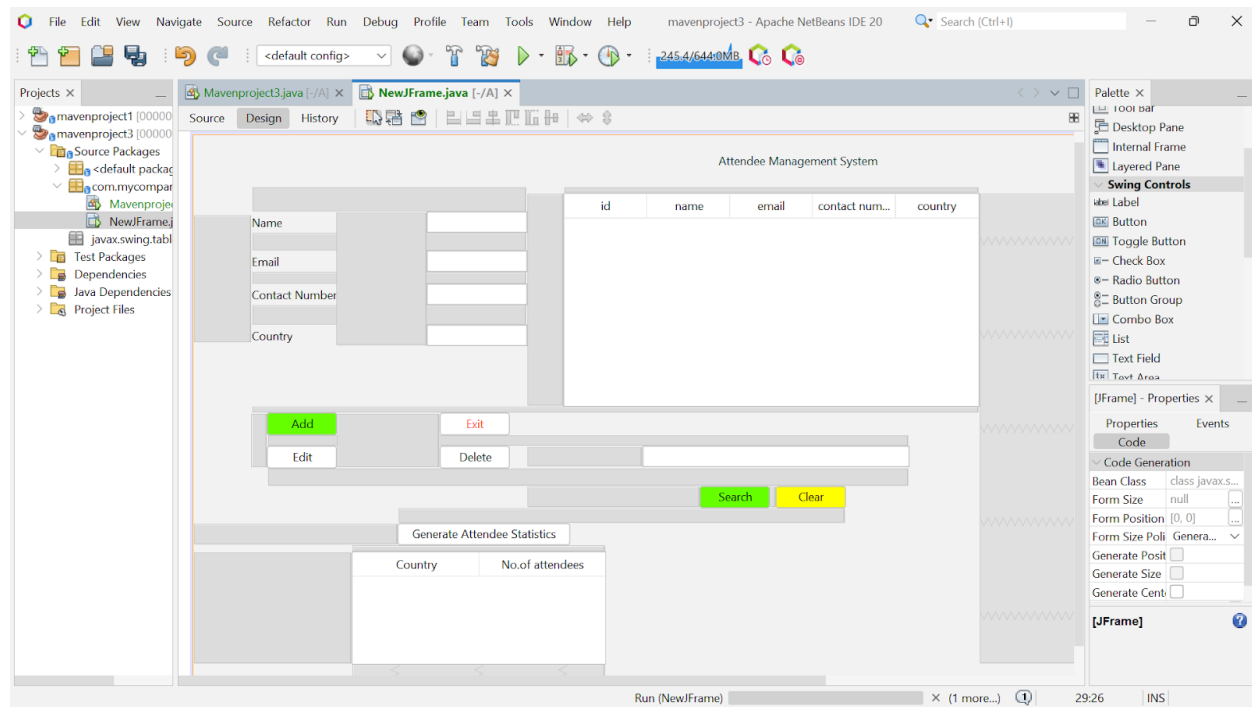
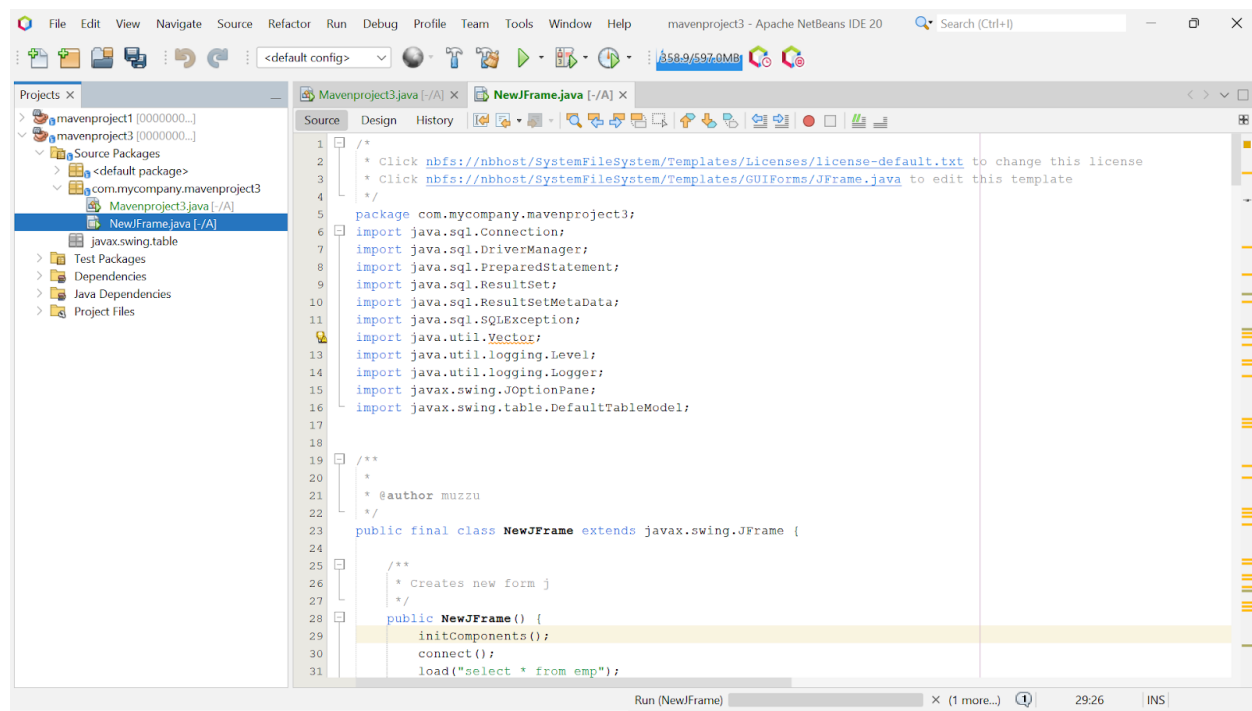
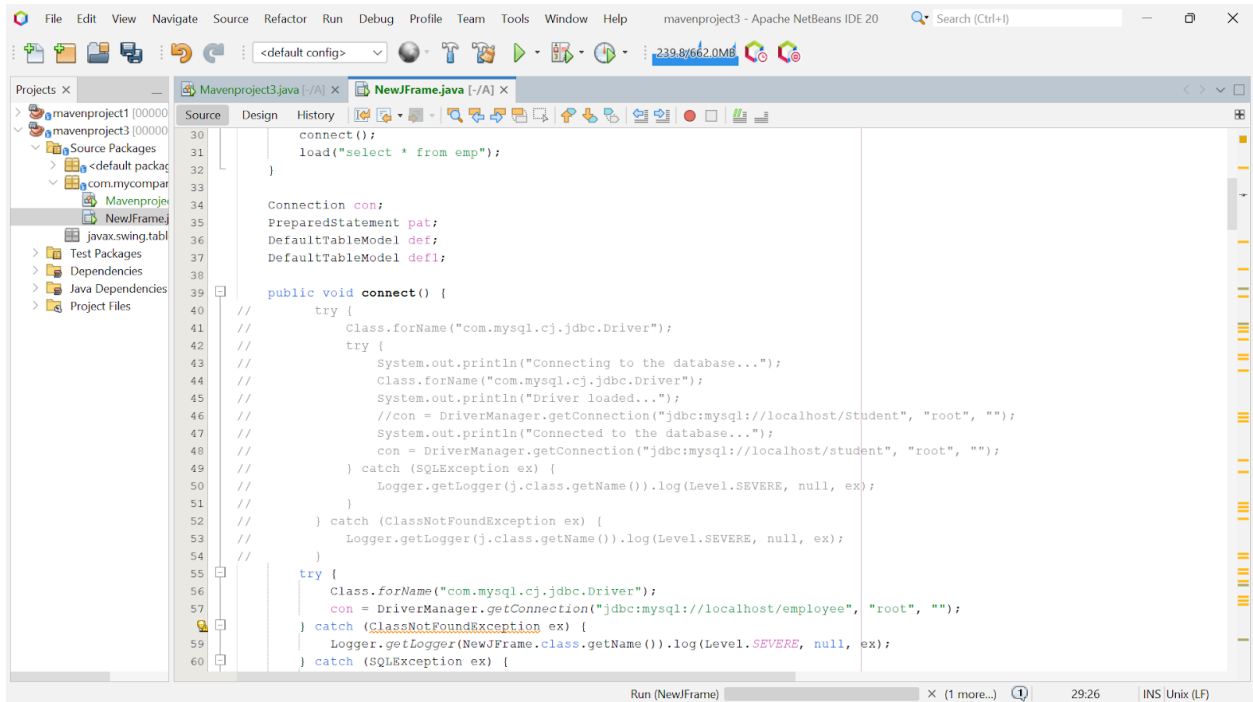


