**CHAPTER-1**

**INTRODUCTION**

**CHAPTER -1**

**INTRODUCTION**

* 1. **Overview of the Project**

The main goal of this project “**VEHICLE INSURANCE CLAIMING AND MANAGEMENT**” is to completely automate the insurance claim for customer and maintain insurance claims, new insurance, and tracking of insurance claim requested by the customer for various branches that are included within the company.

In proposed application the employee can fill the form online and submit it so that the form is sent to CPD (Claim Processing Unit) through Internet. At CPD, the form needs to be checked automatically by a program which will compute the amount that needs to be reimbursed to the employee for the treatment undertaken.

The Claim processing system is an automated insurance facility for all walks of life. The root of the system starts from registration of the customer. This follows with a series of procedures like inspection and other verifications which ultimately end up with either registration or rejection.

This system is enhancing the facilities provided to the customers by converting it into an automated and online system. The employers can directly register under the Insurance firm online and as far as the rest of the process is concerned, the entire processing of transactions is automated. The access of the data is restricted to un-authorized users of the head office and the branch office. And the official details are kept confidential.

The time lag involved in the manual system is completely wiped out. Registering, declaration by the employees and ultimately the claim, payment etc. is computerized. Retrieving information and follow up about each aspect becomes easier. The Objective of this system is to provide the functionalities to the policyholders, which is helpful to claim their auto accident and damages of the vehicle online. It facilitates companywide integrated system covering all functional areas of a claim processing.

**MODULES:**

1. Administrators
2. Insurer
3. Surveyor
4. Customer (Policy Holder)
5. Authentication
6. Search
7. Reports–

**Administrator:**

Admin user holds high privilege to manage all the activities of this application. Admin can edit customer policy holder detail, edit customer profile, change customer plans, assign roles to employees, and suspend users from their regular activities. Admin rights are reserved by their login credential, so every admin login will be administered by the application.

**Insurer:**

Insurers are employees working in the insurance office, they approach customers, explain policy details and make them to register with company. They will be responsible person to collect renewal amount and insist the customer to make payment in prior.

**Surveyor:**

Surveyor is also an employee working in the concern whom visits customer vehicle while they buy a new vehicle, or in case of an accident. They report to admin about vehicle and its worth and owner details in the case of new purchase or estimates the damage and insurance claiming details to admin as well as to customer. They will explain all these details to customers and make them to convince upon accidental claiming.

**Customer:**

Customers are policy holders of the company they will register policy and renew policy with the company. In case of accident the customer can apply for accident claiming via online with their customer policy ID. In CPD, the details will be verified against given policy ID and claiming procedure and eligibility will be considered against customer.

**Authentication:**

Authentication process authorizes the logon user by their login credentials. Every user has different type of operations and roles in this application to work on. The application will show menus based on logon user, and the user has limited options based on their privileges.

**Search:**

The admin or user can search about their insurance details. Admin or Insurer or surveyor has privilege to search user profiles, and admin possess rights to edit and update policy details. Customer can navigate and knows about their policy and accidental claiming procedure.

**Reports:**

Reports will be generated from admin side based on customer, claiming, date wise, and actions taken by the admin in case of particular customer scenario. If need, the admin can take print out from the generated report.

**1.2 Company Profile**

**TRITON SOFT** was founded in Coimbatore in 2007 on the premise that building commercial software is fundamentally different than building internal IT applications and requires specialized knowledge and skills. We started by building our own Mirco ERP and utility software that we still sell and service in some markets today. Since then, we have deepened our knowledge and now offer our software product skills to other software enabled business who need to builds compelling software offerings to attract and retain customers and revenues. We combine our strong product development knowledge with a business model built on honesty and integrity so that software enabled businesses can rely on Triton Soft to deliver winning software.

**ENTERPRISE SOLUTIONS**

**Complete Solution for the industry**

ERP is an effective way to integrate data and business processes across the organization. Today it has become defacto standard to manage business operations. New generations ERP systems not only address the need of a transaction based systems but also bring functions like Business Intelligence, Scorecard, MIS, helping management in right decision making. At Triton Soft we recognize this potential of ERP and have created a team of functional and technical experts, who can help business organization starting from product selection to implementation.

**BUSINESS INTELLIGENCE**

**Provides Faster and Better Results**

Triton Soft Business Intelligence gives you the information you need, when you need it, in the right format. By integrating data from across your enterprise and delivering self-service reporting and analysis, IT spends less time responding to requests and business users spend less time looking for information. Triton Soft Business Intelligence also offers an integrated, robust and flexible presentation layer for the full breadth of Triton Soft Analytics capabilities, including statistics, predictive analytics, data and text mining, forecasting, and optimization – all integrated within the business context for better, faster decision making.

* Mobility

Any Time, Any Place, Any Where

Enterprise Mobility offerings make it easy to build and deploy mobile applications that are differentiated by rich user experience and works across multiple mobile platforms with seamless enterprise integration. These solutions are flexible, adaptable and targeted to maximize business efficiency while reducing integration expenses.

* Web 2.0

Electrocare provides individual web 2.0 application a centralized console for managing application clusters and distributed-application environments. Our platform provides advanced features like Dashboards, Alerts, Graphs, Connectors, Security, SNMP support and more. Our mission is to provide you a platform to centrally manage and monitor your web 2.0 application environments.

**CHAPTER-2**

**SYSTEM STUDY & ANALYSIS**

**CHAPTER -2**

**SYSTEM STUDY & ANALYSIS**

* 1. **Existing System**

Previously the company admin manages all records in excel format. The customer detail and their policy details are kept in excel format. Survey reports and claiming details will be recorded in hand written format. Admin have office setup with huge manpower to maintain customer records and maintenance of official records.

Admin has to take all controls and responsibilities in order to channelize the works. So every orders and customer relationship needs to be acknowledged by admin and employee should get confirmation from management to proceed further.

* 1. **Disadvantages of Existing System**
* Record navigation and finding customer renewal information gets delayed in manual work
* Excel sheet maintenance even though convenient, it leads to mistake when the customer base in high in number
* Sometimes customer relationship may be unsatisfactory due to long process involved in huge manpower
  1. **Proposed System**

The proposed system will be developed using ASP.NET and SQL Server as back end to run from anywhere. The employees will be classified by admin, surveyor, and insurer to manipulate the whole application. Customer policy login, insurance renewal, accidental claiming, customer support and customer queries will be managed in online itself from anywhere and anytime. The application prepares sophistication phase to work on and login credential to prevent unauthorized access.

* 1. **Advantages of Proposed System**
* The proposed system can be accessed from anywhere to minimize the workload and retains customer relationship
* Improves customer relationship by providing online complaint posting and customer query section to give suggestions and clarifications from admin
* Policy calculation, policy renewal calculation will be integrated with policy calculator to calculate customer charges instantly
* Improves record navigation and access their details as quick as possible by store details in SQL and have full- fledged report generation facility

**CHAPTER-3**

**SYSTEM SPECIFICATION**

**CHAPTER -3**

**SYSTEM SPECIFICATION**

* 1. **Hardware Requirements**

Processor : Pentium IV

Clock speed : 2.4 GHz

RAM : 512 MB

HDD : 80 GB

CD Drive : 52x Reader

Pointing device : Scroll Mouse

Keyboard : 101 Standard Key-board

* 1. **Software Requirements**

Front end : ASP .NET

Back end : SQL Server 2000

Operating System : Windows XP

* + 1. **Front End**
       1. **Overview of visual studio .NET 2005**

Visual Studio.net offers a rich set of development tools that hide a lot of the complexity inherent in working with the .NET Framework. This reduces time spent both learning the product and developing applications. When Visual Studio .NET is installed, the CLR and the .NET Framework classes are also installed. Visual Studio.NET defines four different types of applications that can be built on the .NET Framework:

**Console**

These are command line applications that are built without a graphical user interface (GUI) and which can be executed from, and interact with, a DOS Window. In addition to their intrinsic value, they can be particularly useful for testing sections of code to ensure proper functionality.

**Windows Forms**

These are rich-client applications that are built around a GUI similar to desktop applications written using Visual FoxPro 7.0. New features for client applications developed in the .NET Framework include visual inheritance, code-free resizing, automatic control updates, and new controls.

**Web Forms**

These are browser-based applications that are built around a GUI with special controls. Web form applications are written using ASP.NET.

**XML Web services**

These are applications that define XML Web services that can be consumed by other XML Web services or applications either on a local network or exposed on the Internet. Because they are based on HTTP and XML, the information they transfer can be passed through firewalls.

* + - 1. **Overview of ASP .NET**

ASP.NET is used to create programmable Web pages as part of Web form applications. Not only can ASP.NET produce dynamic Web pages, but it can also tailor them to the browser being used by the user. Development in ASP.NET is much easier because it contains a wide range of object-oriented server-side controls ready for use in applications. ASP.NET actually supports four different types of control:

**HTML Server Controls**

These are programmable HTML elements that are exposed to the server. HTML server controls expose an object model that maps very closely to the HTML elements that they render.

**Web Server Controls**

These controls have more built-in features than HTML server controls. Web server controls include not only form-type controls, such as buttons and text boxes, but also special-purpose controls, such as a calendar. Web server controls are more abstract than HTML server controls in that their object model does not necessarily reflect HTML syntax.

**Validation Control**

These are controls that incorporate logic and that can be attached to an input control to test what the user enters. Some validation controls check for a required field, a specific value or pattern of characters, that a value falls within a specified range, and so on.

**User Control**

These are custom controls created as Web Forms pages. Web Forms user controls can be embedded in other Web Forms pages and provide an easy way to create menus, toolbars, and other reusable elements.

Internet Information Server (IIS) is the core window NT service provides Internet services. IIS is the underpinning that provides information publishing capabilities on the internet. IIS is now its fourth version. IIS 1.0 become available in 1995, but it was IIS 3.0 that really caught on in the summer of 1996. IIS 4.0 became available as a public beta in June 1997. With IIS 4.0 Microsoft introduces a number of new features. Many of the application for IIS 4.0 will be hardcore c++ components held together by VBScript.

Microsoft Internet Information Server (IIS) has seen many changes with the Version of IIS on the Windows XP Professional FD. This IIS is only applicable in systems which operate on Windows XP Professional OS. This new IIS 5.1 enables the following

* World Wide Web (WWW) and File Transfer Protocol (FTP) servers.
* TCP/IP based electronic mail using standard SMTP and POP3 protocols.
* Microsoft Management Console (MMC) that provides a central administration point for IIS and other WinNT management utilities.
  + - 1. **Overview of HTML**

HTML stands for Hyper Text Markup Language. Hypertext is ordinary text that has been dressed up with extra features, such as formatting, images, multimedia, and links to other documents. Mark up is the process of taking ordinary text and adding extra symbols. Language is actually a key point to remember about HTML. HTML is a universal language for classifying the function of different sections of a document. It is neither a page-layout language nor a printing language. This allows documents to be displayed on many different kinds of platforms. HTML is flexible to work on the website with their variety of commands.As a formatting language, HTML utilizes SGML (Standard General Markup Language) and Document Type Declarations (DTD). SGML document has three main parts. The first part defines the character set to be used and tells which character i that set distinguishes text for markup tags. Markup tags specify how the viewer application or browser should present the text to the user.

The second part specifies the document and states which markup tags are legal. The third part called the document instance contains the actual text and markup tags. Because there is no requirement that the three parts of an SGML document reside in the same physical file, we can concentrate on the document instance. The web pages created are document instances.

**3.2.2 Back End**

Microsoft SQL Server 2000 includes powerful features to support international operations and environments. Extensive multilingual features make SQL Server 2000 a compelling database product and applications platform. This article provides a complete overview of how to use these features in a global context. This article is not limited to a list of features but also will explain how international/multilingual requirements can affect many aspects of a project.

**ACCESSING DATA WITH ASP.NET:**

ASP.NET includes data access tools that make it easier than ever for you to design sites that allow your users to interact with databases through Web pages.

The .NET Framework includes two data provider for accessing enterprise databases: the .NET Framework Data Provider for OLE DB and the .NET Framework Data Provider for SQL Server. This section focuses on accessing SQL Server (version 7.0 or later) databases using the .NET Framework Data Provider for SQL Server, but you can adapt the code examples to other databases with only minor changes.

