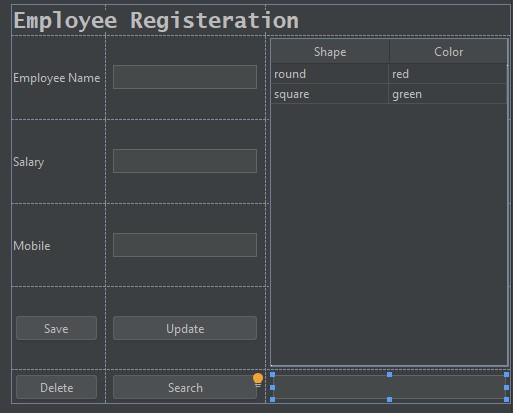
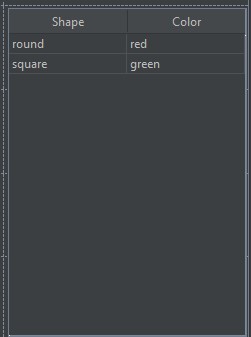
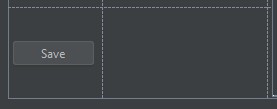
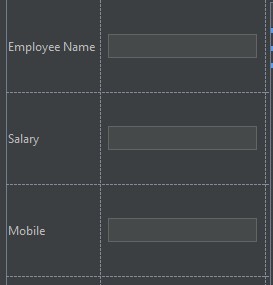
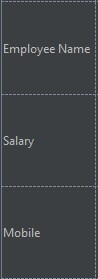
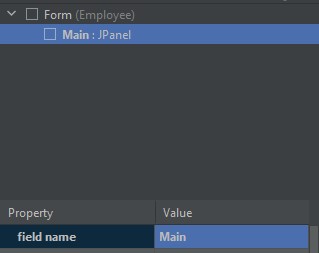
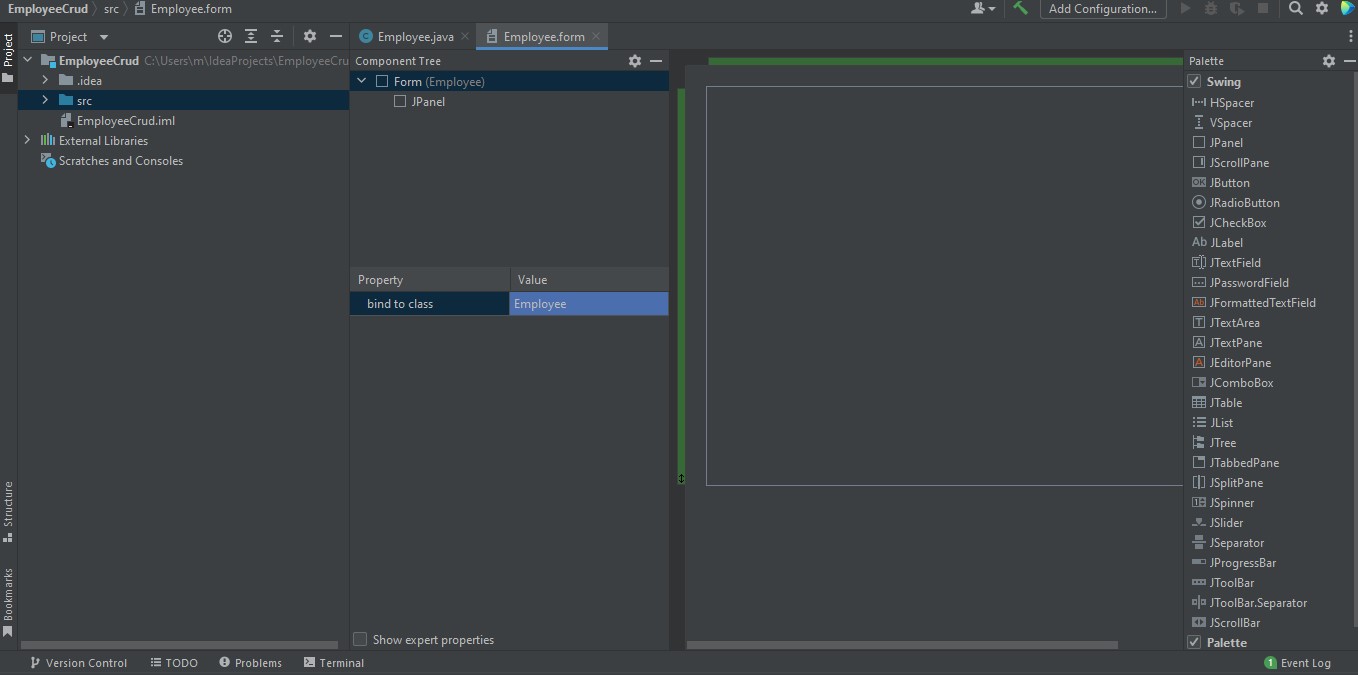
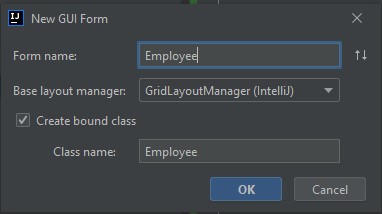
Java MySQL Crud Project Using