DESIGN:

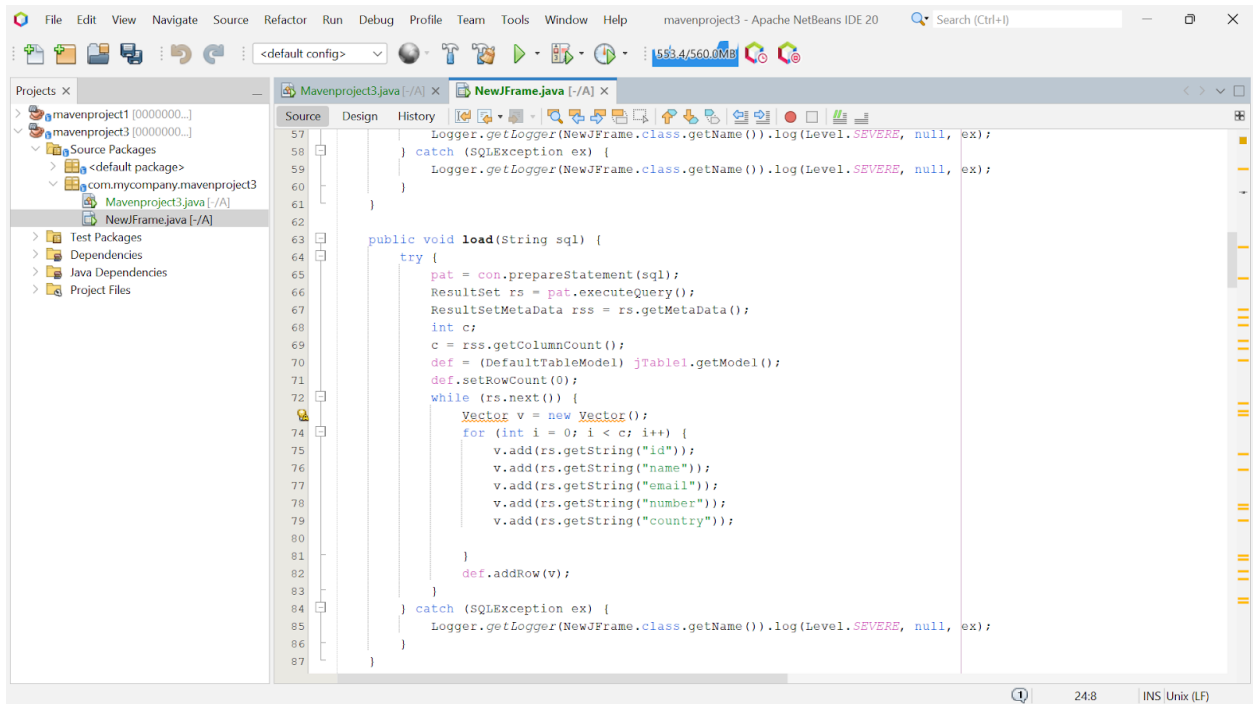


CODE:

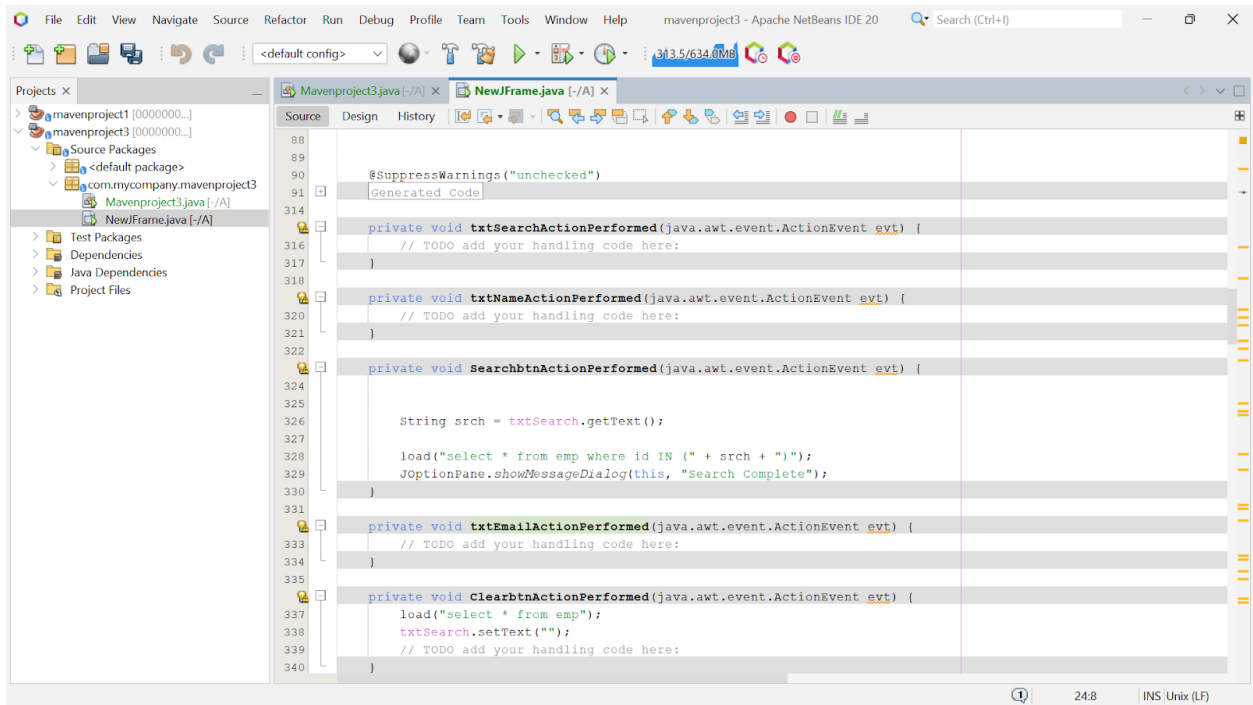
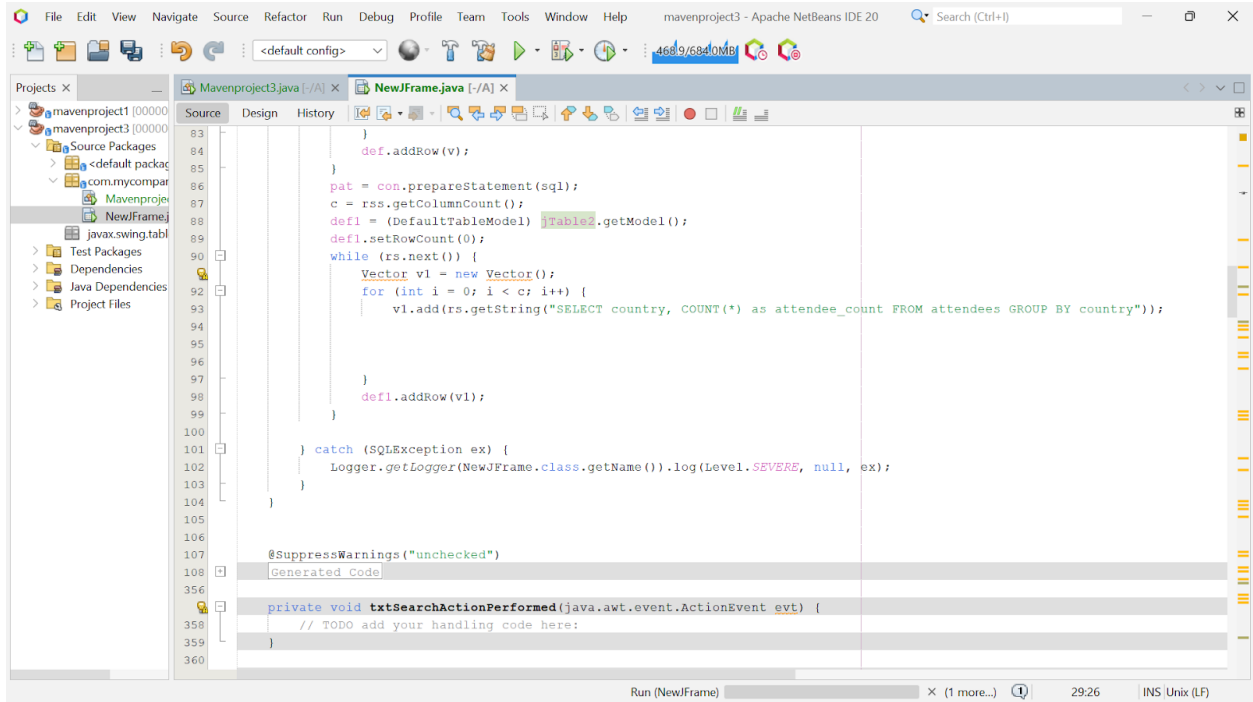