**To access SQL SERVER 2000 databases from ASP.NET**

1. Create a database connection using the SQL Connection class.
2. Select a set of records from the database using the SQL Data Adapter class.
3. Fill a new Data Set using the SQL Data Adapter class.
4. If you are selecting data from a database for non-interactive display only, it is recommended that you use a read-only, forward-only SQL Data Reader (or OleDbDataReader for non-SQL databases) for best performance. When using a SQL Data Reader, select the records using a SQL Command query and create a SQL Data Reader that is returned from the SQL Command object's Execute Reader method. In some cases, such as when you want to sort or filter a set of data, you might also want to create a new Data View based on a Data Set for the desired table.
5. Bind a server control, such as a Data Grid, to the Data Set, SQL DataReader, or DataView.

**SQL Connection:**

A SQL Connection object represents a unique session to a SQL Server data source. In the case of a client/server database system, it is equivalent to a network connection to the server. SQL Connection is used in conjunction with SQL Data Adapter and SQL Command to increase performance when connecting to a Microsoft SQL Server database. For all third-party SQL server products, as well as other OLE DB-supported data sources, use OleDbConnection.

**SQL Data Adapter:**

The SQL Data Adapter serves as a bridge between a Data Set and SQL Server for retrieving and saving data. The SQL Data Adapter provides this bridge by mapping Fill, which changes the data in the Data Set to match the data in the data source, and Update, which changes the data in the data source to match the data in the Data Set, using the appropriate Transact-SQL statements against the data source.

**Data Set:**

The Data Set, which is an in-memory cache of data retrieved from a data source, is a major component of the ADO.NET architecture. The Data Set consists of a collection of Data Table objects that you can relate to each other with Data Relation objects. You can also enforce data integrity in the Data Set by using the Unique Constraint and ForeignKey Constraint objects. For further details about working with DataSet objects, see Creating and Using Datasets.

**SQL Data Reader:**

TheSQLDataReader class is the SQL server version of the OleDbDataReader. Only one SQLDataReader per associated SQLCommand may be open at a time. & any attempt to open another will fail until the first one is closed. The ExecuteReader is one of the methods of SQLDataReader. The ExecuteReader is used to execute the statements.

**SQL Command:**

The SQLCommand class represents an SQL Statements or stored procedures for use in a database using SQL Server.The CommandText is used to gets/sets the SQL Statement for this command to execute. The CommandType is used to gets/sets the type of CommandText property.

**CLIENT/SERVER ENVIRONMENT**

**Benefits offered by client/server computing:**

* Increased end user productivity because of flexible data access.
* Graphically oriented, highly interactive user interface.
* Increased developer productivity through usage of easy to use easy tools.
* Improved access to information because of networking.
* Better control of corporate data through centralized data, systems & network management.
* Easier maintenance of application & data.
* Protection of hardware investments by making use of existing installations of Hardware, software &network and at same time getting maximum leverage out of the available desktop technology.

**CHAPTER-4**

**SYSTEM DESIGN AND DEVELOPMENT**

**CHAPTER -4**

**SYSTEM DESIGN AND DEVELOPMENT**

* 1. **Input Design**

Input designs are done in such a way that it caters to the requirement of the end user. Extensive care has to be taken to ensure the validity of the input data. Error messages will be displayed immediately.

The input design involves providing an interface between the user and the computer system. It is essential to design the interface in such a way that it makes the user entry easy and makes input free from errors. The design of an input focuses on gathering the sufficient data, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple and user friendly.

* 1. **Output Design**

In any system, results of processes are communicated to the users and to the user and to the other systems through output. In output design, it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct use source of information to the user. Efficient and intelligent output design improves the system relationships with the user and helps in decision-making. The output includes reports in specific formats, displays of enquiries as well as simple profile of the database. The output screen is cluster feed and provisions are made for multiple page browsing.

* 1. **Table Design**

Table Name: tblUserReg

Primary key: userId

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| userId | Varchar | 30 | User Id |
| Gender | Varchar | 7 | Gender |
| Address | Varchar | 50 | Address |
| conNum | Varchar | 15 | Contact number |
| Mail\_Id | Varchar | 50 | Mail Id |
| logId | Varchar | 25 | Login Id |
| Pass | Varchar | 20 | Password |

Table Name: customer\_Claim

Primary key: claimed

Foreign Key: userId, policyNo

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| userId | Varchar | 30 | User Id |
| clmDate | Date/Time | 8 | Claim Date |
| policyNo | Varchar | 10 | Policy Number |
| vehNo | Varchar | 12 | Vehicle Reg. No |
| insCmp\_name | Varchar | 30 | Insured Company |
| clmReason | Varchar | 50 | Claim Reason |
| dmgDesc | Varchar | 100 | Damage description |
| Photo | Varchar | 50 | Accident image |
| Status | Varchar | 30 | Claim Status |

Table Name: branch\_det

Primary key: branchId

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| branched | Varchar | 10 | Branch Id |
| brName | Varchar | 30 | Branch Name |
| Location | Varchar | 50 | Location |
| conPerson | Varchar | 30 | Contact person |
| conNum | Varchar | 15 | Contact Number |
| eMail | Varchar | 50 | Mail Id |
| Address | Varchar | 50 | Address |

Table Name: Emp\_reg

Primary key: empId

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| empId | Varchar | 10 | Employee Id |
| empName | Varchar | 30 | Employee Name |
| branchId | Varchar | 10 | Employee Branch |
| conNum | Varchar | 15 | Contact number |
| Mail\_Id | Varchar | 50 | Mail Id |
| Gender | Varchar | 10 | Gender |
| Address | Varchar | 50 | Address |
| Dept | Varchar | 30 | Employee dept |
| Desg | Varchar | 20 | Employee designation |

Table Name: Customer\_policy

Primary key: policyNo

Foreign Key: cusId

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| policyNo | Varchar | 10 | Policy Number |
| cusId | Varchar | 20 | Customer Id |
| vehNo | Varchar | 15 | Vehicle reg no |
| makeDett | Varchar | 30 | Make and model |
| engineNo | Varchar | 20 | Engine Number |
| Chas\_no | Varchar | 25 | Chasis Numberr |
| mangYr | Date/Time | 8 | Manufacture Year |
| Pol\_date | Date/Time | 8 | Policy Date |
| Iss\_by | Varchar | 30 | Issued By |
| Period | Varchar | 10 | Valid period |

Table Name: Claim\_Approval

Primary key: appr\_id

Foreign Key: policy\_no

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Appr\_id | Varchar | 10 | Approval Id |
| Policy\_no | Varchar | 10 | Policy Number |
| Veh\_Re\_no | Varchar | 15 | Vehicle Register no |
| clmStatus | Varchar | 20 | Claim Status |
| clmAmt | Float | 8 | Approved Amount |

* 1. **ER Diagram**

tblbranch

Have

tblemployee

tblcustomer

tblpolicy

tblcliam

tblcliamstatus

Has

Approves

Has

* 1. **Data Flow Diagram**

**LEVEL 0**

ADMIN

Insurance Claiming

Surveyor

Customer

Insurer

**LEVEL 1**

Reports

Admin

Employee

tblPolicy

tblEmp

tblClaim

tblCustomer

tblSurvey

**LEVEL 2**

Customer

Admin

surveyDet

claimDetails

View Result

View Claims

Claim Process

loginDet

surveyDet

Survey Report

custRegDet

**CHAPTER-5**

**SYSTEM TESTING AND IMPLEMENTATION**

**CHAPTER -5**

**SYSTEM TESTING AND IMPLEMENTATION**

* 1. **Testing**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because web-based systems and applications resides on a network and interoperate with many different operating systems, browsers, hardware platforms, and communication protocols, the search for errors represents a significant challenge for web applications.

The distributed nature of client-server environments, the performance issues associated with transaction processing, the potential presence of number of different hardware platform, the complexities of network communications, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client-server architectures.

**5.2 Unit Testing**

Unit is the testing changes made in an existing or new program. This test is carried out during the programming and each module is found to be working satisfactory. In the registration form user details are stored in database without any change. This module is tested for its input and validated while employee login.

Every module is tested for proper input and the problems faced in inputs are eradicated after tested. For example, in registration module the user id, password and mail id details are said to be mandatory. So the user without having this proper information cannot sign up with this system. At the same time the crop interested to select is made optional, if the farmer is interested then they may select this field.

**5.3 Acceptance Testing**

Acceptance testing generally involves running a suite of tests on the completed system. Each individual test, known as a case, exercises a particular operating condition of the user's environment or feature of the system, and will result in a pass or fail, or Boolean, outcome. There is generally no degree of success or failure. The test environment is usually designed to be identical, or as close as possible, to the anticipated user's environment, including extremes of such. These test cases must each be accompanied by test case input data or a formal description of the operational activities (or both) to be performed—intended to thoroughly exercise the specific case—and a formal description of the expected results.

This project is developed and tested in its each phase and verified for its output. While integrating registration module, employee module, category details module with online test module initially the details are not driven from the database properly. After studied carefully the output of each module is tested in its module and finally tested in renewal module. At present the results produced from this module satisfies the needs of admin and yields correct output.

**5.4 System Implementation**

Implementation is the stage in the project where the theoretical design is turned a working system. The most crucial stage is achieving a successful new system and giving the user confidence in that the new system will work efficiently and effectively in the implementation stage.

The stage consists of

* Testing the developed program with simple data.
* Detections and corrections.
* Creating whether the system is meets user requirements.
* Testing whether the system.
* Making necessary changes as desired by the user.
* Training user personnel.

After the system is implemented and conversion is completed, a review should be conducted to determine whether the system is meeting expectations and where improvements are needed. A post implementation review measures the system’s performance against predefined requirements. It determines how well the system continues to meet performance specification. It also provides information to determine whether major design or modification is required.

A post implementation review is an evaluation of a system in terms of the extent to which the system accomplishes stated objectives and actual project costs exceed initial estimates.

The post implementation study begins with the review team, which gathers and reviews request for evaluation. Unexpected change in the system that affects the user or system performance is a primary factor that prompts system review. Once request is filled, the user is asked how well the system is functioning to specifications or how well the measured benefits have been realized. Suggestions regarding changes and improvements are also asked for.

**CHAPTER-6**

**CONCLUSION**

**CHAPTER -6**

**CONCLUSION**

The **VEHICLE INSURANCE CLAIMING AND MANAGEMENT** is implemented and works in a good manner. The Administrator has the complete control over the system. The Insurer gets the data from the customer and makes the customers to register into the company. The Surveyor is given some preferences to manage the data given by the insurer and customer to process to input data of the insurer. The Verification of the customer’s vehicle and authentication is made by the surveyor into the system. Entire customer details are stored into the database and are used for future references. This helps the complete company to work now in faster manner to reach more customers.

**SCOPE FOR FUTURE ENHANCEMENT**

Enhancements are the perquisite for development of a system. Every existing system has proposed enhancements which make it better and easier to use and more secure. The enhancements that have been proposed for this system are listed here. The project developed is tested in all modules for its consistency and errors are rectified to output the correct results.

In Future, Online payment can be incorporated for the customers to pay the amount directly to the company’s account while the insurer is along with them. This can also be made easier using mobile application for the administrator to monitor the data from anywhere outside the office and to verify the resorts.

**CHAPTER-7**

**BIBLOGRAPHY**

**CHAPTER -7**

**BIBLOGRAPHY**

* Richard Mansfield,”**ASP.Net**”, Hungry minds Publications, First Edition. Theron Millis,”**ASP.Net Databases**”, Wiley-Dreamtech Press, Second Edition.
* Jeffery P Mc Manuns nd Chris Kinsman,”**C#. Net**”, Enterprise Publications, 2006 Edition.
* Dan Appleman,”**C#. Net Programming**”, Paralyph Press and Dreamtech, 2005 Edition.
* Richard E Fairly, **“Software Engineering Concepts**”, Tata Mc-Grawhill Publications, 2000 Edition.
* Futrell Shafer, **“Quality Software Project Management**” Galgotia Publications, 1996.
* Anil Desai,”**SQL Server 2000 Backup and Recovery**”, Mc-Grawhill, 2005 Edition.

