PUSL 2020

QUALITY ASSURANCE REPORT

SOFTWARE DEVELOPMENT TOOLS & PRACTICES

(GROUP 54) TEAM LEADER- DEWMITH KIRIELLA

5-19-2022

PUSL 2020

SOFTWARE DEVELOPMENT TOOL AND PRACTICES

QUALITY ASSURANCE TESTING REPORT

PREPARED BY

DT KIRIELLA, EAYI EDIRISINGHE, MYM YUSRY,

MPV BANDARANAYAKE, SMA DHARMASENA, PHN KAVINDYA

SUPERVISED BY

PAVITHRA KANKANAMGE

TABLE OF CONTENTS

CHAPTER	TOPIC	PAGE
1	**DEVELOPMENT BACKGROUND**	3
2	**ELABORATING THE LOGIC BEHIND THE CODE**	
2.1	MAIN ASSETS AVALIABLE IN THE SYSTEM	4
2.2	EXPLAINING THE LOGIC BEHIND EACH AND EVERY METHOD () USED	5
3	**DESCRIBING THE TEST CASES**	
3.1	TEST CASE ID'S AND PURPOSE	25
3.2	TESTING USING DUMMY VALUES	26
4	**TEST CASE REPORT**	36
5	**INDIVIGUAL CONSTRIBUTION**	38

CHAPTER 1

DEVELOPMENT BACKGROUND

TASK

To develop an online form submission website letting people with a driving license to report any accident occurred during the day. So that users who have access to view this information namely 'The Police', 'Road Development Authority' and the 'Insurance Company Agents' can analyze these data and conclude whether to approve or reject the relevant cause in order to let the driver claim from the Insurance company to compensate with the following accidents and also Road Development Authority can obtain a clearer understanding the damages caused to public infrastructure that needs to be repaired, while the police will be able to investigate on who's liable for the specific incident with the available resources.

REQUIRED FUNCTIONS

- Account creation for the all the 4 types of users representing the General Public, The Police, Road Development Authority and Insurance Company
- Letting the drivers submit forms with the required details and facilitating with options of delete and update to alter any incorrect information and upload images
- A responsive UI such as (List view, click to reveal etc.) for the Police, RDA and Insurance users to thoroughly analyze the submitted information
- Graph Illustrations for better understanding about the statistics of the outcomes
- Letting users view the results regarding to the relevant incident

AVAILABLE FUNCTIONS IN THE DEVELOPED SYSTEM

- ✓ Account Registration /Login Logout for all the different type users
- ✓ Form submission and with option of uploading images
- ✓ Update option to rectify the erroneous data submitted to the system
- ✓ Results showing for the users enrolled with the system
- ✓ A pie chart representing the number of passengers involved in a certain area and a list view option to select the passenger count to reveal the collision date and exact location

CHAPTER 2

ELABORATING THE LOGIC BEHIND THE CODE

2.1 MAIN ASSETS AVALIABLE IN THE SYSTEM

FOR GENERAL PUBLIC DRIVERS (WEB FORM NAME, TASK)

- **LoginPage.aspx**- Driver Registration page
- **Loginother.aspx**- Separate Login for the drivers
- **PUBLIC.aspx**- To submit details regarding the accident
- **ProfilePublic.aspx**- To update information in the system
- **RESULTS.aspx**- To view the case results after analysis

FOR THE POLICE (WEB FORM NAME, TASK)

- **Policereg.aspx**- Registration page for the police
- **PoliceandRda.aspx**- Common page for the police to view information (common page)
- **Loginother.aspx**- Separate Login for the police
- **Statistics.aspx**-To view the images relevant to the accidents (common page)
- **Analysis.aspx-** The details of the users and accident information (common page)
- **Charts.aspx-** Graph representation of certain details (common page)
- **AccountPol.aspx-** To update any mistaken details in the system
- **select.aspx**- To obtain more information about the incident (common page)

FOR THE INSURANCE COMPANY MEMBERS (WEB FORM NAME, TASK)

- **Insurancereg.aspx-** Registration page for the insurance company members
- **LoginInsurance.aspx-**Login page for the members
- **PoliceandRda.aspx**- View information from the system (common page)
- **Analysis.aspx-** Details of the users and accident information (common page)
- **Charts.aspx-** Graph representation of certain details (common page)
- **Statistics.aspx**-To view the images relevant to the accidents (common page)
- **DATASCANNING.aspx-**To approve/rejected the Incident Complain No
- **select.aspx**- To obtain more information about the incident (common page)

FOR THE ROAD DEVELOPMENT AUTHORITY (WEB FORM NAME, TASK)

- **RDAreg.aspx**-Registration page for the RDA members
- **Loginother.aspx**- Separate Login for the RDA
- **Analysis.aspx-** The details of the users and accident information (common page)
- **Charts.aspx-** Graph representation of certain details (common page)
- **Statistics.aspx**-To view the images relevant to the accidents (common page)
- **AccountRda.aspx-** To update any mistaken details in the system
- **PoliceandRda.aspx**- View information from the system (common page)
- **select.aspx**- To obtain more information about the incident (common page)

2.2 EXPLAINING THE LOGIC BEHIND EACH AND EVERY METHOD () USED

PAGE RELEVANT TO GENERAL PUBLIC

Under LogicPage.aspx.cs

Once the user fills out the registration and click submit the data will be transferred to the database (SQL SERVEER)

```
protected void Button1_Click(object sender, EventArgs e)
    {
        string NIC_NO = txtNiC.Text;
        string FULL_Name = txtName.Text;
        string Class_of_Vehicle = txtClassVehcile.Text;
        string Status_Reg = txtStatReg.Text;
        string Insurance_No = txtInsNo.Text;
        string Insurance_Category = txtInsCat.Text;
        string Vehicle_No = txtVehicleNo.Text;
        string Engine_No = txtEngNo.Text;

        if (NIC_NO != "" && FULL_Name != "" && Class_of_Vehicle != "" && Status_Reg
!= "" && Insurance_No != ""
         && Insurance_Category != "" && Vehicle_No != "" && Engine_No != "")
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-
RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string sql = "Insert Into Pub_RegInfo Values('" + NIC_NO + "','" + FULL_Name +
"','" + Class_of_Vehicle + "','"
            + Status_Reg + "','" + Insurance_No + "','" + Insurance_Category + "','" +
Vehicle_No + "','" + Engine_No + "')";
```

```
            SqlCommand cmd =new SqlCommand(sql,con);
            cmd.Parameters.AddWithValue("@Insurance_Category",
txtInsCat.SelectedItem.Value);
            try
            {
               con.Open();
               cmd.ExecuteNonQuery();
               Response.Write("Data Insertion carried out Successfully");
            }
            catch(Exception ex)
            {
               Response.Write(ex.Message);
            }


         }
         else
         {
            Response.Write("Please Enter Data to all the Fields");
                  }
```

Under Loginother.aspx.cs

When the relevant login details are submitted, the parameters assigned to the values will be searched through the database until the relevant values are found hence if the desired values are not found, it would indicate a message saying that login credentials are incorrect

```
         static string connectionString = @"Data Source=DESKTOP-RASQODU;Initial
         Catalog=ASPDB;Integrated Security=True";

protected void Button1_Click(object sender, EventArgs e)
      {


         try
         {
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = "sp_login";
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@NIC_No", txtLogNIC.Text.ToString());
            cmd.Parameters.AddWithValue("@Insurance_No", txtLogIns.Text.ToString());
            cmd.Connection = con;
            con.Open();
```

```
            SqlDataReader reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                Session["NIC_No"] = txtLogNIC.Text.ToString();
                txtInfo.Text = "Login Successful!";

                reader.Close();
                con.Close();

                Response.Redirect("~/PUBLIC.aspx");
            }
            else
            {
                txtInfo.Text = "Invalid Credentials";
            }

            reader.Close();

            con.Close();

        }
        catch (Exception ex)
        {

        }
    }
```

Under PUBLIC.aspx.cs

Since all the text fields are validated once the form is fully completed with the details and with the attached images, all the data will be transferred to the relevant tables in the database

```
protected void Button1_Click(object sender, EventArgs e)
    {
        string Incident_Time = txtIncidentTime.Text;
        string Collision_Date = txtDate.Text;
        string Desc = txtDesc.Text;
        string Name = txtAcName.Text;
        string Passengers_No = txtNoPassengers.Text;
        string Injury_Count = txtInjPpl.Text;
        string Incident_Location = txt_location.Text;
        string Complain_No = txtInsComplain.Text;


        if (Incident_Time != "" && Collision_Date != "" && Collision_Date != "" && Desc != "" && Name != ""
         && Passengers_No != "" && Injury_Count != "" && Incident_Location != "" && Complain_No != "")
```

```
        {


            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-
RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string sql = "Insert Into AccidentInfo_Pub  Values('" + Incident_Time + "','" +
Collision_Date + "','" + Desc + "','"
            + Name + "','" + Passengers_No + "','" + Injury_Count + "','" + Incident_Location +
"','" + Complain_No + "')";
            SqlCommand cmd = new SqlCommand(sql, con);
            try
            {
               con.Open();
               cmd.ExecuteNonQuery();
               Response.Write("Data Insertion is Carried out Perfectly");
            }
            catch (Exception ex)
            {
               Response.Write(ex.Message);
            }

        }
        else
        {
            Response.Write("Please Insert Data to each and every field");
        }


        byte[] bytes;
        using (BinaryReader br = new BinaryReader(FileUpload1.PostedFile.InputStream))
        {
            bytes = br.ReadBytes(FileUpload1.PostedFile.ContentLength);
        }
        string                                    constr                                    =
ConfigurationManager.ConnectionStrings["ASPDBConnectionString"].ConnectionString;
        using (SqlConnection conn = new SqlConnection(constr))
        {
            string sql = "INSERT INTO Images VALUES(@Name, @Image, @ContentType)";
            using (SqlCommand cmd = new SqlCommand(sql, conn))
            {
                cmd.Parameters.AddWithValue("@Name",
Path.GetFileName(FileUpload1.PostedFile.FileName));
                cmd.Parameters.AddWithValue("@ContentType",
FileUpload1.PostedFile.ContentType);
                cmd.Parameters.AddWithValue("@Image", bytes);
                conn.Open();
                cmd.ExecuteNonQuery();
                conn.Close();
            }
        }
```

```
        Response.Redirect(Request.Url.AbsoluteUri);


    }
```

In this method it would basically display the uploaded images in a grid view for the users as well

```
protected void Page_Load(object sender, EventArgs e)
    {

        if (!this.IsPostBack)
        {
            string constr = ConfigurationManager.ConnectionStrings["ASPDBConnectionString"].ConnectionString;
            using (SqlConnection conn = new SqlConnection(constr))
            {
                using (SqlDataAdapter sda = new SqlDataAdapter("SELECT * FROM Images", conn))
                {
                    DataTable dt = new DataTable();
                    sda.Fill(dt);
                    gvImages.DataSource = dt;
                    gvImages.DataBind();
                }
            }
        }

    }
```