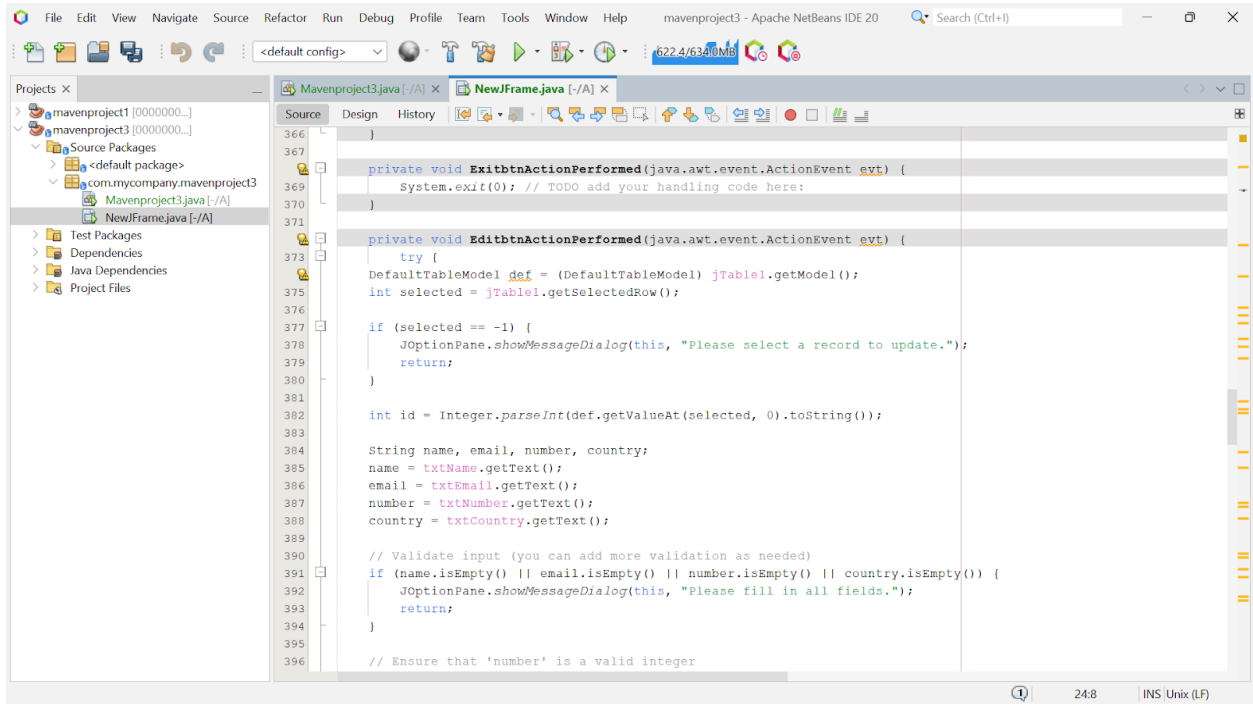
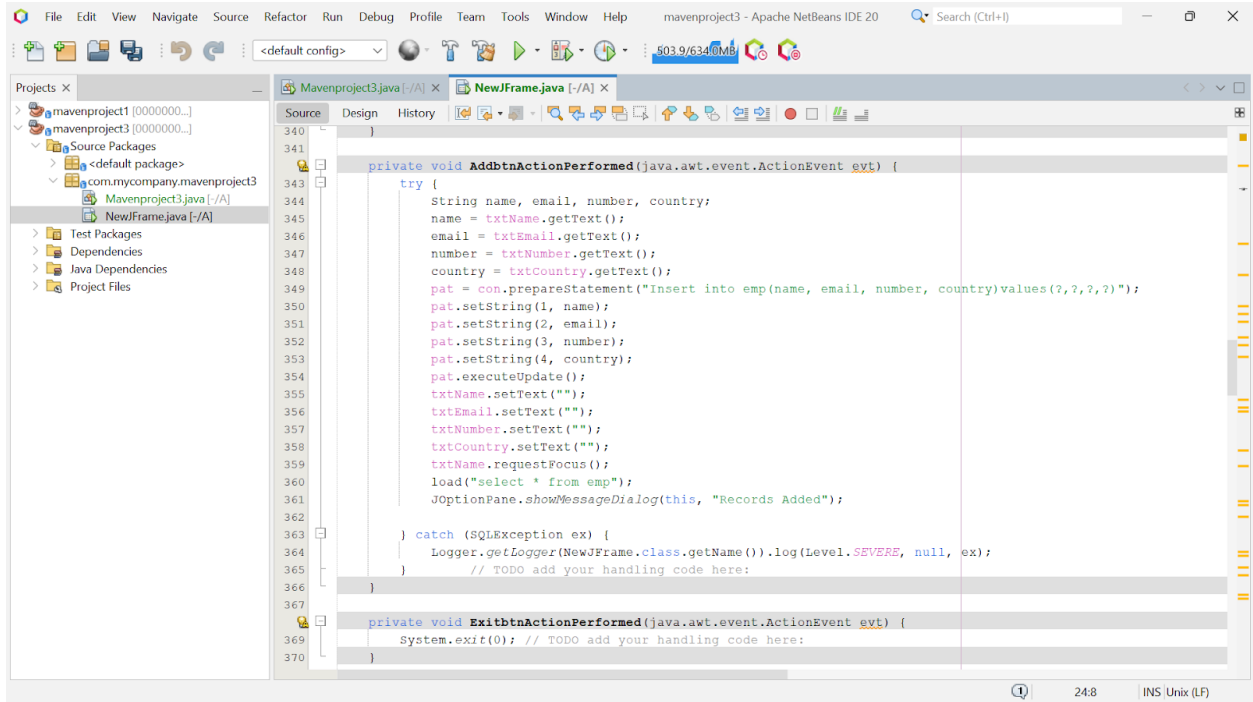


