MAT335E
PROGRAMMING ALGORTIHMS

ITU STUDENT REGISTRATION PROGRAM

Instructor: Assist. Prof. Dr. Burcu TUNGA

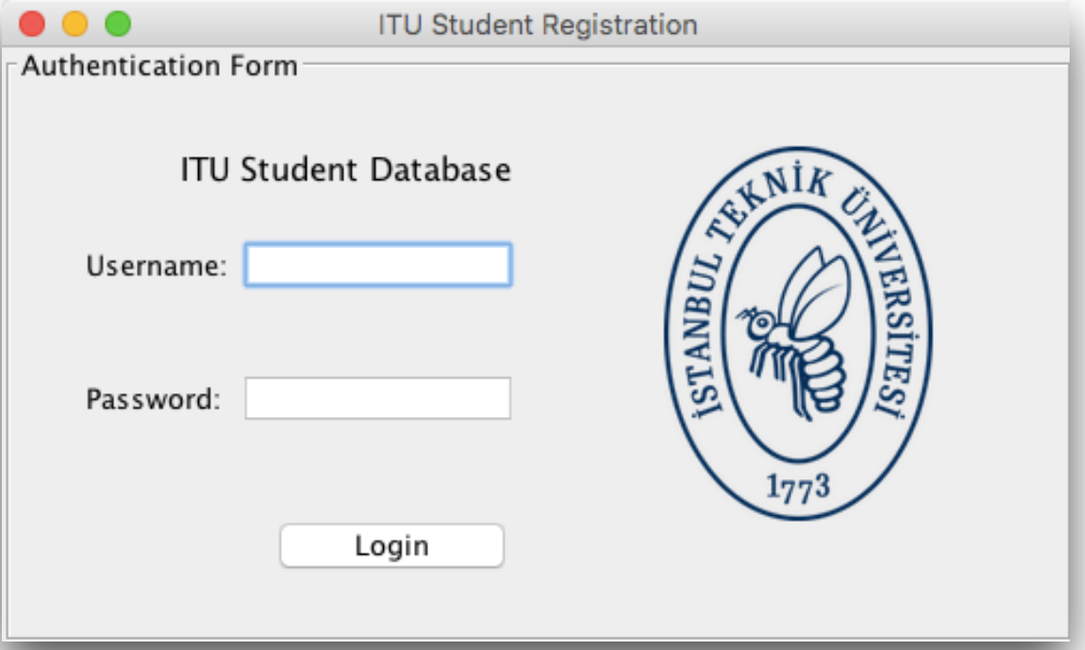
090140328 - Nur Sinem DERE
090140354 - Beyza ŞENGÜL

ITU STUDENT REGISTRATION

- ITU Student Registration is the program that can be logged in with two different types of users. It is written to be used by students and lecturers.

- Different utilizations are defined for each type of the users. While students are able to view contact information and exam results of all students, lecturers are able to insert new student records and grades for each student.

- MySQL, Java and Swing are used to create this program.

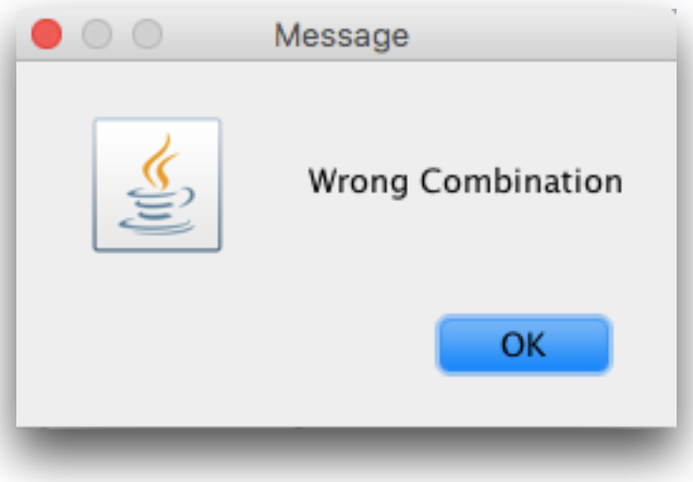


Authentication Form

The screen that can be seen on the left is the main page of the ITU Student Database program.

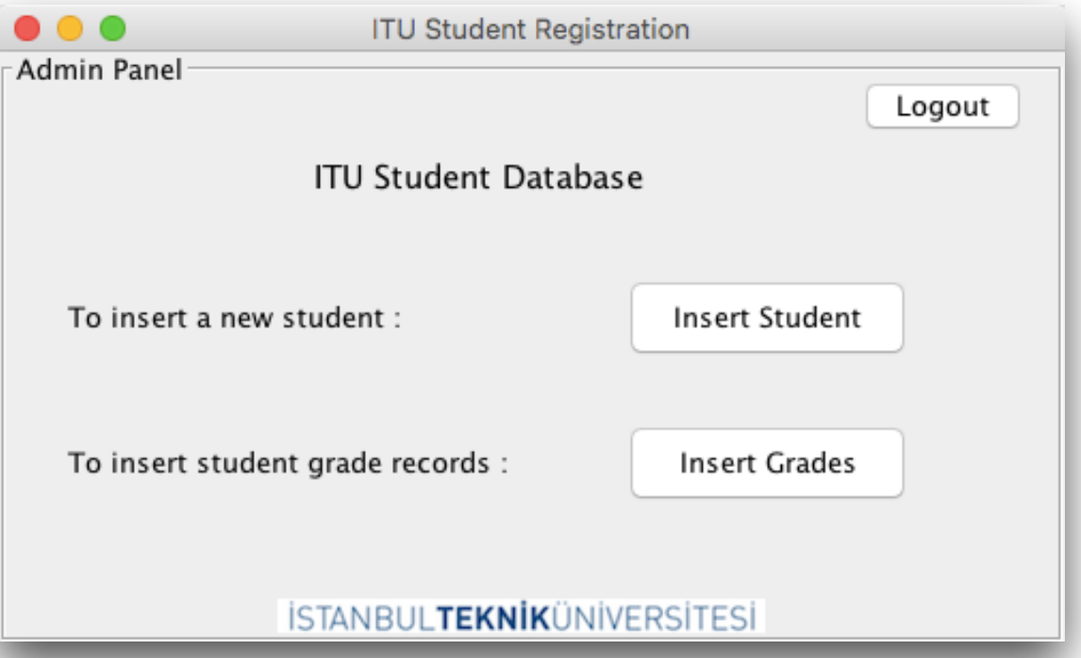
There can be logged in as two different types, one is admin, other one is user. The difference between the type of logins gives different access permissions to the person.