So the images are stored in the database using bytes in order to retrieve it from the database the below method is being used

```
protected void OnRowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            DataRowView dr = (DataRowView)e.Row.DataItem;
            string imageUrl = "" + Convert.ToBase64String((byte[])dr["Image"]);
            (e.Row.FindControl("Image1") as Image).ImageUrl = imageUrl;
        }
    }
```

Under ProfilePublic.aspx.cs

Users can Update the information fed to the system using the available controllers in site

```
protected void UpdateButton(object sender, EventArgs e)
    {

        string NewComplain_No = txtNewComNo.Text;
        string NewIncident_Time = txtNewIncidentTime.Text;
        string NewCollision_Date = txtNewtDate.Text;
        string Newdesc = txtNewDesc.Text;
        string NewName = txtNewAcName.Text;
        string NewPassNo = txtNewNoPassengers.Text;
        string NewInjNo = txtNewInjPpl.Text;
        string NewLoc = txtNew_location.Text;



        SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
        string qry = "UPDATE AccidentInfo_Pub SET Incident_Time='" + NewIncident_Time + "',Collision_Date='" + NewCollision_Date + "',describe='" + Newdesc + "', Name='" + NewName + "' , Passenger_No='" + NewPassNo + "',Injury_Count='" + NewInjNo + "',Incident_Location='" + NewLoc + "' WHERE Complain_No='" + NewComplain_No + "' ";
        SqlCommand cmd = new SqlCommand(qry, con);

        try
        {
           con.Open();
           cmd.ExecuteNonQuery();
           Response.Write("Information Updating is Completed");
        }

        catch (Exception ex)
        {
           Response.Write("Exception Occurred" + ex);
        }
            }
```

Under RESULTS.aspx.cs

From this method data will be retrieved and displayed from the database, the data are fed by the insurance agents to the system

```
protected void Page_Load(object sender, EventArgs e)
```

```
    {
        string connStr = ConfigurationManager.ConnectionStrings["ASPDBConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(connStr);


        conn.Open();
        string sql = "SELECT Complain_No,AprRej,ClaimingDocs  from DataAnalysis";
        SqlCommand cmd = new SqlCommand(sql, conn);
        SqlDataReader reader = cmd.ExecuteReader();


        GridView3.DataSource = reader;
        GridView3.DataBind();

        conn.Close();


          }
```

PAGE RELEVANT TO POLICE

Under Policereg.aspx.cs

Registration details are transferred once the submit button pressed to the database table relevant to gathering information of the users

```
protected void Button1_Click(object sender, EventArgs e)
    {
        string SSN=txtssn.Text;
        string Nic=txtNICP.Text;
        string Name=txtPOName.Text;
        string Loc_Dep=txtLocDep.Text;
        string District =txtDis.Text;
        string Dep_Name=txtNameDep.Text;


        if(SSN!="" && Nic!="" && Name!="" && Loc_Dep!="" && District!="" && Dep_Name!="" && Dep_Name!="" )
        {
            SqlConnection con =new SqlConnection(@"Data Source=DESKTOP-RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string sql = "Insert Into Police_RegInfo Values('" + SSN + "','" + Nic + "','" + Name + "','"+Loc_Dep+"','" + District + "','" + Dep_Name + "')";
            SqlCommand cmd = new SqlCommand(sql, con);

            try
            {
```

```
            con.Open();
            cmd.ExecuteNonQuery();
            Response.Write("Data has been Inserted Successfully");

        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
        }
    }
    else
    {
        Response.Write("Please Insert Data to All Fields");
    }
}
```

Under Loginother.aspx.cs

A method to identify the details from the tables from the database relevant for the specific registration event in order to log the user in to the system

```
protected void Button2_Click(object sender, EventArgs e)
    {
        try
        {
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = "spPolice_login";
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@SSN", txtLogSSN.Text.ToString());
            cmd.Parameters.AddWithValue("@Loc_Dep", txtDepName.Text.ToString());
            cmd.Connection = con;
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                Session["SSN"] = txtLogNIC.Text.ToString();
                txtPoliceInfo.Text = "Login Successful!";

                reader.Close();
                con.Close();

                Response.Redirect("~/PoliceandRda.aspx");
            }
            else
            {
                txtPoliceInfo.Text = "Invalid Credentials";
            }
```

```
            reader.Close();

            con.Close();

        }
        catch (Exception ex)
        {

        }
    }
```

Under AccountPol.aspx.cs

A method which creates and allows the users to their own profile Details in the database

```
protected void Button1_Click(object sender, EventArgs e)
    {
        string NewSSN = txtNewRDASSN.Text;
        string  NewName= txtNewPOName.Text;
        string  NewLoc_Dep= txtNewLocDep.Text;
        string  NewDistrict= txtNewDis.Text;
        string  NewDep_Name= txtNewNameDep.Text;
        if (NewSSN != "" && NewName != "" && NewLoc_Dep != "" && NewDistrict != "" && NewDep_Name != "" )
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string qry = "UPDATE Police_RegInfo SET Name='" + NewName + "',Loc_Dep='" + NewLoc_Dep + "',District='" + NewDistrict + "', Dep_Name='" + NewDep_Name + "' WHERE SSN='" + NewSSN + "' ";
            SqlCommand cmd = new SqlCommand(qry, con);


            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
                Response.Write("Information Updating is Completed");
            }

            catch (Exception ex)
            {
                Response.Write("Exception Occurred" + ex);
            }
        }
```

PAGE RELEVANT FOR THE ISURANCE AGENTS

Under Insurancereg.aspx.cs

Once the submit button is pressed the method is created to transfer the details to relevant the table in the database

```
protected void Button1_Click(object sender, EventArgs e)
    {
        string SSN_INS = txtInsASSN.Text;
        string NIC_INSNO = txtInsNIC.Text;
        string AGENT_NAME = txtInsName.Text;
        string DEP_NAME = txtInsDEPName.Text;
        string BRANCH_NAME = txtInsBRN.Text;
        string DISTRICT = txtInsDis.Text;
        string JOB_TITLE = txtInsJobTitle.Text;

        if (SSN_INS != "" && NIC_INSNO != "" && AGENT_NAME != "" &&
DEP_NAME != "" && BRANCH_NAME != "" && DISTRICT != "" && JOB_TITLE !=
"")
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-
RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string sql = "Insert Into Ins_RegInfo Values('" + SSN_INS + "','" + NIC_INSNO +
"','" + AGENT_NAME + "','" + DEP_NAME + "','" + BRANCH_NAME + "','" + DISTRICT
+ "','"+ JOB_TITLE + "')";
            SqlCommand cmd = new SqlCommand(sql, con);


            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
                Response.Write("Data has been Inserted Successfully");

            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
        else
        {
            Response.Write("Please Insert Data to All Fields");
        }
    }
```

Under LoginInsurance.aspx.cs

Login method to log the user to the relevant main page of the site

```
protected void Button3_Click(object sender, EventArgs e)
    {
        try
        {
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = "spIns_login";

            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@SSN_INS", txtLogInsSSN.Text.ToString());
            cmd.Parameters.AddWithValue("@BRANCH_NAME",
txtInsBRNNAME.Text.ToString());

            cmd.Connection = con;
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                Session["SSN_INS"] = txtLogInsSSN.Text.ToString();

                txtInsInfo.Text = "Login Successful!";

                reader.Close();
                con.Close();

                Response.Redirect("~/PoliceandRda.aspx");
            }
            else
            {
                txtInsInfo.Text = "Invalid Credentials";

            }

            reader.Close();

            con.Close();}
        catch (Exception ex)
        {

        }
            }
```

Under DATASCANNING.aspx.cs

Method to store the details of the analyzed photos in order to publish it to the drivers of the system

```
protected void Button2_Click(object sender, EventArgs e)
    {
        string SSN_INS = txtInsReSSN.Text;
        string Complain_No = txtInsCom.Text;
        string AprRej = txtInsapp.Text;
        string ClaimingDocs = txtInsDoc.Text;



        if (SSN_INS != "" && Complain_No != "" && AprRej != "" && ClaimingDocs !=
"")
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-
RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string sql = "Insert Into DataAnalysis Values('" + SSN_INS + "','" + Complain_No
+ "','" + AprRej + "','" + ClaimingDocs + "')";
            SqlCommand cmd = new SqlCommand(sql, con);


            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
                Response.Write("Data has been Inserted Successfully");

            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
        else
        {
            Response.Write("Please Insert Data to All Fields");
        }
            }
```

PAGE RELEVANT FOR THE RDA

Under RDAreg.aspx.cs

Method to store the registration details of the users separately in the database

```
protected void Button1_Click(object sender, EventArgs e)
    {
        string SSN=txtRDASSN.Text;
        string NIC_No=txtRDANIC.Text;
        string Name=txtRDAName.Text;
        string Dep_Name=txtRDADEPName.Text;
        string Branch=txtRDABRN.Text;
        string District = txtRDADis.Text;

        if (SSN != "" && NIC_No != "" && Name != "" && Dep_Name != "" && Branch != "" && Dep_Name != "" && District != "")
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string sql = "Insert Into RDA_RegInfo Values('" + SSN + "','" + NIC_No + "','" + Name + "','" + Dep_Name + "','" + Branch + "','" + District+ "')";
            SqlCommand cmd = new SqlCommand(sql, con);


            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
                Response.Write("Data has been Inserted Successfully");

            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
        else
        {
            Response.Write("Please Insert Data to All Fields");
        }

    }
```

Under Loginother.aspx.cs

A method to log the specific user to the system using the 'session method'

```
protected void Button3_Click(object sender, EventArgs e)
    {
        try
        {
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = "spRDA_login";

            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@SSN", txtLogRDASSN.Text.ToString());
            cmd.Parameters.AddWithValue("@Branch_No",
txtRDABRNNAME.Text.ToString());

            cmd.Connection = con;
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                Session["SSN"] = txtLogRDASSN.Text.ToString();

                txtRDAInfo.Text = "Login Successful!";

                reader.Close();
                con.Close();

                Response.Redirect("~/PoliceandRda.aspx");
            }
            else
            {
              txtInsInfo. Text = "Invalid Credentials";

            }

            reader.Close();

            con.Close();

        }
        catch (Exception ex)
        {

        }

            }
```