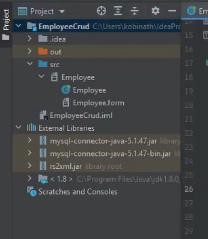
IntelliJ Idea

* **Making a form:**

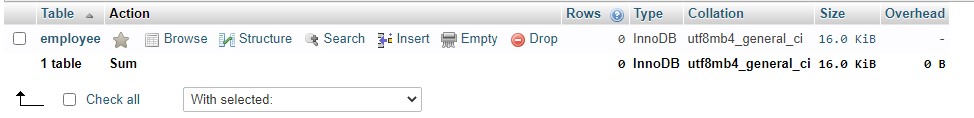
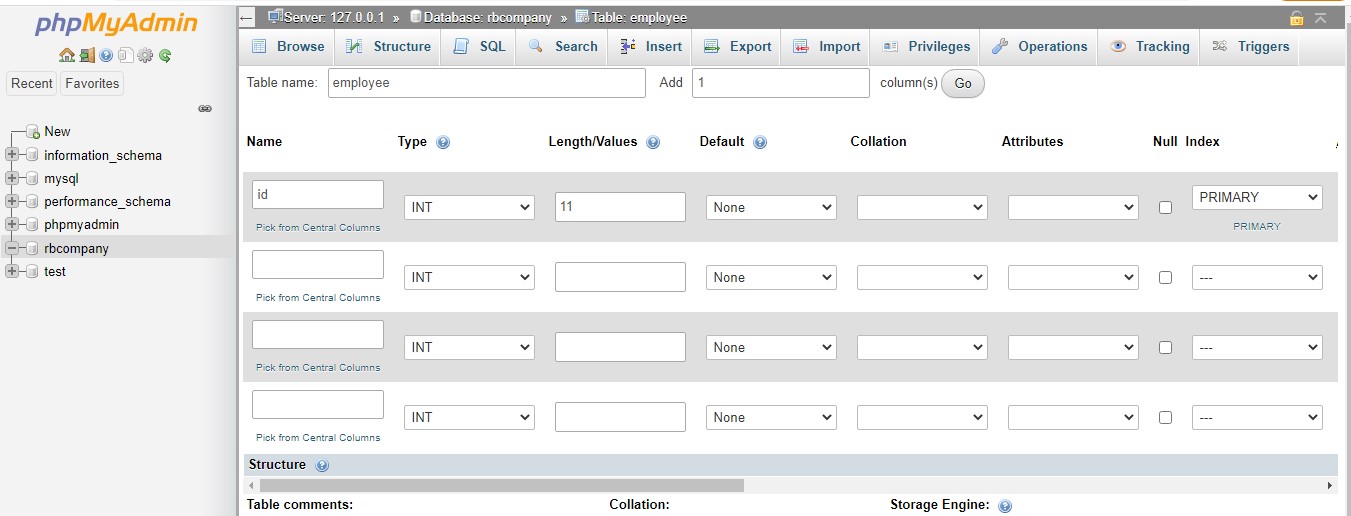
First of all, a form is made to understand the GUI design:



