
6.         jLabel2.getHeight()));
7.     jLabel2.setIcon(icon);
8. } catch (IOException e) {
9.     JOptionPane.showMessageDialog(this, e.getMessage());
10. }
11. int y1 = (int) Math.round(Integer.parseInt(weightlabel.getText()) / Integer.pa
12.     rseInt(jLabel4.getText()));
13. Double extension = (double) Math.round((Double.parseDouble(weightlabel.getText
14.     ()) * 1000) / Double.parseDouble(jLabel4.getText())) / 1000;
15. jLabel11.setText(extension + "");
16. int y = y1 + 90;

```

```

14. jLabel3.setLocation(jLabel3.getX(), y1 + 180);
15. weightlabel.setLocation(weightlabel.getX(), y1 + 240);
16. jLabel2.setBounds(jLabel2.getX(), jLabel2.getY(), jLabel2.getWidth(), y);

```

JAVA CODE[Add Observation button]: -

```

1. DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
2. model.addRow(new Object[] {
3.     weightlabel.getText(), jLabel11.getText()
4. });

```

JAVA CODE[Calculate button]: -

```

1. if (jTable1.getRowCount() > 5) jLabel10.setText("spring constant is " + jLabel
4.getText() + " g/cm");
2. else jLabel10.setText("Add more observation");

```

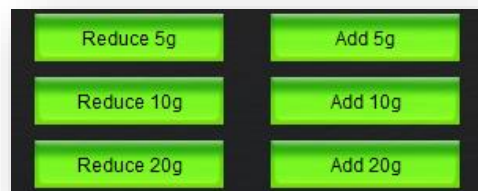
JAVA CODE[Save button]:-

```

1. Double imgcode = (Math.random() * 1000);
2. screenshot.screenshot(imgcode);
3. int n = 1;
4. int m = 0;
5. DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
6. while (n <= model.getRowCount()) {
7.     String a = String.valueOf(model.getValueAt(n - 1, 0));
8.     String b = String.valueOf(model.getValueAt(n - 1, 1));
9.     conection.exup("insert into helicalspring values('" + conection.uid() +
    "',''" + ((double) Math.round(Math.random() * 100000)) + "',''" + "cur
    date()," + a + "," + b + "','" + imgcode + "',''" + "experiment");
10.     n++;
11. }
12. JOptionPane.showMessageDialog(this, "data saved succesfully");

```

JAVA CODE[wchange actionperformed]



All these buttons have same action performed wchange

```
1. int change = Integer.parseInt(evt.getActionCommand());
2. weightlabel.setText((Integer.parseInt(weightlabel.getText()) + change) + "
");
```

Meter Bridge[mbridge.java]

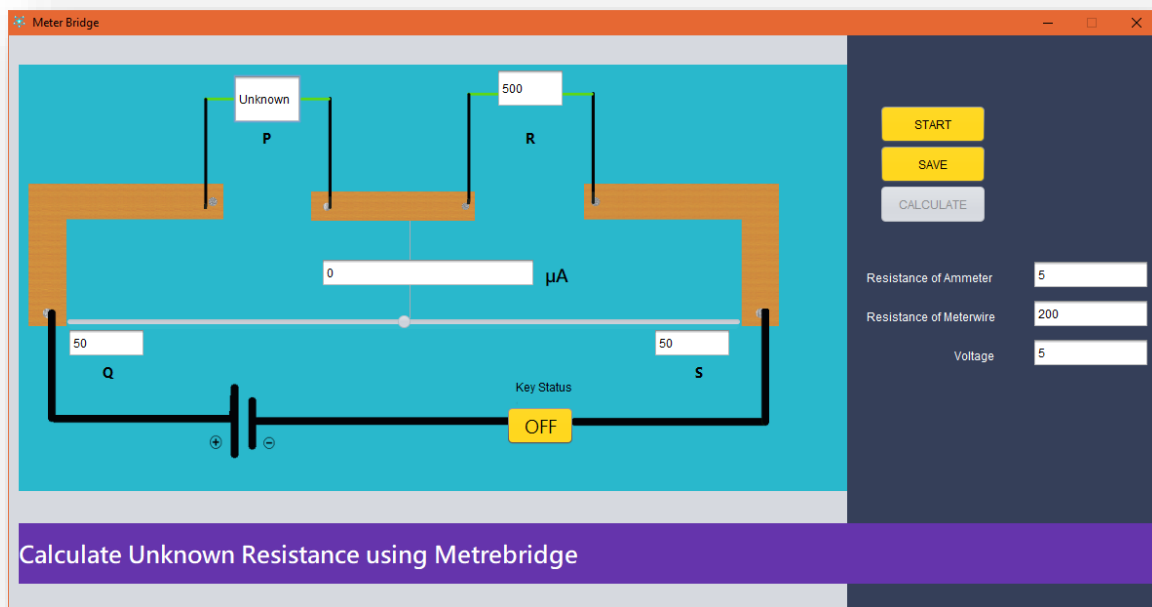
JAVA CODE[jSlider1StateChanged]:-

```
1. int l = jSlider1.getValue();
2. int w = (int) Math.abs(Math.round(350 - 7 * l));
3. int x = (int) Math.round(7 * l + 50);
4. int x2 = jSeparator2.getX();
5. if (x <= 400) {
6.     x2 = x;
7. }
8. else if (x > 400) x2 = 395 + w;
9. if (x >= 400) {
10.    x = 400;
11. }
12. int h = jSeparator3.getHeight();
13. int Y = jSeparator3.getY();
14. int y2 = jSeparator2.getY();
15. jSeparator2.setLocation(x2, y2);
16. jSeparator3.setBounds(x, Y, w, h);
17. jTextField4.setText("" + (100 - l));
18. jTextField5.setText("" + l);
19. double q = Integer.parseInt(jTextField4.getText());
20. double s = Integer.parseInt(jTextField5.getText());
21. double u0 = Double.parseDouble(jTextField7.getText());
22. double rs = Double.parseDouble(jTextField2.getText());
23. double r1 = (rs * q) / 100;
24. double r2 = rs - r1;
25. double r3 = Double.parseDouble(jLabel1.getText());
26. System.out.println(r3);
27. double r4 = Double.parseDouble((resistancebox.getText()));
28. double r5 = Double.parseDouble(jTextField6.getText());
29. double denom = r1 * (r2 * r3 + r2 * r4 + r3 * r4 + r3 * r5
    + r4 * r5) + r2 * (r3 * r4 + r3 * r5 + r4 * r5);
```

```

30. double i = Math.round((r1 * r4 - r2 * r3) * u0 * 1000000 / deno
    m);
31. jTextField3.setText(i + "");
32. if (Math.round(q * 10 / s) == Math.round(r3 * 10 / r4)) {
33.     jLabel15.setText("Now You Can Calculate Unknown Resistance");
34.     jButton4.setEnabled(true);
35. }
36. else {
37.     jLabel15.setText("Try to balance bridge by getting minimum value of curren
        t in ammeter(near to zero)");
38.     jButton4.setEnabled(Boolean.FALSE);
39. }

```



JAVA CODE[start button]:-