If the username and the password are entered incorrectly, the screen on the left will be shown up.



The code on the right provides the username and password matching by using the database with the query as can be seen.
The pop up is provided by the JOptionPane line.

```
public void actionPerformed(ActionEvent e) {
        try {
            Connection con = DriverManager.getConnection(url,user,password);
            PreparedStatement pst = con.prepareStatement("Select * from Permission where username=? and password=?");

            pst.setString(1, textField.getText());
            pst.setString(2, passwordField.getText());
            ResultSet rs = pst.executeQuery();

            if(rs.next()){
                System.out.println("Login Succesful");
                String s = rs.getString("type");

                if(s.equals("admin") ){
                    authPanel.setVisible(false);
                    adminPanel.setVisible(true);
                }
                else{
                    authPanel.setVisible(false);
                    userPanel.setVisible(true);
                }
            }
            else{
                JOptionPane.showMessageDialog(authPanel, "Wrong Combination");
            }
```



Admin Panel

By logging in with the “admin” type of user, the screen that is seen on the left comes next.

There are two actions that admin type of users can do, and these are inserting a new student and inserting the grades for each student into the database.

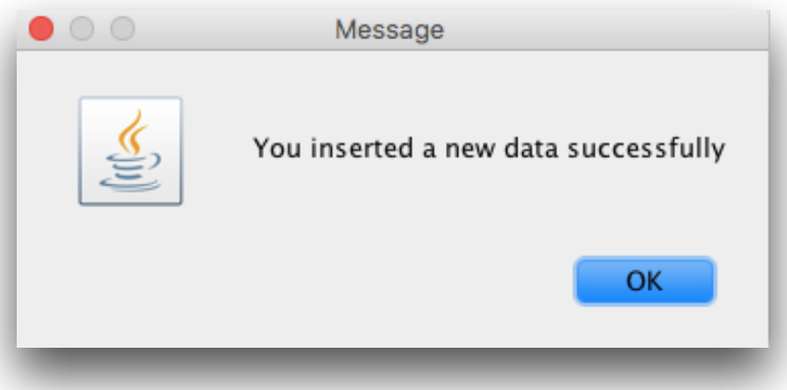
The button on the right top labeled as Logout makes the person be logged out and directs to the Authentication Form screen.

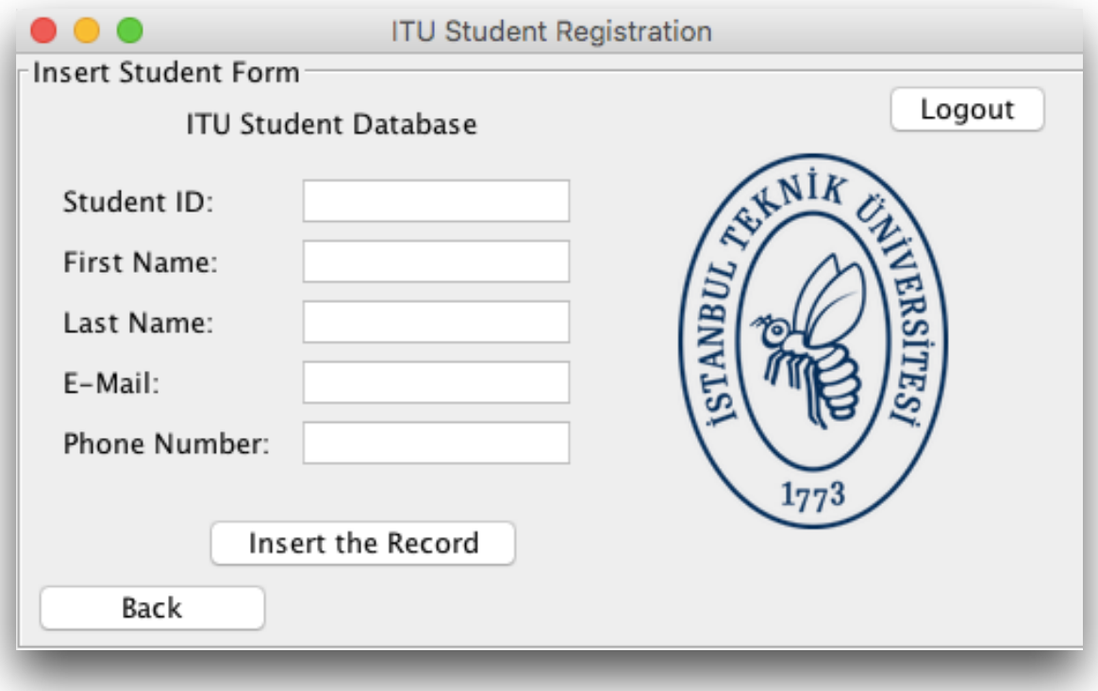
Insert Student Form

The Insert Student button directs the admin to the Insert Student Form screen that is shown by the screen below. By filling the text fields, the record of a new student can be inserted into the database. The Back button directs the admin to the Admin Panel screen.

Student ID must be unique for each student. If the admin tries to insert a student whose student ID is already existed in the database, the record will never be inserted.

If the inserting a new student action is done successfully, the program shows you a pop up indicating the situation that it is done successfully.





The inserting a new student action is provided by the insertStudent() method. With the insert query, the inserting new student into the database is being done.

The method is used when the Insert Student button is clicked.

The method gets its variables from the text fields.