Under AccountRda.aspx.cs

Method enabling the user to update the information in the system

```
protected void Button1_Click(object sender, EventArgs e)
    {
        string NewSSN = txtNewRDASSN.Text;
        string NewName = txtNewRDAName.Text;
        string NewDep_Name = txtNewRDADEPName.Text;
        string NewBranch_No = txtNewRDABRN.Text;
        string NewDistrict = txtNewRDADis.Text;
        if (NewSSN != "" && NewName != "" && NewDep_Name != "" && NewBranch_No
!= "" && NewDistrict != "")
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-
RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            string qry = "UPDATE RDA_RegInfo SET Name='" + NewName + "',Dep_Name='"
+ NewDep_Name + "', Branch_No='" + NewBranch_No + "',District='" + NewDistrict + "'
WHERE SSN='" + NewSSN + "' ";
            SqlCommand cmd = new SqlCommand(qry, con);

            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
                Response.Write("Information Updating is Completed");
            }

            catch (Exception ex)
            {
                Response.Write("Exception Occurred" + ex);
            }
        }

            }
```

Under Analysis.aspx.cs

Method to retrieve information of the drivers and the accidents which was fed to the system through grid views and list view

```
protected void Page_Load(object sender, EventArgs e)
    {
        string connStr = ConfigurationManager.ConnectionStrings["ASPDBConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(connStr);


        conn.Open();
        string sql = "SELECT * from Pub_RegInfo";
        SqlCommand cmd = new SqlCommand(sql, conn);
        SqlDataReader reader = cmd.ExecuteReader();


        GridView1.DataSource = reader;
        GridView1.DataBind();

        conn.Close();

        {
            if (!this.IsPostBack)
            {
                this.BindListView();
            }
        }

    }


    protected void Back_Click(object sender, EventArgs e)
    {
        Response.Redirect("PoliceandRda.aspx");
    }


    private void BindListView()
    {
        string constr = ConfigurationManager.ConnectionStrings["ASPDBConnectionString"].ConnectionString;
        using (SqlConnection con = new SqlConnection(constr))
        {
```

```
            using (SqlCommand cmd = new SqlCommand())
            {
                cmd.CommandText = "SELECT Complain_No, Collision_Date, Name,describe,Incident_Time,Incident_Location,Injury_Count,Passenger_No FROM AccidentInfo_Pub ";
                cmd.Connection = con;
                using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
                {
                    DataTable dt = new DataTable();
                    sda.Fill(dt);
                    lvCustomers.DataSource = dt;
                    lvCustomers.DataBind();
                }
            }
        }
    }
    protected void OnPagePropertiesChanging(object sender, PagePropertiesChangingEventArgs e)
    {
        (lvCustomers.FindControl("DataPager1") as DataPager).SetPageProperties(e.StartRowIndex, e.MaximumRows, false);
        this.BindListView();
    }
```

Under Charts.aspx.cs

Method to represent data in the grid view graphically

```
public partial class charts : System.Web.UI.Page
  {
    private SqlConnection con;
    private SqlCommand com;
    private string constr, query;
    private void connection()
    {
        constr = ConfigurationManager.ConnectionStrings["ASPDBConnectionString"].ToString();
        con = new SqlConnection(constr);
        con.Open();
    }
    protected void Page_Load(object sender, EventArgs e)
    {

        if (!IsPostBack)
        {
            Bindchart();

        }
```

```
    }


    private void Bindchart()
    {
        connection();
        com = new SqlCommand("GetAccidentData", con);
        com.CommandType = CommandType.StoredProcedure;
        SqlDataAdapter da = new SqlDataAdapter(com);
        DataSet ds = new DataSet();
        da.Fill(ds);

        DataTable ChartData = ds.Tables[0];


        string[] XPointMember = new string[ChartData.Rows.Count];
        int[] YPointMember = new int[ChartData.Rows.Count];

        for (int count = 0; count < ChartData.Rows.Count; count++)
        {

            XPointMember[count] = ChartData.Rows[count]["Incident_Location"].ToString();

            YPointMember[count]                                                    =
Convert.ToInt32(ChartData.Rows[count]["Passenger_No"]);

        }

        Chartdetails.Series[0].Points.DataBindXY(XPointMember, YPointMember);


        Chartdetails.Series[0].BorderWidth = 10;

        Chartdetails.Series[0].ChartType = SeriesChartType.Pie;
        foreach (Series charts in Chartdetails.Series)
        {
            foreach (DataPoint point in charts.Points)
            {
                switch (point.AxisLabel)
                {
                    case "Q1": point.Color = Color.RoyalBlue; break;
                    case "Q2": point.Color = Color.SaddleBrown; break;
                    case "Q3": point.Color = Color.SpringGreen; break;
                }
                point.Label = string.Format("{0:0} - {1}", point.YValues[0], point.AxisLabel);

            }
        }
```

```
        con.Close();



    }
}
```

Under Statistics.aspx.cs and ShowImage.ashx.cs

Method to view the images from the database which were uploaded by the drivers who were registered to the system

```
public class ShowImage : IHttpHandler
  {

    public void ProcessRequest(HttpContext context)
    {
      string Id = context.Request.QueryString["Id"].ToString();
      string sConn =
System.Configuration.ConfigurationManager.ConnectionStrings["ASPDBConnectionString"
].ToString();
      SqlConnection objConn = new SqlConnection(sConn);
      objConn.Open();
      string sTSQL = "SELECT Image from Images where Id=@Id";
      SqlCommand objCmd = new SqlCommand(sTSQL, objConn);
      objCmd.CommandType = CommandType.Text;
      objCmd.Parameters.AddWithValue("@Id", Id.ToString());
      object data = objCmd.ExecuteScalar();
      objConn.Close();
      objCmd.Dispose();
      context.Response.BinaryWrite((byte[])data);
    }



    public bool IsReusable
    {
      get
      {
        return false;
      }
    }
```

```
        }
protected void txtGetIamge_Click(object sender, EventArgs e)
    {

        {
            Image1.ImageUrl = "ShowImage.ashx?id=" + Id.Text;
        }
            }
```

Under select.aspx.cs

A separate method to view information in a list view

```
public partial class select : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

        if (!Page.IsPostBack)
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
            SqlCommand cmd = new SqlCommand("select * from AccidentInfo_Pub ", con);
            SqlDataAdapter sda = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            DropDownList1.DataSource = dt;
            DropDownList1.DataBind();
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {

        SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-RASQODU;Initial Catalog=ASPDB;Integrated Security=True");
        SqlCommand cmd = new SqlCommand("select * from AccidentInfo_Pub where Passenger_No=" + DropDownList1.SelectedItem.Value, con);
        SqlDataAdapter sda = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        sda.Fill(dt);
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
        }
```

CHAPTER 3

DESCRIBING THE TEST CASES

3.1 TEST CASE ID'S AND IT'S PURPOSE

ARW- Accident Reporting Website

- REGARW001 – Public Registration (**PR_1)**
- REGARW002 – Police Registration **(PR_2)**
- REGARW003 – Road Development Authority Registration **(PR_3)**
- REGARW004 – Insurance Registration **(PR_4)**

Form Submission – **FS** (For Public)

- FSARW005 – Submitting Accident Reports

Updating Information (Submissions) – **UI** (For Public)

- UIARW006 – Updating form Submission

Upload More Images – **UMI** (For Public)

- UMIARW007 – Uploading Extra Images

View Information – **VI** (For Police, RDA and Insurance)

- VIARW008 – Viewing displayed Information

View Images – **VIMG** (For Police, RDA and Insurance)

- VIMGARW009 – Viewing uploaded Images

Updating Police Information – **UPI** (For Police)

- UPIARW010 – Updating form submission

Updating RDA Information – **URDAI** (For RDA)

- URDAARW011 – Updating form submission

Submitting Results – **SR** (For Insurance)

- SRARW012 – Submitting Case Results of Accidents

Viewing Case Results – **VCR** (For Public)

- VCRARW013 – Viewing Case Reports of Accidents

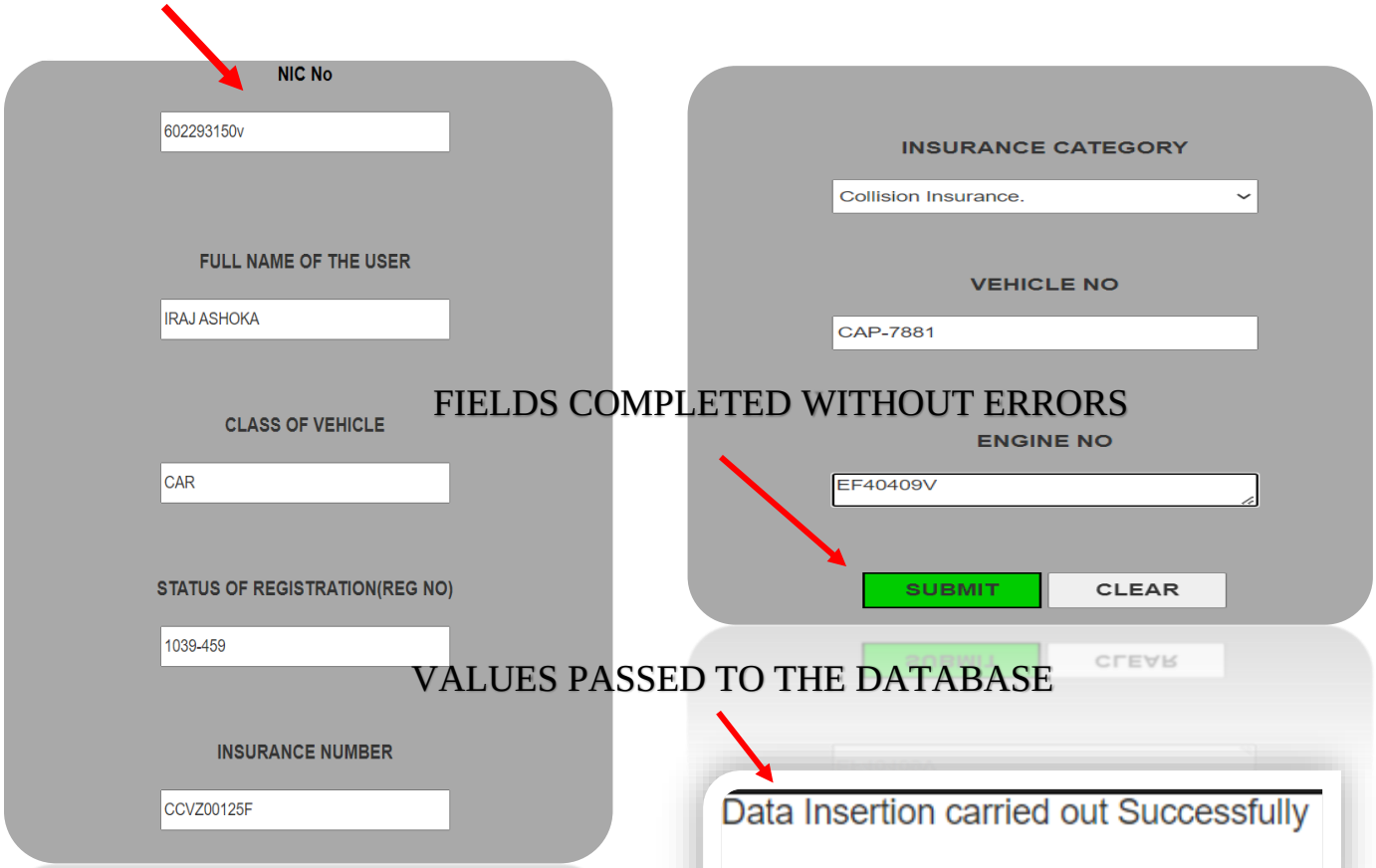
Login into the system

- LGPUBARW014 – Logging (**PUB**)
- LGPOLARW015 – Logging (**POL**)
- LGRDAARW016 – Logging (**RDA**)
- LGINSARW017 – Logging (**INS**)