```

1. jLabel1.setText(Math.round(10 + Math.random() * 990) + "");
2. jTextField3.setText("0");
3. jTextField1.setText("Unknown");
4. jButton1.setText("ON");
5. jSlider1.setEnabled(true);
6. jTextField2.setEditable(false);
7. jTextField7.setEditable(false);
8. jTextField6.setEditable(false);

```

JAVA CODE[save buttton]:-

```

1. double imgcode = (Math.random() * 1000);
2. double rx = Double.parseDouble(jLabel1.getText());
3. double q = Integer.parseInt(jTextField4.getText());
4. double s = Integer.parseInt(jTextField5.getText());

```

```

5. double rb = Double.parseDouble(resistancebox.getText());
6. double u0 = Double.parseDouble(jTextField7.getText());
7. double rs = Double.parseDouble(jTextField2.getText());
8. double r5 = Double.parseDouble(jTextField6.getText());
9. screenshot.screenshot(imgcode);
10. conection.exup("INSERT INTO mbridge VALUES('" + conection.uid() + "','" +
    ((double) Math.round(Math.random() * 100000)) + "','curdate()'," + q +
    "," + s + "," + rb + "," + rx + "," + rs + "," + r5 + "," +
    " + u0 + "','" + imgcode + "')", "experiment");
11. JOptionPane.showMessageDialog(this, "Data saved successfully!");

```

JAVA CODE [Calculate button]: -

```

1. jLabel15.setText("Unknown resistance is " + jLabel1.getText() + " ohm");
2. jTextField1.setText(jLabel1.getText());
3. jButton1.setEnabled(true);
4. jButton1.setText("OFF");
5. jSlider1.setEnabled(false);
6. jButton4.setEnabled(false);
7. jTextField2.setEditable(true);
8. jTextField7.setEditable(true);
9. jTextField6.setEditable(true);

```

JAVA CODE[On/Off button]:-

```

1. if (jButton1.getText().equals("OFF")) {
2.     jButton1.setText("ON");
3. } else if (jButton1.getText().equals("ON")) {
4.     jButton1.setText("OFF");
5. }

```

JAVA CODE[Resistance Box label mouseclicked]:-

```

1. new resistancebox().setVisible(true);

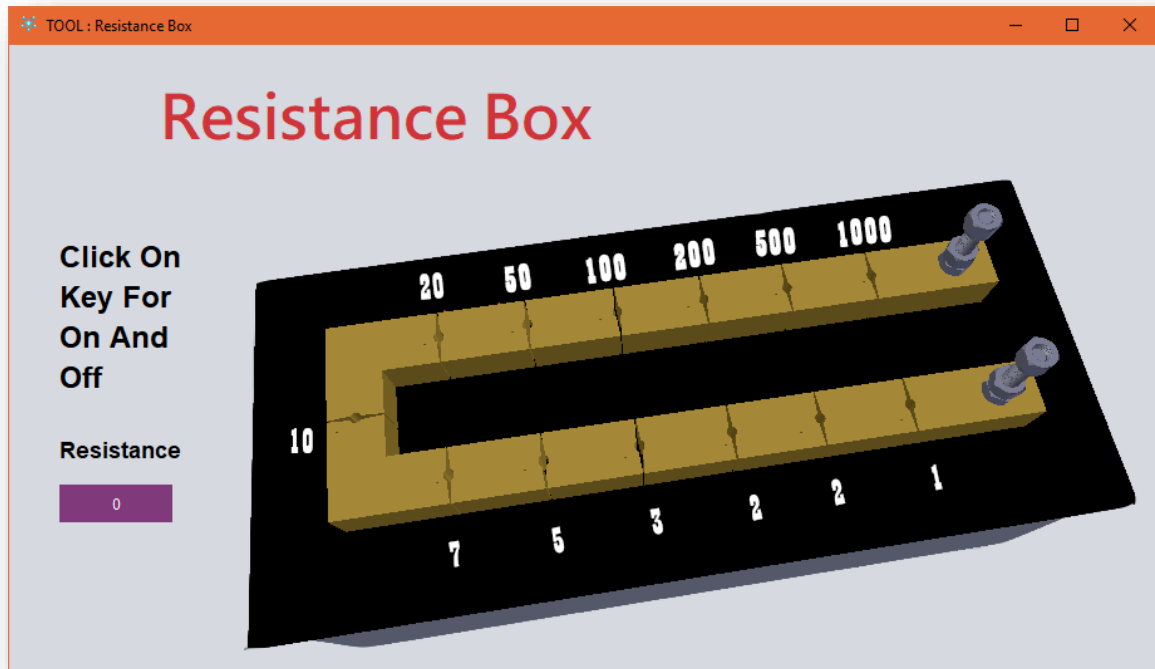
```

RESISTANCE BOX[resistancebox.java]

Algorithm

I have Created 13 JLabels and named them as r[x] wher x is there value

And set there text value to 0 signifiying off once user clicks them values is changed to 1 signifiying on and there value adds to a textfield



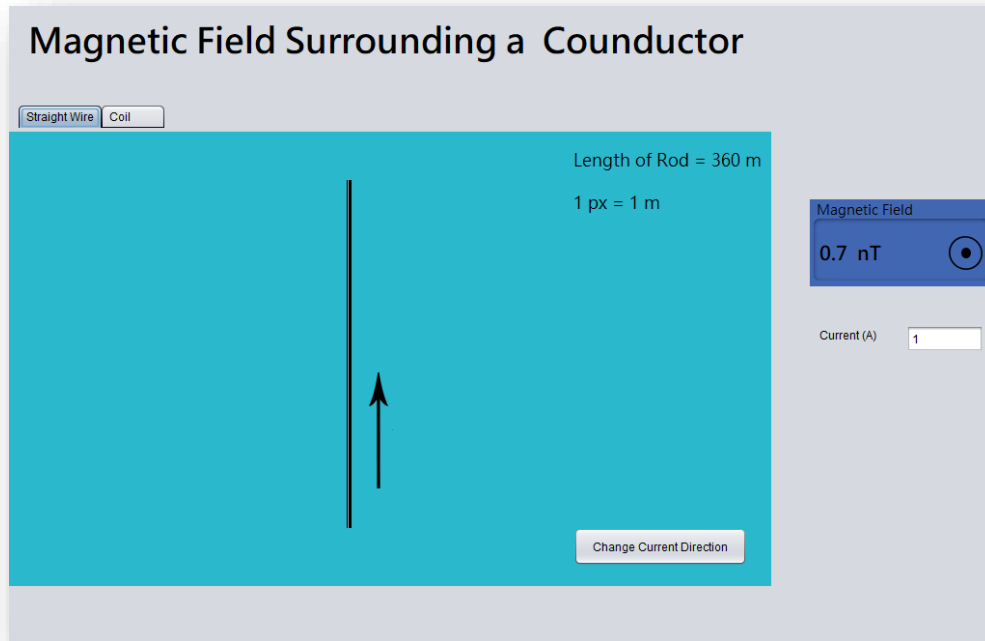
JAVA CODE[r1 label MouseClicked]: -

