CS-2003L-CS-201-BS-CS-4B

Database Lab Fall 2020

TRAX a Toll Tax Management System

Project Report By

Muhammad Danish (cs182019)

Group Members:

Winona Kristein	cs182032
Aaisha Motan	cs182009
Iqra Anwar	cs182015

Table of Contents

1. Introduction

TRAX a Toll Tax Management Application is a Windows Desktop Application, designed with Windows Forms using C#. Backend of this Application is running with SQL Server Express, which is a RDBMS designed for embedded and smaller scale Applications.

IDE used: Visual Studio 2010 Professional.

RDBMS used: SQL Express Server 2008, Connected with Visual Studio 2010 Professional.

GUI: GUI of this Application is designed using Windows Forms using Visual Studio 2010 Professional.

2. Scope

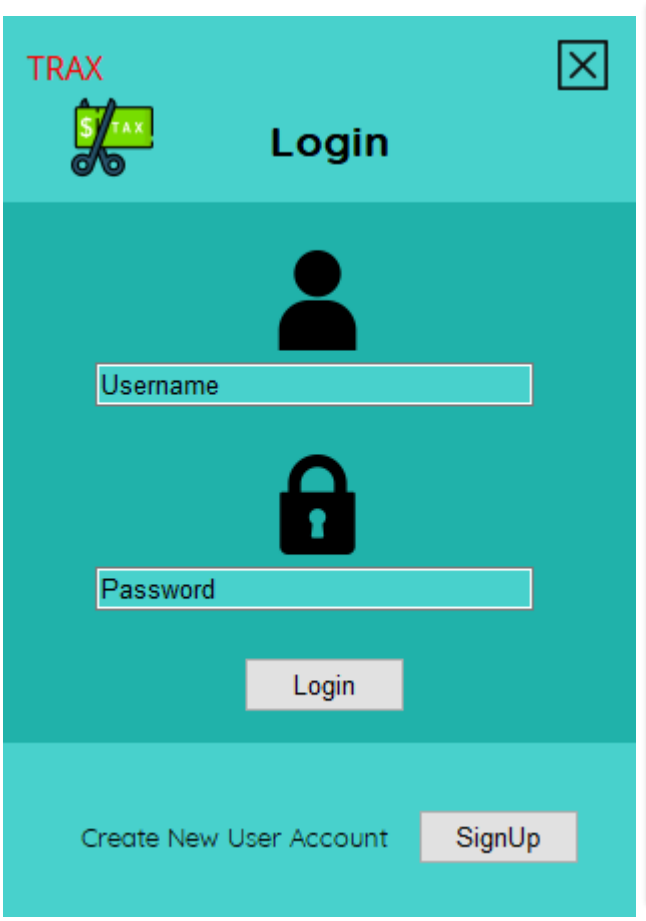
The Application keeps registry of Vehicle and Drivers Data which have passed from Toll Plaza. It can be used on Toll Stations and on private roads.

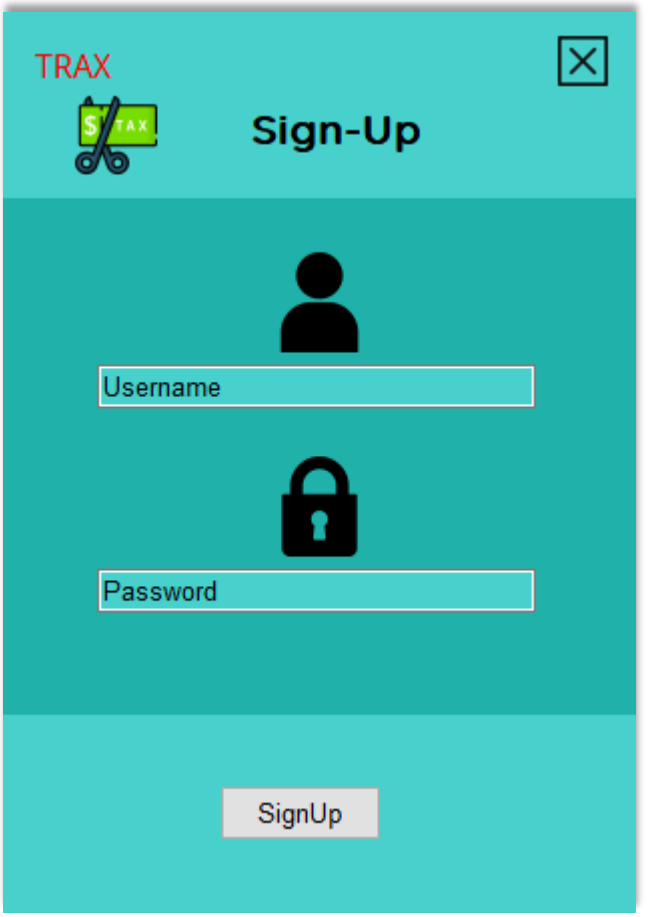
In Future, Counter-Based Point of sales (POS) machines can be setup with this application for printing of bills to the drivers.

3. Module Description

Application Start:

Login Page Appears at the start of the App which checks the user entered credentials. If a user have no Login ID he can Sign-up a Login ID. Only one username is given to a person, the same person can create multiple ID but with different username.





Data Entry and Record:

After Login Page, A main Page opens which is the main data collection point. At this page, Driver's Information and its vehicle Information is collected than it is stored in a database.

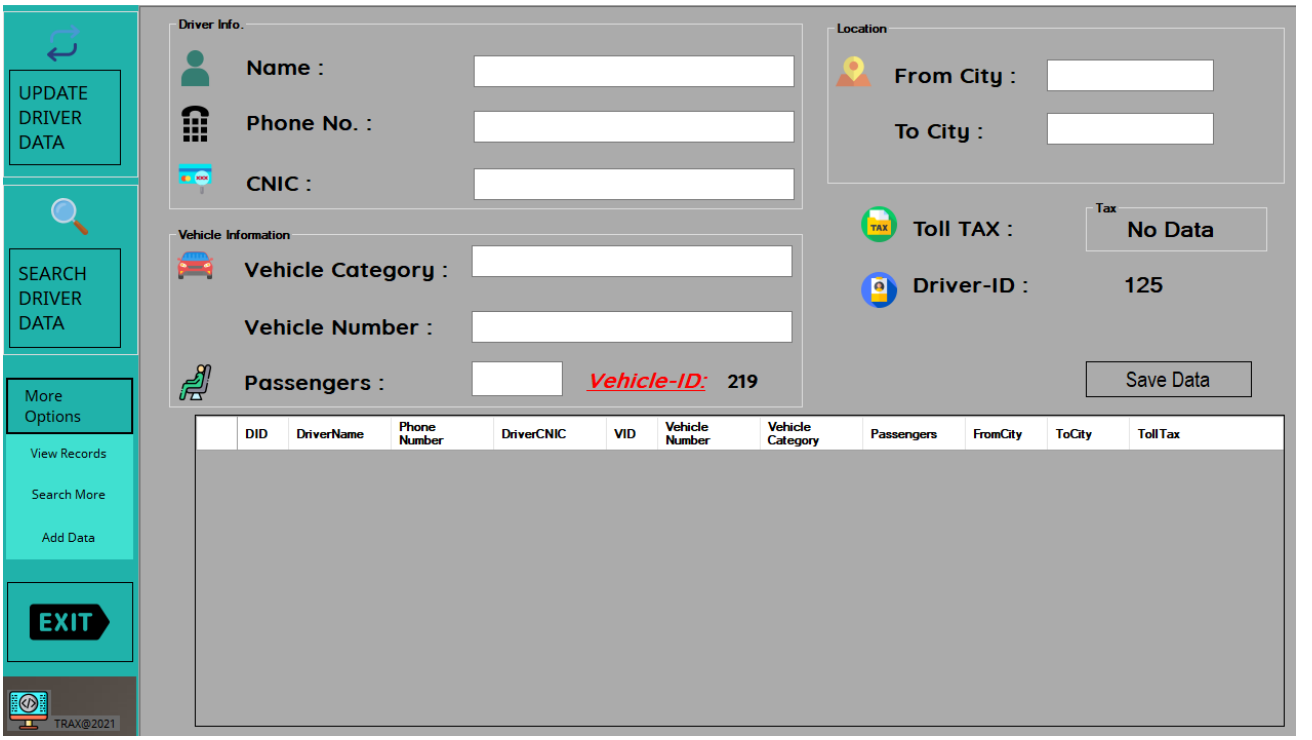
Tax on a Vehicle is according to its category, a driver can only travel to cities which are registered in the database and on main page current(whose data is yet to be inserted) driver and its vehicle ID is always shown.