3.2 TESTING USING DUMMY VALUES

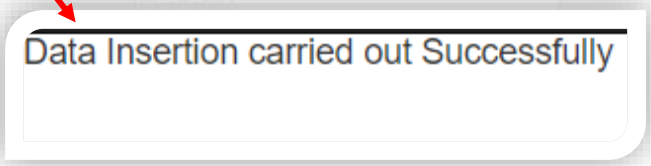
PUBLIC

ADDING VALUES TO TEXTFIELDS



FIELDS COMPLETED WITHOUT ERRORS

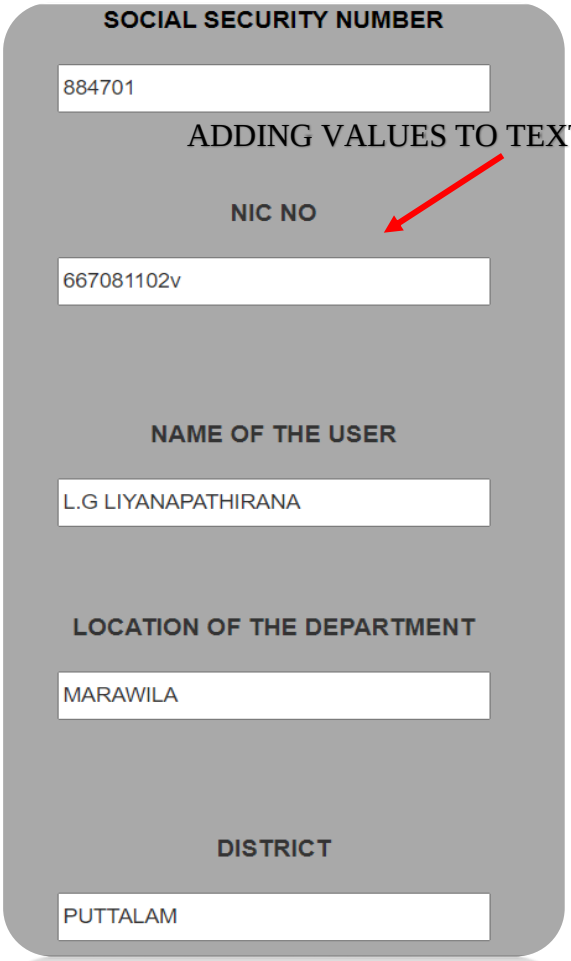
VALUES PASSED TO THE DATABASE



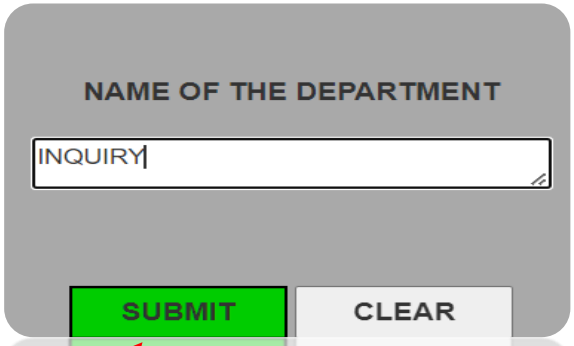
DATA IN THE DATABASE

NIC_No	FULL_Name	Class_of_Vehicle	Status_Reg	Insurance_No	Insurance_Category	Vehicle_No	Engine_N
200080905000	Isuru Paranavitana	Passenger car	749-500	LL349500C	Collision Insurance	CAS-7008	GW249669
602293150v	IRAJ ASHOKA	CAR	1039-459	CCVZ00125F	Collision Insurance	CAP-7881	EF40409V
655002001v	Amith Fernando	Passenger Car	741-100	WA600125	Collision Insurance	CAA-6008	E56-3001
678090020v	Sarath Mahanama	Passenger Car	86-1208	AC500122	Collision Insurance	AO-5885	DV51123
697981300v	Kasuni Amarajiva	Passenger Car	89-500	CWL300125	Collision Insurance	GH-1115	E600L49
8002009V	Damien Perera	Van	56-500	VLL2003001	Liability Coverage	HH-1002	300LV49

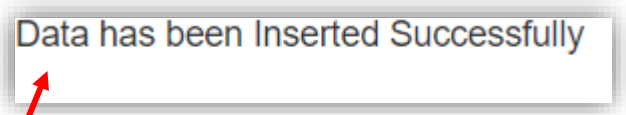
POLICE



ADDING VALUES TO TEXTFIELDS



FIELDS COMPLETED WITHOUT ERRORS

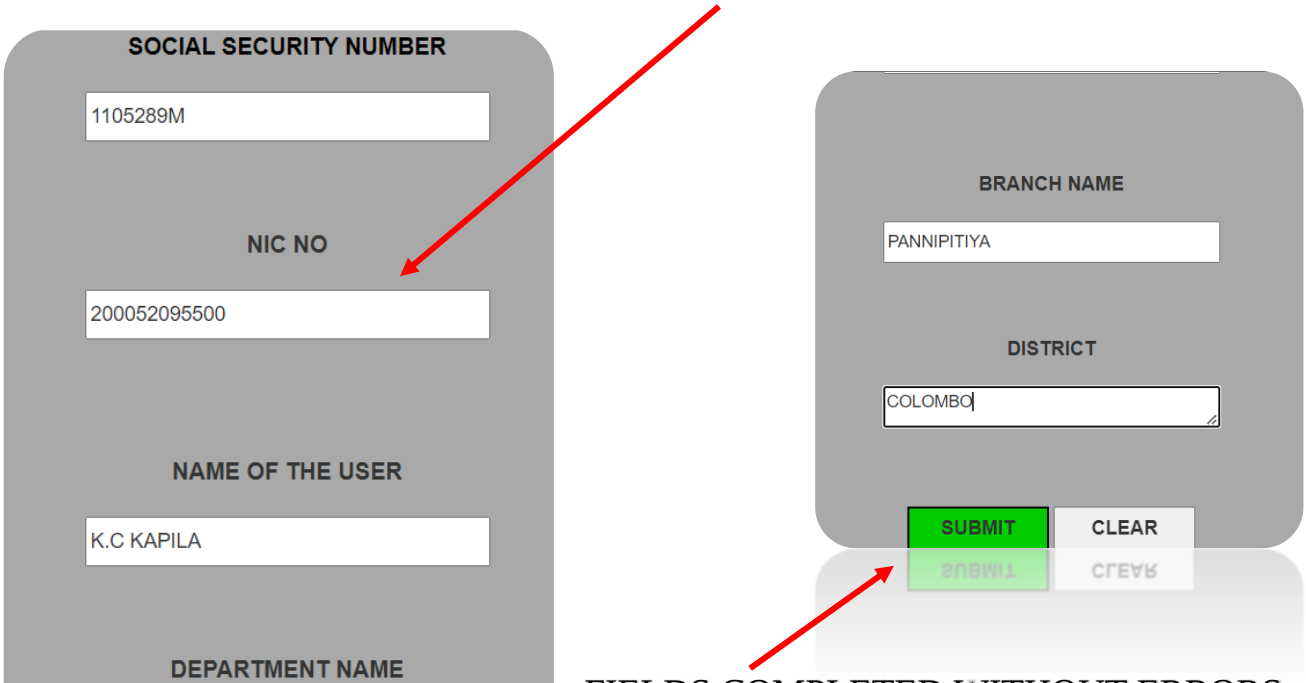


VALUES PASSED TO THE DATABASE

DATA IN THE DATABASE

SSN	Nic	Name	Loc_Dep	District	Dep_Name
749980	63385001v	J.K.C SOMAPALA	MIRIHANA	COLOMBO	DIG
554301	616009002V	S.W.R.P Gunathilaka	Awissawella	Colombo	Inquiry Seargent
449201	600021935v	L.L.P Sumanapala	Kandewatta	Gampaha	Inquiry Seargent
555649	68865000v	G.V Karunananda	Ambanpola	Kandy	Inquiry Seargent
667401	699910025v	T.E.C.K Eldeniya	Kirihetihena	Matale	Inquiry Seargent
`4701	667081102v	L.G LIYANAPATHIRANA	MARAWILA	PUTTALAM	INQUIRY

RDA ADDING VALUES TO TEXTFIELDS



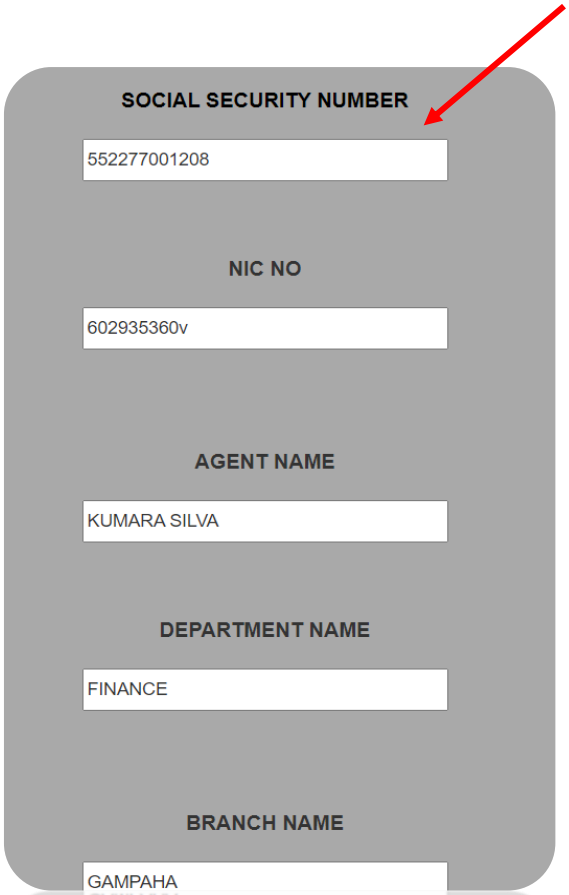
FIELDS COMPLETED WITHOUT ERRORS

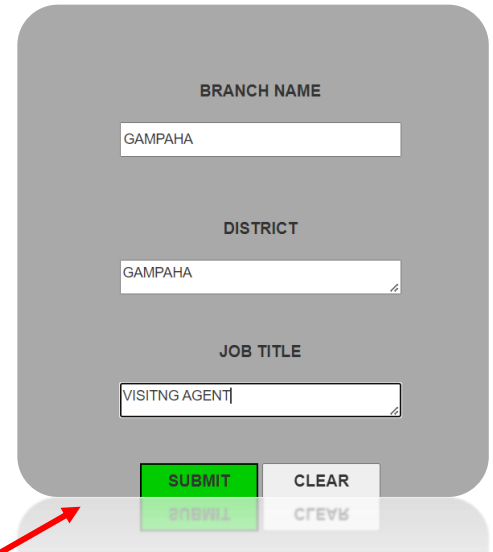
VALUES PASSED TO THE DATABASE

DATA IN THE DATABASE

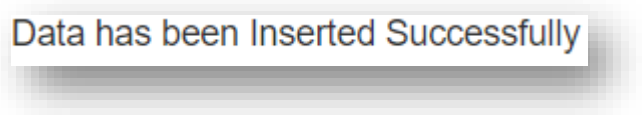
SSN	NIC_No	Name	Dep_Name	Branch_No	District
1105289M	200052095500	K.C KAPILA	CANAL SEC 1	PANNIPITIYA	COLOMBO
4376589M	900020031v	G.P.C ARIYARATHNE	CANALSEC 2 PART 2	KOTTAWA	COLOMBO
5203851C	798720051v	G.P.C Ariyarathne	Canal Sec 1	Nawala	Colombo
5525269R	608090200v	P.B Sirithilaka	Roads Sec 11	Kurunegala	Kurunegala
8090205I	661002318v	C.T Padmakumari	Irrigation Sec 1	Nawala	Colombo
28200M	699001005V	T.A Tanya	Roads Sec 1 Manager	Nawala	Colombo