```
1. if (r1.getText().equals("off")) {
2.     r1.setText("on");
3.     r1.setBackground(Color.BLACK);
4.     r1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/key.png")));
5.     value.setText("" + (Integer.parseInt(value.getText()) + 1));
6. } else if (r1.getText().equals("on")) {
7.     r1.setText("off");
8.     r1.setIcon(null);
9.     value.setText("" + (Integer.parseInt(value.getText()) - 1));
10. }
11. mbridge.rbox2 = Integer.parseInt(value.getText());
```

NOTE: -

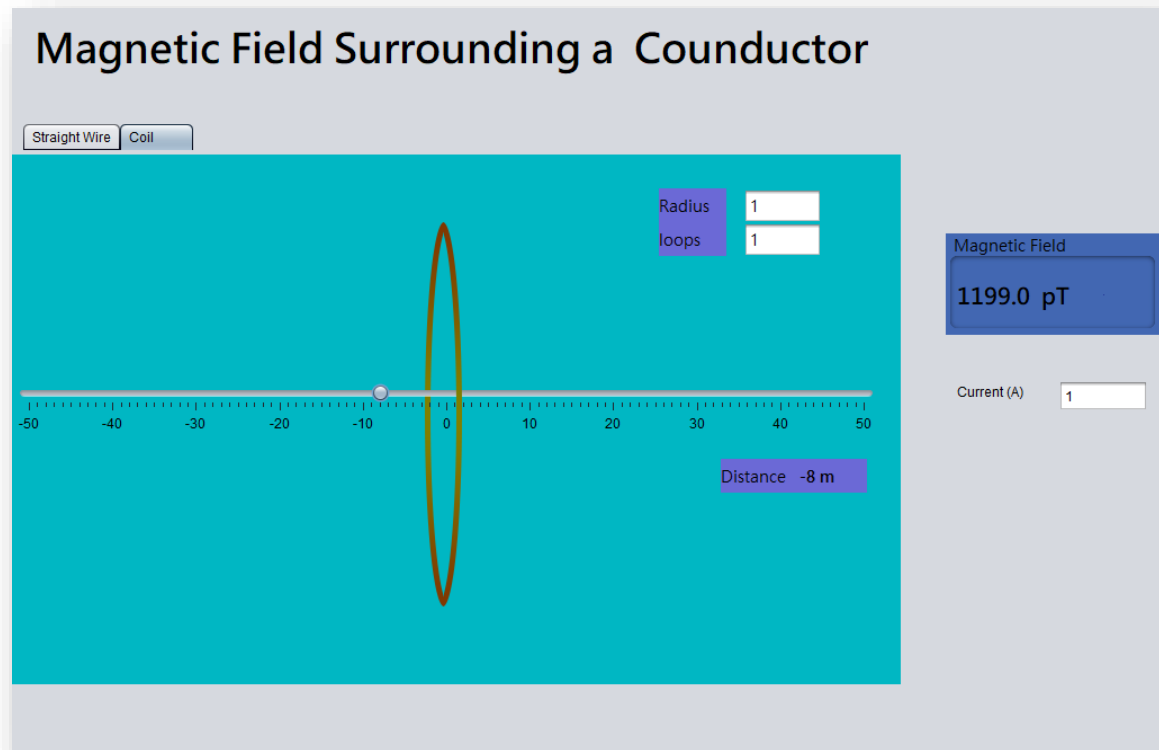
java code for other jlabel mouse click is written in simmilar manner by just chnging r1 to r2, r3, etc.

Magnetic field[MFWIREFINITE.java]



JAVA CODE[jPanel1MouseClicked]: -

```
1. jTabbedPane1.repaint();
2. int I = Integer.parseInt(jTextField1.getText());
3. int x = evt.getX();
4. int y = evt.getY();
5. int x2 = jSeparator1.getX();
6. int y2 = jSeparator1.getY();
7. int d = Math.abs(x - x2);
8. double A = Math.sqrt((y - 50) * (y - 50) + d * d);
9. double sina = d / A;
10. double B = Math.sqrt((410 - y) * (410 - y) + d * d);
11. double sinb = d / B;
12. double field = Math.round(I * (sina + sinb) * 1000 / d);
13. jLabel14.setText("" + field / 10 + " nT");
14. Graphics2D g = (Graphics2D) getGraphics();
15. g.setColor(Color.red);
16. g.drawLine(x2 + 10, y2 + 155, x + 10, y + 155);
17. g.drawLine(x2 + 10, y2 + 155 + 360, x + 10, y + 155);
18. if (jLabel13.getText().equals("0")) {
19.     jLabel16.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/cu
    rrentdir.png")));
20. } else if (jLabel13.getText().equals("1")) {
21.     jLabel16.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/cu
    rrentdir2.png"))); }
```



JAVA CODE[jSlider1StateChanged]: -

```

1. int I = Integer.parseInt(jTextField1.getText());
2. int r = jSlider1.getValue();
3. int rad = Integer.parseInt(jTextField3.getText());
4. double di = 2 * Math.sqrt(Math.pow(rad * rad + r * r, 3));
5. double field = Math.round((I * Integer.parseInt(jTextField2.getText()) * rad *
    rad * 400000 * Math.PI) / di);
6. jLabel4.setText("" + field + "pT");
7. jLabel14.setText("" + r + " m");

```

JAVA CODE[jTabbedPane1MouseClicked]: -

```

1. jLabel4.setText("");
2. jLabel6.setIcon(null);

```

JAVA CODE[Change current direction button]: -

```

1. if (jLabel3.getText().equals("0")) {
2.     jLabel6.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/cu
    rrentdir2.png")));
3.     jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/Au
    p.png")));

```

```

4.     jLabel13.setText("1");
5. } else if (jLabel13.getText().equals("1")) {
6.     jLabel16.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/cu
rrentdir.png")));
7.     jLabel13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/data/Ad
own.png")));
8.     jLabel13.setText("0");
9. }

```

Resistor Color Code[Rsistorcode.java]

Public Variables And Methods:-