When a record is inserted it is shown in a table (at the bottom of main page). This table is used to only show the live inserted data. For viewing all the inserted records click on view records (left side of main page).

Data is safely stored in database but after passing required checks and conditions.

- Null record is not inserted.
- Vehicle Category, departure and Arrival City must be Valid.

Main Page



Application Usage and Control

This Main page is also control page and contains features of Application (left side of main page), including updating, searching, viewing of total registry and some additional features.

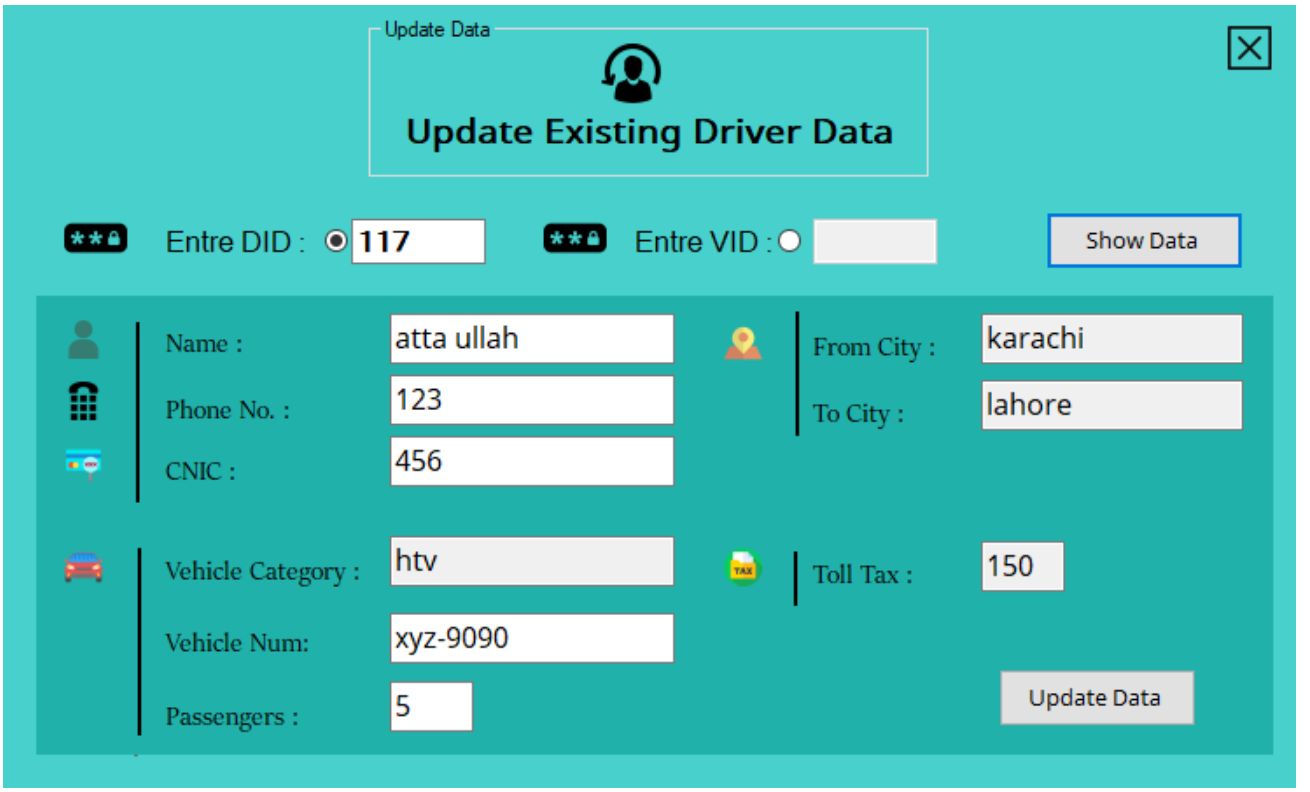
Updating, Search, All record View and Additional Features

- **UPDATING**

When driver data is inserted and collected into database than it can be updated in case of any mistakes. Only name, phone number, CNIC, vehicle number and number of passengers are updatable. For the rest of data like Tax, Vehicle Category, departure city and arrived city are not updatable.

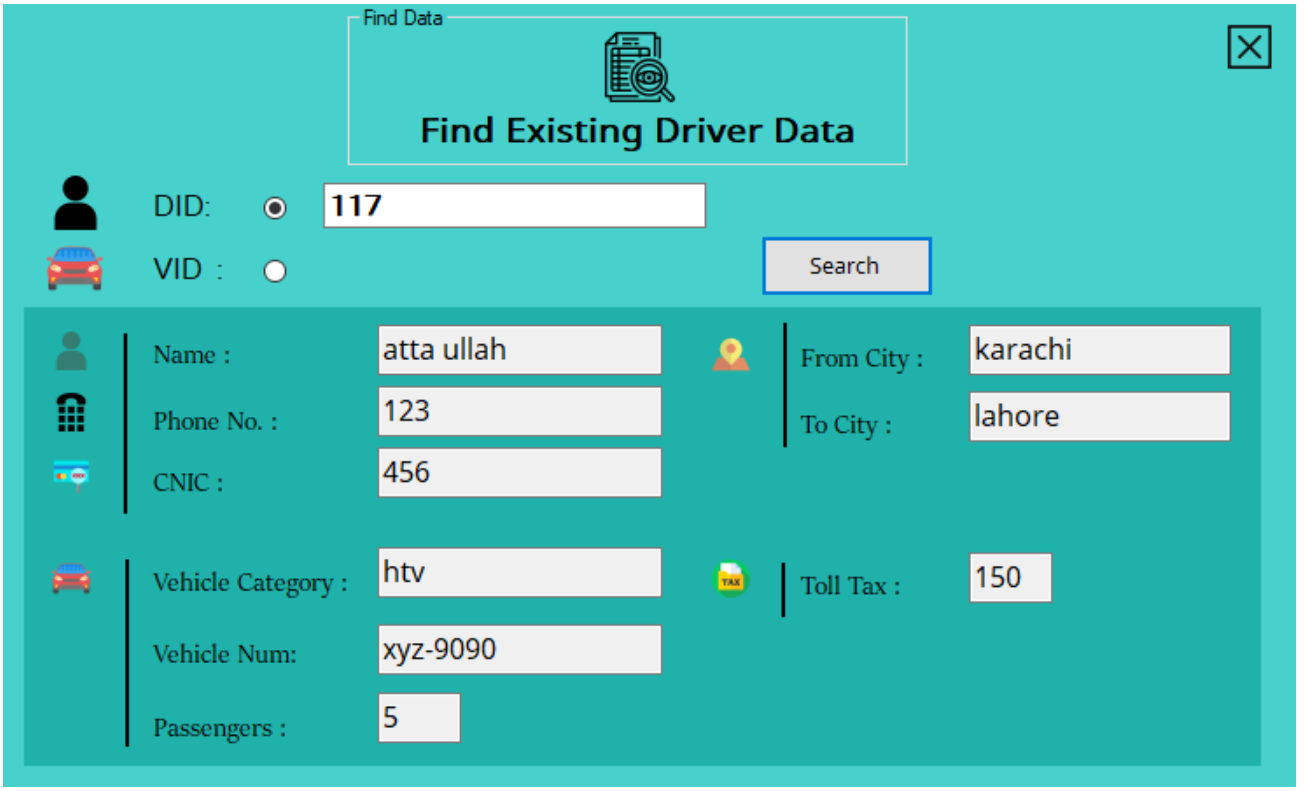
Updating and searching is done using Driver or Vehicle ID's.