INSURANCE ADDING VALUES TO TEXTFIELDS





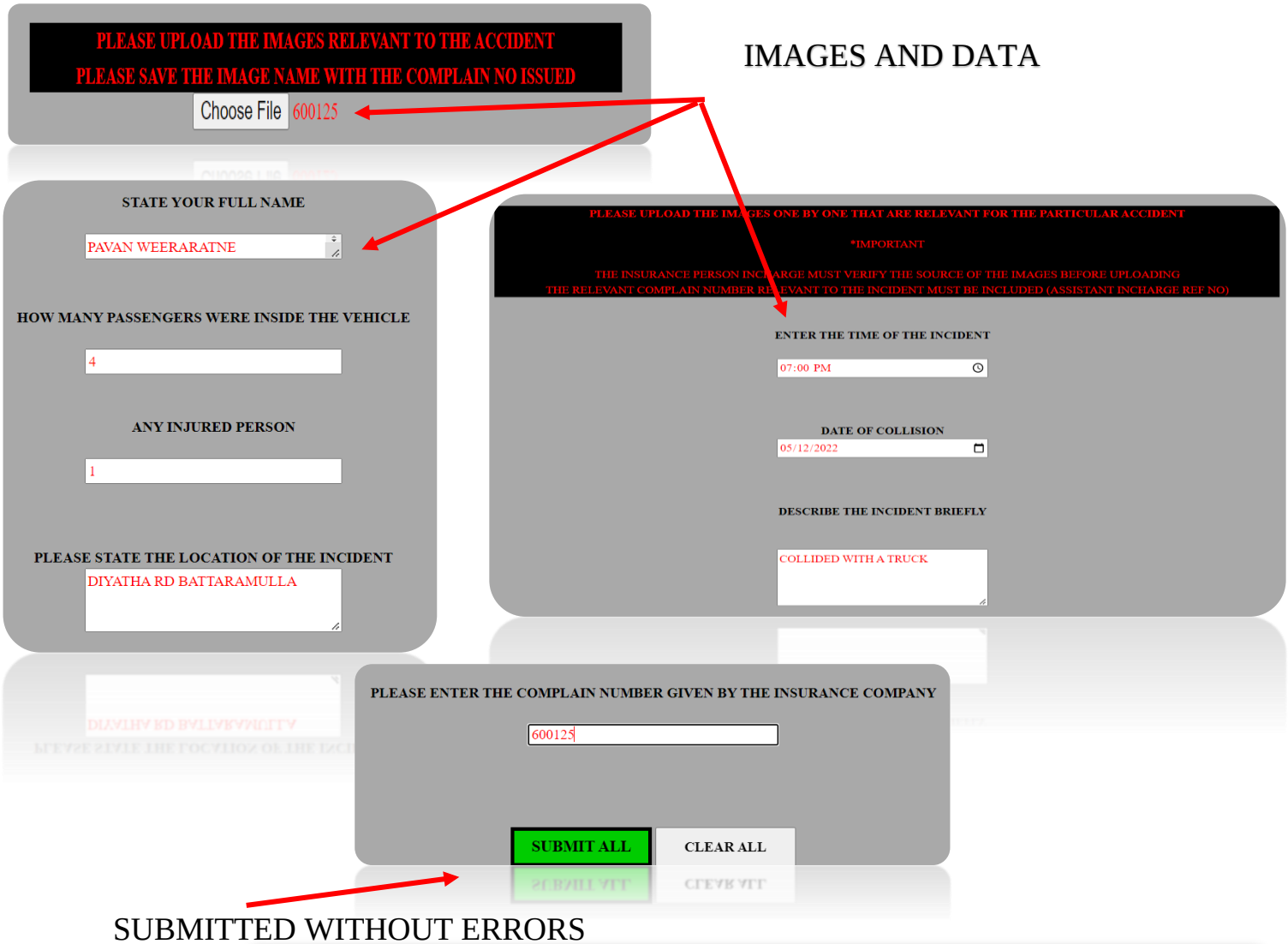
FIELDS COMPLETED WITHOUT ERRORS



VALUES PASSED TO THE DATABASE

SSN_INS	NIC_INSNO	AGENT_NAME	DEP_NAME	BRANCH_NAME	DISTRICT	JOB_TITLE
112233445566	102938475610	Amaya	Finance	Kollupitiya	Colombo 8	Marketing Manager
432050931093	200150002039	Hemaka Edirisinghe	Finance	Battaramulla	Colombo	Assistant Accountant
552277001208	602935360v	KUMARA SILVA	FINANCE	GAMPAHA	GAMPAHA	VISITNG AGENT
732964664210	61124492v	Parindi Soysa	Finance	Maharagama	Colombo	Executive Manager
800024935000	609011147v	Suresh Kasun	Finance	Borella	Colombo 7	Marketing Manager
922461090	688050701v	Piyumi Samarasekara	Finance	Nugegoda	Colombo	Visiting Agent

ACCIDENT REPORT SUBMISSION (PUBLIC)



SUBMITTED WITHOUT ERRORS

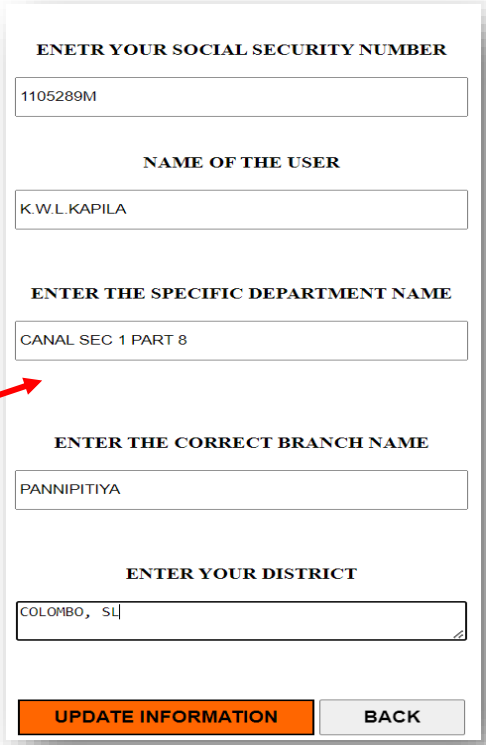
Incident_Time	Collision_Date	describe	Name	Passenger_No	Injury_Count	Incident_Location	Compla...
16:20:00.0000000	2022-04-12	Collided with a bus	Dhanuka Subasinghe	3	No	Rajagiriya	45567
11:51:00.0000000	2022-05-07	A TRUCK CAME AND COLLIDED	SURANGA SILVA	2	NO	ATHURUGIRIYA	51251
17:51:00.0000000	2022-04-30	COLLISION WITH A NOTHER CAR	TIASHA PERERA	4	NO	MAHARAGAMA	56662
19:00:00.0000000	2022-05-12	COLLIDED WITH A TRUCK	PAVAN WEERARATNE	4	1	DIYATHA RD BATTARAMULLA	600125
13:03:00.0000000	2022-04-28	COLLISION WITH ANOTHER CAR	AROSH WICKARAMANAYAKA	3	NO	KATUNAYAKA	665223
...00.0000000	2022-05-09	A three-wheeler came and collided with my car	Amani Karunaratne	1	No	Homagama	67731

IMAGES AND DATA IN THE DATABASE

Id	Name	Image	ContentType
1020	600125	0xFFD8FFE000104A4649460001010100000000000FFE1004...	application/octet-stream
1012	67731	0xFFD8FFE000104A4649460001010100000000000FFE1004...	application/octet-stream
1013	51251	0xFFD8FFE000104A4649460001010100000000000FFE1004...	application/octet-stream
1017	45567	0xFFD8FFE1001845786966000049492A0008000000000000...	application/octet-stream
1018	56662	0xFFD8FFE000104A4649460001010100000000000FFE1004...	application/octet-stream
1019	665223	0xFFD8FFE000104A4649460001020000640064000FFE1006...	application/octet-stream

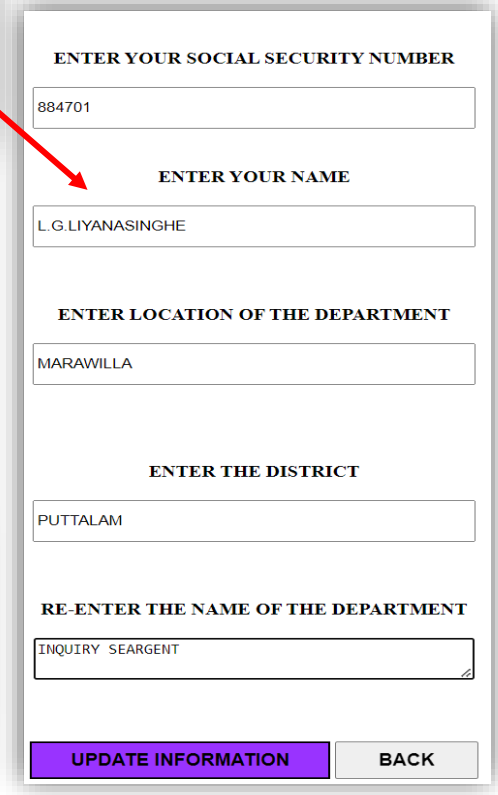
PROFILE UPDATING (PUBLIC, POLICE, RDA)