```

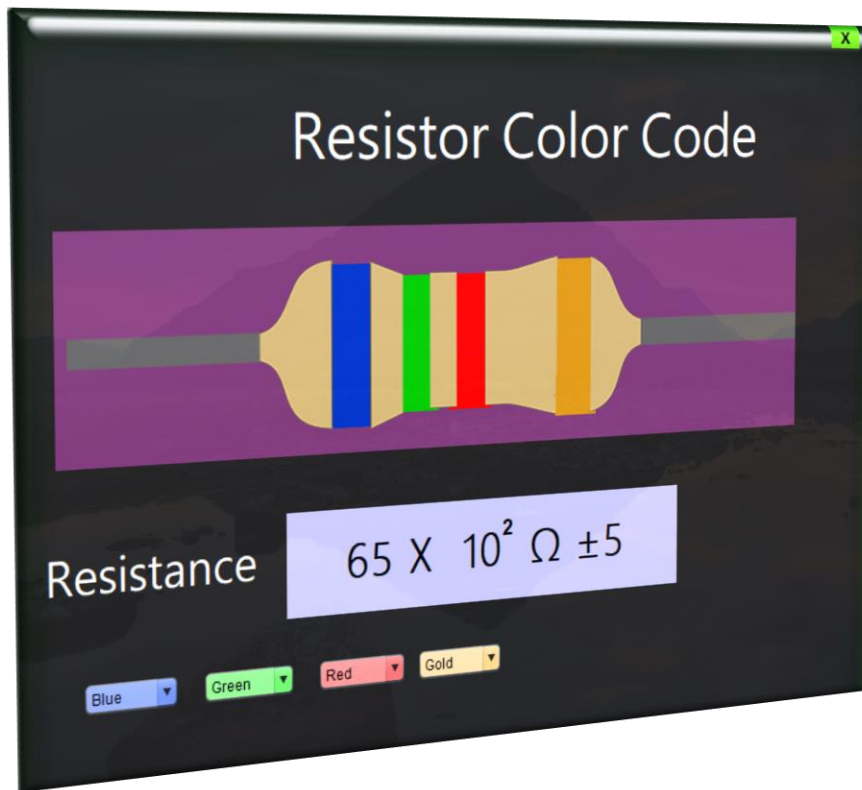
1. Color c0 = new java.awt.Color(0, 0, 0);
2. Color c1 = new java.awt.Color(153, 51, 0);
3. Color c2 = new java.awt.Color(255, 0, 0);
4. Color c3 = new java.awt.Color(255, 153, 0);
5. Color c4 = new java.awt.Color(255, 255, 0);
6. Color c5 = new java.awt.Color(0, 204, 0);
7. Color c6 = new java.awt.Color(0, 51, 204);
8. Color c7 = new java.awt.Color(153, 51, 153);
9. Color c8 = new java.awt.Color(153, 153, 153);
10. Color c9 = new java.awt.Color(255, 255, 255);
11. Color c10 = new java.awt.Color(226, 153, 20);
12. Color c11 = new java.awt.Color(205, 205, 205);
13. Color c = null;
14. public Color c(int n) {
15.     switch (n) {
16.         case 0:
17.             c = c0;
18.             break;
19.         case 1:
20.             c = c1;
21.             break;
22.         case 2:
23.             c = c2;
24.             break;
25.         case 3:
26.             c = c3;
27.             break;
28.         case 10:
29.             c = c10;
30.             break;
31.         case 11:
32.             c = c11;
33.             break;
34.         case 4:
35.             c = c4;
36.             break;
37.         case 5:
38.             c = c5;
39.             break;
40.         case 6:
41.             c = c6;
42.             break;
43.         case 7:
44.             c = c7;
45.             break;

```

```

46.         case 8:
47.             c = c8;
48.             break;
49.         case 9:
50.             c = c9;
51.             break;
52.     }
53.     return c;
54. }

```



JAVA CODE[Jcombo box1 ItemStateChanged]:-

```

1. int n = jComboBox1.getSelectedIndex();
2. jComboBox1.setBackground(c(n));
3. jLabel13.setBackground(c(n));
4. int n1 = jComboBox1.getSelectedIndex();
5. int n2 = jComboBox3.getSelectedIndex();
6. int n3 = jComboBox2.getSelectedIndex();
7. int n4 = jComboBox4.getSelectedIndex();
8. switch (n4) {
9.     case 0:
10.         n4 = 1;
11.         break;
12.     case 1:
13.         n4 = 2;
14.         break;
15.     case 2:
16.         n4 = 5;
17.         break;
18.     case 3:

```

```

19.         n4 = 10;
20.         break;
21.     case 4:
22.         n4 = 20;
23.         break;
24. }
25. jLabel17.setText("" + n1 + n2 + " X " + " 10" + " Ω" + " ±" + n4);
26. jLabel19.setText("" + n3);

```

JAVA CODE[Jcombo box2 ItemStateChanged]:-

```

1. int n = jComboBox2.getSelectedIndex();
2. jComboBox2.setBackground(c(n));
3. jLabel15.setBackground(c(n));
4. int n1 = jComboBox1.getSelectedIndex();
5. int n2 = jComboBox3.getSelectedIndex();
6. int n3 = jComboBox2.getSelectedIndex();
7. int n4 = jComboBox4.getSelectedIndex();
8. switch (n4) {
9.     case 0:
10.         n4 = 1;
11.         break;
12.     case 1:
13.         n4 = 2;
14.         break;
15.     case 2:
16.         n4 = 5;
17.         break;
18.     case 3:
19.         n4 = 10;
20.         break;
21.     case 4:
22.         n4 = 20;
23.         break;
24. }
25. jLabel17.setText("" + n1 + n2 + " X " + " 10" + " Ω" + " ±" + n4);
26. jLabel19.setText("" + n3);

```

JAVA CODE[Jcombo box3ItemStateChanged]:-

```

1. int n = jComboBox3.getSelectedIndex();
2. jComboBox3.setBackground(c(n));
3. jLabel12.setBackground(c(n));
4. int n1 = jComboBox1.getSelectedIndex();
5. int n2 = jComboBox3.getSelectedIndex();
6. int n3 = jComboBox2.getSelectedIndex();
7. int n4 = jComboBox4.getSelectedIndex();
8. switch (n4) {
9.     case 0:

```

```

10.         n4 = 1;
11.         break;
12.     case 1:
13.         n4 = 2;
14.         break;
15.     case 2:
16.         n4 = 5;
17.         break;
18.     case 3:
19.         n4 = 10;
20.         break;
21.     case 4:
22.         n4 = 20;
23.         break;
24. }
25. jLabel17.setText("" + n1 + n2 + " X " + " 10" + " Ω" + " ±" + n4);
26. jLabel19.setText("" + n3);

```

JAVA CODE[Jcombo box4 ItemStateChanged]:-

```

1. int n = jComboBox4.getSelectedIndex();
2. switch (n) {
3.     case 0:
4.         jComboBox4.setBackground(new java.awt.Color(153, 51, 0));
5.         jLabel14.setBackground(new java.awt.Color(153, 51, 0));
6.         break;
7.     case 1:
8.         jComboBox4.setBackground(new java.awt.Color(255, 0, 0));
9.         jLabel14.setBackground(new java.awt.Color(255, 0, 0));
10.        break;
11.    case 2:
12.        jComboBox4.setBackground(new java.awt.Color(226, 157, 19));
13.        jLabel14.setBackground(new java.awt.Color(226, 153, 20));
14.        break;
15.    case 3:
16.        jComboBox4.setBackground(new java.awt.Color(205, 205, 205));
17.        jLabel14.setBackground(new java.awt.Color(205, 205, 205));
18.        break;
19.    case 4:
20.        jLabel14.setBackground(new java.awt.Color(217, 187, 122));
21.        break;
22. }
23. int n1 = jComboBox1.getSelectedIndex();
24. int n2 = jComboBox3.getSelectedIndex();
25. int n3 = jComboBox2.getSelectedIndex();
26. int n4 = jComboBox4.getSelectedIndex();
27. switch (n4) {
28.     case 0:
29.         n4 = 1;
30.         break;
31.     case 1:
32.         n4 = 2;

```