On update page data is visible and updatable.



- ## SEARCHING

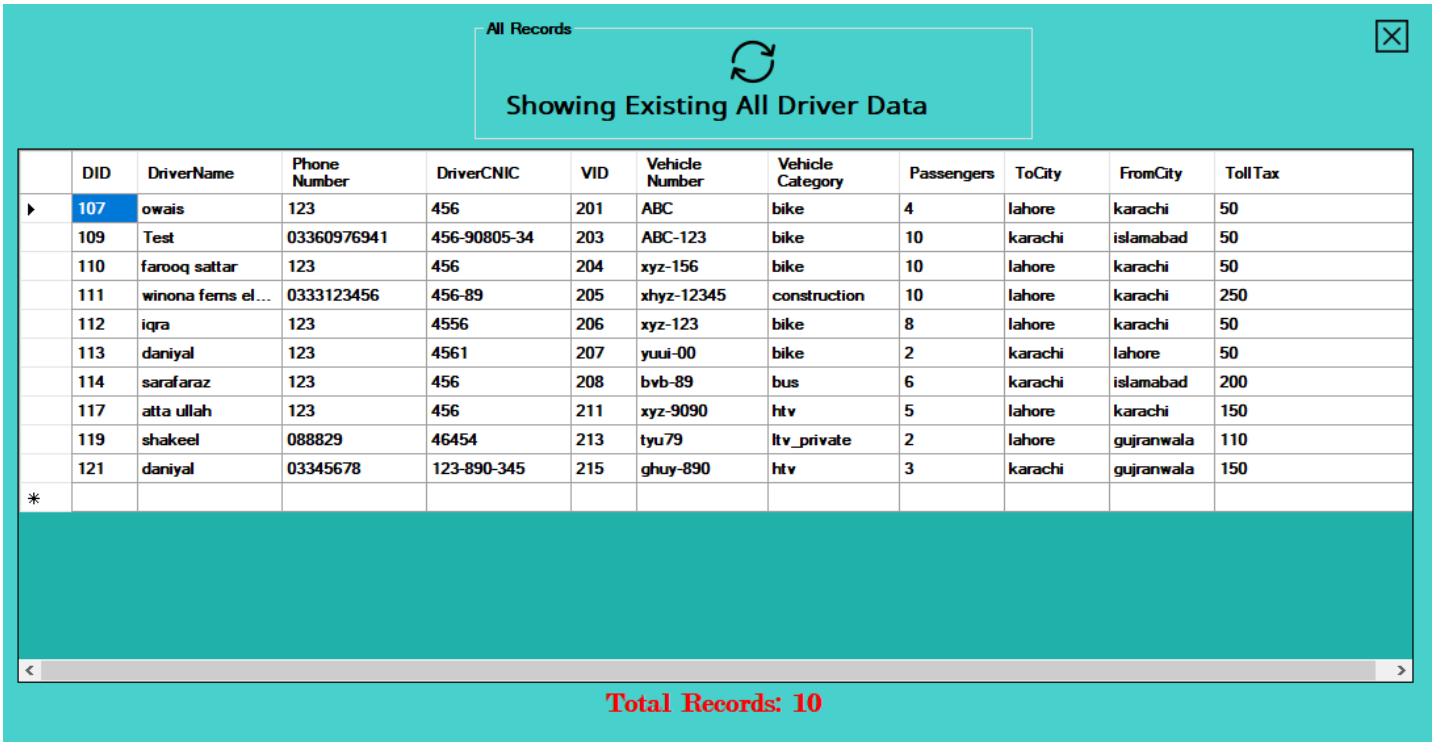
On Search page data is visible but not updatable. Here Text boxes are only used to show data and are in write-only mode.



Continued…..

- ## ALL RECORD VIEW

To view all the records, all the table are joined in a view and this view is used to display data in a table.

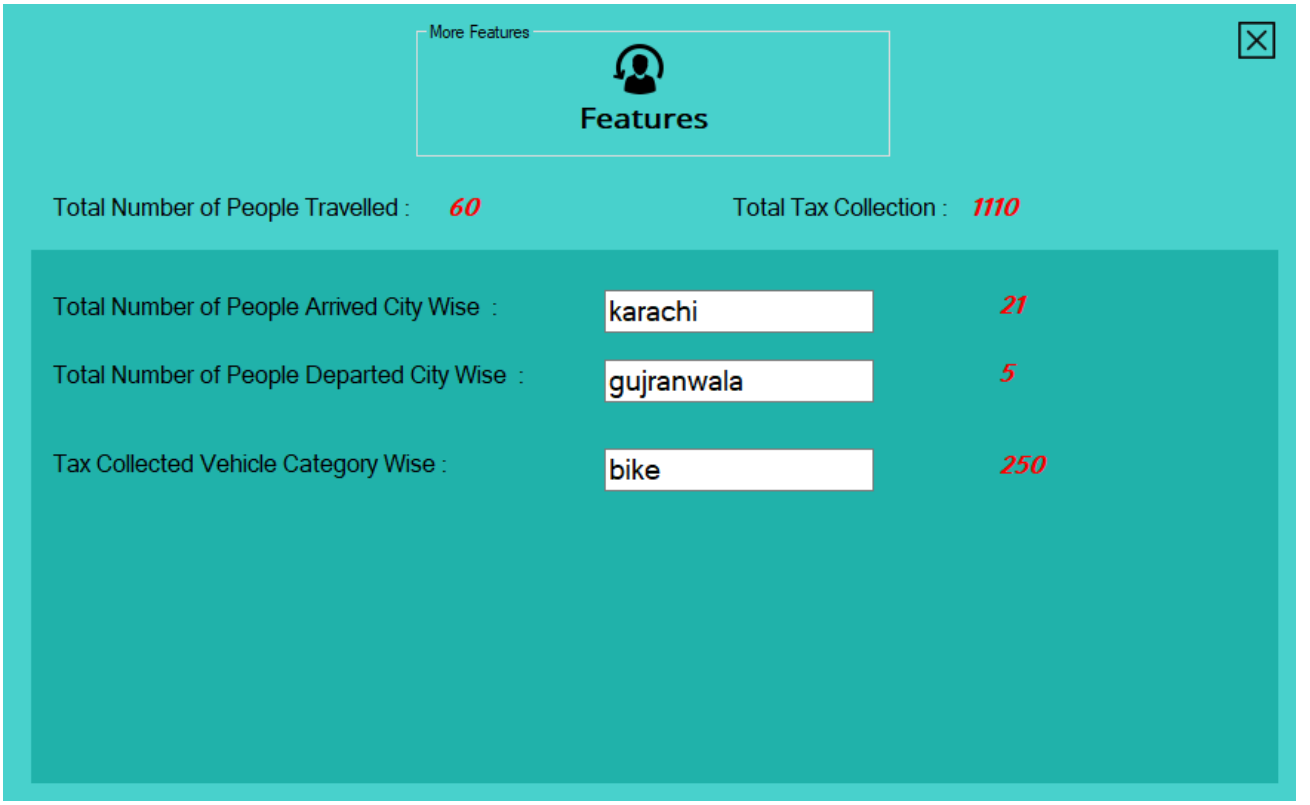


DID	DriverName	Phone Number	DriverCNIC	VID	Vehicle Number	Vehicle Category	Passengers	ToCity	FromCity	TollTax
107	owais	123	456	201	ABC	bike	4	lahore	karachi	50
109	Test	03360976941	456-90805-34	203	ABC-123	bike	10	karachi	islamabad	50
110	farooq sattar	123	456	204	xyz-156	bike	10	lahore	karachi	50
111	winona ferns el...	0333123456	456-89	205	xhyz-12345	construction	10	lahore	karachi	250
112	iqra	123	4556	206	xyz-123	bike	8	lahore	karachi	50
113	daniyal	123	4561	207	yuui-00	bike	2	karachi	lahore	50
114	sarafaraz	123	456	208	bvb-89	bus	6	karachi	islamabad	200
117	atta ullah	123	456	211	xyz-9090	htv	5	lahore	karachi	150
119	shakeel	088829	46454	213	tyu79	ltv_private	2	lahore	gujranwala	110
121	daniyal	03345678	123-890-345	215	ghuy-890	htv	3	karachi	gujranwala	150