NEW VALUES INSERTED



DATABASES UPDATED

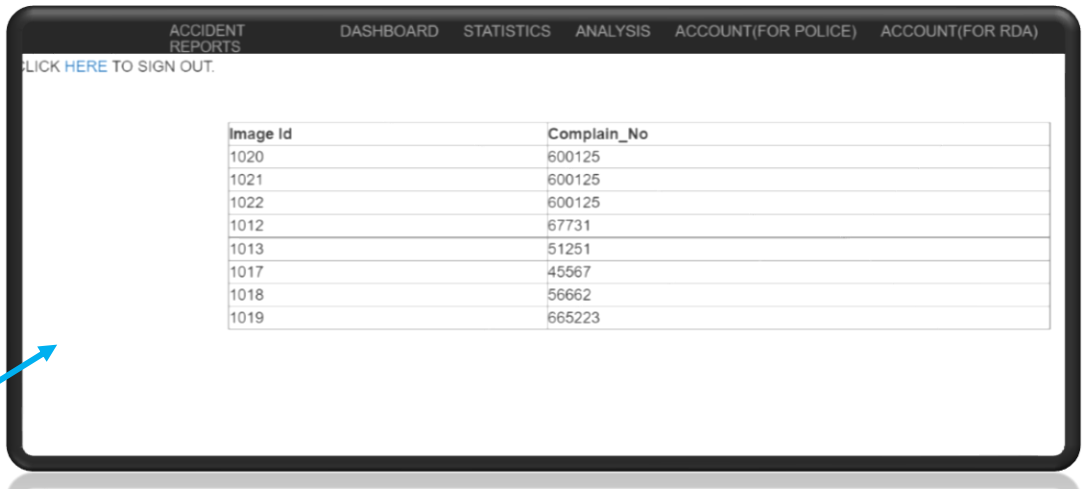
ncident_Time	Collision_Date	describe	Name	Passenger_No	Injury_Count	Incident_Location	Complain_No
16:20:00.0000000	2022-04-12	Collided wi bus	Dhanuka Subasinghe	3	No	Rajagiriya	45567
11:51:00.0000000	2022-05-07	A TRUCK CA AND COLLIDED	SURANGA SILVA	2	NO	ATHURUGIRIYA	51251
17:51:00.0000000	2022-04-30	COLLISION WITH A NOTHER CAR	TIASHA PERERA	4	NO	MAHARAGAMA	56662
19:00:00.0000000	2022-05-12	COLLIDED WITH CAR	PAVAN WEERARATNE	3	NO	DIYATHA RD	600125
13:03:00.0000000	2022-04-28	COLLISION WITH ANOTHER CAR	AROSH WICKARAMANAYAKA	3	NO	KATUNAYAKA	665223
07:50:00.0000000	2022-05-09	A three-wheeler came and collided with my car	Amani Karunaratne	1	No	Homagama	67731



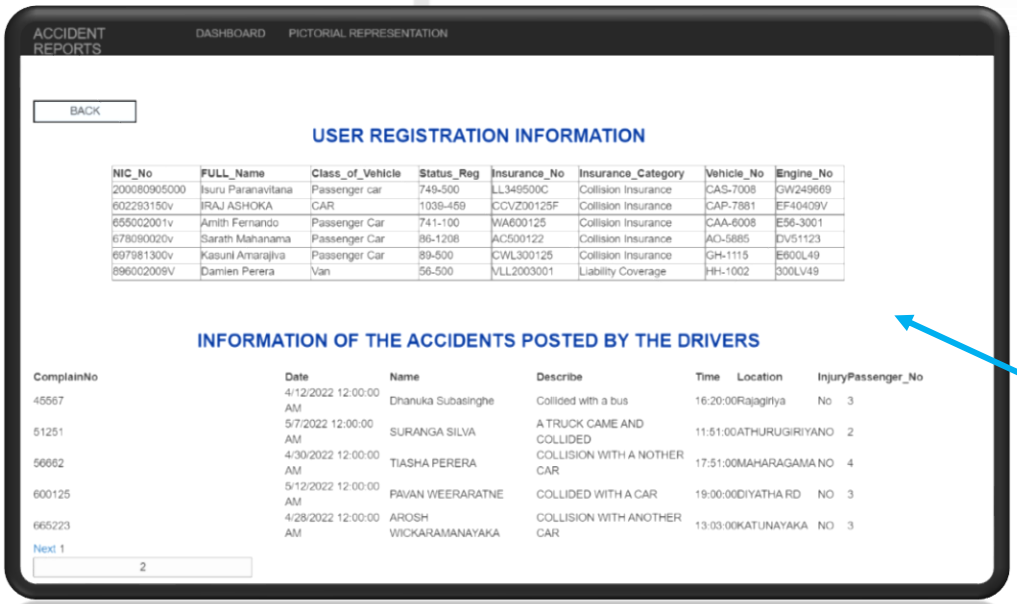
SSN	Nic	Name	Loc_Dep	District	Dep_Name
749980	63385001v	J.K.C SOMAPALA	MIRIHANA	COLOMBO	DIG
554301	616009002V	S.W.R.P Gunathilaka	Awissawella	Colombo	Click to select the wh
449201	600021935v	L.L.P Sumanapala	Kandewatta	Gampaha	Inquiry Seargent
555649	68865000v	G.V Karunanandha	Ambanpola	Kandy	Inquiry Seargent
667401	699910025v	T.E.C.K Eldeniya	Kirihetihena	Matale	Inquiry Seargent
884701	667081102v	L.G.LIYANASINGHE	MARAWILLA	PUTTALAM	INQUIRY SEARGENT

SSN	NIC_No	Name	Dep_Name	Branch_No	District
1105289M	200052095500	K.W.L.KAPILA	CANAL SEC 1 PART 8	PANNIPITIYA	COLOMBO, SL
4376589M	900020031v	G.P.C ARIYARATHNE	CANALSEC 2 PART 2	KOTTAWA	COLOMBO
5203851C	798720051v	G.P.C Ariyarathne	Canal Sec 1	Nawala	Colombo
5525269R	608090200v	P.B Sirithilaka	Roads Sec 11	Kurunegala	Kurunegala
8090205I	661002318v	C.T Padmakumari	Irrigation Sec 1	Nawala	Colombo
8928200M	699001005V	T.A Tanya	Roads Sec 1 Manager	Nawala	Colombo

VIEWING REPORTS, USER DETAILS AND IMAGES (POLICE, RDA AND INSURANCE)