**WEBSITES:**

* <http://www.codeproject.com>
* <http://www.aspdotnetheaven.com>
* <http://www.a1vbcode.com>
* [http://www.support.microsoft.com](http://www.support.microsoft.com/)
* <http://www.startaspdotnet.com>
* <http://www.c-sharpcorner.com>
* <http://www.dotnetspider.com>

**CHAPTER-8**

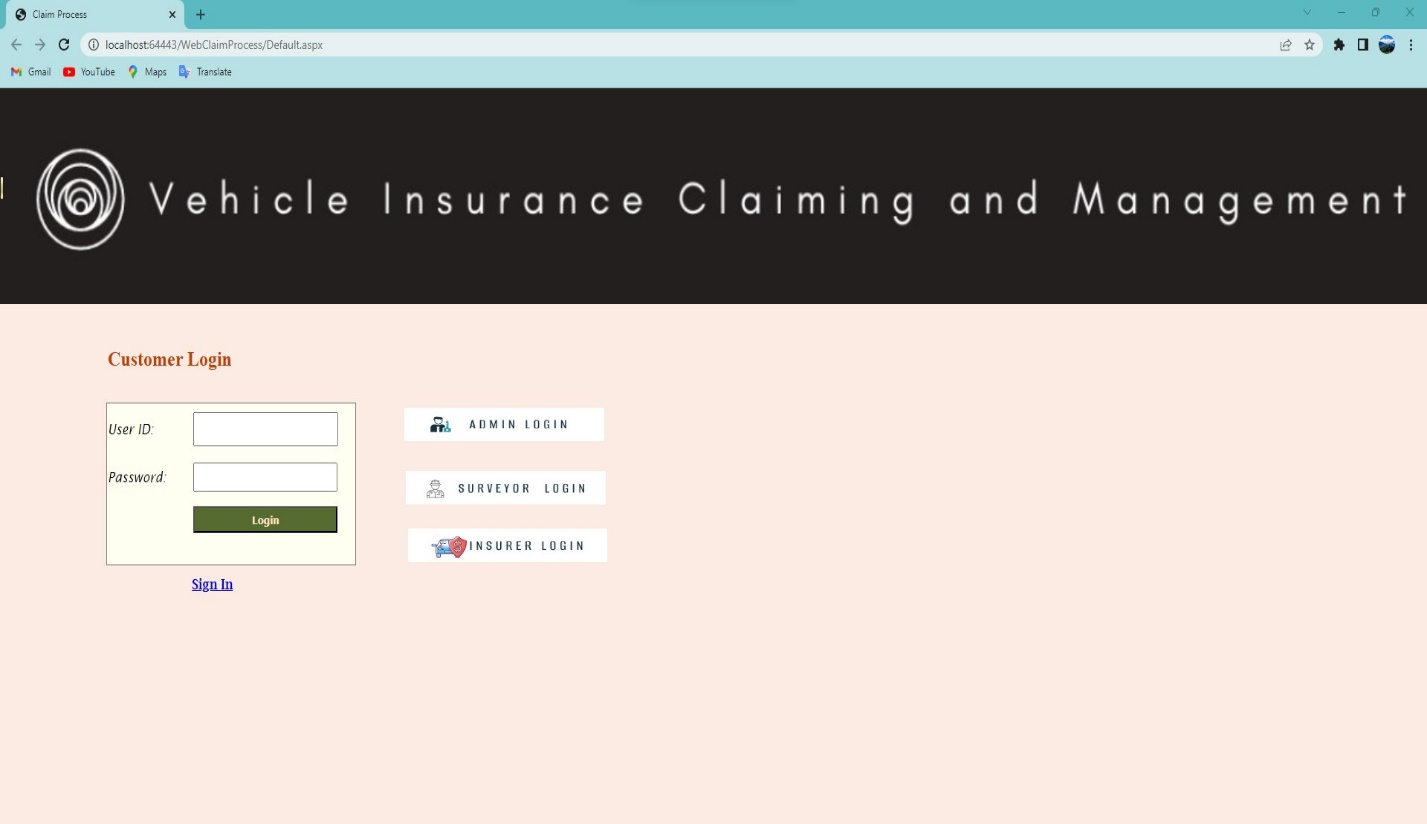
**APPENDICES**

**CHAPTER -8**

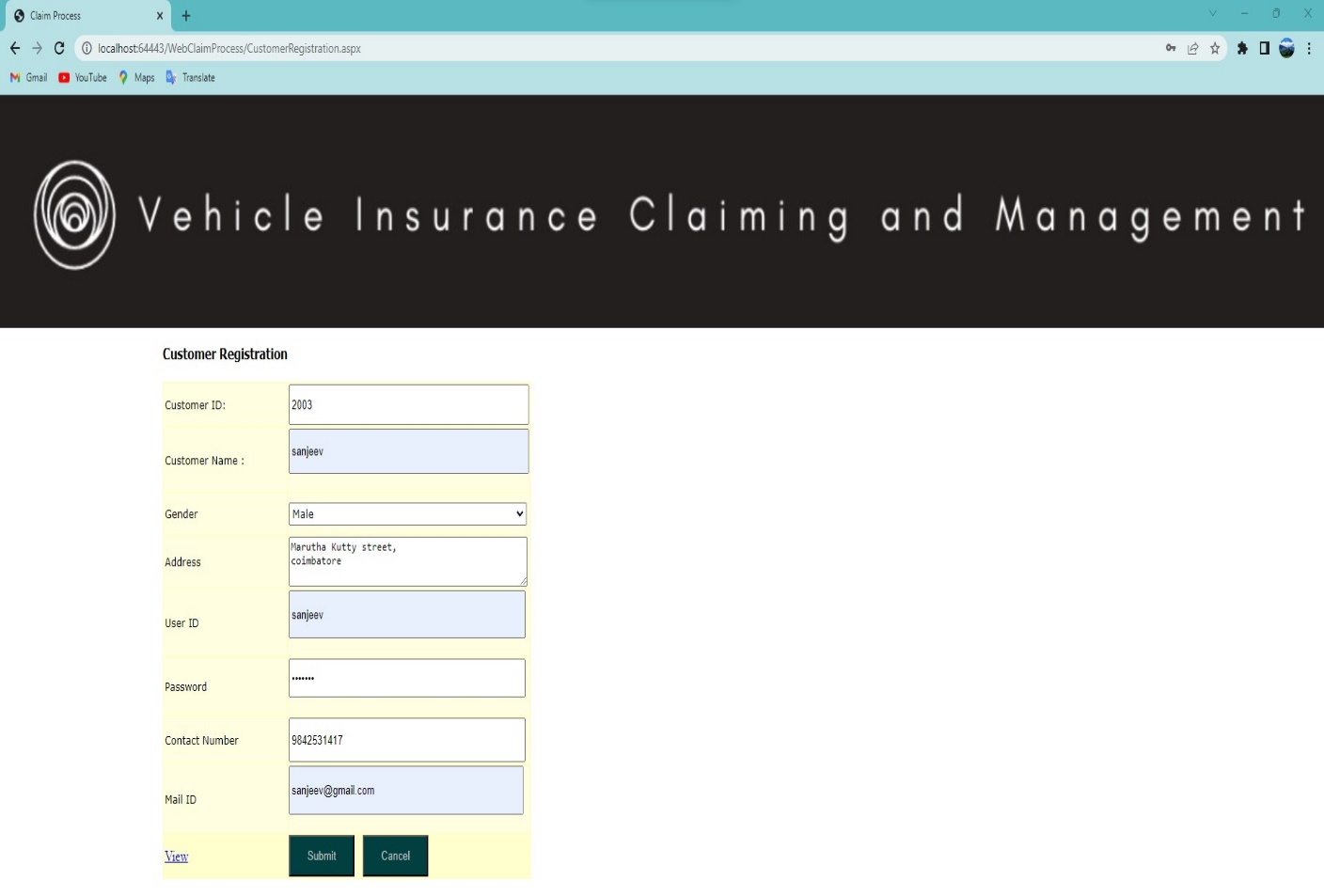