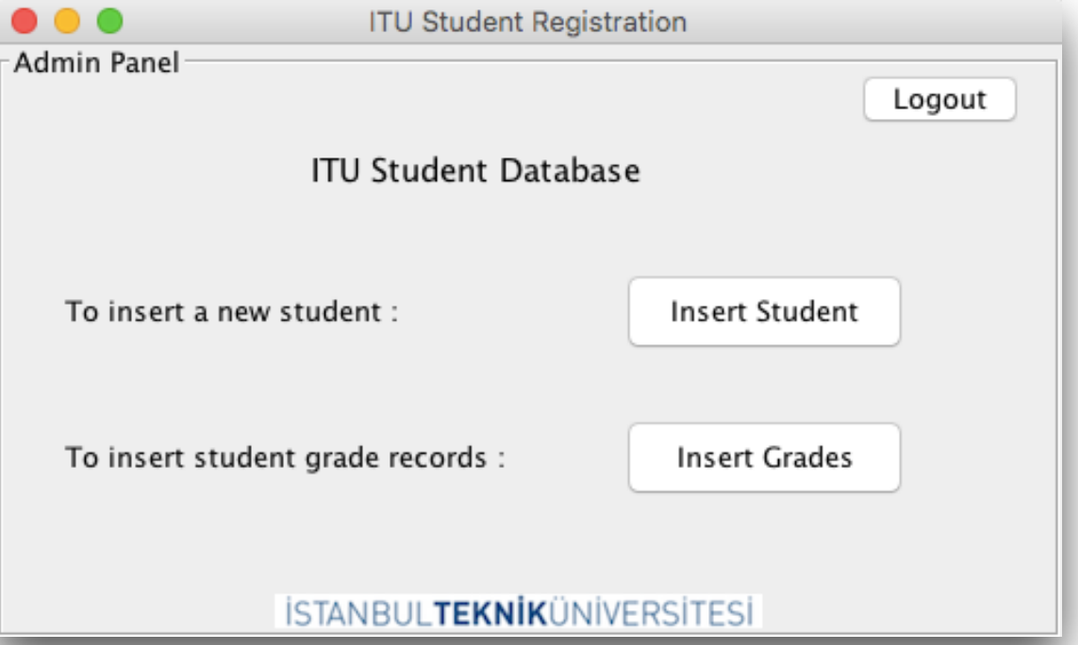
```
public void actionPerformed(ActionEvent e) {

    try {
        insertStudent(insertID.getText(), insertFirst.getText(), insertLast.getText(), insertMail.getText(), insertPhone.getText());
        JOptionPane.showMessageDialog(insertStudentPanel, "You inserted a new data successfully");

    } catch (SQLException e1) {
```

The insertStudent() method can be seen from the code below.

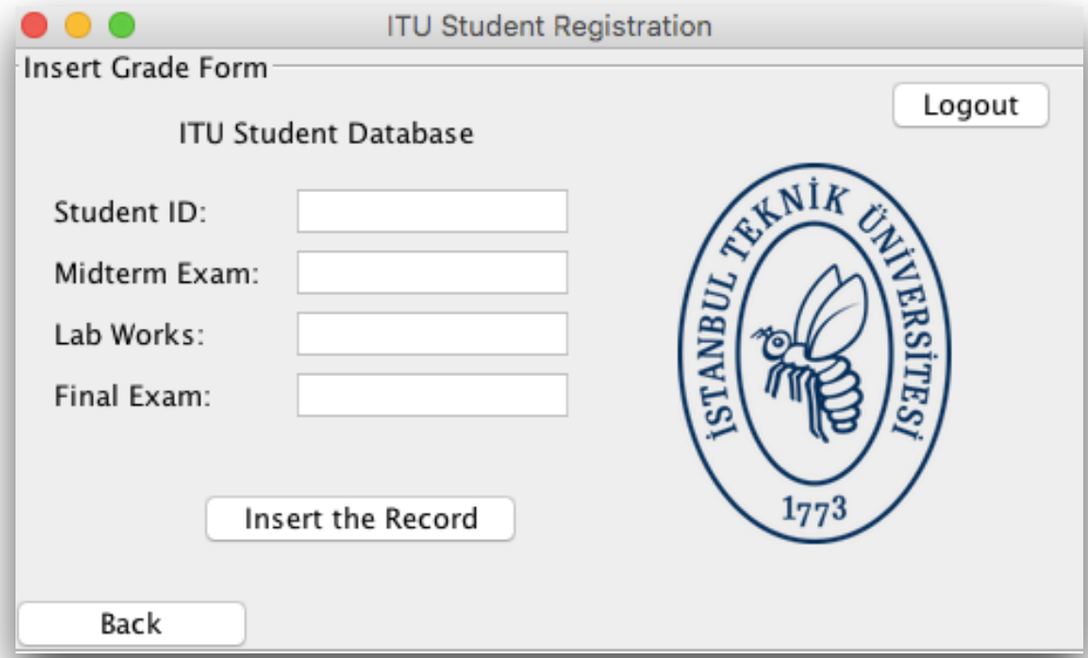
The method needs variables that are provided from the text fields.

By using the variables in the SQL query, the program will be able to insert a new student record into the database.

```
//to insert a new Student data
 public static void insertStudent(String id,String firstName,String lastName,String email,String phoneNumber) throws SQLException{


      String query = "INSERT INTO users(student_id, first_name, last_name, e_mail, phone_number)"+
      "VALUES('"+id+"','"+firstName+"','"+lastName+"','"+email+"','"+phoneNumber+"');";

      Connection con = DriverManager.getConnection(url, user, password);
      PreparedStatement preparedStmt = con.prepareStatement(query);
      preparedStmt.execute();
```



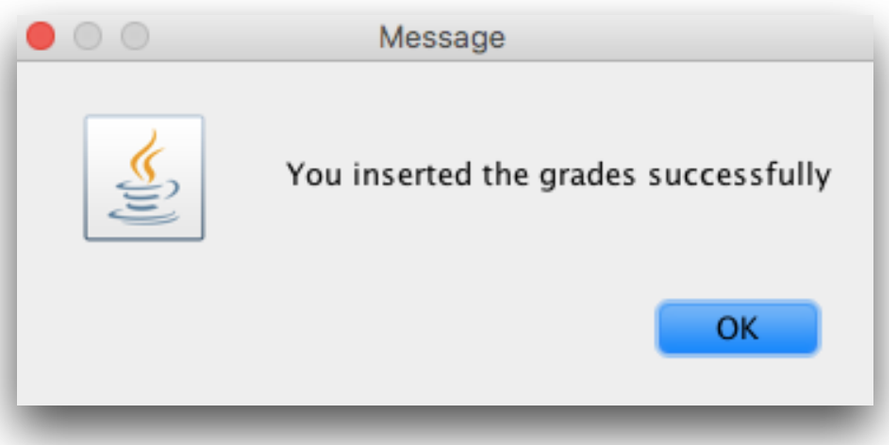
The Insert Grades button in the Admin Panel directs the admin to the Insert Grades Form screen that is shown in the screen below.