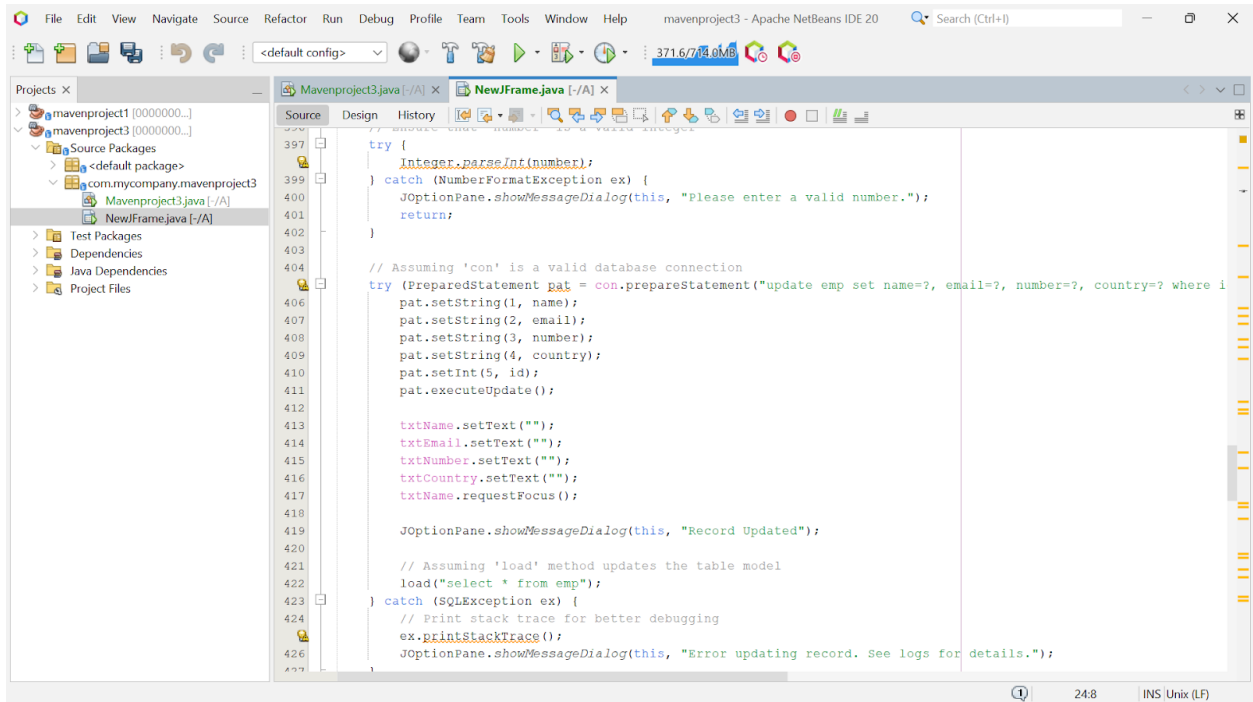
```
30      connect();
31      load("select * from emp");
32  }
33
34      Connection con;
35      PreparedStatement pat;
36      DefaultTableModel def;
37      DefaultTableModel def1;
38
39      public void connect() {
40          try {
41              // Class.forName("com.mysql.cj.jdbc.Driver");
42              try {
43                  System.out.println("Connecting to the database...");
44                  Class.forName("com.mysql.cj.jdbc.Driver");
45                  System.out.println("Driver loaded...");
46                  //con = DriverManager.getConnection("jdbc:mysql://localhost/Student", "root", "");
47                  System.out.println("Connected to the database...");
48                  con = DriverManager.getConnection("jdbc:mysql://localhost/student", "root", "");
49              } catch (SQLException ex) {
50                  Logger.getLogger(j.class.getName()).log(Level.SEVERE, null, ex);
51              }
52          } catch (ClassNotFoundException ex) {
53              Logger.getLogger(j.class.getName()).log(Level.SEVERE, null, ex);
54          }
55      }
56      try {
57          Class.forName("com.mysql.cj.jdbc.Driver");
58          con = DriverManager.getConnection("jdbc:mysql://localhost/employee", "root", "");
59      } catch (ClassNotFoundException ex) {
60          Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
61      } catch (SQLException ex) {
62          Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
63      }
```