**APPENDIX**

* 1. **Appendix-A Screenshots**

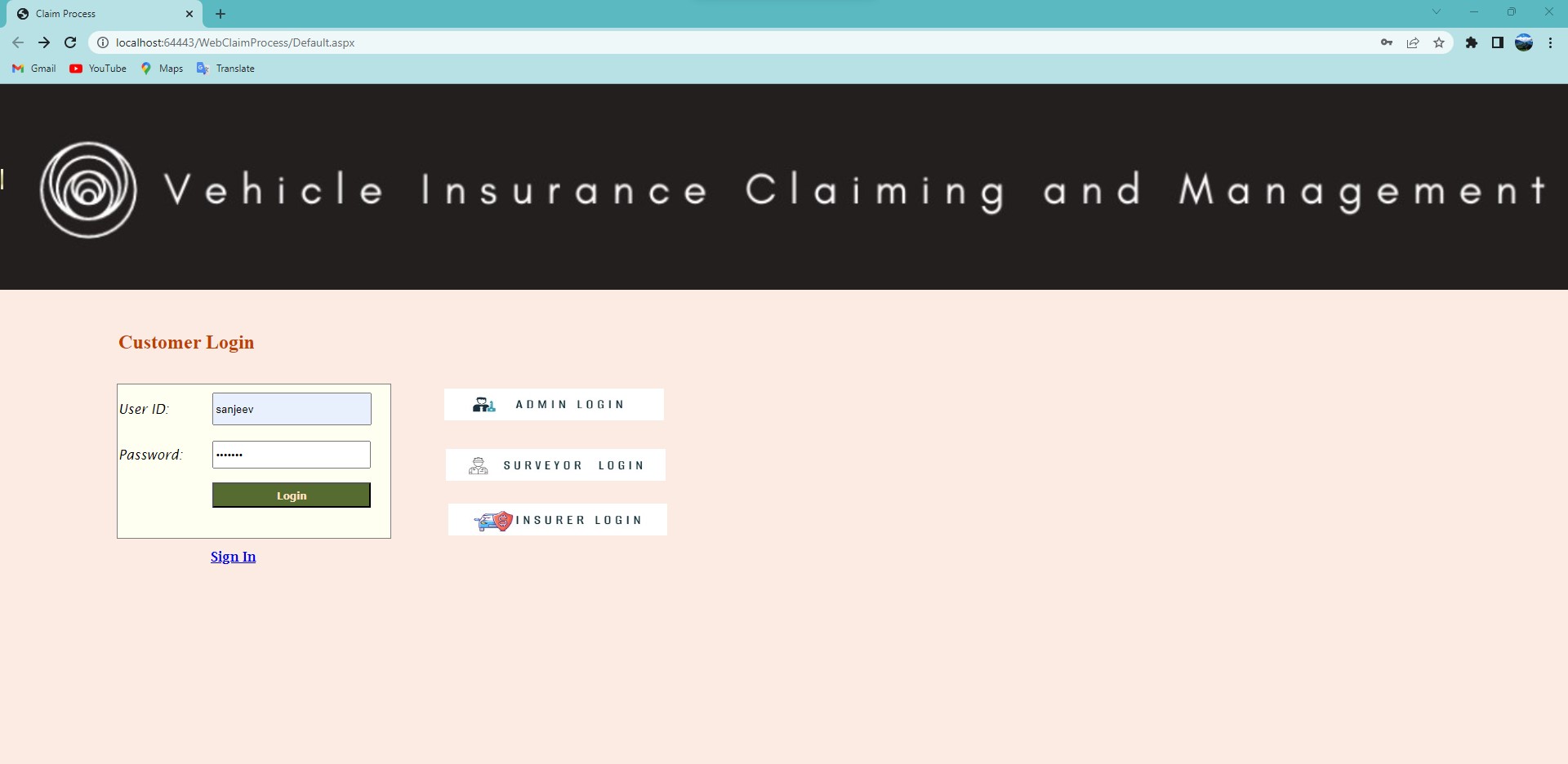
**HOME PAGE:**



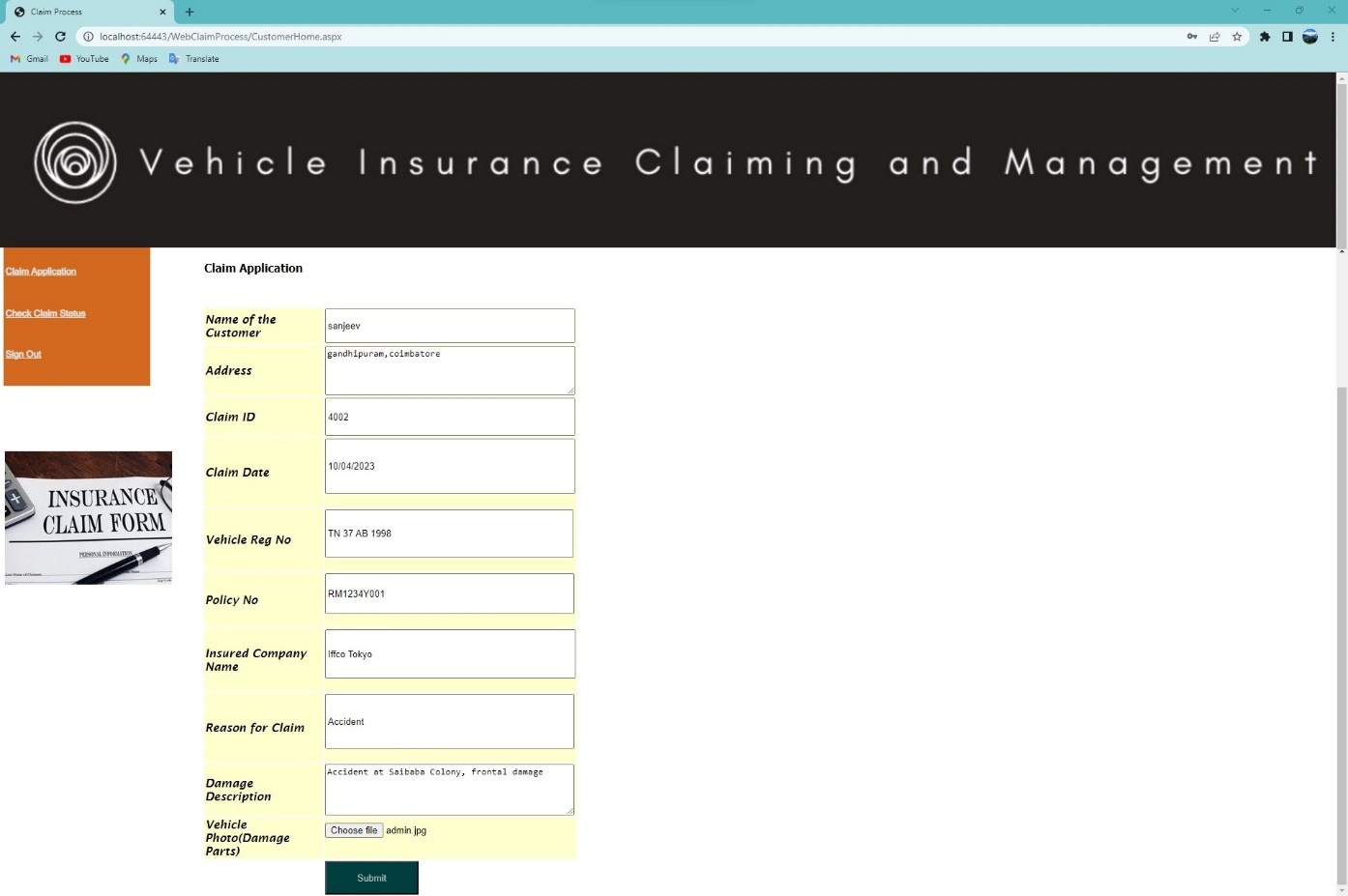
**CUSTOMER REGISTRATION:**



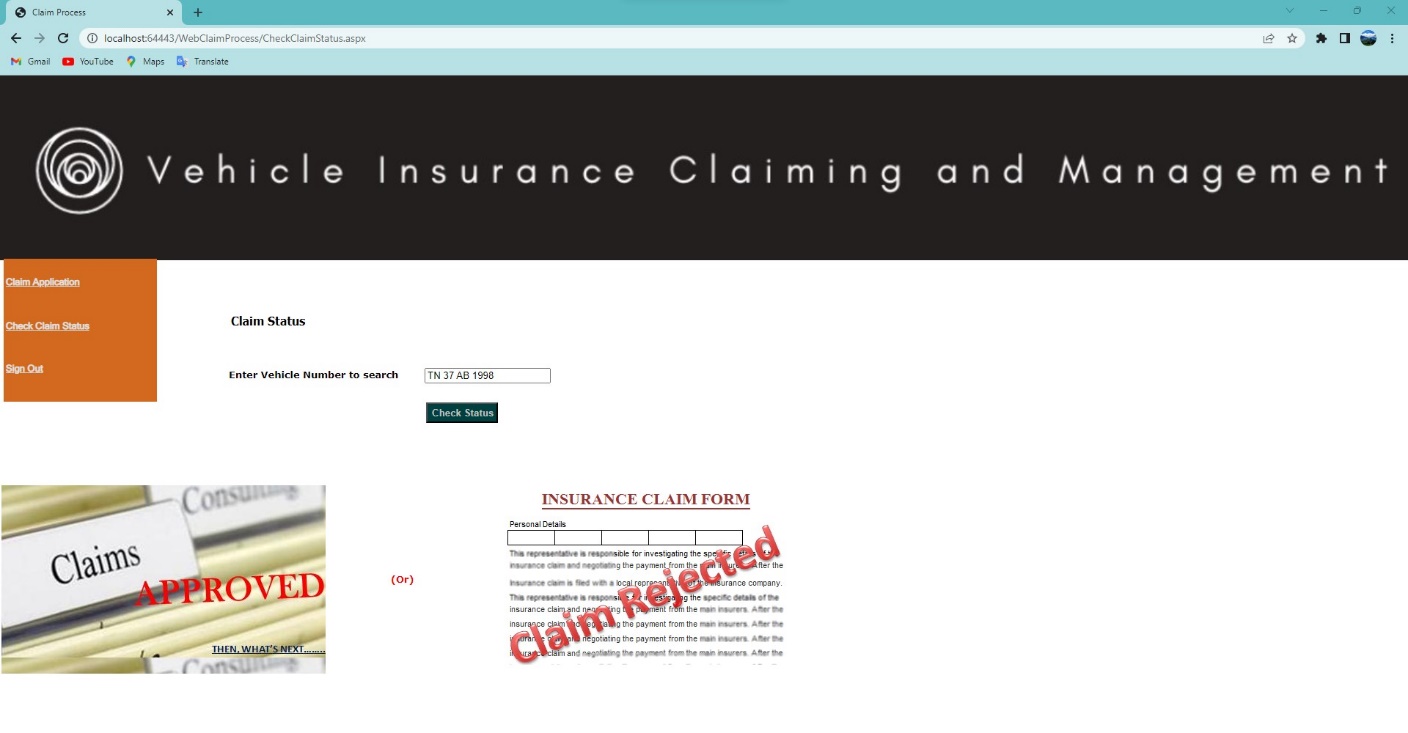
**CUSTOMER LOGIN:**



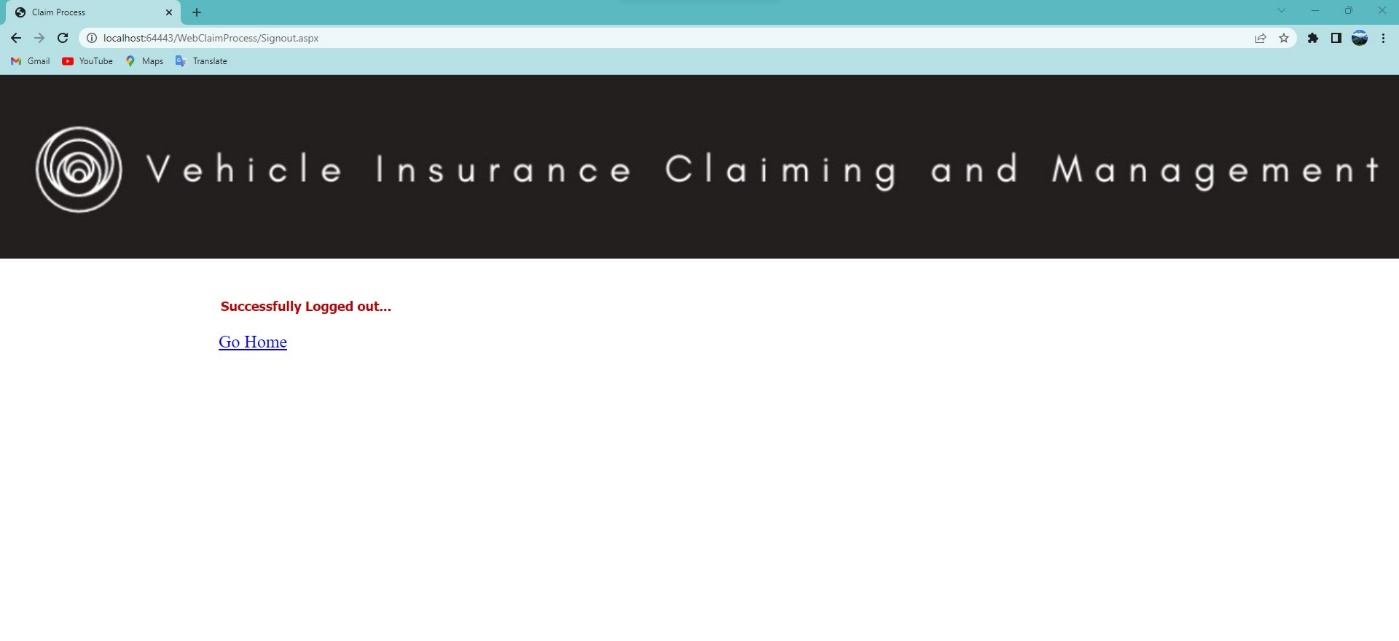
**USER APPLYING FOR CLAIM:**



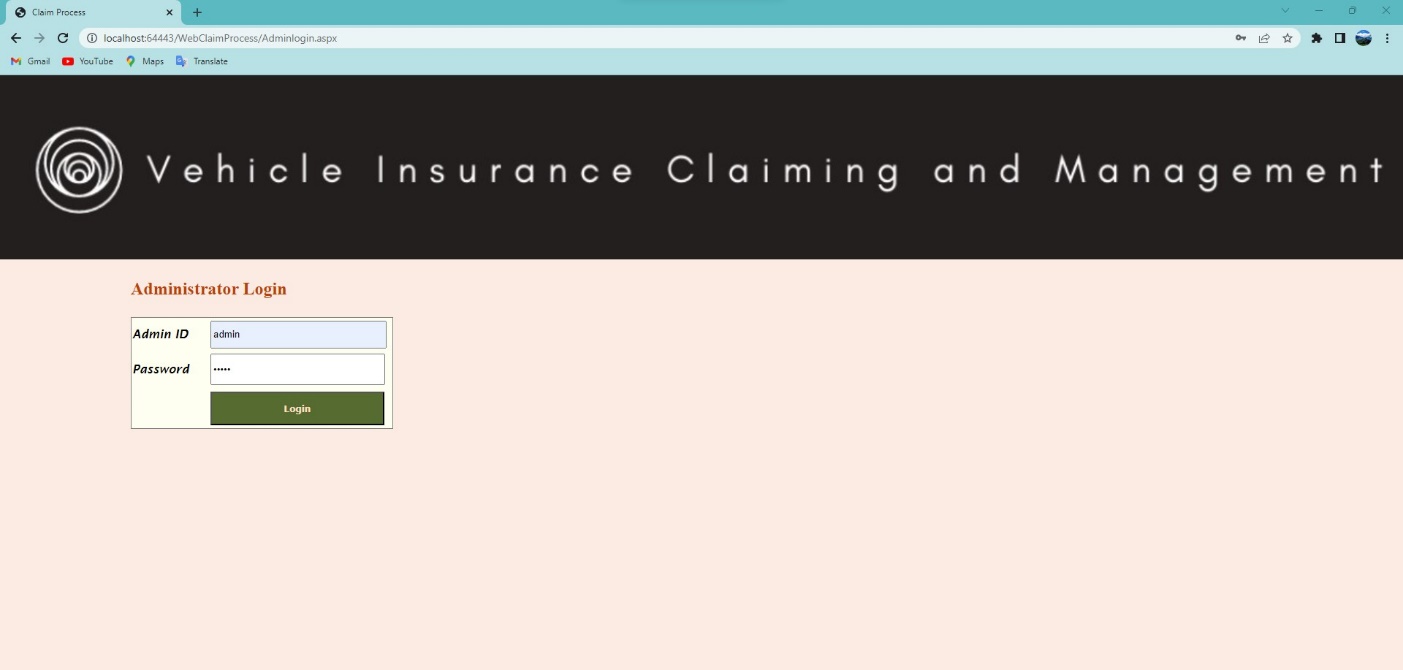
**VEHICLE CLAIM STATUS BY GIVEN REGISTRATION NUMBER:**