Insert Grades Form

By filling the text fields, the record of the grades of each student will be inserted into the database.

Again, the Back button directs the admin to the Admin Panel screen.



The student ID must belong to one of the existing students.

The grades can be updated by inserting each of them once again. If the action is done successfully, the pop up will be shown up to indicate it.

The inserting the grades of the students action is provided by the insertGrades() method. With the insert query, the inserting new records into the database is being done.

The method is used when the Insert Grades button is clicked.

The method gets its variables from the text fields.

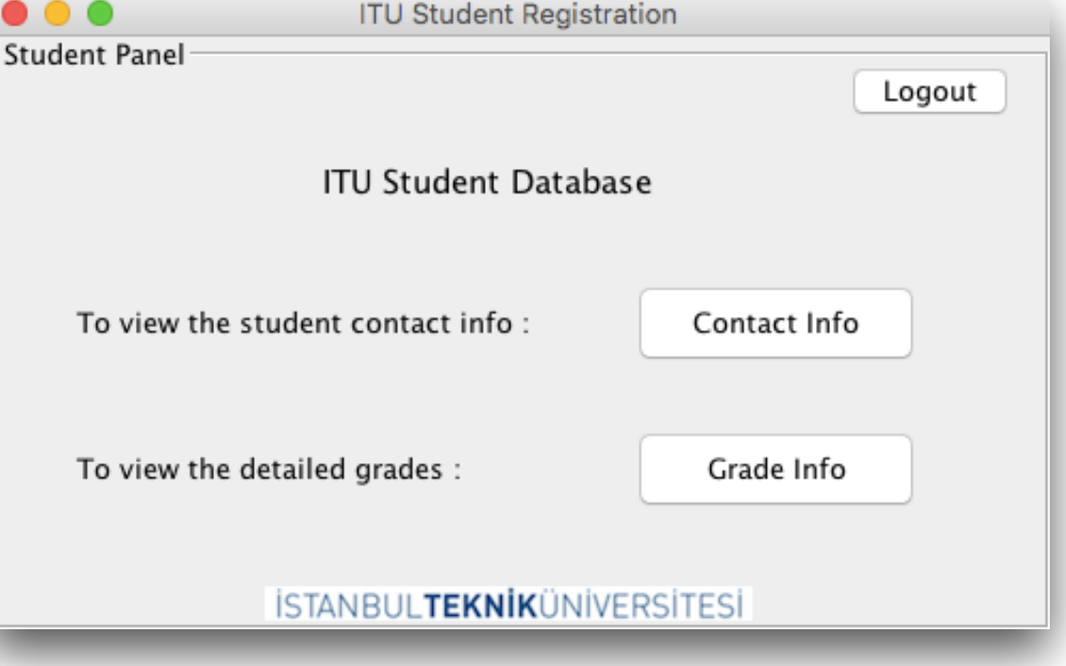
```
public static void insertGrades(String id,String midtermGrade,String labGrade,String finalGrade) throws SQLException{


    String query = "UPDATE users SET midtermResult = "+midtermGrade+", labResult = "+labGrade+", finalResult = "+finalGrade+" "
          + "where student_id = "+id;

Connection con = DriverManager.getConnection(url, user, password);
    PreparedStatement preparedStmt = con.prepareStatement(query);
    preparedStmt.execute();
```

The insertGrades() method can be seen in the code below.

The method needs variables that are provided from the text fields.

By using the variables in the SQL query, the program will be able to insert new records into the database.

```
public void actionPerformed(ActionEvent e) {
    try {
        insertGrades(textField_7.getText(), textField_6.getText(), textField_5.getText(), textField_4.getText());
        JOptionPane.showMessageDialog(insertGradePanel, "You inserted the grades successfully");
```



Student Panel

By logging in with the “user” type, the screen that is seen on the left comes next.

There are two actions that user types of logins can do, and these are viewing contact information of any students and viewing detailed grades information that has been inserted by the admin.

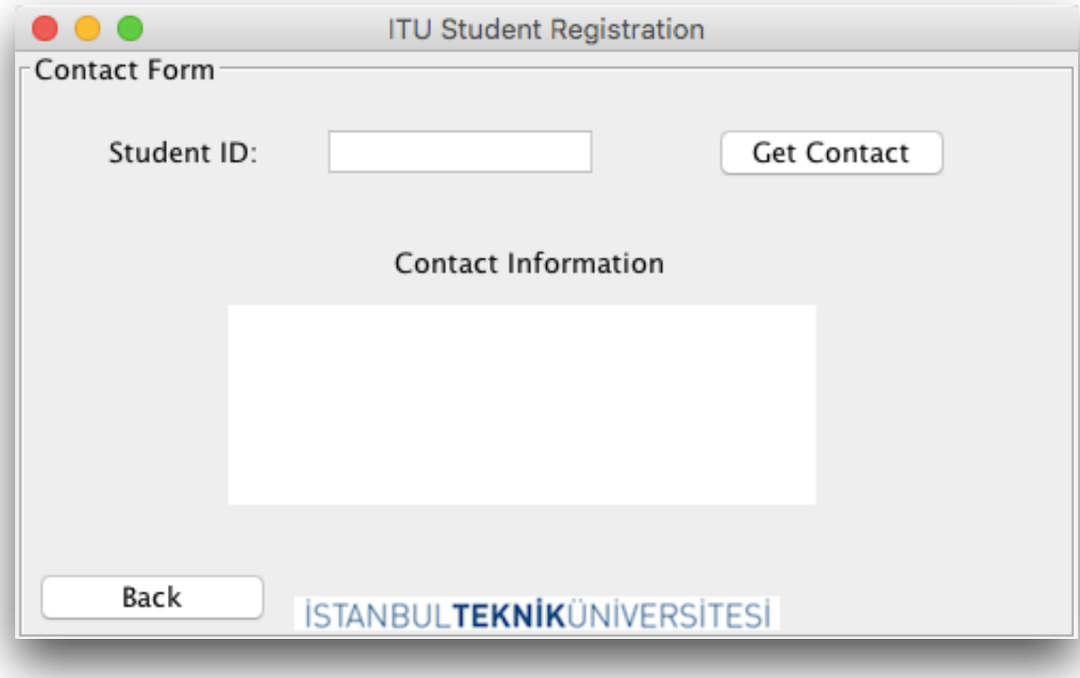
The button on the right top labeled as Logout makes the person be logged out and directs to the Authentication Form screen.