```

33.         break;
34.     case 2:
35.         n4 = 5;
36.         break;
37.     case 3:
38.         n4 = 10;
39.         break;
40.     case 4:
41.         n4 = 20;
42.         break;
43. }
44. jLabel17.setText("" + n1 + n2 + " X " + " 10" + " Ω" + " ±" + n4);
45. jLabel19.setText("" + n3);

```

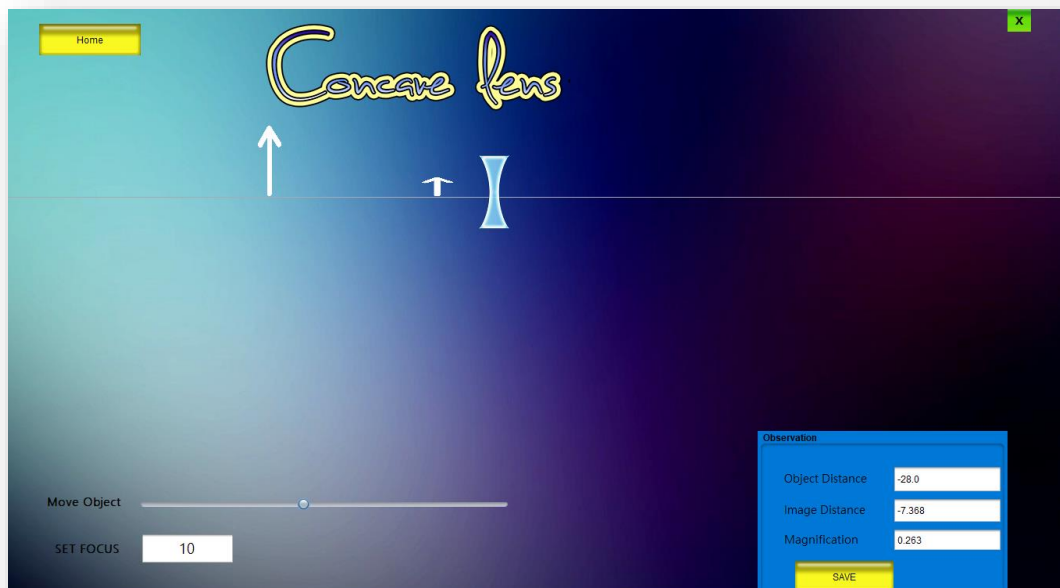
JAVA CODE[Home and X button]: -

```

1. new Mainmenu().setVisible(true);
2. this.dispose();

```

Concave Lens[Concavelens.java]



JAVA CODE[Jslider State changed]: -

```

1. int y1 = jLabel1.getY();
2. double x = 5 * jSlider1.getValue() + 100;

```



```

3. int x2 = (int) Math.round(x);
4. jLabel1.setLocation(x2, y1);
5. int x1 = jLabel1.getX();
6. double u = (x1 - 600) / 10;
7. double f = -1 * Integer.parseInt(jTextField1.getText());
8. double v = u * f / (u + f);
9. double v2 = Math.round(v);
10. int v3 = (int) v2 * (10) + 600;
11. int y2 = jLabel3.getY();
12. jLabel3.setLocation(v3, y1);
13. int heightx = jLabel1.getHeight();
14. double m = Math.abs(v / u);
15. double CASE = v / u;
16. if (CASE >= 0) {
17.     try {
18.         jLabel3.setIcon(new ImageIcon(ImageIO.read(getClass().getResource("/data/OBJECT.png"))));
19.     } catch (IOException ex) {}
20. }
21. double height1 = m * heightx;
22. int height = (int) Math.round(height1);
23. int width = jLabel3.getWidth();
24. if ((v / u) >= 0) jLabel3.setBounds(v3, 240 - height, width, height);
25. else jLabel3.setBounds(v3, y2, width, height);
26. jTextField2.setText("" + ((double) Math.round(u * 1000)) / 1000);
27. jTextField3.setText("" + ((double) Math.round(v * 1000)) / 1000);
28. jTextField4.setText("" + ((double) Math.round((v / u) * 1000)) / 1000);
29. BufferedImage pic = null;
30. BufferedImage pic2 = null;
31. try {
32.     pic = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
33.     if ((v / u) >= 0) pic2 = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
34.     else pic2 = ImageIO.read(getClass().getResource("/data/IMAGE.png"));
35.     ImageIcon icon = new ImageIcon(scaledImage(pic, jLabel1.getWidth(), jLabel1.getHeight()));
36.     ImageIcon icon2 = new ImageIcon(scaledImage(pic2, jLabel3.getWidth(), jLabel3.getHeight()));
37.     jLabel1.setIcon(icon);
38.     jLabel3.setIcon(icon2);
39. } catch (IOException e) {
40.     JOptionPane.showMessageDialog(this, e.getMessage());
41. }

```

JAVA CODE[Home and X button]: -

```

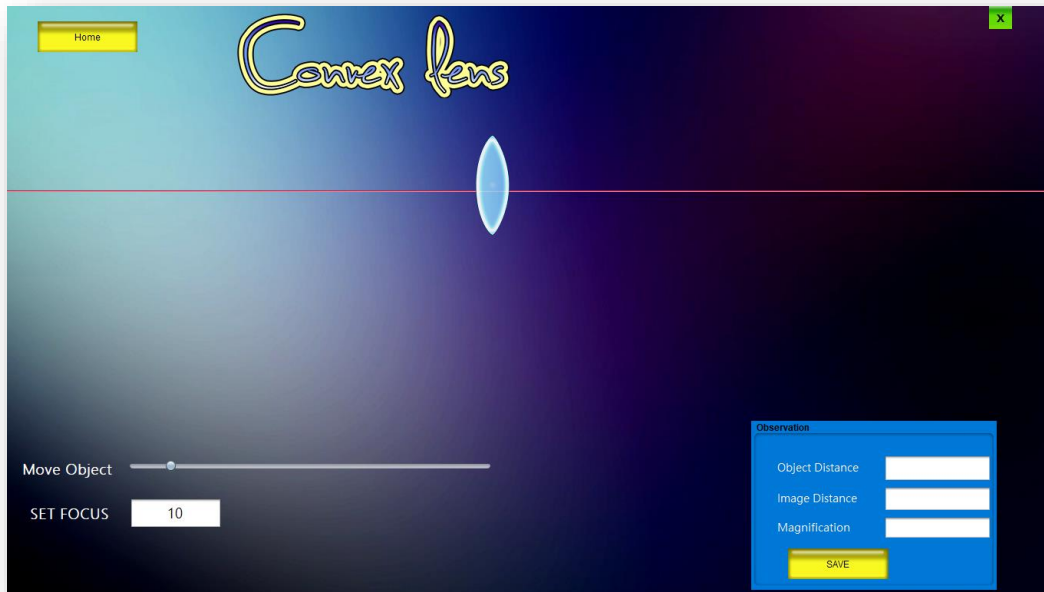
1. new Mainmenu().setVisible(true);
2. this.dispose();

```

JAVA CODE[Save Button]: -