Continued…

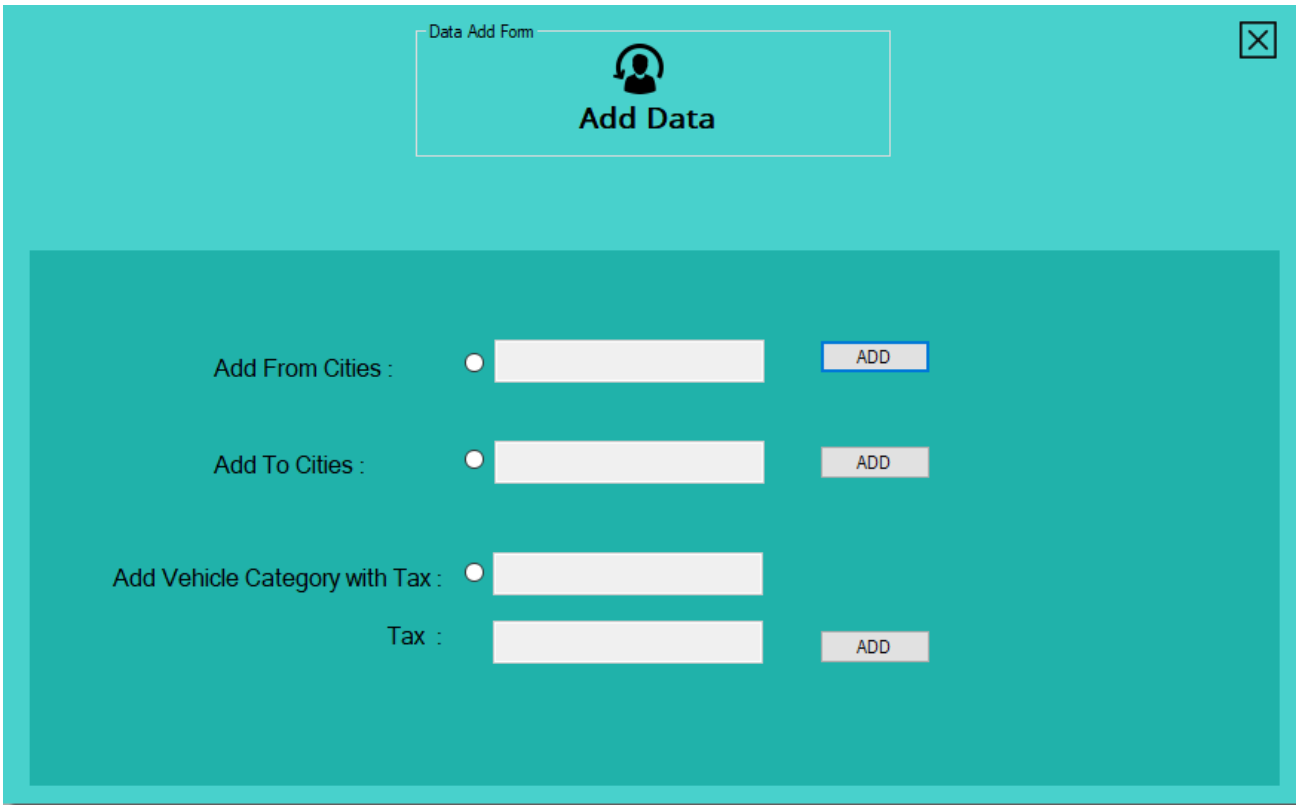
- ## ADDITIONAL FEATURES INCLUDES

- Calculating Total Tax collection and total number of people travelled.
- Calculating Number of people Departed or Arrived in a city.
- Calculating of Tax collection according to a Vehicle Category.



- **ADDING DATA**

Using this form user can enter cities and vehicle category and can fix tax on them.



4. Hardware/Software Requirements

4.1 Tools

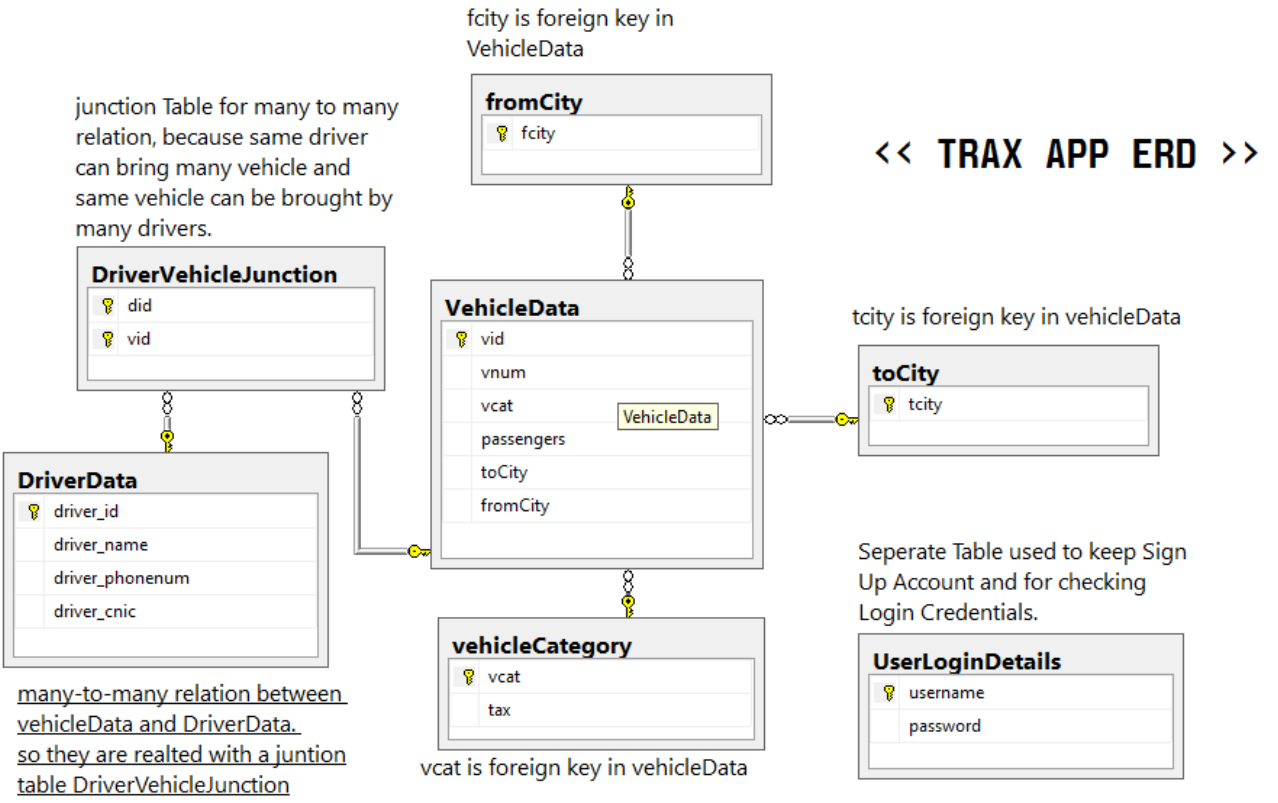
Hardware Requirements: A Windows Machine.

Software Requirements: Windows OS.

