

Here is the Repository that proves that I haven't copied it from anywhere but wrote it all myself =>

[GitHub](#)

I used all the best practices that I've gained through out my 1 year of development

Question 1

Student Bio Data Form

Source Code

Main Form

```
1 reference
private void btn_insert_Click(object sender, EventArgs e){
    frm_insert insertion_form = new frm_insert();
    insertion_form.Show();
}

1 reference
private void btn_update_Click(object sender, EventArgs e){
    FRM_update_rec updation_form = new FRM_update_rec();
    updation_form.Show();
}

1 reference
private void btn_view_Click(object sender, EventArgs e){
    FRM_view_rec view_Rec = new FRM_view_rec();
    view_Rec.Show();
}

1 reference
private void btn_delete_Click(object sender, EventArgs e)
{
    FRM_delete_rec delete_Rec = new FRM_delete_rec();
    delete_Rec.Show();
}

1 reference
private void btn_search_Click(object sender, EventArgs e){
    FRM_search search = new FRM_search();
    search.Show();
}

1 reference
private void btn_exit_Click(object sender, EventArgs e){Application.Exit();}
```

Insert

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void frm_insert_Load(object sender, EventArgs e){txtbx_id.Focus();}

1 reference
private void btn_insert_record_Click(object sender, EventArgs e){
    conn.Open();
    if (conn.State == ConnectionState.Open){
        int id = int.Parse(txtbx_id.Text);
        string first_name = txtbx_fname.Text;
        string last_name = txtbx_lname.Text;
        int age = int.Parse(txtbx_age.Text);
        string sql = "INSERT INTO `final_exam`.`student_table` (`student_id`, `student_first_name`, `student_last_name`, `student_age`) " +
            "VALUES " + "(" + id + ", '" + first_name + "', '" + last_name + "', " + age + ")";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
        txtbx_id.Clear();
        txtbx_fname.Clear();
        txtbx_lname.Clear();
        txtbx_age.Clear();
        conn.Close();
    }else{MessageBox.Show("Connection Failed");}
}

```

Update

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void FRM_update_rec_Load(object sender, EventArgs e){txtbx_id.Focus();}
1 reference
private void btn_update_Click(object sender, EventArgs e){
    conn.Open();
    if(conn.State == ConnectionState.Open){
        int id = int.Parse(txtbx_id.Text);
        string first_name = txtbx_fname.Text;
        string last_name = txtbx_lname.Text;
        int age = int.Parse(txtbx_age.Text);
        string sql = "UPDATE `final_exam`.`student_table` SET `student_first_name` = '"+first_name+"', `student_last_name` = '" +
            last_name+"', `student_age`='"+age+"' WHERE (`student_id` = '" + id+"')";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
        txtbx_age.Clear();
        txtbx_fname.Clear();
        txtbx_id.Clear();
        txtbx_lname.Clear();
        conn.Close();
    }else MessageBox.Show("Connection Failed");
}

```

View

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void FRM_view_rec_Load(object sender, EventArgs e){
    try{
        MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT * FROM final_exam.student_table", conn);
        conn.Open();
        DataSet ds = new DataSet();
        adapter.Fill(ds, "student_table");
        dataGridView1.DataSource = ds.Tables["student_table"];
        conn.Close();
    }catch (Exception ex){
        MessageBox.Show(ex.Message);
    }
}

```

Delete

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void FRM_delete_rec_Load(object sender, EventArgs e){txtbx_id.Focus();}

1 reference
private void btn_dlt_Click(object sender, EventArgs e){
    conn.Open();
    if (conn.State == ConnectionState.Open){
        int id = int.Parse(txtbx_id.Text);
        string sql = "DELETE FROM final_exam.student_table WHERE `student_id` = '"+id+"'";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
        txtbx_id.Clear();
        conn.Close();
    }else MessageBox.Show("Connection Failed");
}