```
1. double imgcode = (Math.random() * 1000);
2. screenshot.screenshot(imgcode);
3. String f = jTextField1.getText();
4. double u = Double.parseDouble(jTextField2.getText());
5. double v = Double.parseDouble(jTextField3.getText());
6. double m = Double.parseDouble(jTextField4.getText());
7. conection.exup("INSERT INTO cavel VALUES('\" + conection.uid() + \"',\" +
  ((double) Math.round(Math.random() * 100000)) + \"',\" + \"curdate(),\" +
  f + \",\" + u + \",\" + v + \",\" + m + \",\" + imgcode + \"')\", \"
  experiment\");
8. JOptionPane.showMessageDialog(this, \"Data saved successfully!\");
```

Convex lens[convexlens.java]



JAVA CODE[Jslider State changed]: -

```
1. int y1 = jLabel1.getY();
2. double x = 5 * jSlider1.getValue() + 100;
3. int x2 = (int) Math.round(x);
4. jLabel1.setLocation(x2, y1);
5. double u = (x2 - 600) / 10;
6. double f = Integer.parseInt(jTextField1.getText());
7. double v = u * f / (u + f);
```

```

8. double v2 = v * (10) + 600;
9. int v3 = (int) v2;
10. int y2 = jLabel3.getY();
11. int heightx = jLabel1.getHeight();
12. double m = Math.abs(v / u);
13. double CASE = v / u;
14. if (CASE >= 0) {
15.     try {
16.         jLabel3.setIcon(new ImageIcon(ImageIO.read(getClass().getResource("/data/OBJECT.png"))));
17.     }
18.     catch (IOException ex) {}
19. }
20. double height1 = m * heightx;
21. int height = (int) Math.round(height1);
22. int width = jLabel3.getWidth();
23. if ((v / u) >= 0) jLabel3.setBounds(v3, 240 - height, width, height);
24. else jLabel3.setBounds(v3, y2, width, height);
25. jTextField2.setText("" + ((double) Math.round(u * 1000)) / 1000);
26. jTextField3.setText("" + ((double) Math.round(v * 1000)) / 1000);
27. jTextField4.setText("" + ((double) Math.round((v / u) * 1000)) / 1000);
28. BufferedImage pic = null;
29. BufferedImage pic2 = null;
30. try {
31.     pic = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
32.     if ((v / u) >= 0) pic2 = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
33.     else pic2 = ImageIO.read(getClass().getResource("/data/IMAGE.png"));
34.     ImageIcon icon = new ImageIcon(scaledImage(pic, jLabel1.getWidth(),
        jLabel1.getHeight()));
35.     ImageIcon icon2 = new ImageIcon(scaledImage(pic2, jLabel3.getWidth(),
        jLabel3.getHeight()));
36.     jLabel1.setIcon(icon);
37.     jLabel3.setIcon(icon2);
38. }
39. catch (IOException e) {
40.     JOptionPane.showMessageDialog(this, e.getMessage());
41. }

```

JAVA CODE[Save Button]: -

```

1. double imgcode = (Math.random() * 1000);
2. screenshot.screenshot(imgcode);
3. String f = jTextField1.getText();
4. double u = Double.parseDouble(jTextField2.getText());
5. double v = Double.parseDouble(jTextField3.getText());
6. double m = Double.parseDouble(jTextField4.getText());

```

```

7. conection.exup("INSERT INTO cvexl VALUES('" + conection.uid() + "','" +
  ((double) Math.round(Math.random() * 100000)) + "','" + "curdate()," +
  f + "','" + u + "','" + v + "','" + m + "','" + imgcode + "',')", "
  experiment");
8. JOptionPane.showMessageDialog(this, "Data saved successfully!");

```

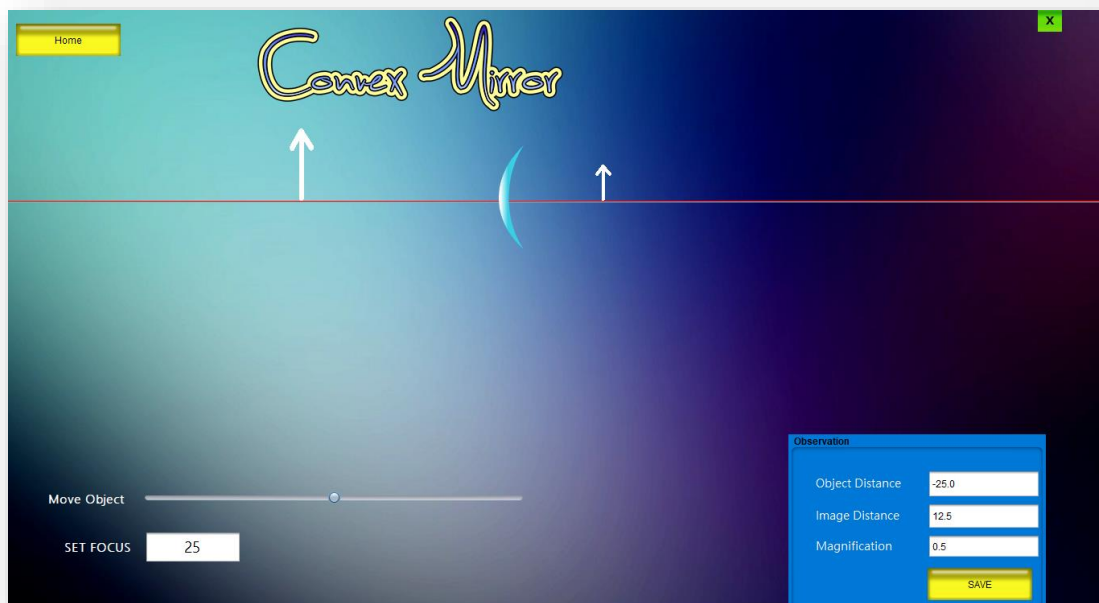
JAVA CODE[Home and X button]: -

```

1. new MainMenu().setVisible(true);
2. this.dispose();

```

Convex Mirror[convexmirror.java]



JAVA CODE[Jslider State changed]: -

```

1. int y1 = jLabel1.getY();
2. double x = 5 * jSlider1.getValue() + 100;
3. int x2 = (int) Math.round(x);
4. jLabel1.setLocation(x2, y1);
5. int x1 = jLabel1.getX();
6. double u = (x1 - 600) / 10;
7. double f = Integer.parseInt(jTextField1.getText());
8. double v = u * f / (u - f);
9. double v2 = Math.round(v);
10. int v3 = (int) v2 * (10) + 600;
11. int y2 = jLabel3.getY();

```