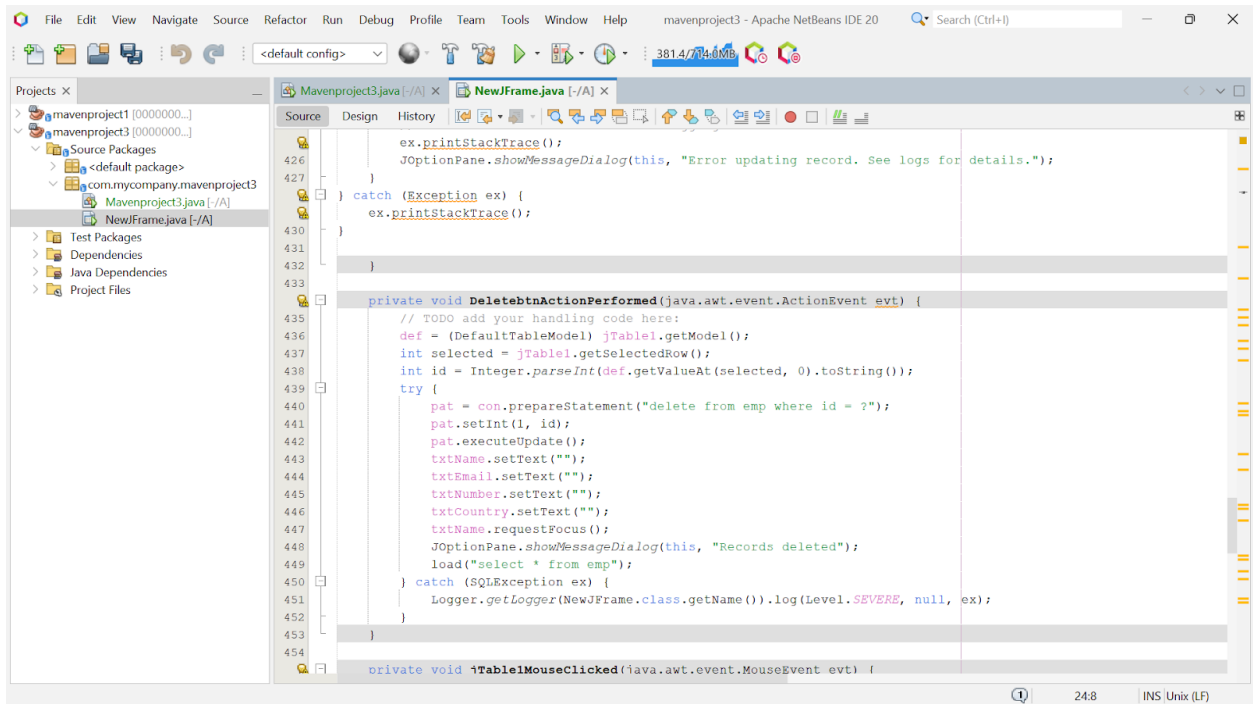
```
57      Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
58  } catch (SQLException ex) {
59      Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
60  }
61  }
62
63      public void load(String sql) {
64          try {
65              pat = con.prepareStatement(sql);
66              ResultSet rs = pat.executeQuery();
67              ResultSetMetaData rss = rs.getMetaData();
68              int c;
69              c = rss.getColumnCount();
70              def = (DefaultTableModel) jTable1.getModel();
71              def.setRowCount(0);
72              while (rs.next()) {
73                  Vector v = new Vector();
74                  for (int i = 0; i < c; i++) {
75                      v.add(rs.getString("id"));
76                      v.add(rs.getString("name"));
77                      v.add(rs.getString("email"));
78                      v.add(rs.getString("number"));
79                      v.add(rs.getString("country"));
80                  }
81                  def.addRow(v);
82              }
83          } catch (SQLException ex) {
84              Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
85          }
86      }
87  }
```



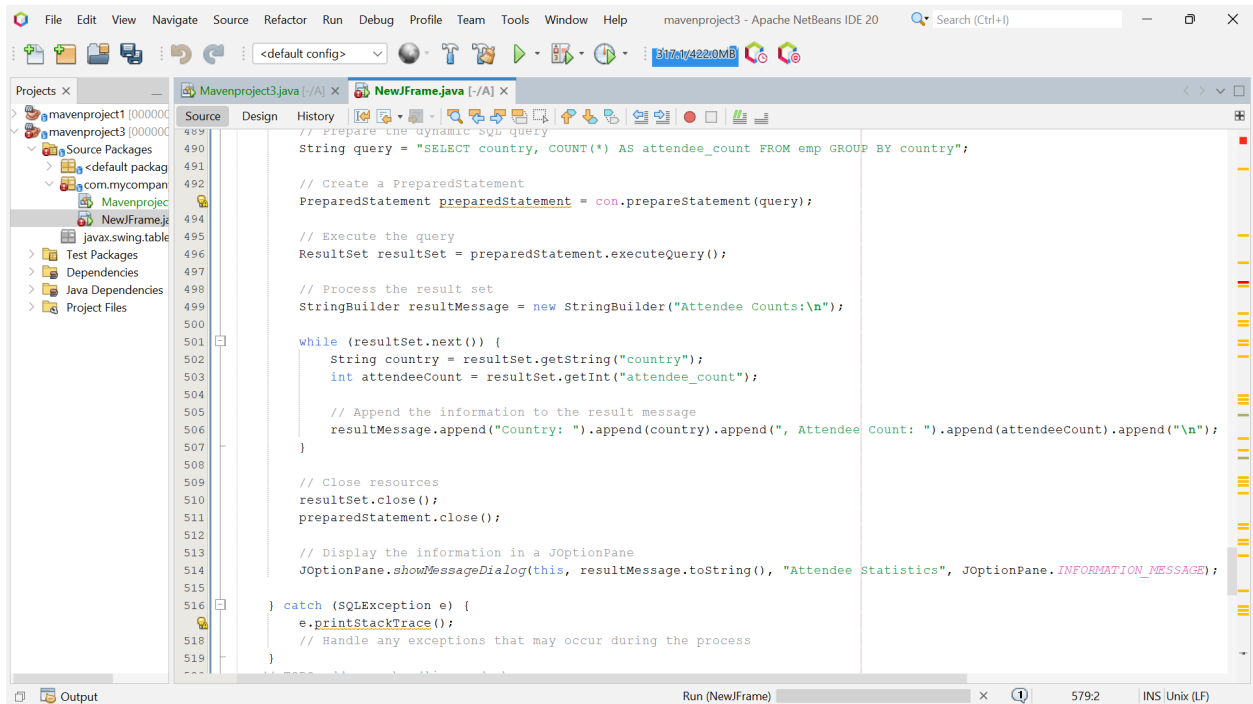
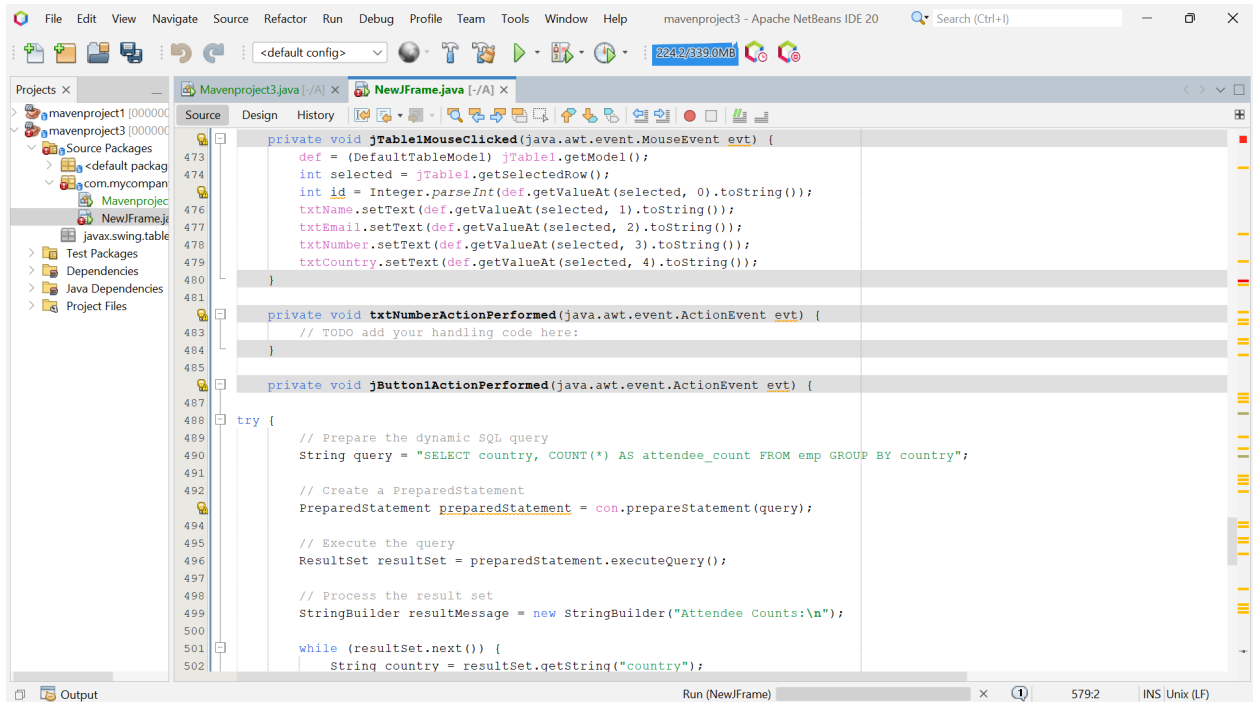