```

Search

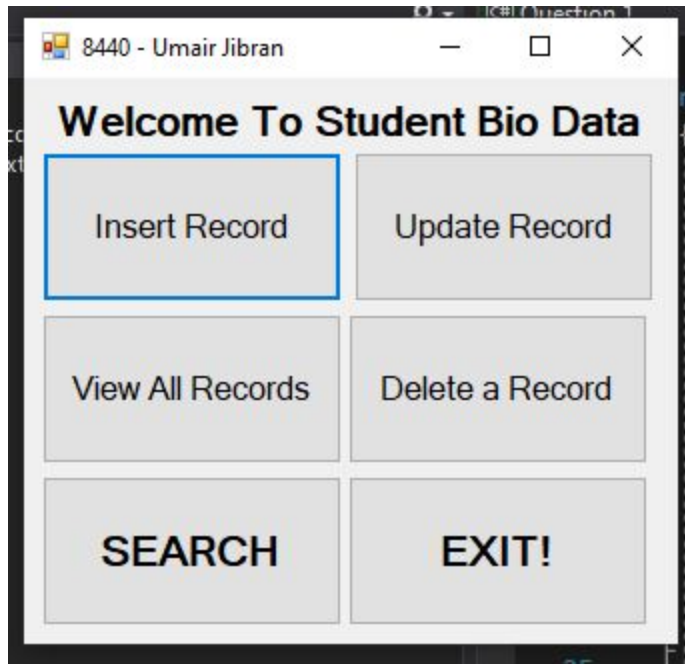
```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void FRM_search_Load(object sender, EventArgs e){}
1 reference
private void button1_Click(object sender, EventArgs e){
    string name = txtbx_name.Text;
    try{
        MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT * FROM final_exam.student_table WHERE student_first_name LIKE '"+name+
        "%' OR student_last_name LIKE '"+name+"'", conn);
        conn.Open();
        DataSet ds = new DataSet();
        adapter.Fill(ds, "student_table");
        dataGridView1.DataSource = ds.Tables["student_table"];
        conn.Close();
    }catch (Exception ex){MessageBox.Show(ex.Message);}
}

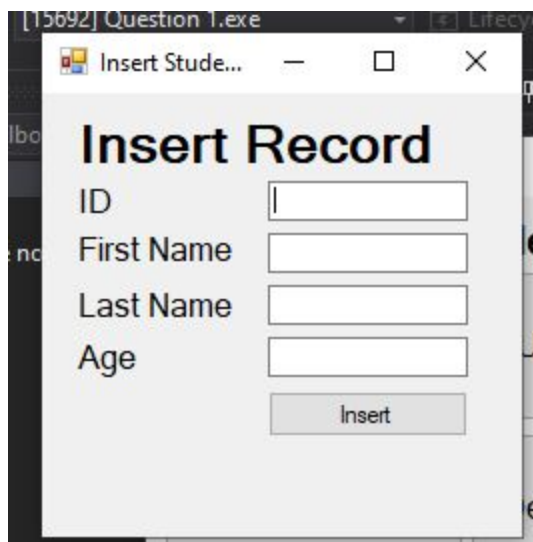
```