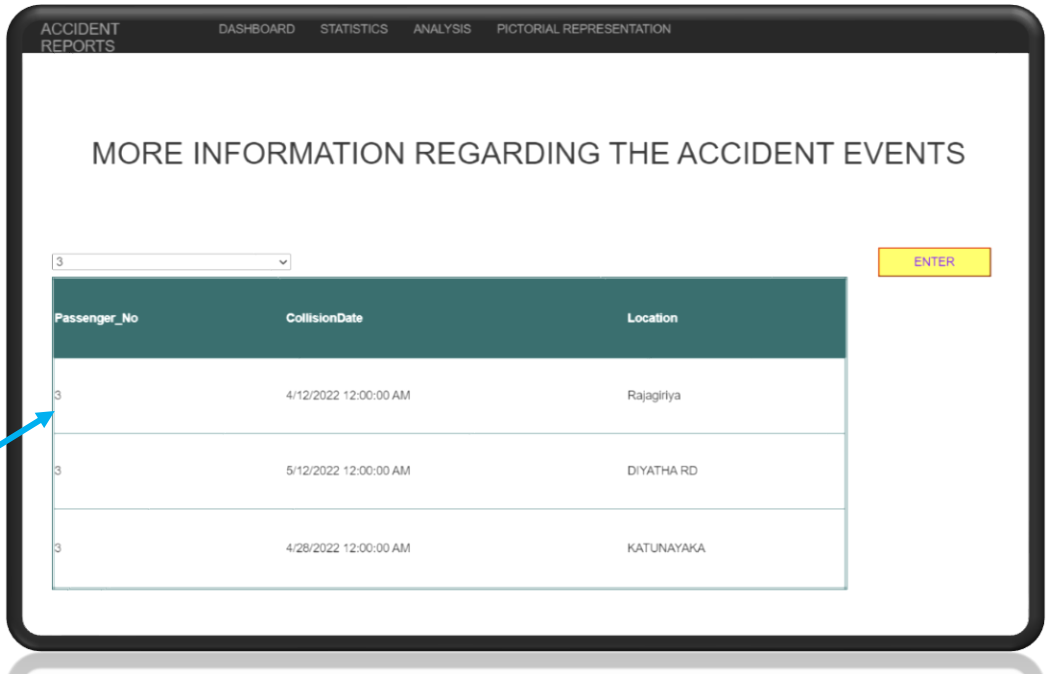


DASHBOARD VIEW

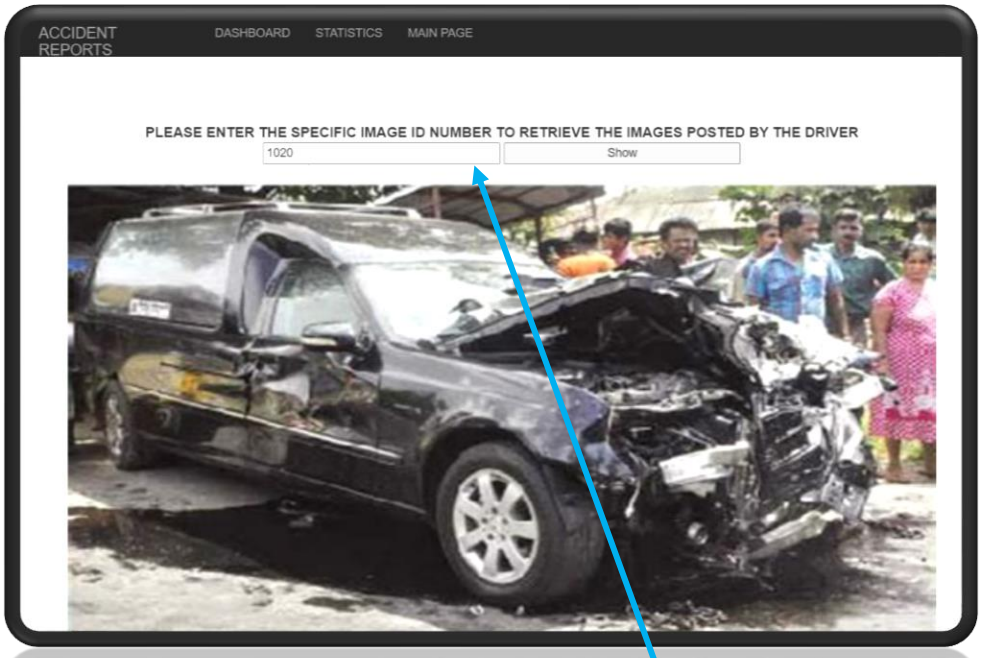


DESIGN VIEW FOR RETRIEVING INFORMATION FROM THE DATABASE

GRAPHICAL VIEW OF DETAILS REGARDING THE ACCIDENT

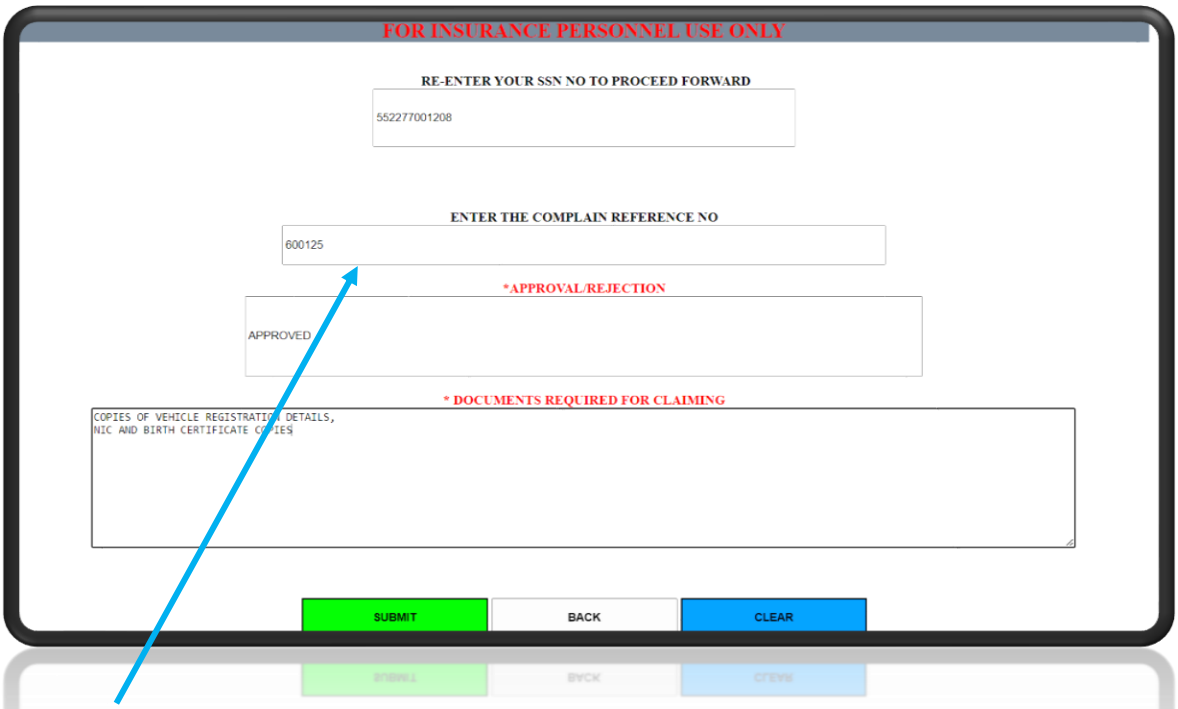


LIST VIEW TO GET MORE DETAILS REGARDING THE ACCIDENT



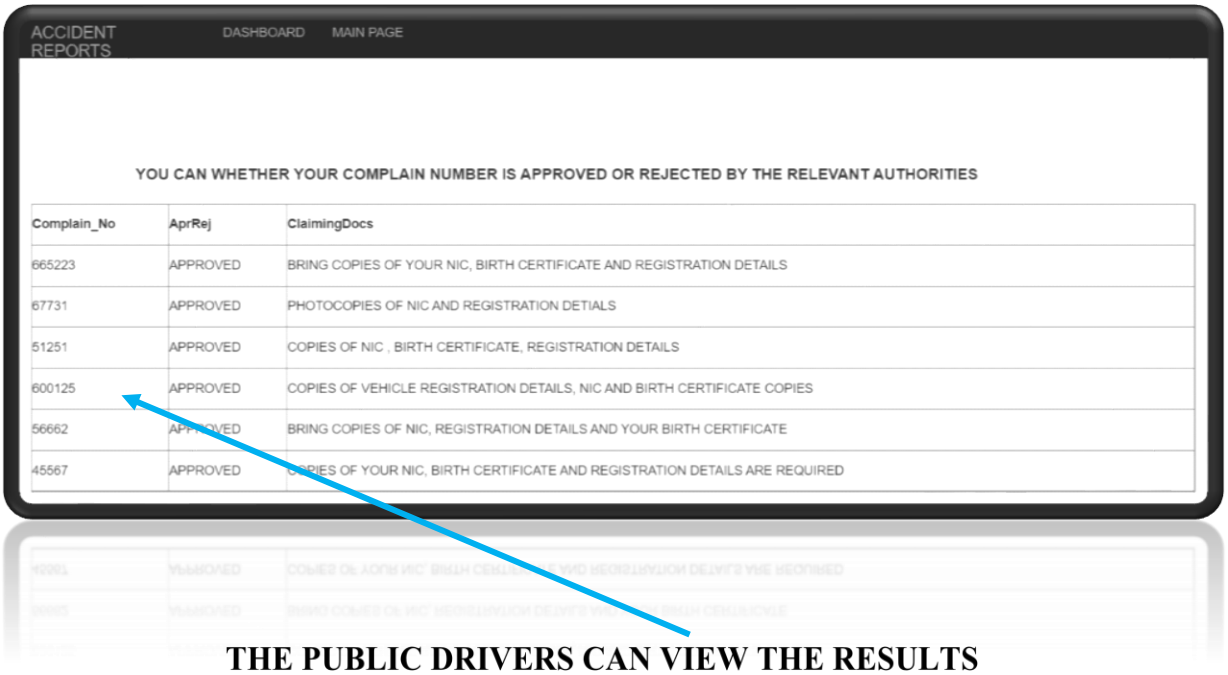
IMAGES ARE DISPLAYED ONCE THE RELEVANT IMAGE ID IS SELECTED AND ENTERED IN THE TEXTBOX ABOVE

INSURANCE ANALYSIS PAGE TO PUBLISH RESULTS



ONCE AUTHORIZED INSURANCE AGENT SSN IS INCLUDED THE FORM IS SUBMITTED

VIEWING PAGE FOR THE CASE RESULT

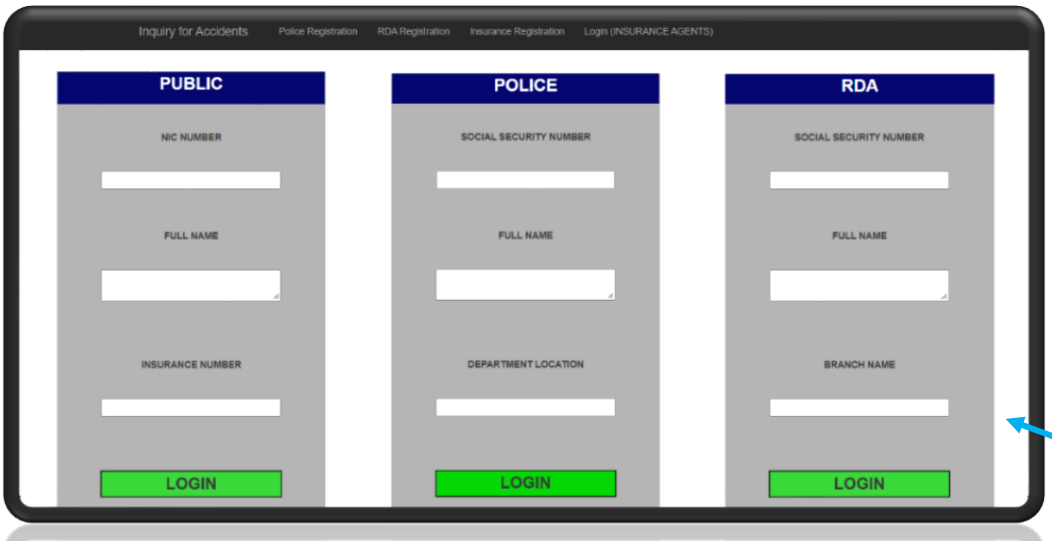


Complain_No	AprRej	ClaimingDocs
665223	APPROVED	BRING COPIES OF YOUR NIC, BIRTH CERTIFICATE AND REGISTRATION DETAILS
67731	APPROVED	PHOTOCOPIES OF NIC AND REGISTRATION DETIALS
51251	APPROVED	COPIES OF NIC , BIRTH CERTIFICATE, REGISTRATION DETAILS
600125	APPROVED	COPIES OF VEHICLE REGISTRATION DETAILS, NIC AND BIRTH CERTIFICATE COPIES
56662	APPROVED	BRING COPIES OF NIC, REGISTRATION DETAILS AND YOUR BIRTH CERTIFICATE
45567	APPROVED	COPIES OF YOUR NIC, BIRTH CERTIFICATE AND REGISTRATION DETAILS ARE REQUIRED

THE PUBLIC DRIVERS CAN VIEW THE RESULTS

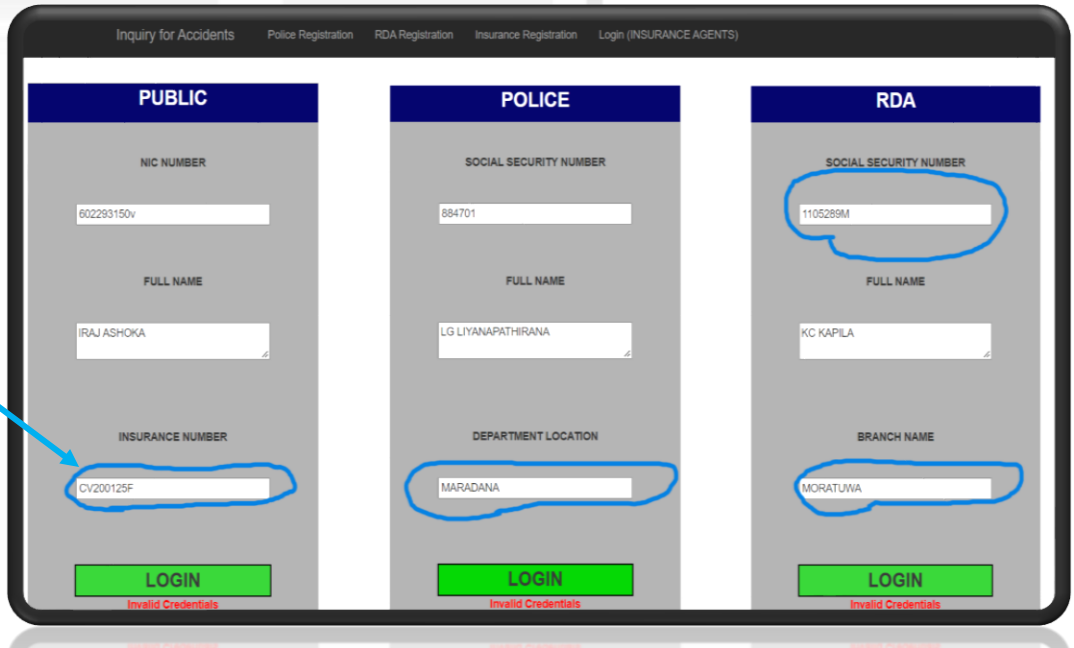
TO THE RELEVANT CASES FILED

LOGIN UI'S FOR EACH REGISTRATION TYPES



DESIGN VIEW FOR THE LOGIN PROCESS

IF INCORRECT DATA IS SUBMITTED THE FOLLOWING WARNING WILL BE INDICATED ELSE USERS ARE DIRECTED TO THE DASHBOARD



602293150v	IRAJ ASHOKA	CAR	1039-459	CCVZ00125F	Collision Insurance	CAP-7881	EF40409V