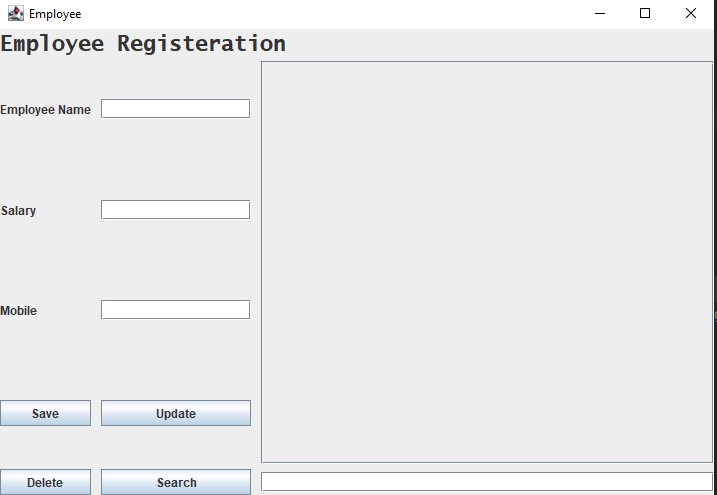
* Adding mysql connectors to the the bench:



* Creating a database on xampp:



* Basic GUI is made up till now:



* **Code of the whole program is as follows:**

|  |
| --- |
| • import javax.swing.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener;  public class Employee { private JPanel Main; private JTextField txtName; private JTextField txtSalary; private JTextField txtMobile; private JTable table1; private JButton updateButton; private JButton deleteButton; private JButton searchButton; private JTextField txtid; private JButton saveButton;    public static void main(String[] args) { JFrame frame = new JFrame("Employee"); frame.setContentPane(new Employee().Main);  frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*); frame.pack();  frame.setVisible(true);  }  String dbURL = "jdbc:mysql://localhost:3306/sampledb";  String username = "root"; String password = "secret";  try {  Connection conn = DriverManager.getConnection(dbURL, username, password);    if (conn != null) { |

|  |
| --- |
| System.*out*.println("Connected");  }  } catch (SQLException ex) { ex.printStackTrace();  } conn.close();  try (Connection conn = DriverManager.getConnection(dbURL, username, password)) {    // code to execute SQL queries goes here...    } catch (SQLException ex) { ex.printStackTrace();  }  String sql = "INSERT INTO Users (username, password, fullname, email) VALUES (?, ?, ?, ?)";    PreparedStatement statement = conn.prepareStatement(sql); statement.setString(1, "bill"); statement.setString(2, "secretpass"); statement.setString(3, "Bill Gates");  statement.setString(4, "bill.gates@microsoft.com");    int rowsInserted = statement.executeUpdate(); if (rowsInserted > 0) {  System.*out*.println("A new user was inserted successfully!"); }  String sql = "SELECT \* FROM Users";    Statement statement = conn.createStatement(); ResultSet result = statement.executeQuery(sql);    int count = 0;    while (result.next()){  String name = result.getString(2);  String pass = result.getString(3);  String fullname = result.getString("fullname"); String email = result.getString("email");  String output = "User #%d: %s - %s - %s - %s";  System.*out*.println(String.*format*(output, ++count, name, pass, fullname, email));  }  String sql = "UPDATE Users SET password=?, fullname=?, email=? WHERE username=?";    PreparedStatement statement = conn.prepareStatement(sql); statement.setString(1, "123456789");  statement.setString(2, "William Henry Bill Gates"); statement.setString(3, "bill.gates@microsoft.com"); statement.setString(4, "bill");  int rowsUpdated = statement.executeUpdate(); if (rowsUpdated > 0) {  System.*out*.println("An existing user was updated successfully!");  }  String sql = "DELETE FROM Users WHERE username=?";  PreparedStatement statement = conn.prepareStatement(sql); |
| statement.setString(1, "bill");    int rowsDeleted = statement.executeUpdate(); if (rowsDeleted > 0) {  System.*out*.println("A user was deleted successfully!");  }  public Employee() {  saveButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  }  });  }  private void createUIComponents() {  // *TODO: place custom component creation code here*  }  } |

* Some main statements are:

Using a Statement for a static SQL query.

Using a PreparedStatement for a parameterized SQL query and using setXXX() methods to set values for the parameters.

Using execute() method to execute the general query.

Using executeUpdate() method to execute INSERT, UPDATE or DELETE query Using executeQuery() method to execute SELECT query.