Output

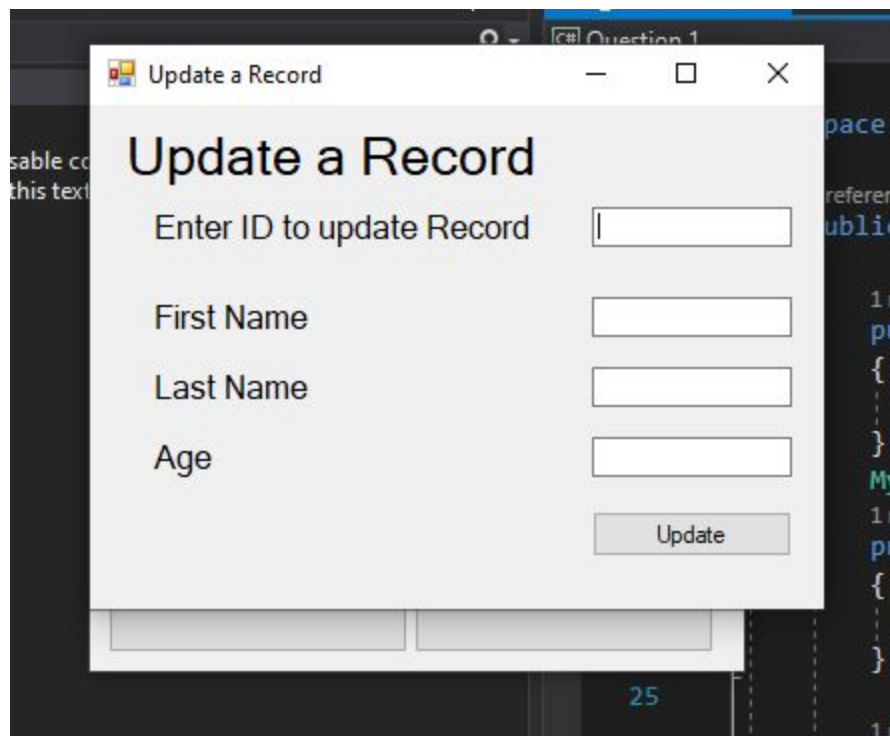
Main Form



Insert

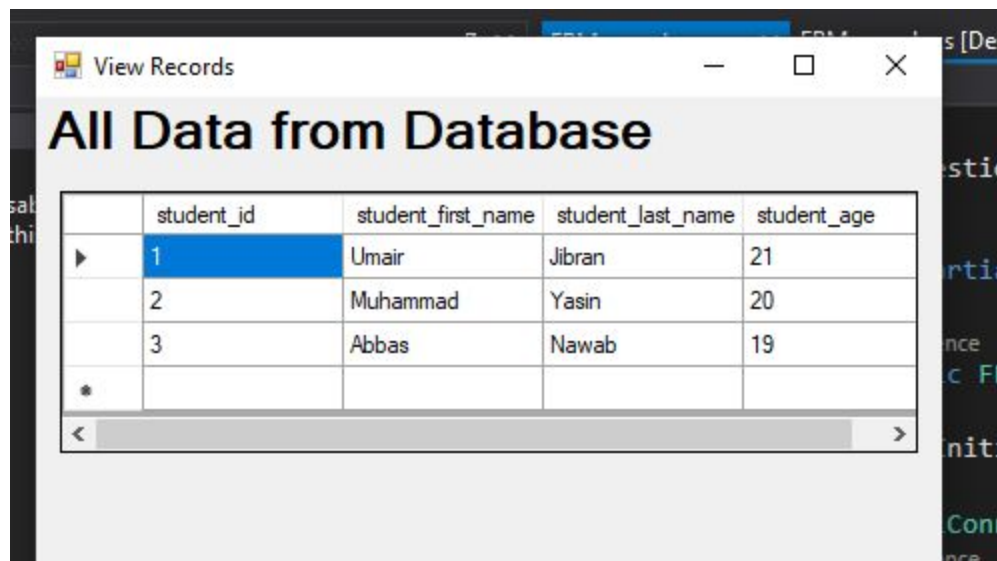


Update



The screenshot shows a Windows-style dialog box titled "Update a Record". It contains four input fields for "Enter ID to update Record", "First Name", "Last Name", and "Age". An "Update" button is located at the bottom right of the dialog. The dialog is overlaid on a background showing a code editor with C# code.

View

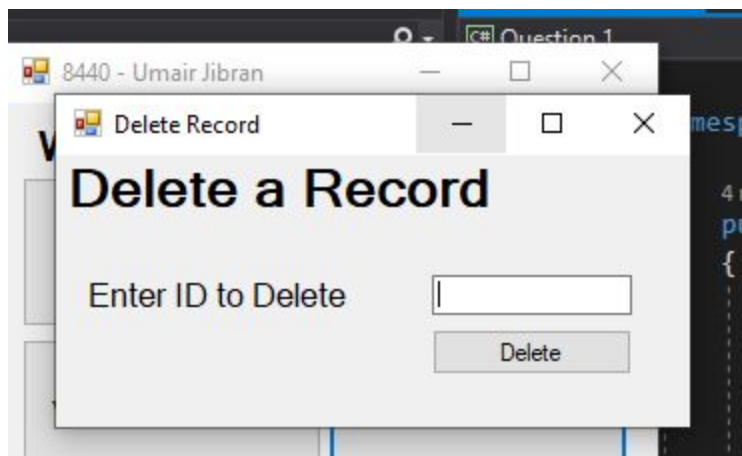


The screenshot shows a Windows-style dialog box titled "View Records". It displays a table titled "All Data from Database" with the following data:

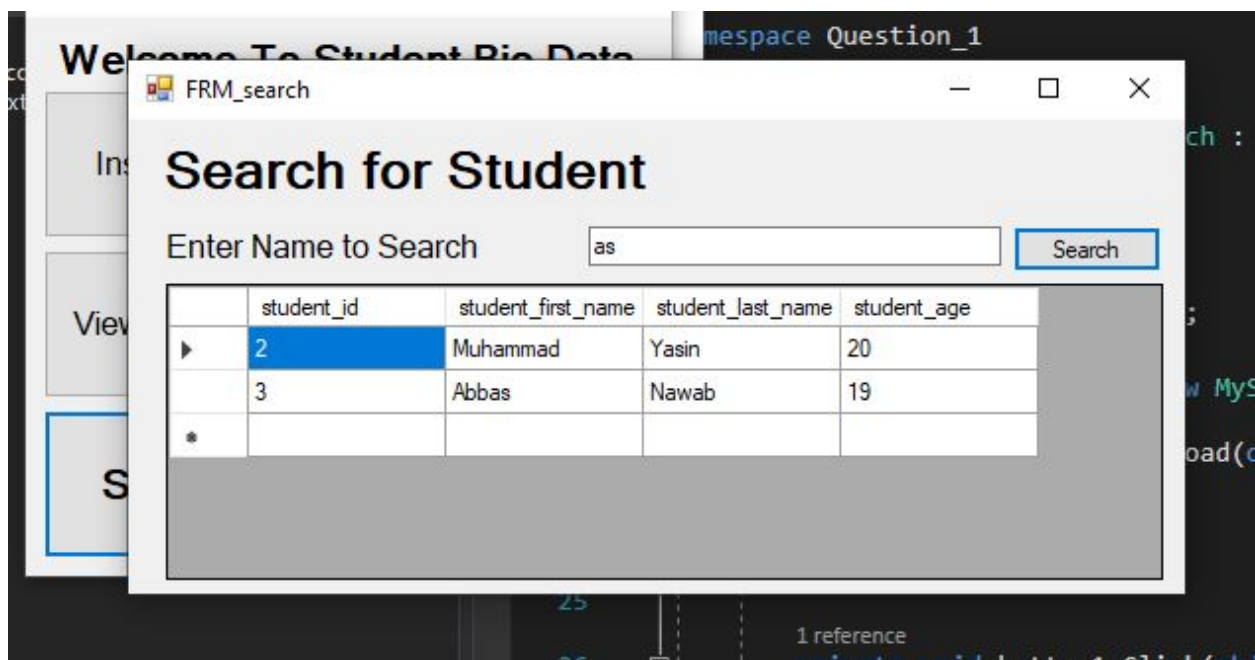
	student_id	student_first_name	student_last_name	student_age
▶	1	Umair	Jibran	21
	2	Muhammad	Yasin	20
	3	Abbas	Nawab	19
*				

The table has a scrollbar at the bottom, and the first row is highlighted in blue.

Delete



Search



Question 2

Stationary Management

Source Code

Main Form

```
1 reference
private void btn_ins_Click(object sender, EventArgs e){
    FRM_insert insert = new FRM_insert();
    insert.Show();
}
1 reference
private void btn_dlt_Click(object sender, EventArgs e){
    FRM_dlt dlt = new FRM_dlt();
    dlt.Show();
}
1 reference
private void btn_upd_Click(object sender, EventArgs e){
    FRM_upd upd = new FRM_upd();
    upd.Show();
}
1 reference
private void btn_view_Click(object sender, EventArgs e){
    FRM_view view = new FRM_view();
    view.Show();
}
1 reference
private void btn_search_Click(object sender, EventArgs e){
    FRM_search search = new FRM_search();
    search.Show();
}
```