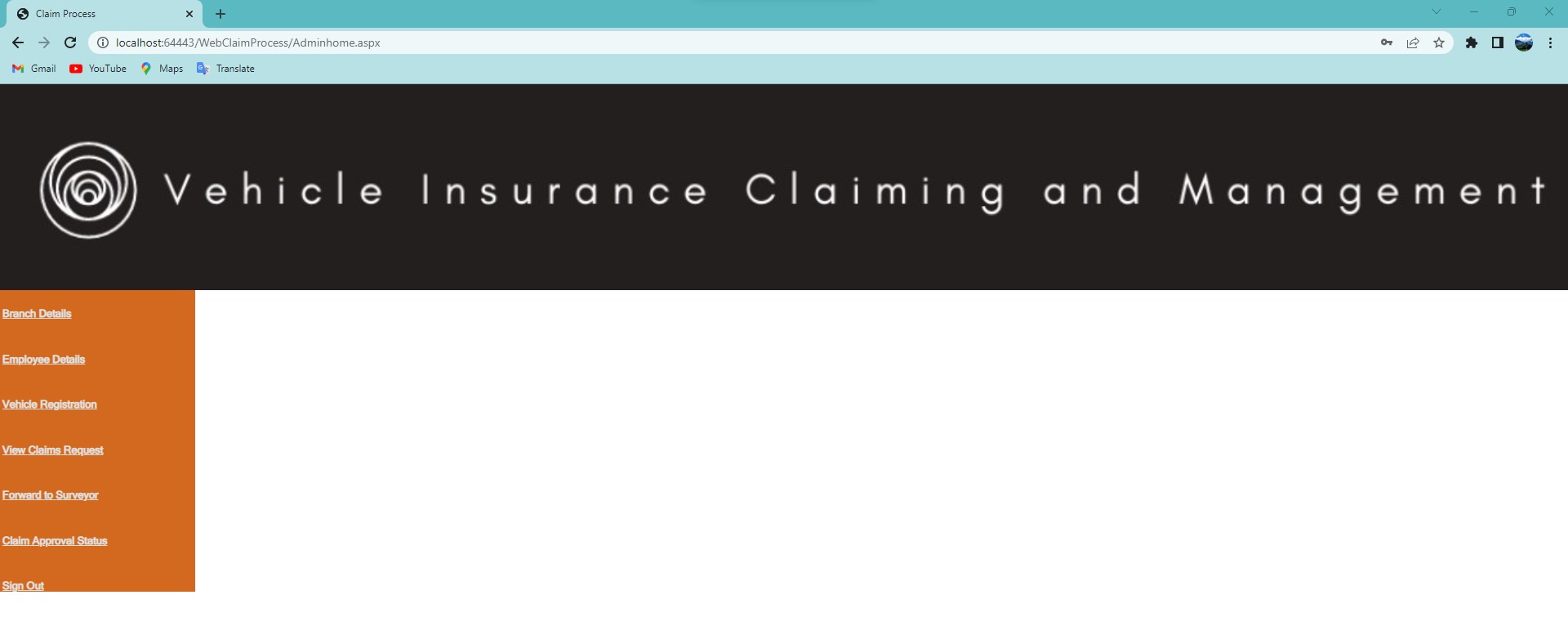
**CUSTOMER SIGNOUT:**



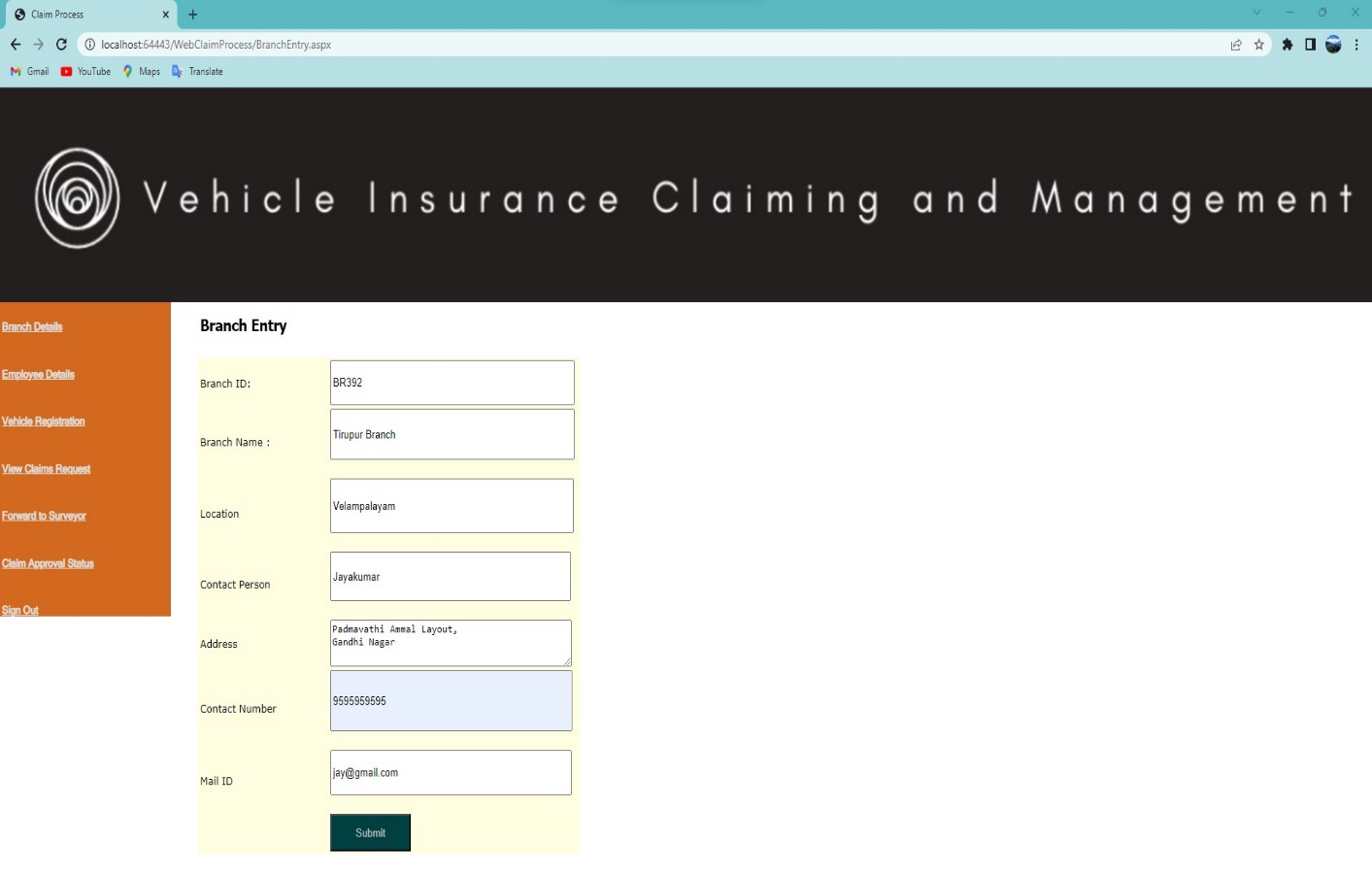
**ADMIN LOGIN:**



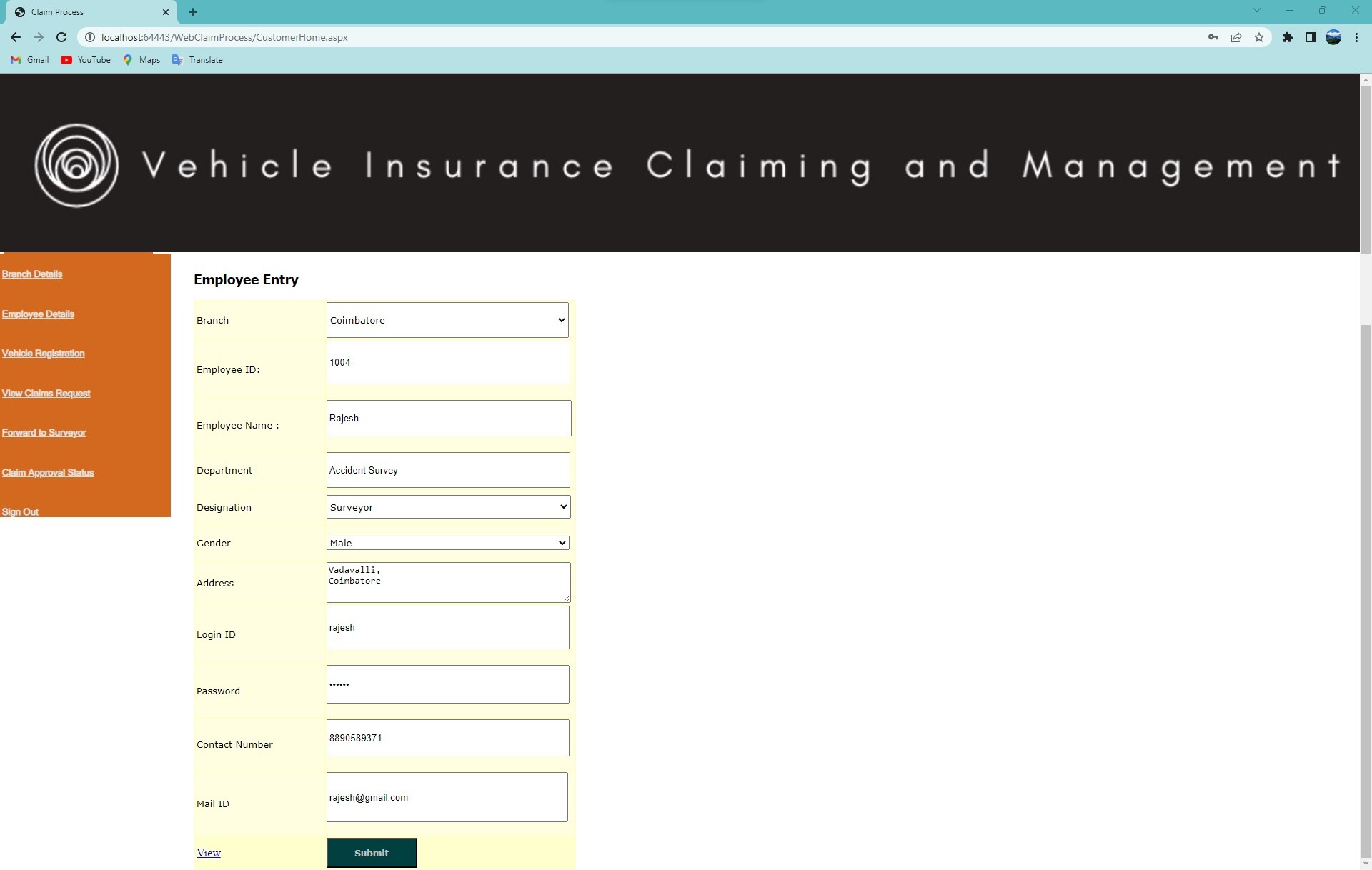
**ADMIN HOME:**



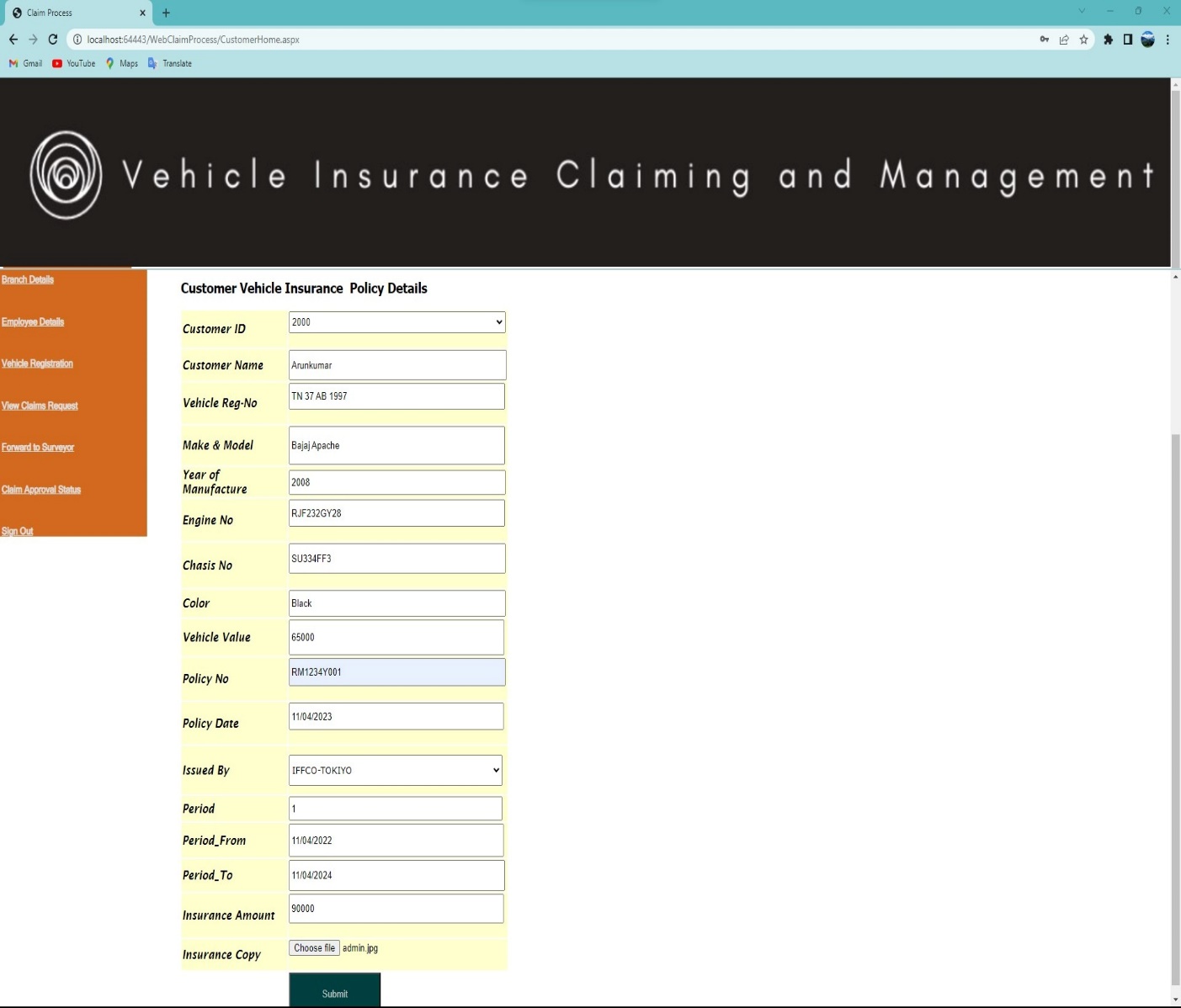
**NEW BRANCH DETAILS:**



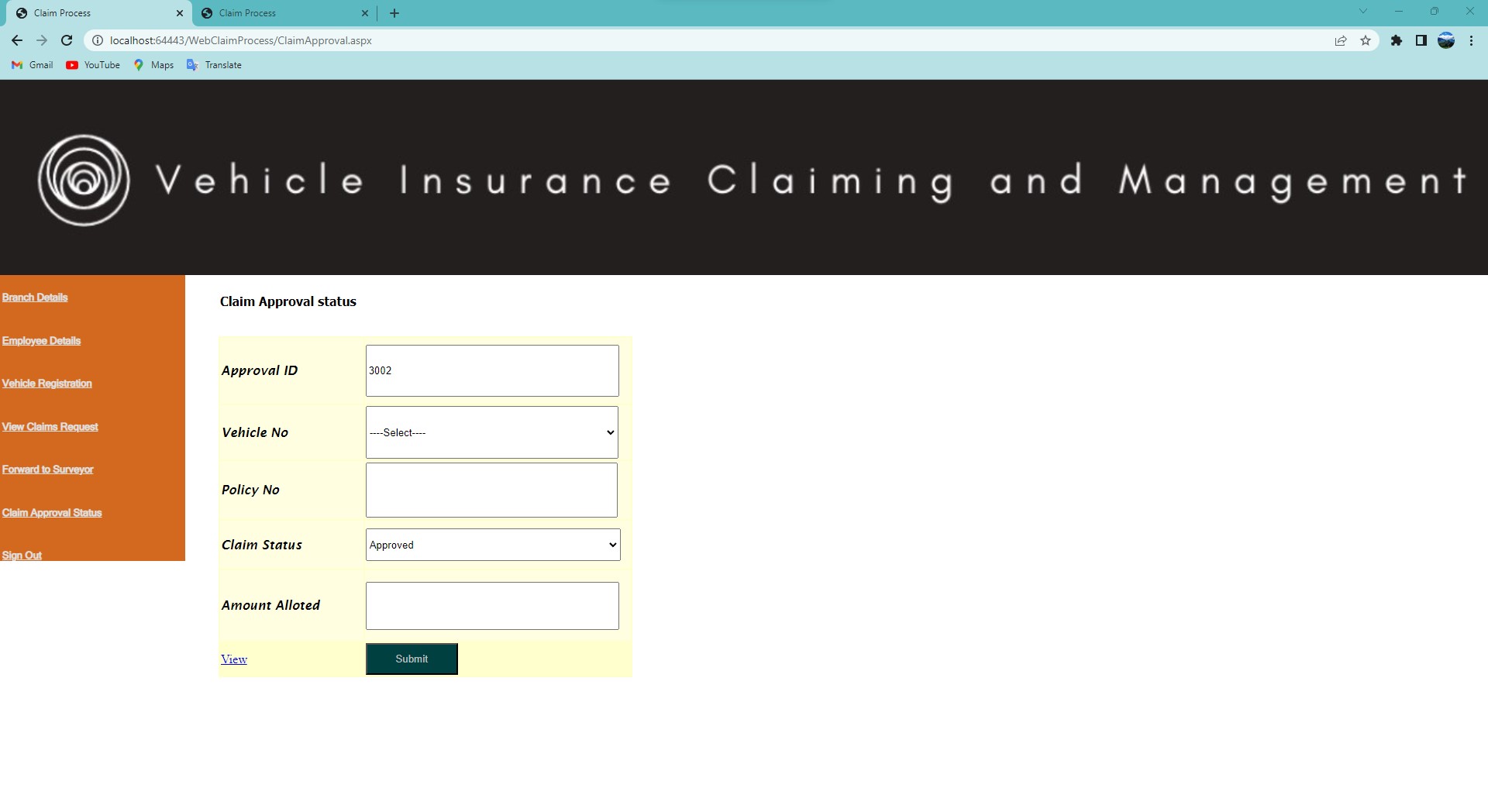
**EMPLOYEE DETAILS:**



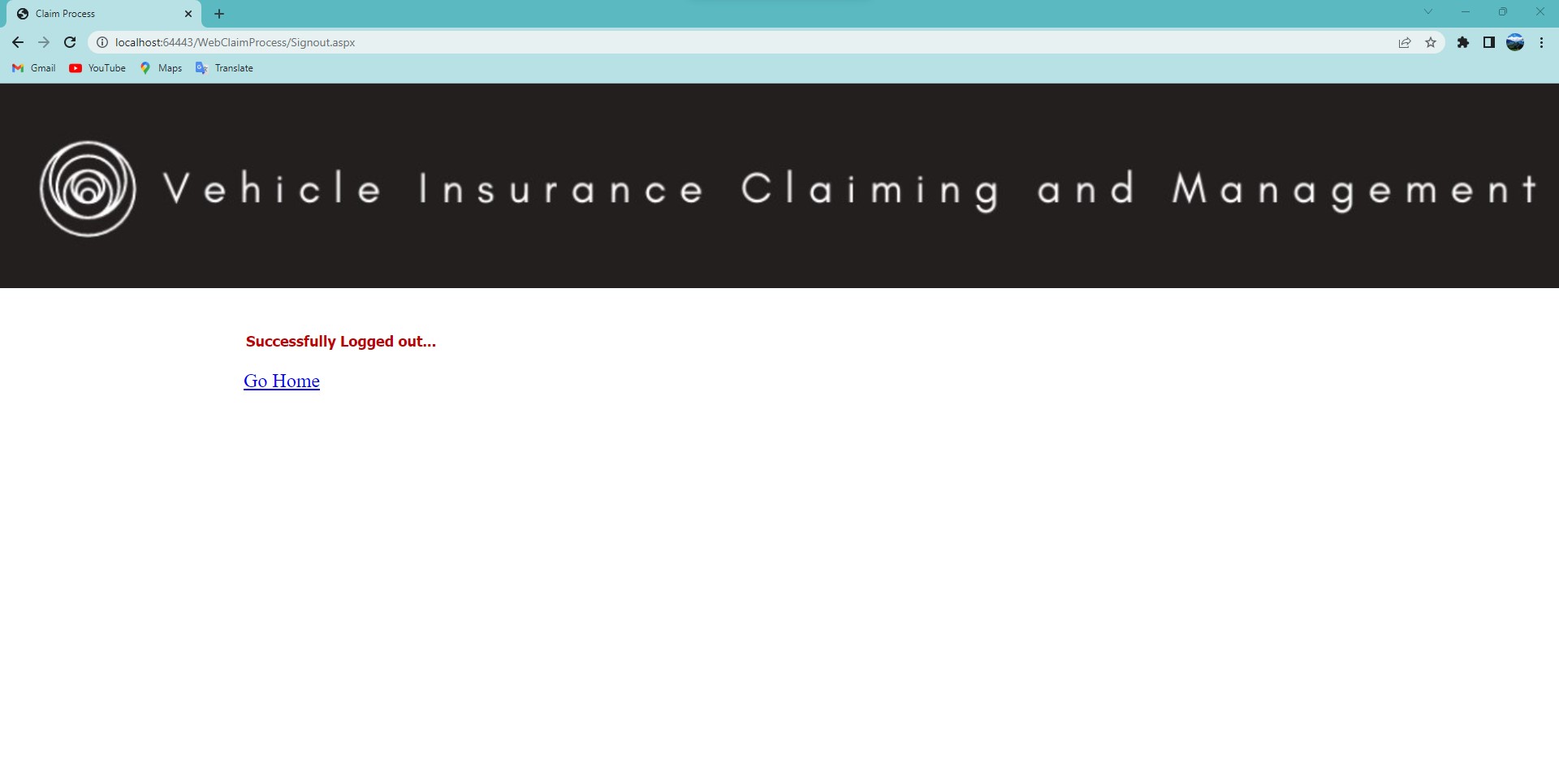
**VEHICLE INSURANCE POLICY DETAILS:**



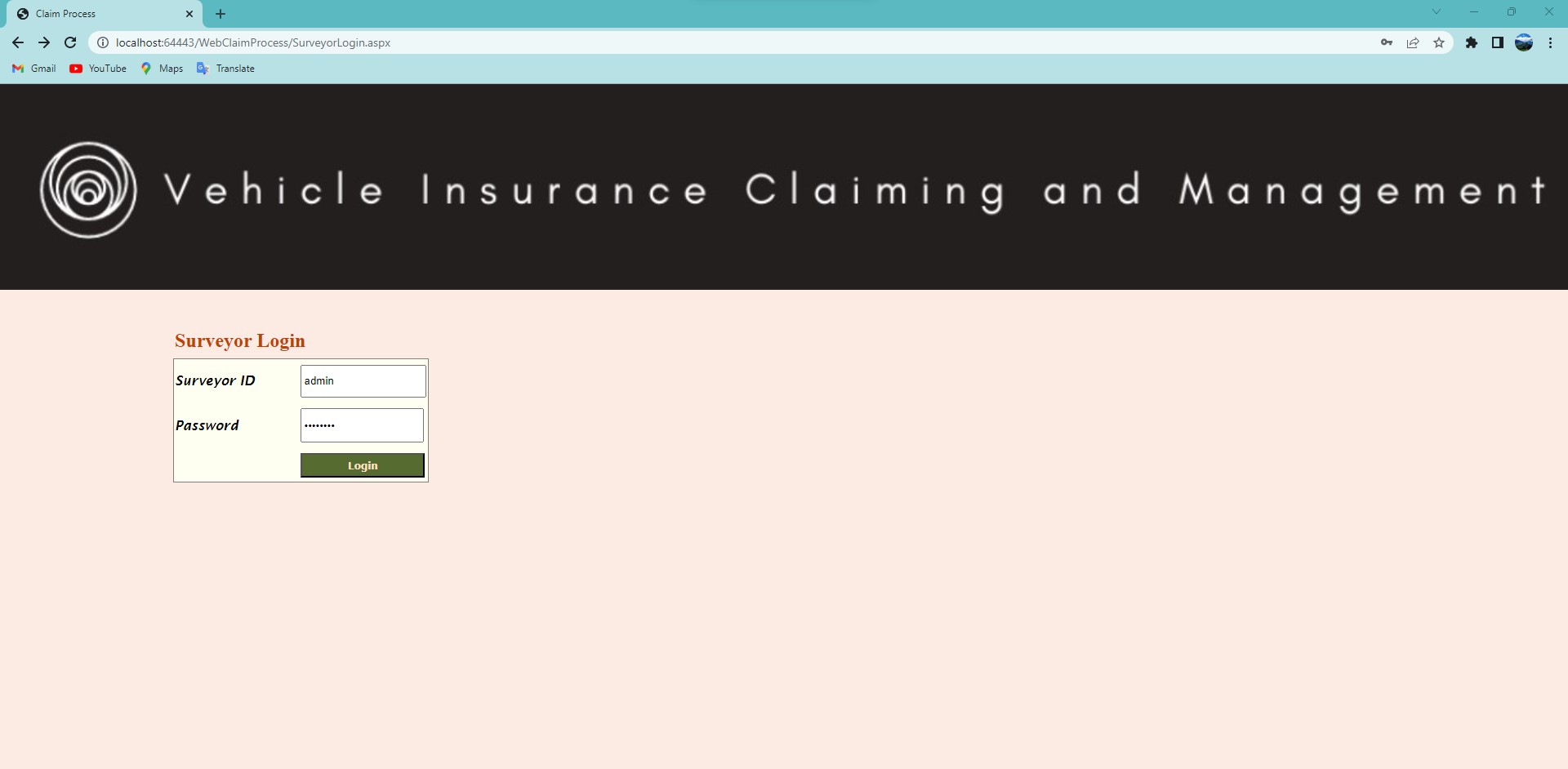
**CLAIM APPROVAL:**



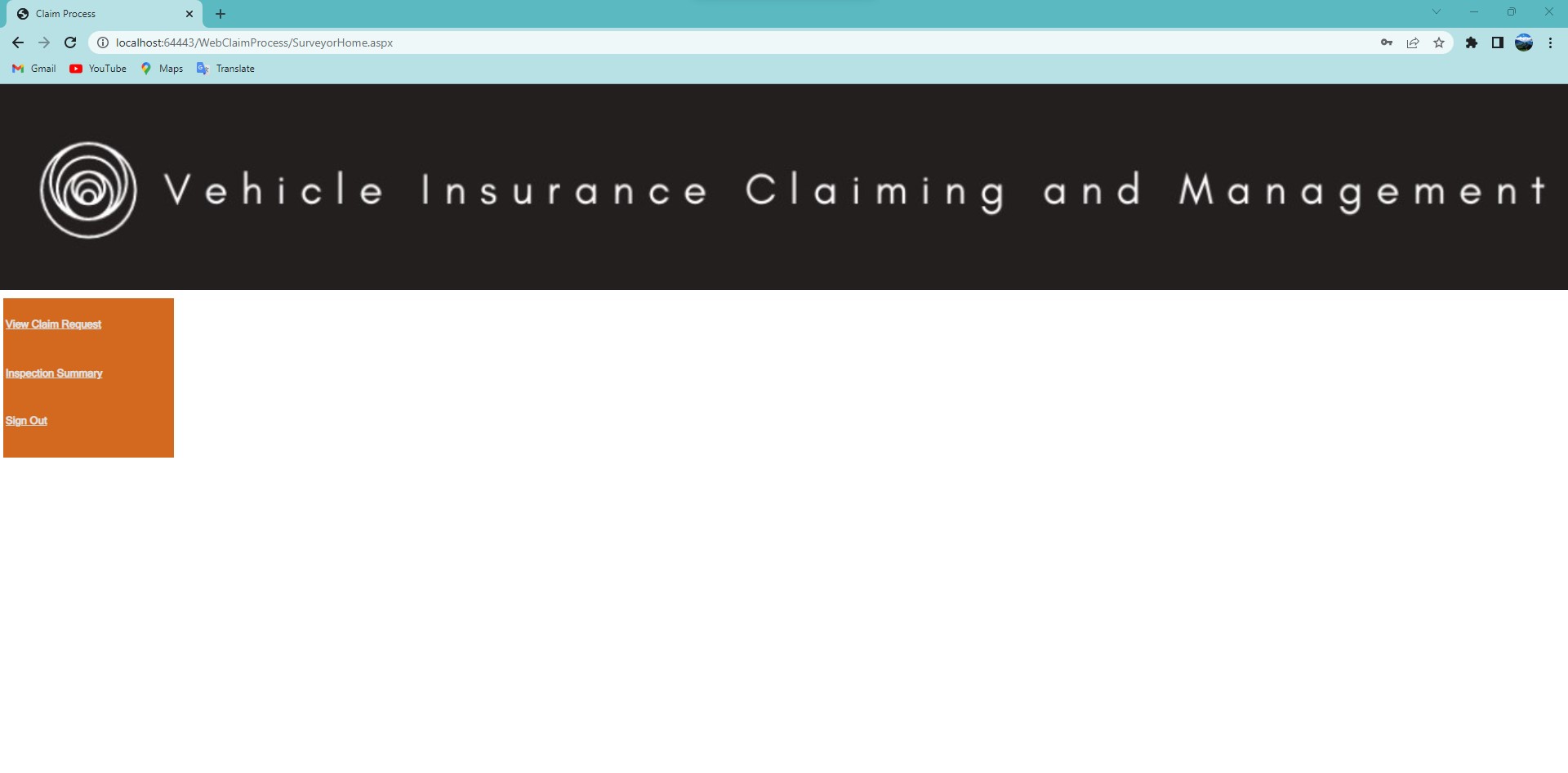
**ADMIN SIGNOUT:**



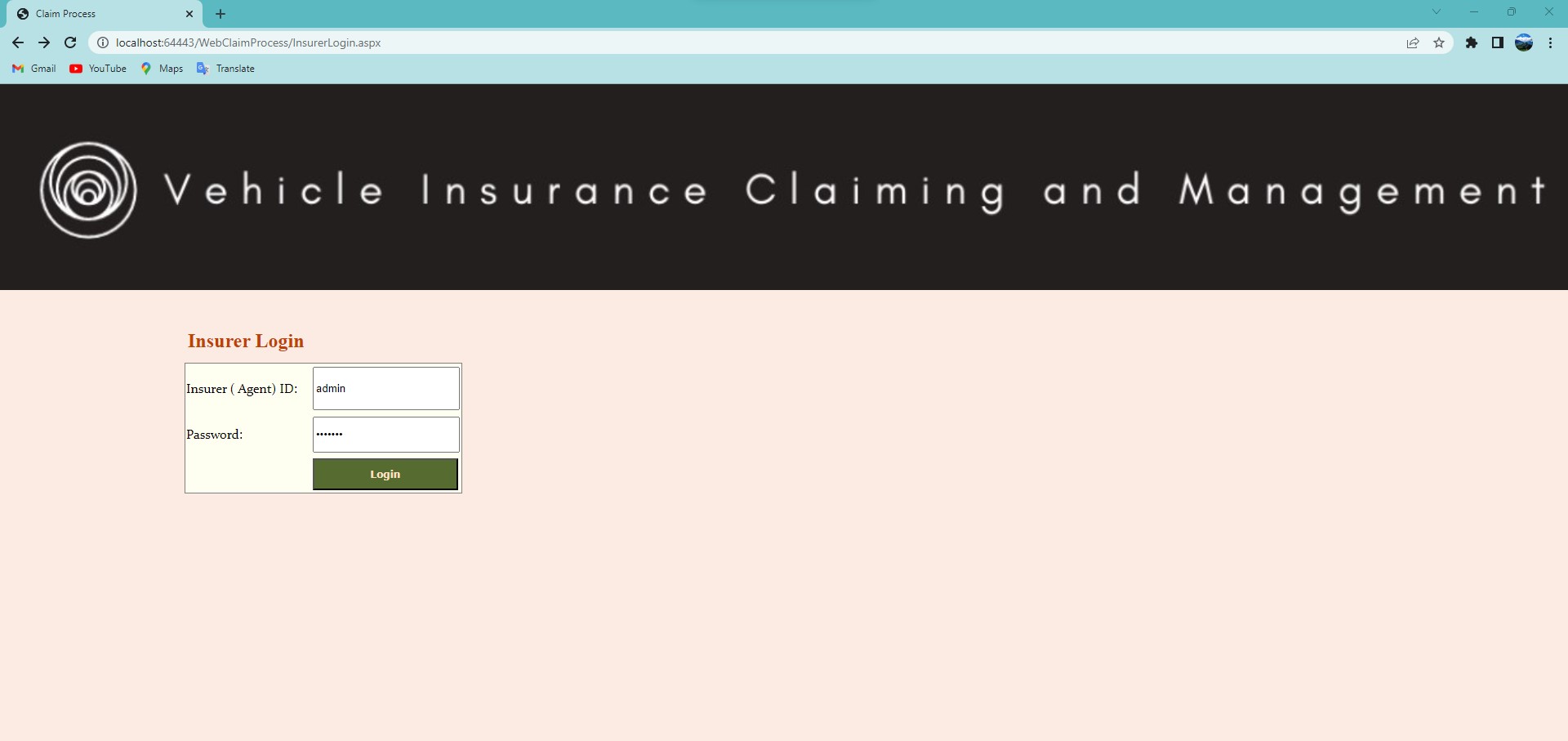
**SURVEYOR LOGIN:**



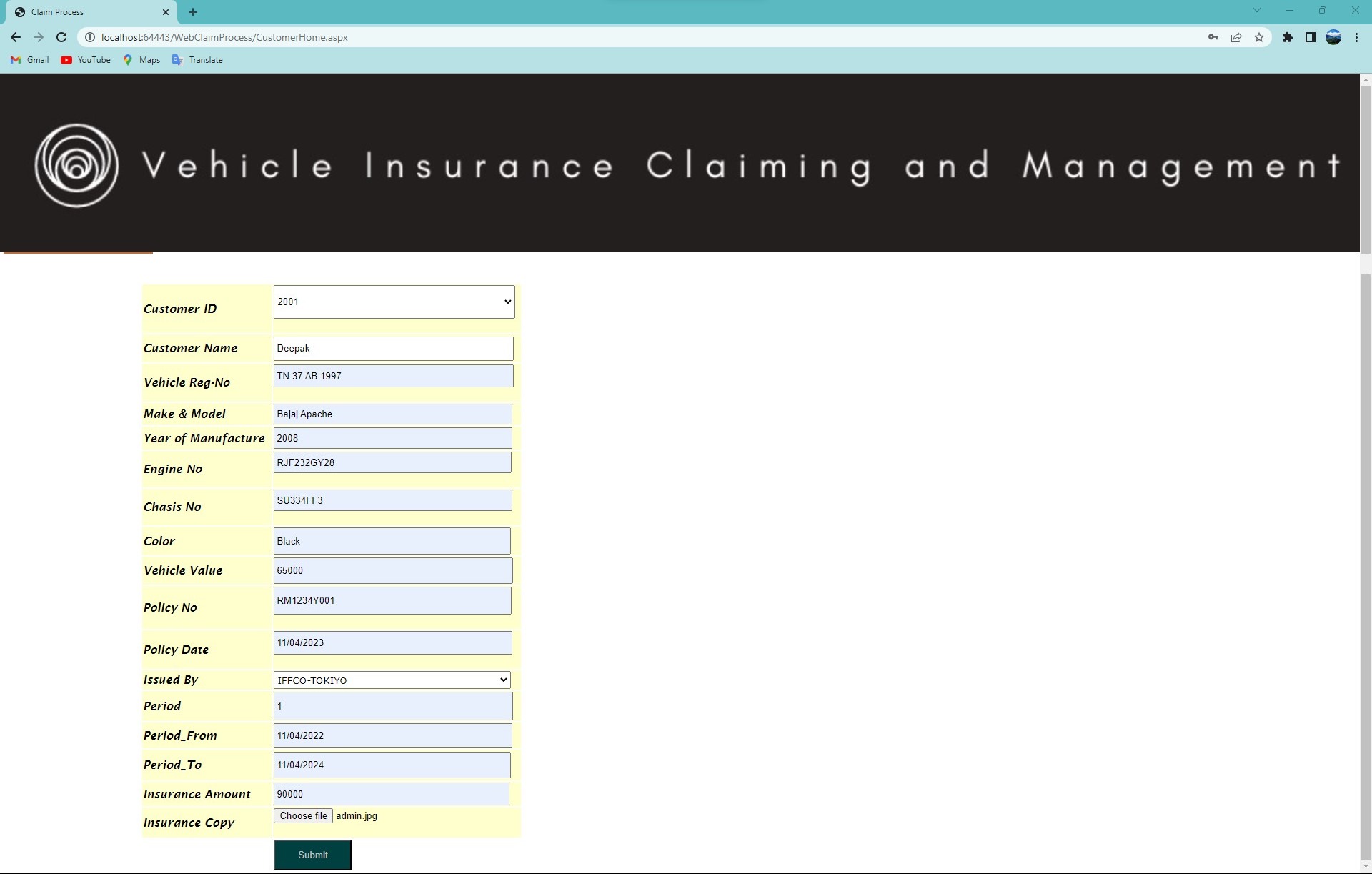
**SURVEYOR HOME:**



**INSURER HOME:**



**INSURER TAKING VEHICLE POLICY:**



* 1. **Appendix-B Sample Code**

**HOME PAGE**

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb" Inherits="\_Default" %>

<%@ Register Src="MainBanner.ascx" TagName="MainBanner" TagPrefix="uc1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<title>Claim Process</title>

</head>

<body>

<form id="form1" runat="server">

<div>

<uc1:MainBanner ID="MainBanner1" runat="server" />

<asp:Panel ID="Panel1" runat="server" BackImageUrl="~/Images/Risk\_Claim\_Management\_crop\_09.jpg" Height="224px"

Style="z-index: 100; left: 0px; position: absolute; top: 259px" Width="454px">

</asp:Panel>

<asp:Panel ID="Panel2" runat="server" BackImageUrl="~/Images/insurance\_claim.jpg"

Height="243px" Style="z-index: 101; left: 524px; position: absolute; top: 250px"

Width="244px">

</asp:Panel>

&nbsp;

<asp:Panel ID="Panel4" runat="server" BackImageUrl="~/Images/Application-Claims-424-177.jpg"

Height="139px" Style="z-index: 102; left: 0px; position: absolute; top: 99px"

Width="1152px">

<asp:ImageButton ID="ImageButton1" runat="server" Height="57px" ImageUrl="~/Images/admin.jpg"

Style="z-index: 100; left: 834px; position: absolute; top: 383px" Width="301px" />

<asp:ImageButton ID="ImageButton2" runat="server" Height="41px" ImageUrl="~/Images/surveyor.jpg"

Style="z-index: 102; left: 836px; position: absolute; top: 448px" Width="284px" />

</asp:Panel>

&nbsp;&nbsp;

<asp:ImageButton ID="ImageButton3" runat="server" Height="40px" ImageUrl="~/Images/insurer.jpg"

Style="z-index: 103; left: 841px; position: absolute; top: 605px" Width="292px" />