4.2 Libraries, IDE

- Visual Studio 2010 Professional.
- SQL Server Express 2008

5. ERD



A view is created using all these tables as "allData"

SELECT dbo.DriverData.driver_id, dbo.DriverData.driver_name, dbo.DriverData.driver_phonenum, dbo.DriverData.driver_cnic, dbo.VehicleData.vid, dbo.VehicleData.vnum, dbo.VehicleData.vcat, dbo.VehicleData.passengers, dbo.VehicleData.toCity, dbo.VehicleData.fromCity, dbo.vehicleCategory.tax

FROM dbo.DriverData INNER JOIN dbo.DriverVehicleJunction ON dbo.DriverData.driver_id = dbo.DriverVehicleJunction.did INNER JOIN dbo.VehicleData ON dbo.DriverVehicleJunction.vid = dbo.VehicleData.vid INNER JOIN dbo.vehicleCategory ON dbo.VehicleData.vcat = dbo.vehicleCategory.vcat

	driver_id	driver_name	driver_phonen...	driver_cnic	vid	vnum	vcat	passengers	toCity	fromCity	tax
▶	107	owais alex	123	456	201	ABC	bike	4	lahore	karachi	50
	109	Test 101	03360976941	456-90805-34	203	ABC-123	bike	10	karachi	islamabad	50
	110	farooq sattar	123	456	204	xyz-156	bike	10	lahore	karachi	50
	111	winona ferns el...	0333123456	456-89	205	xhyz-12345	construction	10	lahore	karachi	250
	112	iqra	123	4556	206	xyz-123	bike	8	lahore	karachi	50

6. Screen Shots (attach screenshot of source code and output)

Code from App Main Page

This function is used to restrict user to only enter vehicle category which is present in database.

```
// vehicle category text box checking
private void vehciletxt_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
{
    // key pressed
    if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Tab)
    {
        // if whats in text box is in databse.
        // we are fetching the data from database and storing it in ArrVehicleCat(a string array collection).
        // than print vehicle category and text in their respective places.

        // the user is restricted to only choose vehicle category which is in database.
        if (ArrVehicleCat.Contains(vehcilecattxt.Text))
        {
            // split whats in text box.
            // create a string array and store the splitted data.
            string[] A = vehcilecattxt.Text.Split('#');
            // print vehicle category and tax in their respective places.
            vehcilecattxt.Text = A[0];
            tolltaxlbl.Text = A[1];
            // focus on next text box.
            vehiclenumtxt.Focus();
        }
        // if user entered(a vehicle category) something else which is not in database than.
        else
        {
            // show error.
            MessageBox.Show("Select Valid Category!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            // empty vehicle text box and tax label.
            vehcilecattxt.Text = string.Empty;
            tolltaxlbl.Text = string.Empty;
            // focus on vehicle category text box.
            vehcilecattxt.Focus();
        }

    }
}
```

When main page loads vehicle category, tax and cities are fetched from database and are displayed for users to select from. Next DriverID and VehicleID is also fetched and is shown on main page.

```
        private void DriverData_Load(object sender, EventArgs e)
        {
            conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
            conTaxAppDB.Open();
            // getting vehicle category.
            daVehicleCat = new SqlDataAdapter("select *from vehicleCategory", conTaxAppDB);
            // getting from cities.
            dafcity = new SqlDataAdapter("select *from fromCity", conTaxAppDB);
            // getting to cities.
            datcity = new SqlDataAdapter("select *from toCity", conTaxAppDB);

            //data table for vehicle cat
            dtVehicleCat = new DataTable();
            // data table for from cities
            dtfcity = new DataTable();
            //data table for to cities
            dttcity = new DataTable();

            // filling all the data adapters into respective data tables
            dafcity.Fill(dtfcity);
            datcity.Fill(dttcity);
            daVehicleCat.Fill(dtVehicleCat);
            // closing connection
            conTaxAppDB.Close();

            // getting next driver id.
            getDriverID();
            // getting next vehicle ID
            getVehicleID();
            // now running all three auto completion fucntions.
            getFromCity();
            getToCity();
            getVehicleCat();
        }
```

Function used to fetch vehicle category and tax from database.

```
        // function for auto completion of text in three boxes
        // arrVehicleCat is used in two fucntions(keyPreviewVehicleCat and getVehicleCat) that's why
 it is global.
        AutoCompleteStringCollection ArrVehicleCat = new AutoCompleteStringCollection();
        private void getVehicleCat()
        {
            string vcat = "";
            string tax = "";
            string concat = "";
            // concatinating and filling and the data from table into auto completion string
colletion(ArrVehicleCat)
            for (int i = 0; i < dtVehicleCat.Rows.Count; i++)
            {
                vcat = dtVehicleCat.Rows[i]["vcat"].ToString();
                tax = dtVehicleCat.Rows[i]["tax"].ToString();
                concat = vcat+"#"+tax ;

                ArrVehicleCat.Add(concat);
            }
            // now creating a custom auto-complete mode on vehicle category text box.
            vehcilecattxt.AutoCompleteCustomSource = ArrVehicleCat;
            vehcilecattxt.AutoCompleteMode = AutoCompleteMode.SuggestAppend;
            vehcilecattxt.AutoCompleteSource = AutoCompleteSource.CustomSource;
        }
```

Function used to fetch cities from database.

```
        // ArrFromCity is used in two fucntions(keyPreviewFromCity and getFromCity) that's why it is
 global.
        AutoCompleteStringCollection ArrFromCity = new AutoCompleteStringCollection();
        private void getFromCity()
        {
            // doing the same thing just not concatinating anything.
            string fromcity = "";
            for (int i = 0; i < dtfcity.Rows.Count; i++)
            {
                fromcity = dtfcity.Rows[i]["fcity"].ToString();
                ArrFromCity.Add(fromcity);
            }

            // creating a custom auto-complete mode on FROM city text box
            FromCitytxt.AutoCompleteCustomSource = ArrFromCity;
            vehcilecattxt.AutoCompleteMode = AutoCompleteMode.SuggestAppend;
            vehcilecattxt.AutoCompleteSource = AutoCompleteSource.CustomSource;
        }

        // ArrToCity is used in two fucntions(keyPreviewToCity and getToCity) that's why it is
global.
        AutoCompleteStringCollection ArrToCity = new AutoCompleteStringCollection();
        private void getToCity()
        {
            // doing the same thing just not concatinating anything.
            string tocity = "";
            for (int i = 0; i < dttcity.Rows.Count; i++)
            {
                tocity = dttcity.Rows[i]["tcity"].ToString();
                ArrToCity.Add(tocity);
            }

            // now creating a custom auto-complete mode on TO city text box
            ToCItytxt.AutoCompleteCustomSource = ArrToCity;
            vehcilecattxt.AutoCompleteMode = AutoCompleteMode.SuggestAppend;
            vehcilecattxt.AutoCompleteSource = AutoCompleteSource.CustomSource;
        }
```

Function used to get next driverID.

```
        // getting next driver id function
        // this function is used at form load and save_data button click
        private void getDriverID()
        {
            conTaxAppDB.Open();

            //getting last updated driver ID value.
            // driver id is primary key and auto_increment .
            // starting from 100 with increment of 1.
            daDriverID = new SqlDataAdapter(@"select IDENT_CURRENT('DriverData') as ID",
conTaxAppDB);

            dtDriverID = new DataTable();
            daDriverID.Fill(dtDriverID);

            conTaxAppDB.Close();

            string driverID = "";
            int convertedDriverID = 0;

            driverID = dtDriverID.Rows[0]["ID"].ToString();
            // converting string to integer
            convertedDriverID = Int32.Parse(driverID);
            // increminting +1 in id
            convertedDriverID++;

            // printing next DriverID in label of GUI
            driveridlbl.Text = convertedDriverID.ToString();
        }
```

Function used to get next vehicleID.

```
        // getting next vehicle id function
        // this function is used at form load and save_data button click
        private void getVehicleID()
        {
            conTaxAppDB.Open();
            daVehicleID = new SqlDataAdapter(@"select IDENT_CURRENT('VehicleData') as ID",
conTaxAppDB);

            dtVehicleID = new DataTable();
            daVehicleID.Fill(dtVehicleID);

            string vehicleID = "";
            int convertedVehicleID = 0;

            vehicleID = dtVehicleID.Rows[0]["ID"].ToString();
            convertedVehicleID = Int32.Parse(vehicleID);
            convertedVehicleID++;

            // printing next DriverID in label of GUI
            vidlbl.Text = convertedVehicleID.ToString();

        }
```