Insert

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void btn_add_Click(object sender, EventArgs e){
    string name = txtbx_name.Text;
    string rprice = txtbx_rprice.Text;
    string pprice = txtbx_pprice.Text;
    string qty = txtbx_qty.Text;
    if (conn.State == ConnectionState.Open){
        string sql = "INSERT INTO `final_exam`.`stationary_store` (`item_name`, `item_purchase_price`, `item_retail_price`, " +
            "`item_quantity`) VALUES ('"+name+"', '"+rprice+"', '"+pprice+"', '"+qty+"')";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
    }
}
1 reference
private void FRM_insert_Load(object sender, EventArgs e){conn.Open();}
1 reference
private void btn_rst_Click(object sender, EventArgs e){
    txtbx_name.Clear();
    txtbx_pprice.Clear();
    txtbx_rprice.Clear();
    txtbx_qty.Clear();
}

```

Update

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void btn_upd_Click(object sender, EventArgs e){
    conn.Open();
    if (conn.State != ConnectionState.Open) MessageBox.Show("Connection Failed");
    else
    {
        int id = int.Parse(txtbx_id.Text);
        string name = txtbx_name.Text;
        string pprice = txtbx_pprice.Text;
        string rprice = txtbx_rprice.Text;
        string qty = txtbx_qty.Text;
        string sql = "UPDATE `final_exam`.`stationary_store` SET `item_name` = '"+name+"', `item_purchase_price` = '"+pprice+"', " +
            "`item_retail_price` = '"+rprice+"', `item_quantity` = '"+qty+"' WHERE (`item_id` = '"+id+"')";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
        conn.Close();
    }
}

```


View

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void FRM_view_Load(object sender, EventArgs e){
    try{
        MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT * FROM final_exam.stationary_store", conn);
        conn.Open();
        DataSet ds = new DataSet();
        adapter.Fill(ds, "stationary_store");
        dataGridView1.DataSource = ds.Tables["stationary_store"];
        conn.Close();
    }catch (Exception ex){
        MessageBox.Show(ex.Message);
    }
}

```

Delete

```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void btn_dlt_Click(object sender, EventArgs e){
    conn.Open();
    if (conn.State != ConnectionState.Open) MessageBox.Show("Connection Failed");
    else{
        int id = int.Parse(txtbx_id.Text);
        string sql = "DELETE FROM final_exam.stationary_store WHERE `item_id` = " + id + ";";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
        conn.Close();
    }
}

```

Search

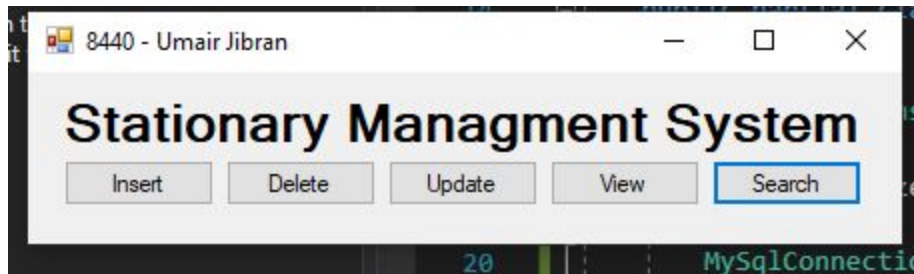
```

MySQLConnection conn = new MySqlConnection("datasource=localhost;port=3306;username=root;password=root");
1 reference
private void btn_search_Click(object sender, EventArgs e)
{
    int id = int.Parse(txtbx_id.Text);
    try
    {
        MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT * FROM final_exam.stationary_store WHERE `item_id` = " + id + " , conn);
        conn.Open();
        DataSet ds = new DataSet();
        adapter.Fill(ds, "stationary_store");
        dataGridView1.DataSource = ds.Tables["stationary_store"];
        conn.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
}