<asp:Panel ID="Panel3" runat="server" Height="180px" Style="z-index: 104; left: 822px;

position: absolute; top: 294px" Width="332px">

<table border="1" style="width: 313px; height: 152px">

<tr>

<td bgcolor="#999966" style="width: 102px">

<span style="font-family: Constantia">User ID:</span></td>

<td bgcolor="#999966" style="width: 150px">

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td bgcolor="#999966" style="width: 102px">

<span style="font-family: Constantia">Password:</span></td>

<td bgcolor="#999966" style="width: 150px">

<asp:TextBox ID="TextBox2" runat="server" TextMode="Password" Width="151px"></asp:TextBox></td>

</tr>

<tr>

<td bgcolor="#999966" style="width: 102px">

</td>

<td bgcolor="#999966" style="width: 150px">

<asp:Button ID="Button1" runat="server" BackColor="DarkOliveGreen" Font-Bold="True"

Font-Names="Tahoma" ForeColor="#FFE0C0" Text="Login" Width="111px" /></td>

</tr>

<tr>

<td bgcolor="#999966" style="width: 102px">

&nbsp;</td>

<td bgcolor="#999966" style="width: 150px">

<asp:Label ID="lblResult" runat="server" ForeColor="#3333FF"></asp:Label>

</td>

</tr>

</table>

<asp:Label ID="Label2" runat="server" Font-Bold="True" Font-Size="Large" ForeColor="#C00000"

Style="z-index: 100; left: -341px; position: absolute; top: 219px" Width="302px"></asp:Label>

<asp:LinkButton ID="LinkButton1" runat="server" Font-Bold="True" Font-Names="Tahoma"

Style="z-index: 102; left: 142px; position: absolute; top: 152px" Width="185px">Sign In?</asp:LinkButton>

</asp:Panel>

<asp:Label ID="Label1" runat="server" Font-Bold="True" Font-Size="Large" ForeColor="#C04000"

Style="z-index: 106; left: 826px; position: absolute; top: 262px" Text="Customer Login"

Width="286px"></asp:Label>

zS</div>

</form>

</body>

</html>

**CUSTOMER DETAILS**

Imports System.Data.SqlClient

Imports System.Data

Partial Class CustomerHome

Inherits System.Web.UI.Page

Dim con As New SqlConnection

Dim constr As String

Dim da As New SqlDataAdapter

Dim ds As New DataSet

Dim cmd As New SqlCommand

Dim dr As SqlDataReader

Dim userval As String

Dim datared As SqlDataReader

Public Sub autoincrement()

Dim qry As String = "select max(claimid) from tblClaimApplication"

cmd = New SqlCommand(qry, con)

da.SelectCommand = cmd

datared = cmd.ExecuteReader

If datared.Read Then

If IsDBNull(datared(0)) Then