```

12. int heightx = jLabel1.getHeight();
13. double CASE = -1 * v / u;
14. if (CASE >= 0) {
15.     try {
16.         jLabel3.setIcon(new ImageIcon(ImageIO.read(getClass().getResource("/data/OBJECT.png"))));
17.     } catch (IOException ex) {}
18. }
19. double m = Math.abs(v / u);
20. double height1 = m * heightx;
21. int height = (int) Math.round(height1);
22. int width = jLabel3.getWidth();
23. if ((-v / u) >= 0) jLabel3.setBounds(v3, 240 - height, width, height);
24. else jLabel3.setBounds(v3, y2, width, height);
25. jTextField2.setText("" + ((double) Math.round(u * 1000)) / 1000);
26. jTextField3.setText("" + ((double) Math.round(v * 1000)) / 1000);
27. jTextField4.setText("" + ((double) Math.round((-v / u) * 1000)) / 1000);
28. BufferedImage pic = null;
29. BufferedImage pic2 = null;
30. try {
31.     pic = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
32.     if ((-v / u) >= 0) pic2 = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
33.     else pic2 = ImageIO.read(getClass().getResource("/data/IMAGE.png"));
34.     ImageIcon icon = new ImageIcon(scaledImage(pic, jLabel1.getWidth(), jLabel1.getHeight()));
35.     ImageIcon icon2 = new ImageIcon(scaledImage(pic2, jLabel3.getWidth(), jLabel3.getHeight()));
36.     jLabel1.setIcon(icon);
37.     jLabel3.setIcon(icon2);
38. } catch (IOException e) {
39.     JOptionPane.showMessageDialog(this, e.getMessage());
40. }

```

JAVA CODE[Save Button]: -

```

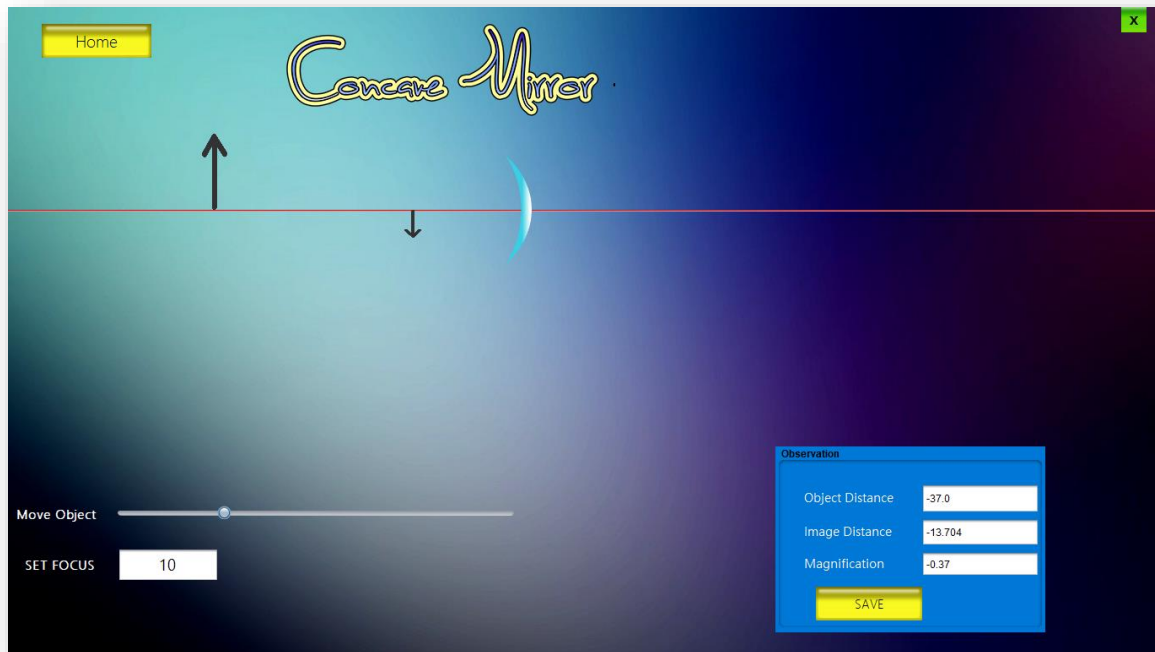
1. double imgcode = (Math.random() * 1000);
2. screenshot.screenshot(imgcode);
3. String f = jTextField1.getText();
4. double u = Double.parseDouble(jTextField2.getText());
5. double v = Double.parseDouble(jTextField3.getText());
6. double m = Double.parseDouble(jTextField4.getText());
7. conection.exup("INSERT INTO cvexm VALUES('" + conection.uid() + "','" +
    ((double) Math.round(Math.random() * 100000)) + "','" + "curdate()," +
    f + "','" + u + "','" + v + "','" + m + "','" + imgcode + "'", "experiment");
8. JOptionPane.showMessageDialog(this, "Data saved successfully!");

```

JAVA CODE[Home and X button]: -

1. `new Mainmenu().setVisible(true);`
2. `this.dispose();`

Concave Mirror[concavemirror.java]



JAVA CODE[Jslider State changed]: -

1. `int y1 = jLabel1.getY();`
2. `double x = 5 * jSlider1.getValue() + 100;`
3. `int x2 = (int) Math.round(x);`
4. `jLabel1.setLocation(x2, y1);`
5. `double u = (x2 - 600) / 10;`
6. `double f = -1 * Integer.parseInt(jTextField1.getText());`
7. `double v = u * f / (u - f);`
8. `int v3 = (int) v * (10) + 600;`
9. `int y2 = jLabel3.getY();`
10. `int heightx = jLabel1.getHeight();`
11. `double CASE = -1 * v / u;`
12. `if (CASE >= 0) {`
13. `try {`
14. `jLabel3.setIcon(new ImageIcon(ImageIO.read(getClass().getResource("/data/OBJECT.png"))));`

```

15.     } catch (IOException ex) {}
16. }
17. double m = Math.abs(v / u);
18. double height1 = m * heightx;
19. int height = (int) Math.round(height1);
20. int width = jLabel3.getWidth();
21. if ((-v / u) >= 0) jLabel3.setBounds(v3, 240 - height, width, height);
22. else jLabel3.setBounds(v3, y2, width, height);
23. jTextField2.setText("" + ((double) Math.round(u * 1000)) / 1000);
24. jTextField3.setText("" + ((double) Math.round(v * 1000)) / 1000);
25. jTextField4.setText("" + ((double) Math.round((-v / u) * 1000)) / 1000);
26. double magni = (double) Math.round(m * 1000) / 1000;
27. BufferedImage pic = null;
28. BufferedImage pic2 = null;
29. try {
30.     pic = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
31.     if ((-
        v / u) >= 0) pic2 = ImageIO.read(getClass().getResource("/data/OBJECT.png"));
32. else pic2 = ImageIO.read(getClass().getResource("/data/IMAGE.png"));
33.     ImageIcon icon = new ImageIcon(scaledImage(pic, jLabel1.getWidth(), jLabel
        1.getHeight()));
34.     ImageIcon icon2 = new ImageIcon(scaledImage(pic2, jLabel3.getWidth(), jLab
        el3.getHeight()));
35.     jLabel1.setIcon(icon);
36.     jLabel3.setIcon(icon2);
37. } catch (IOException e) {
38.     JOptionPane.showMessageDialog(this, e.getMessage());
39. }

```

JAVA CODE[Save Button]: -