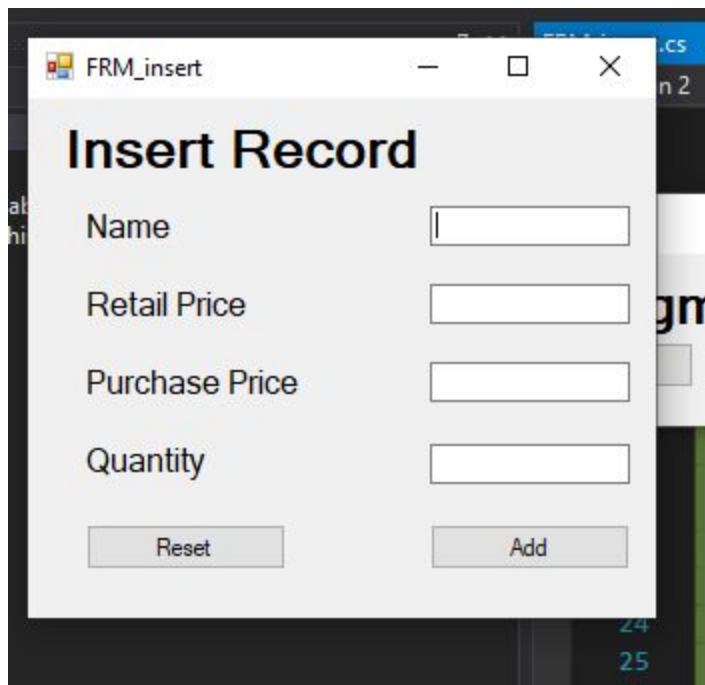
```

Output

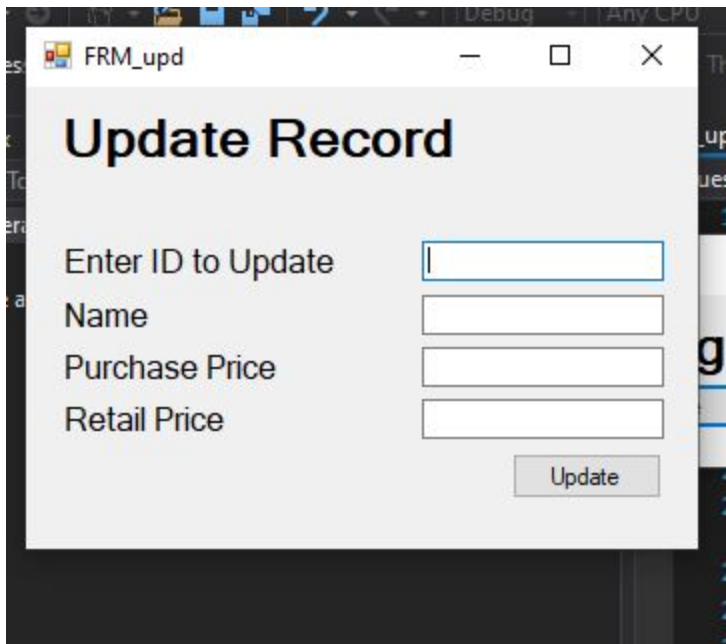
Main Form



Insert

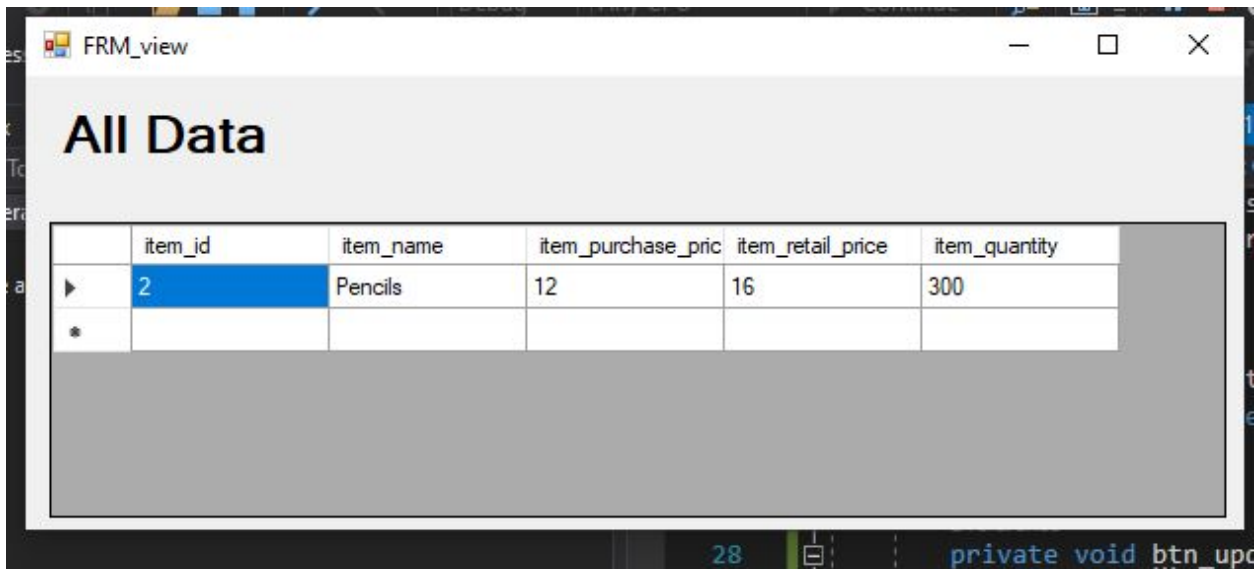


Update



The screenshot shows a Windows form titled "FRM_upd" with the main heading "Update Record". Below the heading, there are four labels with corresponding text input fields: "Enter ID to Update", "Name", "Purchase Price", and "Retail Price". At the bottom right of the form is an "Update" button.

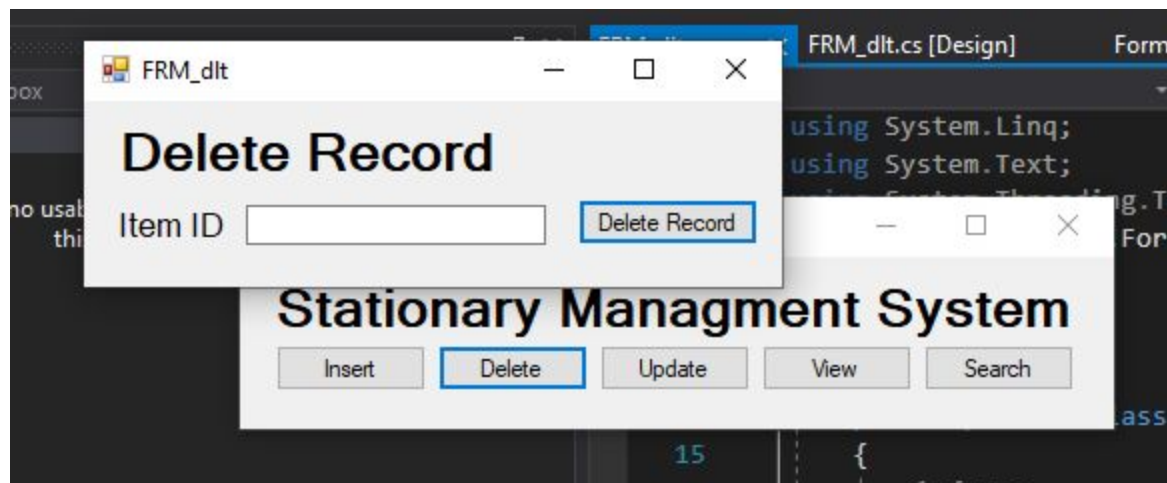
View



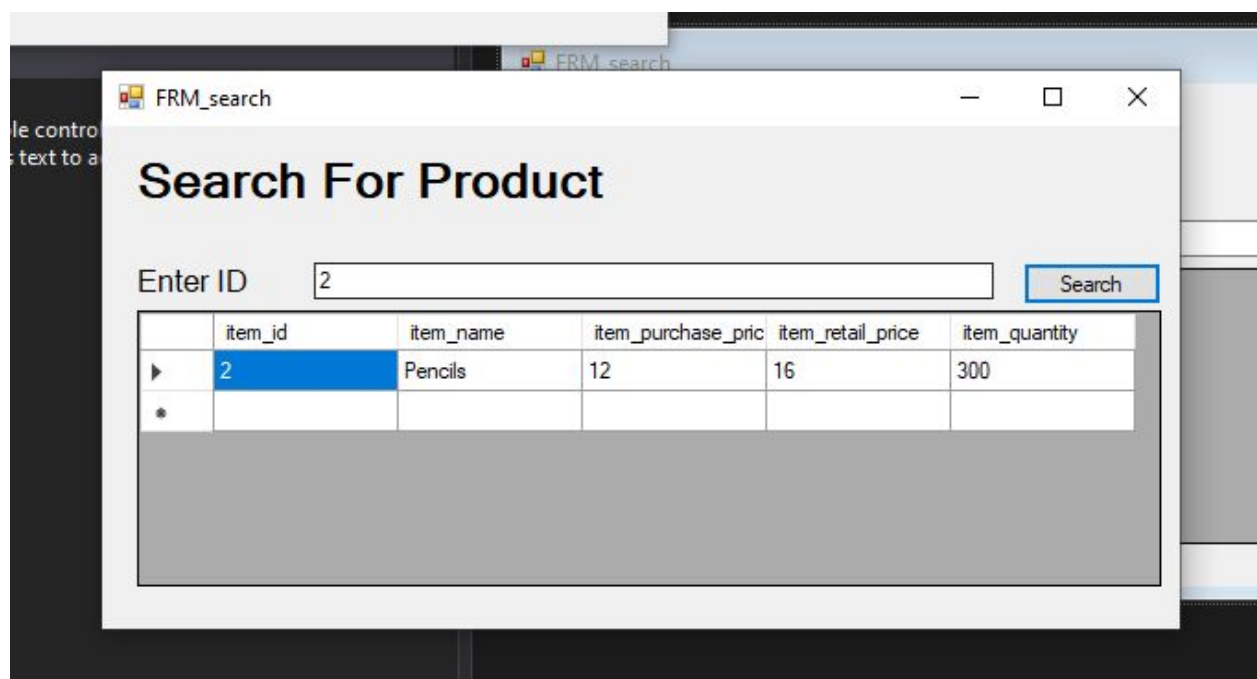
The screenshot shows a Windows form titled "FRM_view" with the main heading "All Data". Below the heading is a table displaying item data. The table has six columns: item_id, item_name, item_purchase_price, item_retail_price, and item_quantity. The first row of data is highlighted in blue.

	item_id	item_name	item_purchase_price	item_retail_price	item_quantity