Function of save data button click.

```
        private void savedataBtn_Click(object sender, EventArgs e)
        {
            // if data entered int any box in empty.
            if (string.IsNullOrEmpty(dnametxt.Text.Trim()) || string.IsNullOrEmpty
(dphonetxt.Text.Trim()) || string.IsNullOrEmpty(dcnictxt.Text.Trim()) || string.IsNullOrEmpty
(vehcilecattxt.Text.Trim()) || string.IsNullOrEmpty(vehiclenumtxt.Text.Trim()) ||
string.IsNullOrEmpty(FromCitytxt.Text.Trim()) || string.IsNullOrEmpty(ToCItytxt.Text.Trim()) ||
string.IsNullOrEmpty(passengertxt.Text.Trim()))
            {
                MessageBox.Show("Insert Proper Data !!", "ERORR!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
            else
            {
                try
                {
                    conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initia
 Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                    conTaxAppDB.Open();

                    // parametrized string
                    //insert in driver table.
                    cmdDriverData = new SqlCommand("INSERT INTO DriverData
(driver_name,driver_phonenum,driver_cnic) values(@driverName,@driverPhoneNum,@driverCNIC)",
conTaxAppDB);
                    cmdDriverData.Parameters.AddWithValue("@driverName", dnametxt.Text);
```

```
                    cmdDriverData.Parameters.AddWithValue("@driverName", dnametxt.Text);
                    cmdDriverData.Parameters.AddWithValue("@driverPhoneNum", dphonetxt.Text);
                    cmdDriverData.Parameters.AddWithValue("@driverCNIC", dcnictxt.Text);
                    cmdDriverData.ExecuteNonQuery();

                    //insert in vehicle table.
                    cmdVehicleData = new SqlCommand("INSERT INTO VehicleData
(vnum,vcat,passengers,toCity,fromCity) values(@vNUM,@vCat,@passengers,@toCity,@fromCity)",
conTaxAppDB);
                    cmdVehicleData.Parameters.AddWithValue("@vNUM", vehiclenumtxt.Text);
                    cmdVehicleData.Parameters.AddWithValue("@vCat", vehcilecattxt.Text);
                    cmdVehicleData.Parameters.AddWithValue("@passengers", passengertxt.Text);
                    cmdVehicleData.Parameters.AddWithValue("@toCity", ToCItytxt.Text);
                    cmdVehicleData.Parameters.AddWithValue("@fromCity", FromCitytxt.Text);
                    cmdVehicleData.ExecuteNonQuery();

                    // get current Driver and Vehicle id.
                    daDriverData = new SqlDataAdapter(@"select IDENT_CURRENT('DriverData') as
driverID", conTaxAppDB);
                    daVehicleData = new SqlDataAdapter(@"select IDENT_CURRENT('VehicleData') as
vehicleID", conTaxAppDB);

                    dtDriverData = new DataTable();
                    dtVehicleData = new DataTable();

                    // sql command line pe jesa output ata hai.
                    // wesa yaha se sochan shuru karo.
                    daDriverData.Fill(dtDriverData);
                    daVehicleData.Fill(dtVehicleData);

                    string did = "";
                    string vid = "";

                    did = dtDriverData.Rows[0]["driverID"].ToString();

                    vid = dtVehicleData.Rows[0]["vehicleID"].ToString();
```

```
                    vid = dtVehicleData.Rows[0]["vehicleID"].ToString();

                    // parsing both the id into integer
                    int didINT = Int32.Parse(did);
                    int vidINT = Int32.Parse(vid);

                    // inserting these id's into junction table.
                    // in junction table did and vid and composite primary key.
                    cmdJunction = new SqlCommand("Insert into DriverVehicleJunction(did,vid) values
(@did,@vid)", conTaxAppDB);
                    cmdJunction.Parameters.AddWithValue("@did", didINT);
                    cmdJunction.Parameters.AddWithValue("@vid", vidINT);
                    cmdJunction.ExecuteNonQuery();

                    conTaxAppDB.Close();

                    // woh jo main page me grid pe data dikh raha hai insert hote we bus yeh woh hai
 ek line.
                    DriverDataGrid.Rows.Add(driveridlbl.Text, dnametxt.Text, dphonetxt.Text,
dcnictxt.Text,vidlbl.Text ,vehiclenumtxt.Text,vehcilecattxt.Text, passengertxt.Text,
ToCItytxt.Text,FromCitytxt.Text, tolltaxlbl.Text);

                    // data insert hogaya and ab data dikh bhi raha hai grid me.
                    // ab sare text boxes khali kardo.
                    dnametxt.Text = string.Empty;
                    dphonetxt.Text = string.Empty;
                    dcnictxt.Text = string.Empty;
                    vehcilecattxt.Text = string.Empty;
                    vehiclenumtxt.Text = string.Empty;
                    passengertxt.Text = string.Empty;
                    FromCitytxt.Text = string.Empty;
                    ToCItytxt.Text = string.Empty;

                    tolltaxlbl.Text = "No Data";

                    //
```

```
                tolltaxlbl.Text = "No Data";

                // next driver id nikalo
                getDriverID();
                getVehicleID();

                // and driver name wala text box pe phir se focus kardo.
                dnametxt.Focus();

            }
            // database kay koi bhi error ko handle karne kay lie ek catch exception.
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
```

Code from login page:

```
        // checking username and password both
        SqlConnection con;
        SqlDataAdapter da;
        DataTable dt;
        // login button click
        private void loginBtn_Click(object sender, EventArgs e)
        {
            // SQL WORK : checking username and password both
            try
            {
                // making and opening a connection
                con = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                con.Open();
                // fetching username and pass from database.
                da = new SqlDataAdapter("select *from UserLoginDetails where username = @uname  and
password = @pass",con);
                da.SelectCommand.Parameters.AddWithValue("@uname",usertxt.Text.Trim());
                da.SelectCommand.Parameters.AddWithValue("@pass", passtxt.Text.Trim());
                // filling the fetched data in a table.
                dt = new DataTable();
                da.Fill(dt);
                con.Close();

                // username is primary key in table.
                // if the login details are in database than there must be a one record(one row) in
table.
                // if not than there is no row in table.

                if (dt.Rows.Count == 1)
                {
                    //close login page
                    this.Close();
                    // open main page
```

```
                    // open main page
                    th = new Thread(opendriverdata);
                    th.SetApartmentState(ApartmentState.STA);
                    th.Start();
                    th.Join();
                }
                else
                {
                    MessageBox.Show("Username or Password is incorrect.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    // close login page
                    this.Close();
                    // open login form again
                    th = new Thread(openloginform);
                    th.SetApartmentState(ApartmentState.STA);
                    th.Start();
                }
            }
            // a catch Exception for handling any return error from database.
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message,"Error",MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
```