Contact Form

The Contact Info button in the Student Panel directs the user to the Contact Form screen that is shown by the screen on the right.

By filling the text field with a student ID that is existed and clicking the button “Get Contact”, fills the text area labeled “Contact Information” with the contact information of the student with the student ID.

By clicking the Back button on the right-bottom, the user will be directed to the Student Panel.



If the student ID that has been entered is not existed in the database, then the program shows a pop up that indicates that the student ID is not right.

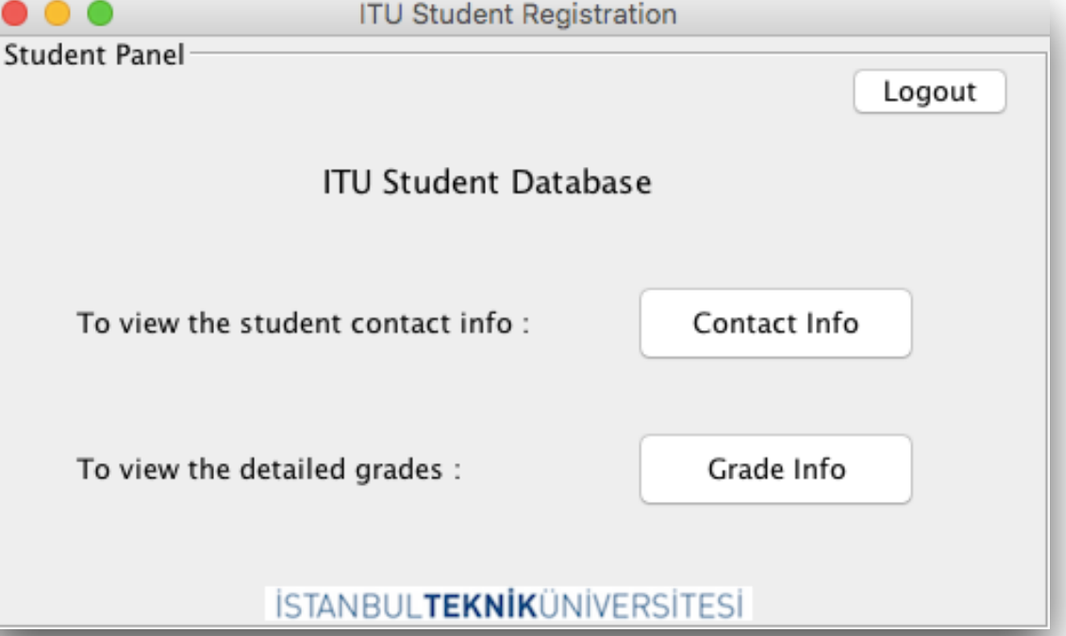
The getting student's contact information and filling the text area action are provided by the getStudentToTextArea() method.
The method gets the student ID from the text field, and then uses it to run the method.
The method is used when the "Get Contact" button is clicked.

```
public void actionPerformed(ActionEvent e) {
    try {
        if(getStudentToTextArea(textField_2.getText()) != null){
        textPane_1.setText(getStudentToTextArea(textField_2.getText()));
        }
        else
            JOptionPane.showMessageDialog(userPanel, "Wrong Student ID");
```

The getStudentToTextArea() method is seen in the code below.

By using the student ID that has been filled, in the SQL query, the program will be able to get the student's contact information, create a string that will be written and return the string to the text area.

```
//to fill the text area with the student information
 public static String getStudentToTextArea(String student_id) throws SQLException{
       Connection con = DriverManager.getConnection(url, user, password);
       Statement stmt = con.createStatement() ;

       String query = "SELECT * FROM USERS WHERE student_id ='"+student_id+"';";
       ResultSet rs = stmt.executeQuery(query);

       String st = null;

       while(rs.next()){
           String first = rs.getString("first_name");
           String last = rs.getString("last_name");
           String email = rs.getString("e_mail");
           String phone = rs.getString("phone_number");

           st = "ID: "+student_id+"\nFirst Name: "+first+"\nLast Name: "+last+"\nE-Mail: "+email+"\nPhone Number: "+phone;
 }
       System.out.println(st);
       return st;
```

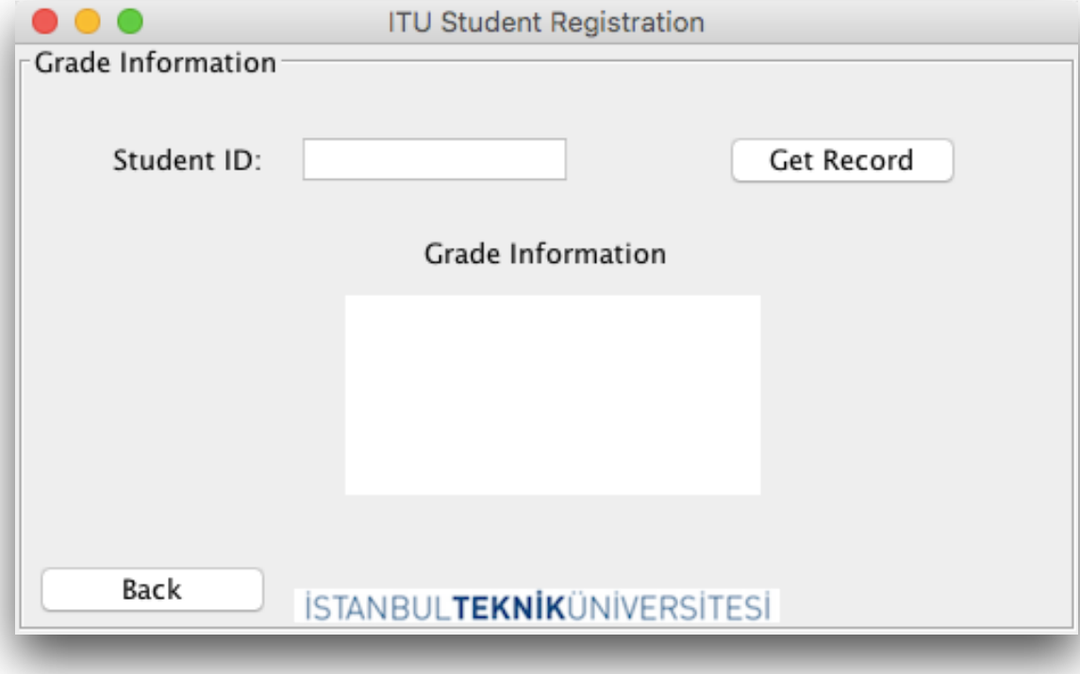


The Grade Info button in the Student Panel directs to the Grade Information screen that is shown by the screen below.