Using a ResultSet to iterate over rows returned from a SELECT query, using its next() method to advance to next row in the result set, and using getXXX() methods to retrieve values of columns.

* **Database Entry:**

|  |
| --- |
| • String empname,salary,mobile; empname = txtName.getText(); salary = txtSalary.getText(); mobile = txtMobile.getText();    try {  pst = con.prepareStatement("insert into employee(empname,salary,mobile)values(?,?,?)"); pst.setString(1, empname); pst.setString(2, salary); pst.setString(3, mobile); pst.executeUpdate();  JOptionPane.showMessageDialog(null, "Record Addedddd!!!!!");  table\_load(); txtName.setText(""); txtSalary.setText(""); txtMobile.setText(""); txtName.requestFocus();  } |

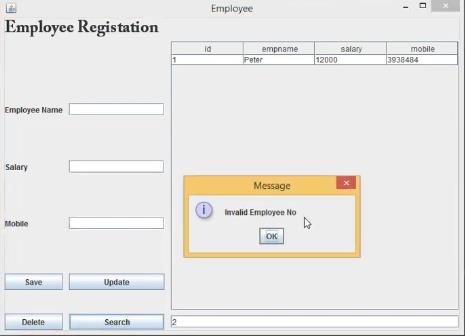
catch (SQLException e1)

{

e1.printStackTrace();

}

}



* **Table Updating:**

|  |
| --- |
| • String empid,empname,salary,mobile; empname = txtName.getText(); salary = txtSalary.getText(); mobile = txtMobile.getText(); empid = txtid.getText();    try {  pst = con.prepareStatement("update employee set empname  = ?,salary = ?,mobile = ? where id = ?"); pst.setString(1, empname); pst.setString(2, salary); pst.setString(3, mobile); pst.setString(4, empid);    pst.executeUpdate();  JOptionPane.showMessageDialog(null, "Record Updateee!!!!!");  table\_load(); txtName.setText(""); |

txtSalary.setText(""); txtMobile.setText(""); txtName.requestFocus();

}

catch (SQLException e1)

{

e1.printStackTrace();

}

}

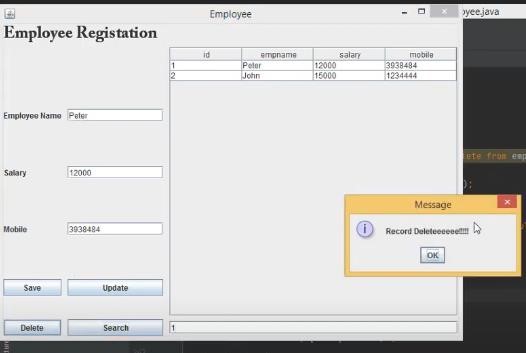


* **Deleting Record:**

|  |  |
| --- | --- |
| • | String empid;  empid = txtid.getText();  try {  pst = con.prepareStatement("delete from employee where id = ?");  pst.setString(1, empid);    pst.executeUpdate();  JOptionPane.showMessageDialog(null, "Record Deleteeeeee!!!!!"); table\_load(); txtName.setText(""); txtSalary.setText(""); txtMobile.setText(""); txtName.requestFocus();  } catch (SQLException e1)  {  e1.printStackTrace(); |

}

}



* **References:**
* [**https://www.tutussfunny.com/java-mysql-crud-project-using-intellij-idea/**](https://www.tutussfunny.com/java-mysql-crud-project-using-intellij-idea/)
* [**https://www.youtube.com/watch?v=e3AKnrTxFFo**](https://www.youtube.com/watch?v=e3AKnrTxFFo)
* [**https://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-selectupdate-and-delete-examples**](https://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-select-update-and-delete-examples)
* [**https://www.javaguides.net/2021/10/spring-boot-crud-rest-api-projectusing-IntelliJ-IDEA.html**](https://www.javaguides.net/2021/10/spring-boot-crud-rest-api-project-using-IntelliJ-IDEA.html)
* [**http://www.mastertheboss.com/eclipse/intellij-idea/intellij-idea-tip-howto-connect-to-a-database-using-database-navigator/**](http://www.mastertheboss.com/eclipse/intellij-idea/intellij-idea-tip-how-to-connect-to-a-database-using-database-navigator/)