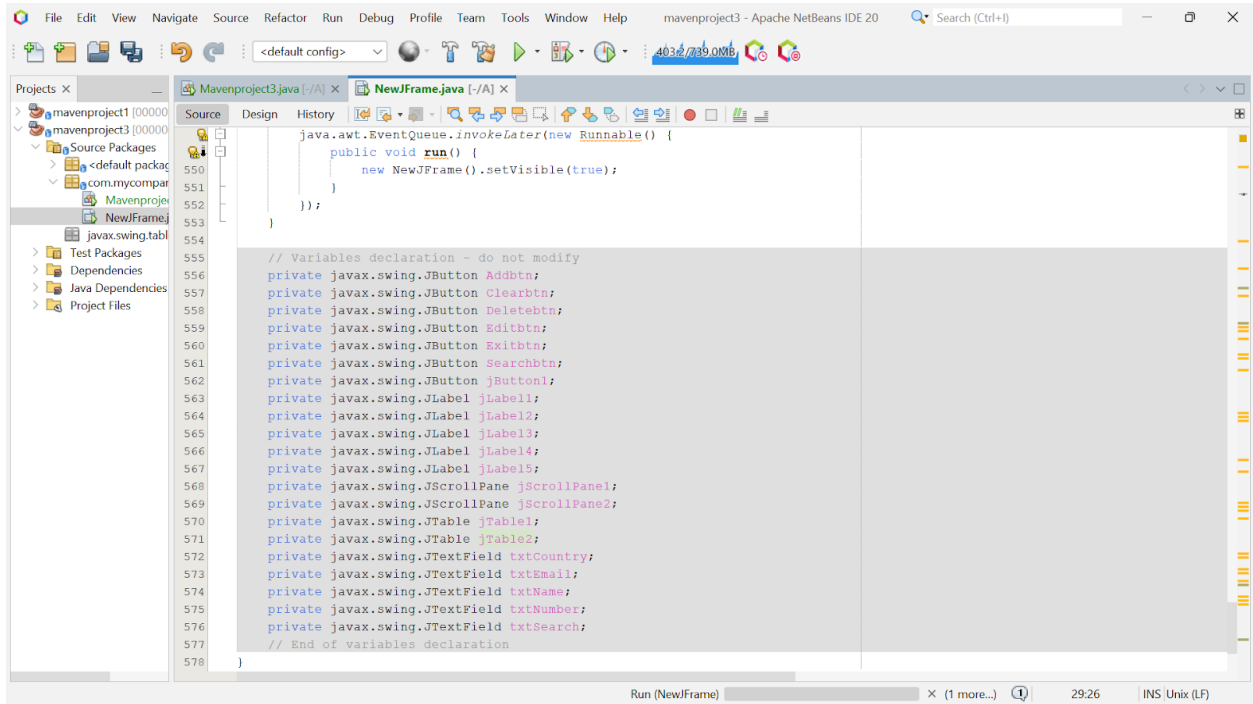
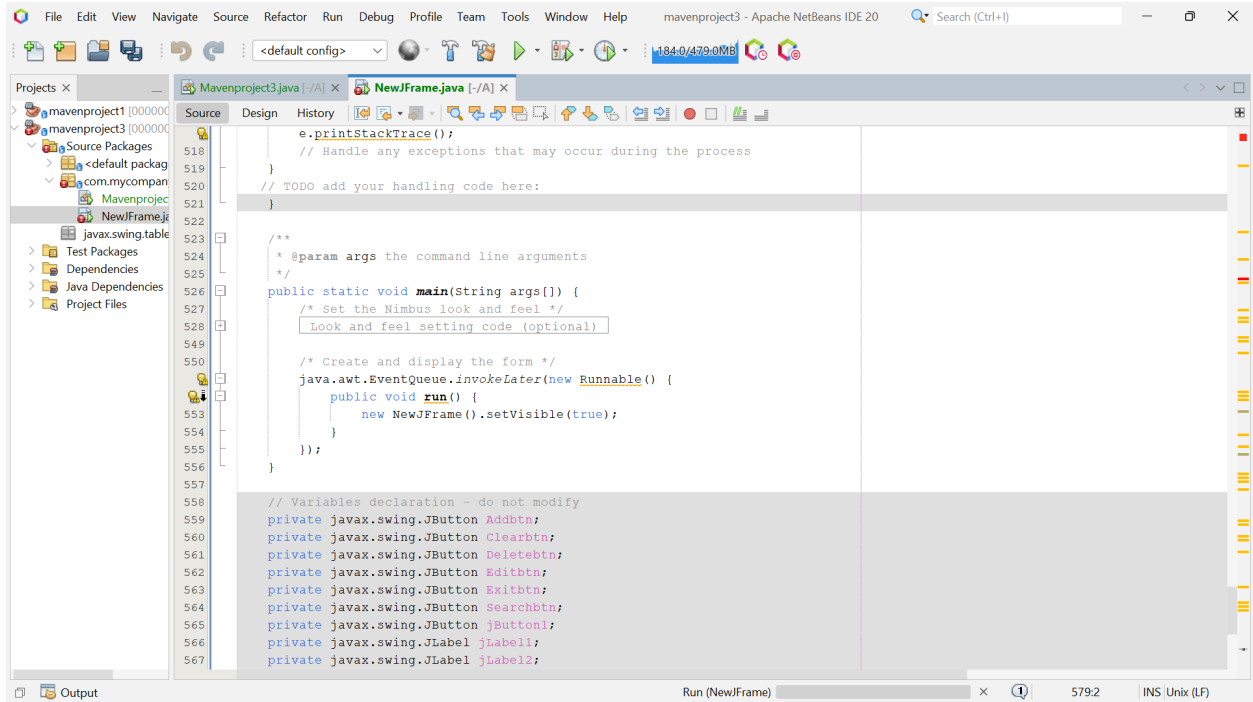


```
397 try {
398     Integer.parseInt(number);
399 } catch (NumberFormatException ex) {
400     JOptionPane.showMessageDialog(this, "Please enter a valid number.");
401     return;
402 }
403
404 // Assuming 'con' is a valid database connection
405 try (PreparedStatement pat = con.prepareStatement("update emp set name=?, email=?, number=?, country=? where i
406     pat.setString(1, name);
407     pat.setString(2, email);
408     pat.setString(3, number);
409     pat.setString(4, country);
410     pat.setInt(5, id);
411     pat.executeUpdate();
412
413     txtName.setText("");
414     txtEmail.setText("");
415     txtNumber.setText("");
416     txtCountry.setText("");
417     txtName.requestFocus();
418
419     JOptionPane.showMessageDialog(this, "Record Updated");
420
421     // Assuming 'load' method updates the table model
422     load("select * from emp");
423 } catch (SQLException ex) {
424     // Print stack trace for better debugging
425     ex.printStackTrace();
426     JOptionPane.showMessageDialog(this, "Error updating record. See logs for details.");
427 }
```



```
426 ex.printStackTrace();
427 JOptionPane.showMessageDialog(this, "Error updating record. See logs for details.");
428 } catch (Exception ex) {
429     ex.printStackTrace();
430 }
431
432
433 private void DeletebtnActionPerformed(java.awt.event.ActionEvent evt) {
434     // TODO add your handling code here:
435     def = (DefaultTableModel) jTable1.getModel();
436     int selected = jTable1.getSelectedRow();
437     int id = Integer.parseInt(def.getValueAt(selected, 0).toString());
438     try {
439         pat = con.prepareStatement("delete from emp where id = ?");
440         pat.setInt(1, id);
441         pat.executeUpdate();
442         txtName.setText("");
443         txtEmail.setText("");
444         txtNumber.setText("");
445         txtCountry.setText("");
446         txtName.requestFocus();
447         JOptionPane.showMessageDialog(this, "Records deleted");
448         load("select * from emp");
449     } catch (SQLException ex) {
450         Logger.getLogger(NewFrame.class.getName()).log(Level.SEVERE, null, ex);
451     }
452 }
453
454 private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
```