Grade Information

By filling the text field, the record of the grades of whose student ID is entered, will be seen in the text area, below the “Grade Information” label.
It also shows the letter grade by using the exam results.

Again, the Back button directs the user to the Student Panel screen.



The getting student's grade results and filling the text area action are provided by the getResultToTextArea() method.
The method gets the student ID from the text field, and then uses it to run the method.
The method is used when the "Get Record" button is clicked.

```
public void actionPerformed(ActionEvent e) {
    try {
        if(getResultToTextArea(textField_3.getText()) != null){
        textPane_2.setText(getResultToTextArea(textField_3.getText()) + " ");
        }
        else
            JOptionPane.showMessageDialog(userPanel, "Wrong Student ID");
```

The getResultToTextArea() method can be seen in the code below.

By using the student ID that has been filled, in the SQL query, the program will be able to get the student's exam results, create a string that will be written and return the string to the text area. (next slide)

```
public static String getResultToTextArea(String student_id) throws SQLException{
      Connection con = DriverManager.getConnection(url, user, password);
      Statement stmt = con.createStatement() ;

      String query = "SELECT student_id, midtermResult, finalResult, labResult FROM users WHERE student_id ='"+student_id+"';";
      ResultSet rs = stmt.executeQuery(query);

      String str = null;


      while(rs.next()){
          String id = rs.getString("student_id");
          String midtermR = rs.getString("midtermResult");
          String finalR = rs.getString("finalResult");
          String labR = rs.getString("labResult");
```


...

```
double labPerc = Integer.valueOf(labR)*0.2;
double finalPerc = Integer.valueOf(finalR)*0.5;
double midtermPerc = Integer.valueOf(midtermR)*0.3;

double sum = labPerc + finalPerc + midtermPerc;
String grade = null;
if(sum>=90)
    grade="AA";
else if(sum>=80 && sum <90)
    grade="BA";
else if(sum>=70 && sum <80)
    grade="BB";
else if(sum>=60 && sum <70)
    grade="CB";
else if(sum>=50 && sum <60)
    grade="CC";
else if(sum>=40 && sum <50)
    grade="DC";
else if(sum>=30 && sum <20)
    grade="DD";
else
    grade ="FF";
```

```
        str = "Midterm Exam: "+midtermR+"\nLab: "+labR+"\nFinal Exam: "+finalR+"\n\nLetter Grade: "+grade;

}
      System.out.println(str);
      return str;
```

The exam results are held as VARCHAR() in the database, after change their data types to Integer, and multiply them by the each of the percentage effects the result and return the sum of them to the if-else statement gets the letter grade.

The method returns the string that will be used and be shown at the “Grade Information “labeled text area.