Signup page:

```
        // sql command for insertion
        SqlCommand cmd;
        SqlConnection con;
        // sign up button CLICK code
        private void signupBtn_Click(object sender, EventArgs e)
        {
            // handling any default entry or any null entry.
            // btw database will also dont let enter any null record.
            // because username is primary key.
            // and password is set to not null.
            if (usertxt.Text.Trim() == "Username" || passtxt.Text.Trim() == "Password" ||
string.IsNullOrEmpty(usertxt.Text.Trim()) || string.IsNullOrEmpty(passtxt.Text.Trim()))
            {
                DialogResult DR_defaultError = MessageBox.Show("Validation Erorr !\n\n1) *Fill all
Fields.\n\n2) *Cannot have default Username and Password.", "ERROR!", MessageBoxButtons.RetryCancel,
 MessageBoxIcon.Error);
                if (DR_defaultError == DialogResult.Retry)
                {
                    // close sign up page
                    this.Close();
                    //th.Abort();
                    // opening again signup page
                    th = new Thread(againOpenSignup);
                    th.SetApartmentState(ApartmentState.STA);
                    th.Start();
                    th.Join();
                }
                else if (DR_defaultError == DialogResult.Cancel)
                {
                    Application.Exit();
                }
            }

            else
```

```
            else
            {
                try
                {
                    // creating and opening a sql connection to database.
                    con = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                    con.Open();
                    // inserting sign up details
                    cmd = new SqlCommand("INSERT INTO UserLoginDetails(username,password) values
(@userName,@password)", con);
                    cmd.Parameters.AddWithValue("@userName", usertxt.Text);
                    cmd.Parameters.AddWithValue("@password", passtxt.Text);
                    cmd.ExecuteNonQuery();
                    //closing connection
                    con.Close();

                    MessageBox.Show("New Account is Created !\nNow you Can
Login.","Success",MessageBoxButtons.OK,MessageBoxIcon.Information);
                    this.Close();
                    th.Abort();
                    th = new Thread(redirectLoginForm);
                    th.SetApartmentState(ApartmentState.STA);
                    th.Start();

                }
                // catch for handling any return error from database.
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }
```

Showing all the records code:

```
        // all work of this form is in the form load event
        // only showing all the data in the database.
        // data is fetched and shown from a view.
        private void AllRecordForm_Load(object sender, EventArgs e)
        {
            AllDriverDataGrid.Rows.Clear();
            conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
            conTaxAppDB.Open();
            daAllDataView = new SqlDataAdapter(@"select * from allData",conTaxAppDB);
            dtAllDataView=new DataTable();
            daAllDataView.Fill(dtAllDataView);

            conTaxAppDB.Close();

            int Count = dtAllDataView.Rows.Count;

            string[] array= new string[11];
            string str;
            for (int i = 0; i < dtAllDataView.Rows.Count; i++) // row
            {
                // is loop kay baad ek array me sara data agaya hai. ek row ka bus.
                for (int j = 0; j < dtAllDataView.Columns.Count; j++) //column
                {
                    str = dtAllDataView.Rows[i][j].ToString();
                    array[j] = str;
                }

                AllDriverDataGrid.Rows.Add(array[0], array[1], array[2], array[3], array[4], array
[5], array[6], array[7], array[8], array[9], array[10]);
            }

                lblCount.Text = "Total Records: " + Count;
        }
    }
}
```

Update data form code:

```
        string driverid;
        string vehicleid;
        private void showdatabtn_Click(object sender, EventArgs e)
        {
            if (didtxt.Enabled == false && vidtxt.Enabled == false)
            {
                MessageBox.Show("Please select a method to
Update !","Error!",MessageBoxButtons.OK,MessageBoxIcon.Error);
            }
            else if (didradiobtn.Checked)
            {
                // checked  hai to yani visible hai.
                // agar visible hai and text box empty hai ya phir spaces dale we hai bus to..
                if (didtxt.Enabled == true && string.IsNullOrEmpty(didtxt.Text.Trim()))
                {
                    MessageBox.Show("Please Insert DID for Search.", "Error!", MessageBoxButtons.OK,
 MessageBoxIcon.Error);
                    didtxt.Text = string.Empty;
                    vidtxt.Text = string.Empty;
                }
                // agar esa nahi hai to database me entered values to check karo.
                else
                {
                    conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
 Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                    conTaxAppDB.Open();
                    daViewAllData = new SqlDataAdapter(@"select *from allData where driver_id =
@did", conTaxAppDB);
                    //string me dala text box wali cheez ko
                    string dids = didtxt.Text.Trim();
                    // phir int me convert kiya kyunke databse me id int hai.
                    int did = Int32.Parse(dids);
```

```
                    int did = Int32.Parse(dids);

                    // yeh wali cheez kar rahe hai similar kaam horaha bus data Adapter par.
                    // cmd wala sql command hai. jo insert kay lie use hoti hai
                    //cmd.Parameters.AddWithValue("@did", did);

                    daViewAllData.SelectCommand.Parameters.AddWithValue("@did", did);
                    conTaxAppDB.Close();

                    // ab data agaya hai. ab baat yeh hai kay har waqt ek row aegi ya koi row nahi
aegi.
                    dtViewAllData = new DataTable();
                    daViewAllData.Fill(dtViewAllData);
                    // agar ek bhi row nahi ayi hai to..
                    if (dtViewAllData.Rows.Count == 0)
                    {
                        nametxt.Visible = false;
                        namelbl.Visible = false;
                        phonenotxt.Visible = false;
                        phonelbl.Visible = false;
                        cnictxt.Visible = false;
                        cniclbl.Visible = false;
                        vehiclecategorytxt.Visible = false;
                        vehiclecategorylbl.Visible = false;
                        vehiclenumtxt.Visible = false;
                        vehiclenumlbl.Visible = false;
                        passengerstxt.Visible = false;
                        passengerlbl.Visible = false;
                        fromcitytxt.Visible = false;
                        fromcitylbl.Visible = false;
                        tocitytxt.Visible = false;
                        tocitylbl.Visible = false;
                        tolltaxtxt.Visible = false;
```

```
                    updatebtn.Visible = false;
                    MessageBox.Show("Invalid DID.", "Error!", MessageBoxButtons.OK,
MessageBoxIcon.Error);

                }

                // nahi to ek to row ayi hi hogi.
                else
                {
                    nametxt.Visible = true;
                    namelbl.Visible = true;
                    phonenotxt.Visible = true;
                    phonelbl.Visible = true;
                    cnictxt.Visible = true;
                    cniclbl.Visible = true;
                    vehiclecategorytxt.Visible = true;
                    vehiclecategorylbl.Visible = true;
                    vehiclenumtxt.Visible = true;
                    vehiclenumlbl.Visible = true;
                    passengerstxt.Visible = true;
                    passengerlbl.Visible = true;
                    fromcitytxt.Visible = true;
                    fromcitylbl.Visible = true;
                    tocitytxt.Visible = true;
                    tocitylbl.Visible = true;
                    tolltaxtxt.Visible = true;
                    tolltaxlbl.Visible = true;

                    updatebtn.Visible = true;

                    // ek hi row hai to uske sare columns kay data ko respective txt boxes me
daal denge.
                    // yaha pe loop ki zaroorat nahi hai wese.
```

```
                    updatebtn.Visible = true;

                    // ek hi row hai to uske sare columns kay data ko respective txt boxes me
daal denge.
                    // yaha pe loop ki zaroorat nahi hai wese.
                    for (int i = 0; i < dtViewAllData.Rows.Count; i++)
                    {
                        driverid = dtViewAllData.Rows[i]["driver_id"].ToString();
                        nametxt.Text = dtViewAllData.Rows[i]["driver_name"].ToString();
                        phonenotxt.Text = dtViewAllData.Rows[i]["driver_phonenum"].ToString();
                        cnictxt.Text = dtViewAllData.Rows[i]["driver_cnic"].ToString();

                        vehicleid = dtViewAllData.Rows[i]["vid"].ToString();
                        vehiclenumtxt.Text = dtViewAllData.Rows[i]["vnum"].ToString();
                        vehiclecategorytxt.Text = dtViewAllData.Rows[i]["vcat"].ToString();
                        passengerstxt.Text = dtViewAllData.Rows[i]["passengers"].ToString();
                        tocitytxt.Text = dtViewAllData.Rows[i]["toCity"].ToString();
                        fromcitytxt.Text = dtViewAllData.Rows[i]["fromCity"].ToString();
                        tolltaxtxt.Text = dtViewAllData.Rows[i]["tax"].ToString();
                    }
                }
            }
        }
```