txtclaimid.Text = 4000

Else

txtclaimid.Text = datared(0) + 1

End If

datared.Close()

End If

datared.Close()

End Sub

Sub userid()

cmd.CommandText = "select \* from tblCustomer where userid='" & Session.Item("username") & "'"

cmd.Connection = con

dr = cmd.ExecuteReader

If dr.Read = True Then

userval = dr("cusid")

TextBox1.Text = dr("addr")

txtcusname.Text = dr("cusname")

End If

dr.Close()

End Sub

Sub empname()

cmd.CommandText = "select \* from tblVehInsrdet where cusid='" & userval & "'"

cmd.Connection = con

dr = cmd.ExecuteReader

If dr.Read Then

txtcompanyname.Text = dr("issuedby")

txtpolicyno.Text = dr("policyno")

txtvehicleno.Text = dr("vregno")

End If

dr.Close()

End Sub

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

constr = "Data Source=KAMAL\SQLEXPRESS1;Initial Catalog=claim;Integrated Security=True"

con = New SqlConnection(constr)

con.Open()

userid()

empname()

autoincrement()

End Sub

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click

cmd.CommandText = "Insert into tblClaimApplication values('" & userval & "','" & txtcusname.Text & "','" & TextBox1.Text & "','" & txtclaimid.Text & "','" & txtdate.Text & "','" & txtvehicleno.Text & "','" & txtpolicyno.Text & "','" & txtcompanyname.Text & "','" & txtreason.Text & "','" & txtdescr.Text & "','" & FileUpload1.FileName & "','" & "REQUEST" & "')"

cmd.Connection = con

cmd.ExecuteNonQuery()

Label1.Text = "Registered"

' MsgBox("Inserted")

referesh()

End Sub

Protected Sub Page\_Unload(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Unload

con.Close()

Label1.Visible = False

End Sub

Private Sub referesh()

txtcusname.Text = ""

txtcompanyname.Text = ""

txtdate.Text = ""

txtdescr.Text = ""

txtpolicyno.Text = ""

txtreason.Text = ""

txtvehicleno.Text = ""

TextBox1.Text = ""

txtclaimid.Text = ""

End Sub

End Class

**ADMIN DETAILS**

Partial Class AdminLinks

Inherits System.Web.UI.UserControl

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton1.Click

Response.Redirect("BranchEntry.aspx")

End Sub

Protected Sub LinkButton2\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton2.Click

Response.Redirect("EmployeeDetails.aspx")

End Sub

Protected Sub LinkButton3\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton3.Click