OUTPUT:

Name

Email

Contact Number

Country

Add

Edit

Exit

Delete

Generate Attendee Statistics

id	name	email	contact num...	country
2	razin	razin@gmail...	9000000	india
3	johnjustine	john@gmail...	8000000	india
5	rahul	rahul@gmail...	8100000	usa

Search

Clear

Add:

Name

Email

Contact Number

Country

Add

Edit

Exit

Delete

Generate Attendee Statistics

id	name	email	contact num...	country
2	razin	razin@gmail...	9000000	uae
3	johnjustine	john@gmail...	8000000	india
5	rahul	rahul@gmail...	8100000	usa
6	ridha	ridha@gmail...	00000000	oman

Search

Clear

Delete:

Name

Email

Contact Number

Country

Add

Edit

Exit

Delete

Generate Attendee Statistics

id	name	email	contact num...	country
2	razin	razin@gmail...	9000000	uae
6	ridha	ridha@gmail...	00000000	oman
7	xx	ss	ss	ss

Message

Records deleted

OK

Edit:

Name

Email

Contact Number

Country

Add

Edit

Exit

Delete

Generate Attendee Statistics

id	name	email	contact num...	country
2	razin	razin@gmail...	9000000	uae
6	ridha	ridha@gmail...	4700000	oman

Message

Record Updated

OK

Search:

Attendee Management System

Name

Email

Contact Number

Country

id	name	email	contact num...	country
6	ridha	ridha@gma...	999999999	oman

Statistics:

Attendee Management System

Name

Email

Contact Number

Country

id	name	email	contact num...	country
2	razin	razin@gmai...	9000000	uae
6	ridha	ridha@gma...	999999999	oman
8	rameena	rameena@...	900000000	uae

Attendee Statistics

Attendee Counts:
Country: oman, Attendee Count: 1
Country: uae, Attendee Count: 2