884701	667081102v	L.G LIYANAPATHIRANA	MARAWILA	PUTTALAM	INQUIRY

1105289M	200052095500	K.C KAPILA	CANAL SEC 1	PANNIPITIYA	COLOMBO

- Likewise, The Same Methodology Is Carried Out for Insurance Login

CHAPTER 4

TEST CASE REPORT

PROJECT NAME	ARW_SQA_TESTING
MODULE NAME	ARW_MODULE
CREATED BY	M.P.V BANDARANAYAKE
REVIEWED BY	D.T. KIRIELLA
REVIEWED DATE	

PLEASE REFER PAGE 25 FOR THE ABBREIVATINON DEFINITION

Test Case Number	Test Case Name	Pre-condition	Test Steps	Test Data	Expected Results	Actual Result	Status	Executed by	Priority	Defect ID	Comments
REG_ ARW_ 001	**(PR_1)**	URL and Internet Access	Navigate to registration tab and fill the form	602293150v, Iraj Ashoka, Car, 1039-459, CCV200125F	Data passing in to the database	"Data Insertion Successful"	**PASS**		High		**Ref Pg 26**
REG_ ARW_ 002	**(PR_2)**	URL and Internet Access	Navigate to registration tab and fill the form	884701, 667081102v, LG Liyanapathirana, Marawila, Puttalam, Inquiry	Data passing in to the database	"Data Insertion Successful"	**PASS**		High		**Ref Pg 27**
REG_ ARW_ 003	**(PR_3)**	URL and Internet Access	Navigate to registration tab and fill the form	1105289M, 200052095500, KC Kapila, Canal Sec 1, Pannipitiya, Colombo	Data passing in to the database	"Data Insertion Successful"	**PASS**		High		**Ref Pg 28**
REG_ ARW_ 004	**(PR_4)**	URL and Internet Access	Navigate to registration tab and fill the form	552277001208, 602935360v, Kumara Silva, Finance, Gampaha. Gampaha, Visiting Agent	Data passing in to the database	"Data Insertion Successful"	**PASS**		High		**Ref Pg 29**
FS_ ARW_ 005	**FS**	Valid ID, Insurance, Drivers Licenses	1.Register 2.Login 3.Select Image 4.Fill the form 5.Submit	600125, 7.00pm, 12/05/2022, Collided with a Truck, Pavan Weeraratne, 4, 1, Diyatha Rd Battaramulla, 600125	Data/Img passing in to the database	"Data And Img Insertion/ upload Successful"	**PASS**		High		**Ref Pg 30**
UI_ ARW_ 006	**UI**	Previous Info.	1.Enter Complain No 2.Fill the Form 3.Submit	600125, 7.00pm, 12/05/2022, Collided with a Truck, Pavan Weeraratne, 3, No, Diyatha Rd	Update Previous Info	"Data Updated"	**PASS**		Low		**Ref Pg 31**

UMI ARW 007	**UMI**	Img format type	1.Select Img 2.Rename Img with ComplainNo	Renamed With 600125	Img Uploaded Seen Through a grid view	"Uploaded"	**PASS**		Low		
VI_ ARW_ 008	**VI**	Data In the database	Grid View Auto Updating Info	Retrieved Data	Grid View With data/img details		**PASS**		High		**Ref Pg 32**
VIMG_ ARW_ 009	**VIMG**	Images In the database	1.Enter desired Img Id 2.Click show button	Img Id- 1020, 1021, 1022	Img Displayed Through The Img window		**PASS**		High		**Ref Pg 33**
UPI_ ARW_ 010	**UPI**	Previous Info.	1.Enter SSN No 2.Fill The form 3.Submit	884701, LG Liyanasinghe, Marawilla, Puttalam, Inquiry Seargent	Update Previous Info	"Data Updated"	**PASS**		Low		**Ref Pg 31**
URDA_ ARW_ 011	**U_ RDA_ I**	Previous Info.	1.Enter SSN No 2.Fill The form 3.Submit	11052889M, KWL Kapila, Canal Sec Part 8, Pannipitiya, Colombo SL	Update Previous Info	"Data Updated"	**PASS**		Low		**Ref Pg 31**
SR_ ARW_ 012	**SR**	Auth User	1.Enter SSN 2.Fill The form	552277001208, 600125, Approved, (Copies Reg details, NIC, Birth Certificate)	Results sent	"Cases Analyzed"	**PASS**		High		**Ref Pg 34**
VCR_ ARW_ 013	**VCR**	Login		Data From The database	Grid View results		**PASS**		High		**Ref Pg 34**
LG_ PUB_ ARW_ 014	**PUB**	Register	Enter NIC, Name And Ins.No	602293150v, Iraj Ashoka, CCV200125F		" Permission Granted "	**PASS**		High		**Ref Pg 35**
LG_ POL_ ARW_ 015	**POL**	Register	Enter SSN, Name And Dep_Loc	884701, LG Liyanasinghe, Puttalam		" Permission Granted "	**PASS**		High		**Ref Pg 35**
LG_ RDA_ ARW_ 016	**RDA**	Register	Enter SSN, Name And Brn_Name	1105289M, KWL Kapila, Pannipitiya		" Permission Granted "	**PASS**		High		**Ref Pg 35**
LG INS ARW 017	**INS**	Register	Enter SSN, Name And Brn_Name	552277001208, Gampaha, Gampaha		" Permission Granted "	**PASS**		High		**Ref Pg 35**

CHAPTER 5

INDIVIGUAL CONSTRIBUTION

NAME	PLYMOUTH ID
D T KIRIELLA	10748147
S M A DHARMASENA	10749195
MP V BANDARANAYAKE	10749841
M Y M YUSRY	10749082
P H N KAVINDYA	10748162
E A Y I EDIRISINGHE	10749143

1. Developing the UI Design and Back End Coding for the Login Process

Firstly, a web form was created by the name of LoginPage.aspx.com and a separate div tag was created, under which button controllers and text boxes were included separately for the Names, NIC number, Branch name, Job title etc. Since separate div tags were created under each button controller, the back-end coding was done. Basically, during the click event, the inserted values in the text boxes are assigned to a specific variable and that variable will act as a search key and search through the database (SQL Server) for the relevant information inserted during the registration process. Once the specific values are found, the user will be able to login to the main page/ dashboard. The technique used in this process is known as Sessions.

"A SQL session is an ***occurrence of a user interacting with a relational database using SQL commands****. When a user initially connects to the database, a session is established. A session may be invoked by either direct connection to the database or through a front-end application."*

There are 4 button click events, namely, for the public, police, Road Development Authority, and the Insurance companies. Once the program is run, the users will be able to see the UI design for the login process and separately once the relevant information are inserted, user will be able to proceed forward.

2. Developing the UI Design and the Back-end coding for the Registration process

Here first of all a web form was created by the name of RegisterPage.aspx.com and a separate div tag was created, under which button controllers and text boxes were included separately for the SSN, NIC No., Name, Dep_Name, Branch_ District, etc. Since separate div tags were created under each button controller, the back-end coding was done. In the event of click the data inserted in the text boxes are assigned to a specific variable and that variable will store the data in the database (SQL Server) in the relevant registration data table. Once the data is stored inside the table the user gets registered to the system.

There are 3 button click events for the Police, Road Development Authority and the Insurance companies. When the program starts to run the users can see UI designs for the 3 types of user registration processes separately. When the user registers themselves for the relevant category, they can login to the system entering those details in the login process.

3. Police and RDA UI Design and Back-end coding

Here first a web page is created to show the images uploaded and the complaints numbers in a table by the users. This is where the officers can see all the reports of accidents and statistics of the data of accidents. The officers can retrieve the images and when they type in the image ids. For that, a Select SQL command is used to retrieve the images of those specific images. Process Request method is used to get those images. The data can be displayed in a pictorial form as in a chart as well. The user registration info and accident details can be retrieved from the database as well. For those SQL commands are used.

4. Database Creation with Tables

For the database we have selected the SQL Server database. The Database is consisting of 7 tables. Images in one table. Police, Road Development Authority, Public and Insurance user details in tables and Data Analysis information in a separate table as well. For Accidents recorded by the public a separate table is being created to store details of them as well. For each table there is an unique primary key. For example, NIC No and SSN in Public and Police Tables for examples. The database is connected to the Project using the connection string of the database. The Connection string is called for each instance where it is used in the project. The tables are fed with information and when an user registers themselves with the system the details provided will be stored in the relevant table of the user.

5. Image Upload and Retrieval and Accident Recording UI designs and backend coding

When reporting an accident, the public has to upload images of the accident scene for better clarity. For that inside a separate div tag, the grid view tag is used. So that they can upload the images and the data of the images can be seen after uploading it. Such as the Image name and the Image ID. When the image has uploaded the image and the details of the image will be stored in the database in the Image table. For that SQL command INSERT is used in the code. Binary Reader class is used to simplify the reading of the primitive data of the image. Other than the Image Upload A webform is created to enter the accident details as well. For that Text, boxes have been used to enter details. At the end of the form, a button is inserted to submit the details. On button click event a SQL command will run to insert the data into the database. Incident Time, date, No. of passengers, and Complaint Number are a few of the details to be filled in the form. Details entered will be recorded in the Accident details table in the database. A try-catch block is used to check if all the fields are filled out before submitting. If all are not filled a message will be shown as “Please Insert Data to each field” or else as “Data Insertion is Carried out Perfectly”.

6. UI designs and back-end coding for User accounts

First of all a web form is created with the name AccountPol.aspx Then text boxes are added to get the details of the Police. Through this form, the Users can update their details. Here Text boxes are added to enter the details of the police. After filling out the form an update button is at the bottom of the page to submit the details. The data will be passed to the database and get updated on the Button Onclick method. An SQL Update query is used to perform this task. The same procedure is done for the Road Development Authority as well. The fields are validated as well to increase efficiency and reduce data abnormality. A try-catch block is used to identify whether the form is filled correctly. If yes, a message will be shown as “Information updating is successful” else a message will be shown as “Exception occurred”.