Response.Redirect("InsurancePolicydet.aspx")

End Sub

Protected Sub LinkButton7\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton7.Click

Response.Redirect("ClaimApproval.aspx")

End Sub

Protected Sub LinkButton5\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton5.Click

Response.Redirect("Signout.aspx")

End Sub

Protected Sub LinkButton4\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton4.Click

End Sub

Protected Sub LinkButton6\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton6.Click

End Sub

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

End Sub

End Class

**INSURER DETAILS**

Imports System.Data.SqlClient

Partial Class InsurancePolicydet

Inherits System.Web.UI.Page

Dim con As New SqlConnection

Dim constr As String

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

constr = "Data Source=KAMAL\SQLEXPRESS1;Initial Catalog=claim;Integrated Security=True"

con = New SqlConnection(constr)

con.Open()

'MsgBox("done")

idcal()

End Sub

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim cmd As New SqlCommand

cmd.CommandText = "Insert into tblVehInsrdet values(" & DropDownList2.SelectedItem.Text & ",'" & txtcusname.Text & "','" & txtvehicleNo.Text & "','" & TextBox1.Text & "','" & TextBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & txtpolicyno.Text & "','" & txtpolicydate.Text & "','" & DropDownList1.SelectedItem.Text & "','" & txtperiod.Text & "','" & txtfrom.Text & "','" & txtto.Text & "'," & txtamount.Text & ",'" & FileUpload1.FileName & "')"

cmd.Connection = con

cmd.ExecuteNonQuery()

Label1.Text = "Registered"

'MsgBox("Inserted")

'referesh()

End Sub

Private Sub idcal()

Dim cmd As New SqlCommand

Dim dr As SqlDataReader

cmd.CommandText = "select cusid from tblCustomer"

cmd.Connection = con

dr = cmd.ExecuteReader

While dr.Read

DropDownList2.Items.Add(dr(0))

End While

dr.Close()

End Sub

Protected Sub Page\_Unload(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Unload

con.Close()

Label1.Visible = False

End Sub

Private Sub referesh()

txtamount.Text = ""

txtcusname.Text = ""

txtfrom.Text = ""

txtperiod.Text = ""

txtpolicydate.Text = ""

txtpolicyno.Text = ""

txtto.Text = ""

txtfrom.Text = ""

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ""

TextBox5.Text = ""

TextBox6.Text = ""

End Sub

End Class