▶	2	Pencils	12	16	300
*					

Delete



Search



MYSQL Workbench Database

Students

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view of databases including 'final_exam', 'sakila', 'sys', and 'world'. Under 'final_exam', the 'Tables' folder is expanded, showing 'stationary_store' and 'student_table'. The 'student_table' is selected, and its columns are listed: 'student_id' (int PK), 'student_first_name' (varchar(45)), 'student_last_name' (varchar(45)), and 'student_age' (int).

The main query editor shows a query titled 'stationary_store - Table' with the following SQL statement:

```
1 • SELECT * FROM final_exam.student_table WHERE 1;
```

The 'Result Grid' pane at the bottom displays the query results in a table format:

student_id	student_first_name	student_last_name	student_age
1	Umair	Jibran	21
2	Muhammad	Yasin	20
3	Abbas	Nawab	19
NULL	NULL	NULL	NULL

Stationary

The screenshot displays a database management interface. On the left, the 'SCHEMAS' pane shows a tree view with 'final_exam' expanded, containing 'stationary_store'. The 'Columns' list for 'stationary_store' includes: item_id, item_name, item_purchase_price, item_retail_price, and item_quantity. Below the tree, the 'Administration' tab is active, showing the 'Schemas' section. The 'Table: stationary_store' is selected, and its 'Columns' are listed with their data types: item_id (int AI PK), item_name (varchar(45)), item_purchase_price (varchar(45)), item_retail_price (varchar(45)), and item_quantity (varchar(45)).

The main query editor shows a query: `SELECT * FROM final_exam.stationary_store WHERE 1;`. The 'Result Grid' pane displays the query results in a table format:

	item_id	item_name	item_purchase_price	item_retail_price	item_quantity
▶	2	Pencils	12	16	300
*	NULL	NULL	NULL	NULL	NULL