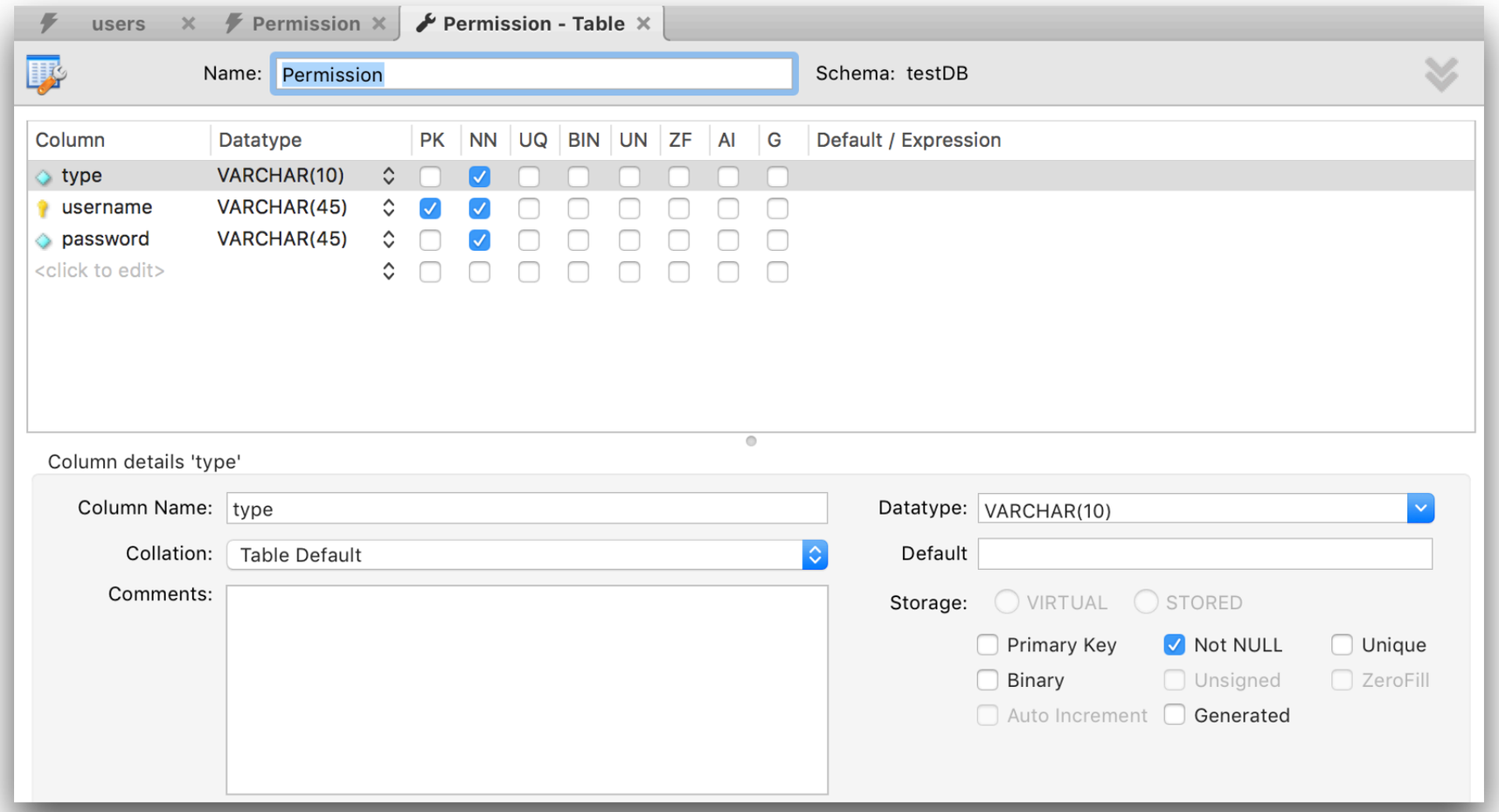
Permission Table

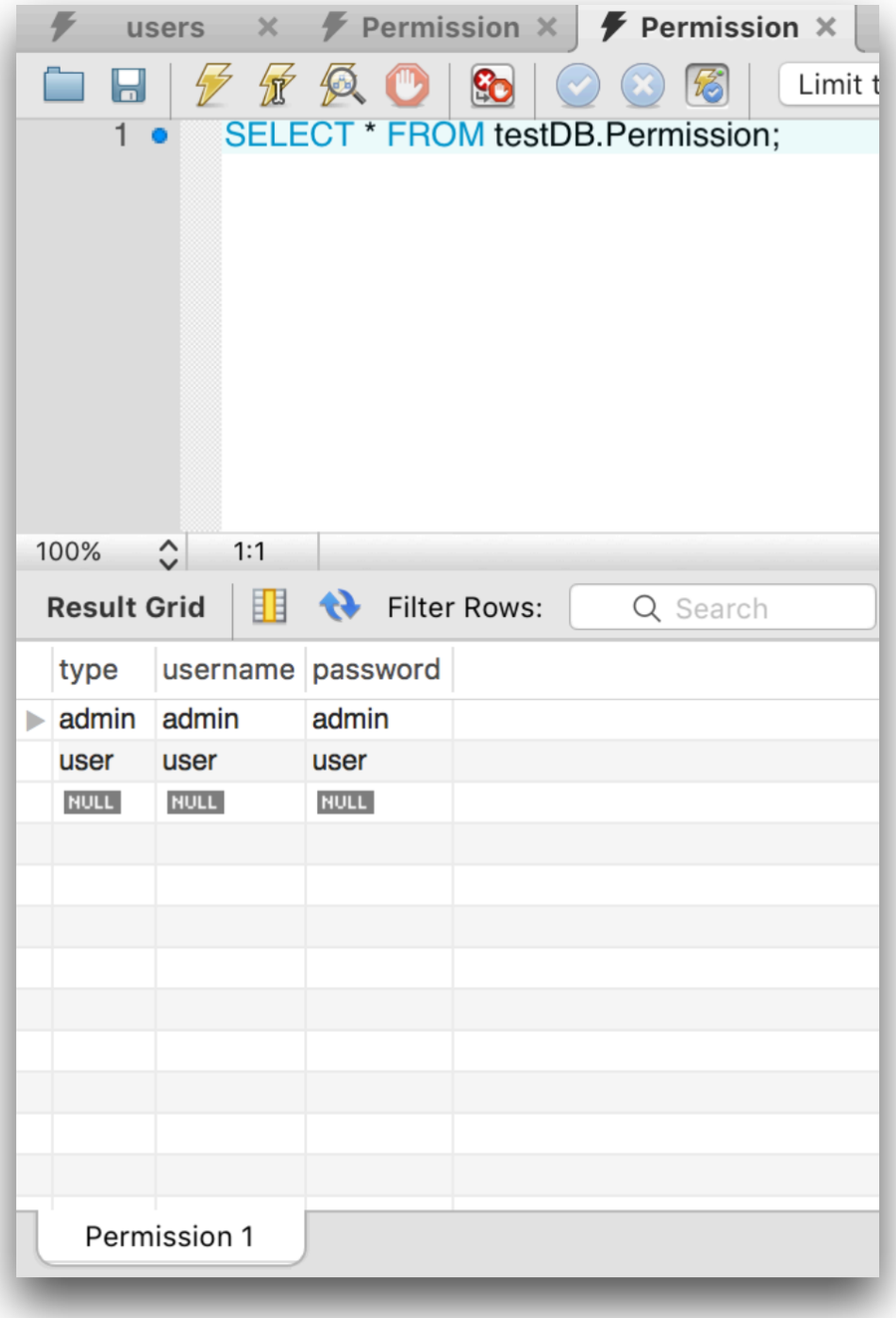
Since there are two login types in the program, a permission table, which includes type, username and password columns, is created.

In the permission table, username is the primary key and can not be null which means username is the unique variable in the permission table.

The datatype of all variables in the users table is VARCHAR().

The type column is to keep the login type so that the program directs the user to the Admin Panel or Student Panel according to the user type.





The permission table is filled with some sample data.