```

1. double imgcode = (Math.random() * 1000);
2. screenshot.screenshot(imgcode);
3. String f = jTextField1.getText();
4. double u = Double.parseDouble(jTextField2.getText());
5. double v = Double.parseDouble(jTextField3.getText());
6. double m = Double.parseDouble(jTextField4.getText());
7. conection.exup("INSERT INTO cavem VALUES('" + conection.uid() + "','" + ((doub
    le) Math.round(Math.random() * 100000)) + "','" + "curdate()," + f + "','" + u +
    "','" + v + "','" + m + "','" + imgcode + "','", "experiment");
8. JOptionPane.showMessageDialog(this, "Data saved successfully!");

```

JAVA CODE[Home and X button]: -

```

1. new Mainmenu().setVisible(true);
2. this.dispose();

```

MySQL

Once you Click Install in Installation Frame the Database named experiment is created in MySQL

```
mysql> Use Experiment;
```

```
Database changed
```

```
mysql> Show Tables;
```

```
+-----+
| Tables_in_experiment |
+-----+
| activestatus          |
| cavel                 |
| cavem                 |
| cvexl                 |
| cvexm                 |
| helicalspring         |
| install               |
| mbridge               |
| userdata              |
+-----+
```

```
mysql> Desc ActiveStatus;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| uid        | varchar(30)   | YES  | MUL | NULL     |                |
| statusno   | int(11)       | NO   | PRI | NULL     | auto_increment |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> Desc cavel;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| uid        | varchar(30)   | YES  | MUL | NULL     |                |
| eid        | varchar(1000) | YES  |     | NULL     |                |
| edate      | date          | YES  |     | NULL     |                |
| focus      | int(11)       | YES  |     | NULL     |                |
| obdis      | decimal(10,3) | YES  |     | NULL     |                |
| imgdis     | decimal(10,3) | YES  |     | NULL     |                |
| mgficton   | decimal(10,3) | YES  |     | NULL     |                |
| imge       | varchar(100)  | YES  |     | NULL     |                |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```



```
mysql> Desc cavem;
```

Field	Type	Null	Key	Default	Extra
uid	varchar(30)	YES	MUL	NULL	
eid	varchar(1000)	YES		NULL	
edate	date	YES		NULL	
focus	int(11)	YES		NULL	
obdis	decimal(10,3)	YES		NULL	
imgdis	decimal(10,3)	YES		NULL	
mgficton	decimal(10,3)	YES		NULL	
imge	varchar(100)	YES		NULL	

8 rows in set (0.00 sec)

```
mysql> Desc cvexl;
```

Field	Type	Null	Key	Default	Extra
uid	varchar(30)	YES	MUL	NULL	
eid	varchar(1000)	YES		NULL	
edate	date	YES		NULL	
focus	int(11)	YES		NULL	
obdis	decimal(10,3)	YES		NULL	
imgdis	decimal(10,3)	YES		NULL	
mgficton	decimal(10,3)	YES		NULL	
imge	varchar(100)	YES		NULL	

8 rows in set (0.00 sec)

```
mysql> Desc cvexm;
```

Field	Type	Null	Key	Default	Extra
uid	varchar(30)	YES	MUL	NULL	
eid	varchar(1000)	YES		NULL	
edate	date	YES		NULL	
focus	int(11)	YES		NULL	
obdis	decimal(10,3)	YES		NULL	
imgdis	decimal(10,3)	YES		NULL	
mgficton	decimal(10,3)	YES		NULL	
imge	varchar(100)	YES		NULL	

8 rows in set (0.00 sec)

```
mysql> desc install;
```

Field	Type	Null	Key	Default	Extra
statusno	int(1)	YES		NULL	

1 row in set (0.00 sec)

```
mysql> Select* from Install;
```

```
+-----+
| statusno |
+-----+
|          1 |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> Desc HelicalSpring;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| uid        | varchar(30)    | YES  | MUL | NULL    |       |
| eid        | varchar(1000)  | YES  |     | NULL    |       |
| edate      | date           | YES  |     | NULL    |       |
| wght       | int(8)         | YES  |     | NULL    |       |
| extension  | decimal(30,10) | YES  |     | NULL    |       |
| imge       | varchar(100)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

```
6 rows in set (0.00 sec)
```

```
mysql> Desc Mbridge;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| uid        | varchar(30)    | YES  | MUL | NULL    |       |
| eid        | varchar(1000)  | YES  |     | NULL    |       |
| edate      | date           | YES  |     | NULL    |       |
| la         | int(2)         | YES  |     | NULL    |       |
| lb         | int(2)         | YES  |     | NULL    |       |
| rbresist   | int(11)        | YES  |     | NULL    |       |
| uresist    | int(11)        | YES  |     | NULL    |       |
| rmeter     | decimal(30,5)  | YES  |     | NULL    |       |
| rammeter   | decimal(30,5)  | YES  |     | NULL    |       |
| vol        | decimal(10,4)  | YES  |     | NULL    |       |
| imge       | varchar(100)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

```
11 rows in set (0.00 sec)
```

```
mysql> Desc UserData;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| uid   | varchar(30)    | NO   | PRI | NULL    |       |
| name  | varchar(70)    | NO   |     | NULL    |       |
| pass  | varchar(15)    | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

Once a user registers by using New User Register frame his data is saved in a table named userdata

```
mysql> select * from userdata;
+-----+-----+-----+
| uid   | name   | pass   |
+-----+-----+-----+
| 100   | tushar | tushar |
| 1000  | Admin  | admin  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

When a user Login using login frame his current session is identified in each frame by using database table activestatus in the following way

```
mysql> select* from activestatus;
+-----+-----+
| uiD   | statusno |
+-----+-----+
| 100   | 1         |
+-----+-----+
1 row in set (0.00 sec)
```

In this statusno is set to 1 which signifies that user with id 100 is currently online

Now I will explain how I am saving screenshots when a user after doing experiment saves his data then along with saving data to database a imgcode is generated which is saved in database and a image with same imgcode is saved in a folder named EXPHYSICS which was created during installation

```
mysql> select * from helicalspring
-> ;
+-----+-----+-----+-----+-----+-----+
| uid   | eid     | edate   | wght | extension | imge |
+-----+-----+-----+-----+-----+-----+
| 100   | 10174.0 | 2017-12-07 | 15   | 1.6670000000 | 507.7170903452018 |
| 100   | 58519.0 | 2017-12-07 | 20   | 2.2220000000 | 507.7170903452018 |
| 100   | 26593.0 | 2017-12-07 | 25   | 2.7780000000 | 507.7170903452018 |
| 100   | 9707.0  | 2017-12-07 | 30   | 3.3330000000 | 507.7170903452018 |
| 100   | 10175.0 | 2017-12-07 | 35   | 3.8890000000 | 507.7170903452018 |
| 100   | 86391.0 | 2017-12-07 | 40   | 4.4440000000 | 507.7170903452018 |
| 100   | 7864.0  | 2017-12-07 | 50   | 5.5560000000 | 507.7170903452018 |
| 100   | 85430.0 | 2017-12-07 | 55   | 6.1110000000 | 507.7170903452018 |
| 100   | 99855.0 | 2017-12-07 | 60   | 6.6670000000 | 507.7170903452018 |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

EXPHYSICS folder