Update data function

```
        SqlCommand cmd;
        private void updatebtn_Click(object sender, EventArgs e)
        {
            // agar jo dabbe update karne kay lie hai agar woh khali hai to...
            if (string.IsNullOrEmpty(nametxt.Text.Trim()) || string.IsNullOrEmpty
(phonenotxt.Text.Trim()) || string.IsNullOrEmpty(cnictxt.Text.Trim()) || string.IsNullOrEmpty
(vehiclenumtxt.Text.Trim()) || string.IsNullOrEmpty(passengerstxt.Text.Trim()))
            {
                MessageBox.Show("Please Fill All Boxes!.", "Error!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

            // agar khali nahi hai to...
            else
            {
                try
                {
                    conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
 Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                    conTaxAppDB.Open();
                    cmd = new SqlCommand(@"UPDATE DriverData SET driver_name = @name,driver_phonenum
 = @phonenum,driver_cnic = @cnic where driver_id = @did", conTaxAppDB);
                    // driver_name ="danish"
                    cmd.Parameters.AddWithValue("@name", nametxt.Text);
                    cmd.Parameters.AddWithValue("@phonenum", phonenotxt.Text);
                    cmd.Parameters.AddWithValue("@cnic", cnictxt.Text);
                    cmd.Parameters.AddWithValue("@did", driverid);
                    cmd.ExecuteNonQuery();

                    cmd = new SqlCommand(@"UPDATE VehicleData SET
vnum=@vnum,vcat=@vcat,passengers=@passengers,toCity=@tcity,fromCity=@fcity where vid = @vid",
conTaxAppDB);
```

```
conTaxAppDB);
                    cmd.Parameters.AddWithValue("@vnum", vehiclenumtxt.Text);
                    cmd.Parameters.AddWithValue("@vcat",vehiclecategorytxt.Text);
                    cmd.Parameters.AddWithValue("@passengers",passengerstxt.Text);
                    cmd.Parameters.AddWithValue("@tcity",tocitytxt.Text);
                    cmd.Parameters.AddWithValue("@fcity",fromcitytxt.Text);
                    cmd.Parameters.AddWithValue("@vid", vehicleid);
                    conTaxAppDB.Close();


                    nametxt.Text = string.Empty;
                    phonenotxt.Text = string.Empty;
                    cnictxt.Text = string.Empty;
                    vehiclecategorytxt.Text = string.Empty;
                    vehiclenumtxt.Text = string.Empty;
                    passengerstxt.Text = string.Empty;
                    fromcitytxt.Text = string.Empty;
                    tocitytxt.Text = string.Empty;
                    tolltaxtxt.Text = string.Empty;

                    updatebtn.Visible = false;

                    didtxt.Text = string.Empty;
                    vidtxt.Text = string.Empty;

                    MessageBox.Show("Data is Safely Updated !", "Success!", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
```

SqlCommand Update.cmd

Search more page code:

Function used to get total travelling people and total tax collection.

```
        //calculating total passengers and total tax collection method. used on load event.
        private void getPassengers_Tax()
        {
            conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
            conTaxAppDB.Open();
            daAllData = new SqlDataAdapter(@"select sum(passengers) as TotalTravel , sum(tax) as
TotalTax from allData", conTaxAppDB);
            conTaxAppDB.Close();

            dtAllData = new DataTable();
            daAllData.Fill(dtAllData);

            totalTravellbl.Text = dtAllData.Rows[0]["TotalTravel"].ToString();
            totaltaxlbl.Text = dtAllData.Rows[0]["TotalTax"].ToString();
        }
```

Function used to calculate arrived people in a city.

```
        // checking if selected city is in database.
        // if in database than calculating the total passengers arrived into that city.
        // using sub query.
        private void arrivedtxt_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
        {
            if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Tab)
            {
                if (ArrToCity.Contains(arrivedtxt.Text))
                {
                    conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
 Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                    conTaxAppDB.Open();
                    daAllData = new SqlDataAdapter(@"select sum(passengers) as TotalArrived from
allData where toCity in (select toCity from allData where toCity=@tcity)", conTaxAppDB);
                    daAllData.SelectCommand.Parameters.AddWithValue("@tcity", arrivedtxt.Text.Trim
());
                    conTaxAppDB.Close();
                    dtAllData = new DataTable();
                    daAllData.Fill(dtAllData);
                    arrivedlbl.Text = dtAllData.Rows[0]["TotalArrived"].ToString();
                }
                else
                {
                    MessageBox.Show("Select Valid City!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    arrivedtxt.Text = string.Empty;
                    arrivedlbl.Text = "No Data";
                }
            }
        }
```

Function used to get total tax collection according to a vehicle category.

```
        private void vcattaxttxt_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
        {
            if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Tab)
            {
                if (ArrVehicleCat.Contains(vcattaxttxt.Text.Trim()))
                {
                    conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
 Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                    conTaxAppDB.Open();
                    davcattax = new SqlDataAdapter(@"select sum(tax) as TotalTax from allData where
vcat in (select vcat from allData where vcat=@vcat)", conTaxAppDB);
                    davcattax.SelectCommand.Parameters.AddWithValue("@vcat", vcattaxttxt.Text.Trim
());
                    conTaxAppDB.Close();
                    dtvcattax = new DataTable();
                    davcattax.Fill(dtvcattax);

                    string totalTax = dtvcattax.Rows[0]["TotalTax"].ToString();

                    if (string.IsNullOrEmpty(totalTax))
                    {
                        MessageBox.Show("No Tax record of this vehicle Category!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                        vcattaxlbl.Text = "0";
                        vcattaxttxt.Focus();

                    }
                    else
                    {
                        vcattaxlbl.Text = totalTax;
                    }
                }
                else
                {
                    MessageBox.Show("Select Valid Vehicle Category!", "Error", MessageBoxButtons.OK,
 MessageBoxIcon.Error);
                    vcattaxttxt.Text = string.Empty;
                    vcattaxlbl.Text = "No Data";
                }
            }
```

Add data page code:

```
        // insertion of data
        SqlConnection conTaxAppDB;
        SqlCommand cmd;

        private void addfcitybtn_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(fromCitytxt.Text.Trim()))
            {
                fromCitytxt.Text=string.Empty;
                MessageBox.Show("Please Insert a From City !", "ERROR!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
            else
            {
                try
                {
                    conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
 Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                    conTaxAppDB.Open();
                    cmd = new SqlCommand(@"Insert into fromCity(fcity) values(@fcity)",
conTaxAppDB);
                    cmd.Parameters.AddWithValue("@fcity", fromCitytxt.Text.Trim());
                    cmd.ExecuteNonQuery();
                    conTaxAppDB.Close();

                    fromCitytxt.Text = string.Empty;
                    MessageBox.Show("Data is Safely Saved.");
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }

        private void addtcitybtn_Click(object sender, EventArgs e)...
```

```
        private void addvcattaxbtn_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(vcatxt.Text.Trim()) || string.IsNullOrEmpty(taxtxt.Text.Trim
()))
            {
                vcatxt.Text = string.Empty;
                taxtxt.Text = string.Empty;
                MessageBox.Show("Please Insert Vehicle Category & Tax !", "ERROR!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            try
            {

                conTaxAppDB = new SqlConnection(@"Data Source=AlexanderAnjelo\sqlexpress;Initial
Catalog=TaxAppDB;Integrated Security=True;Pooling=False");
                conTaxAppDB.Open();
                cmd = new SqlCommand(@"Insert into vehicleCategory(vcat,tax) values(@vcat,@tax)",
conTaxAppDB);
                cmd.Parameters.AddWithValue("@vcat", vcatxt.Text.Trim());
                cmd.Parameters.AddWithValue("@tax", taxtxt.Text.Trim());
                cmd.ExecuteNonQuery();
                conTaxAppDB.Close();

                vcatxt.Text = string.Empty;
                taxtxt.Text = string.Empty;
                MessageBox.Show("Data is Safely Saved.");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
```

Task Sheet:

ERD Design, GUI design, Main page work by.	**Muhammad Danish (cs182019)**
Update and search work by	**Winona Kristein (cs182032)**
View all records, additional features work by	**Aaisha Motan (cs182009)**
Add data, additional features work by	**Iqra Anwar (cs182015)**