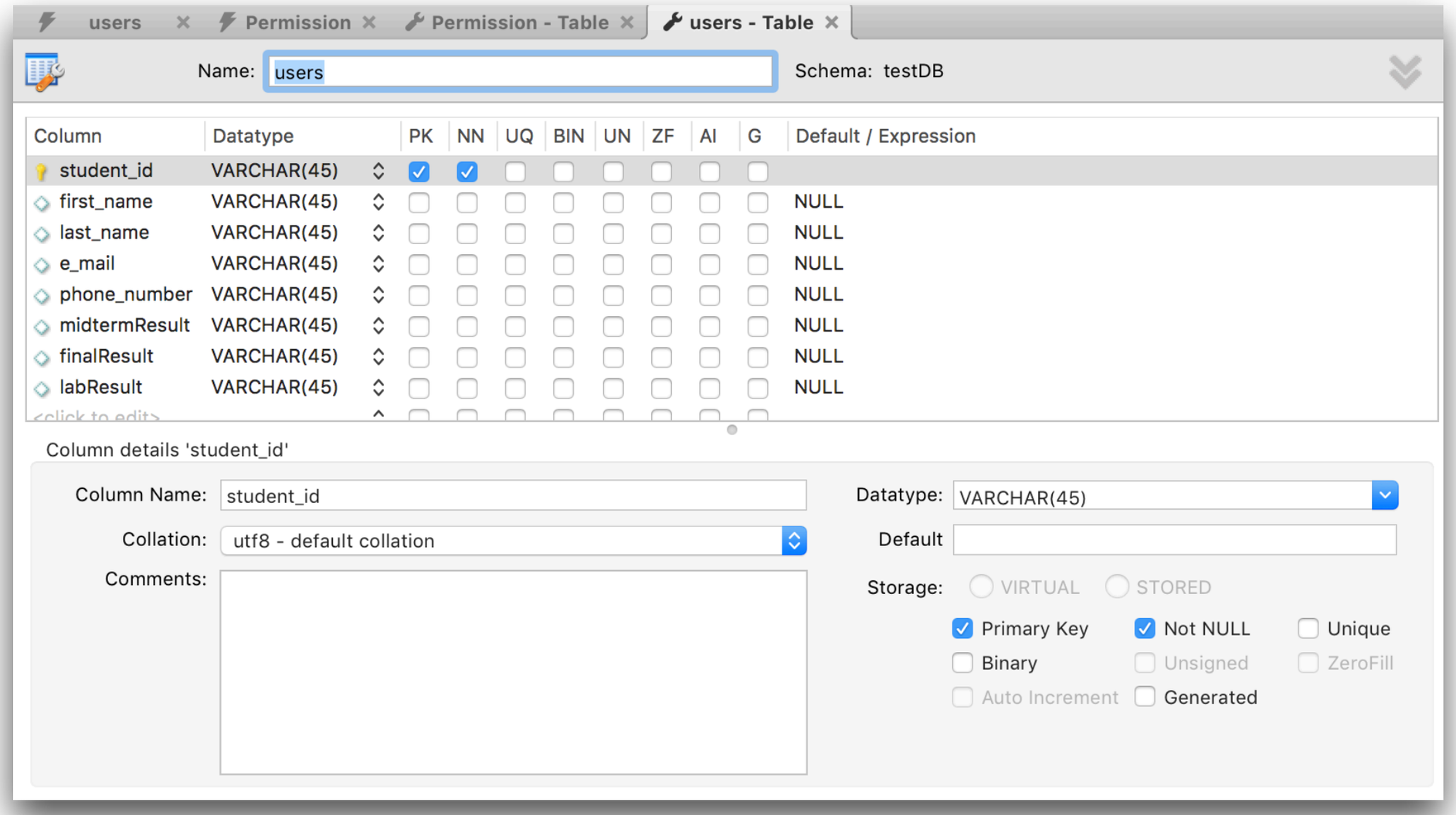
This table can not be edited by the admin.

Users Table

In the users table, there are student_id, which is the primary key, first_name, last_name, e_mail, phone_number, midtermResult, finalResult and labResult columns.

student_id is unique.

The datatype of all variables in the users table is VARCHAR().



users | Permission

Limit to 1000 rows

```
1  SELECT * FROM testDB.users;
```

100% 28:1

Result Grid | Filter Rows: Search | Edit: | Export/Import:

student_id	first_name	last_name	e_mail	phone_number	midtermR...	finalResult	labResult
040130018	Merve	Oral	oralme@itu.edu.tr	05376789032	0	43	75
090030010	Tolga	Yavuz	yavuzto@itu.edu.tr	05646543489	65	78	85
090090419	Basar	Budak	budakba@itu.edu.tr	05238760012	0	42	67
090120158	Selin	Unal	unalse@itu.edu.tr	05546712390	0	89	70
090120424	Semanur	Karabay	karabays@itu.edu.tr	05443215647	35	37	69
090120435	Ozan	Koksal	koksalo@itu.edu.tr	05341113423	60	63	90
090120436	Beril	Darici	daricib@itu.edu.tr	05334568812	30	57	78
090120447	Kagan	Borekci	borekcika@itu.edu.tr	05424789923	60	75	85
090120505	Tugce	Basarslan	basarslan@itu.edu.tr	05315678901	20	45	80
090120520	Ogulcan	Cakir	cakirog@itu.edu.tr	05463219310	0	45	70
090120535	Merve	Etili	etilime@itu.edu.tr	05405612304	0	50	20
090120537	Burcu	Akbalik	akbalikme@itu.edu.tr	05305103206	20	35	75
090120911	A.	Nurtozhiev	nurtozhiev@itu.edu.tr	05396134591	0	50	90
090120917	Siti	Zahrotun	zahrotun@itu.edu.tr	05472083594	25	55	65
090130303	Berk	Gurel	gurelbe@itu.edu.tr	05484044269	60	74	95
090130319	Cagla	Demir	demirca@itu.edu.tr	0535114554	0	65	55
090130321	Sinan	Kekik	kekiksi@itu.edu.tr	05215403249	80	67	95
090130322	Seda	Ipek	ipekse@itu.edu.tr	05674421581	35	70	100
090130327	Senem	Bildirici	bildiricise@itu.edu.tr	05495521204	65	80	100
090130338	Doruk	Ismen	ismendo@itu.edu.tr	05824534392	70	90	85
090140304	Muhamm...	Duran	duranmuh@itu.edu.tr	05286727302	65	70	80
090140311	Isilay	Tuncer	tunceris@itu.edu.tr	05368105633	55	67	83
090140328	Nursinem	Dere	derenu@itu.edu.tr	05316032517	95	90	100
090140329	Umut	Cabuk	cabukum@itu.edu.tr	05428502538	30	45	60
090140333	Rabia	Aydin	aydinra@itu.edu.tr	05306371277	75	78	65
090140340	Anil	Ulusoy	ulusoyan@itu.edu.tr	05279347829	20	50	65
090140343	Ipek	Sarsilmaz	sarsilmazip@itu.edu.tr	05379129939	0	45	20
090140352	Ata	Oz	ozata@itu.edu.tr	05217882853	100	90	100
090140353	Oguz	Oztinaz	oztinazog@itu.edu.tr	05368199124	100	95	100

The users table is filled with some sample data.

When it is inserted data in the Insert Student and Insert Grade Form, this table gets updated.

This table can be inserted and updated by the admin.