**E - BANKING SYSTEM**

**A Project Report submitted to**

**Bishop Heber College (Autonomous)Tiruchirappalli**

**affiliated to Bharathidasan University, Tiruchirappalli-6200024**

**In partial fulfilment of the requirements for the award of the degree of**

**Bachelor of Vocational in Information Technology**

**By**

**NARAIN KARTHIK J**

**(Register No: 205915130)**

**Under the guidance of**

**Mrs. P.USHA, MCA, M.Phil., SET., NET.,**

****

**Department Of Information Technology**

**Bishop Heber College (Autonomous)**

**(Nationality Re-accredited with ,, A’’ Grade by NAAC with a CGPA of 3.58 out of 4)**

**(Recognized by UGC as “College of Excellence”)**

**Tiruchirappalli – 620017**

**NOVEMBER 2021**



**Department of Information Technology Bishop Heber College (Autonomous) Tiruchirappalli – 620 017, Tamilnadu, India**

**Phone No. : 0431 – 277 0136**

**CERTIFICATE**

The Viva-Voce Examination for the candidate **NARAIN KARTHIK J**

**(Reg. No. 205915130)** was held on .

**Signature of the HOD**

**Examiner.**

**1.**

**2.**

**** **Mrs. P. USHA, MCA, M.Phil., SET., NET.,**

**Assistant professor, Department of Information Technology,**

**Bishop Heber College (Autonomous), Tiruchirappalli – 620 017**

**Date:**

**CERTIFICATE**

This is to certify that the project work entitled **“E-BANKING SYSTEM”** is a bonafide work done under my supervision by **NARAIN KARTHIK J (Reg. No. 205915130)** and submitted to Bishop Heber College (Autonomous), Tiruchirappalli – 620017 in partial fulfillment of the requirements of the award of the degree of Bachelor of Vocation in Information Technology during the odd semester of the academic year (2020 – 2021)

**DECLARATION**

I hereby declare that the work presented in this project work report is the original work done by me under the guidance of Mrs. P. USHA, MCA, M.Phil., SET., NET., Assistant Professor and Department of information technology, Bishop Heber College (Autonomous), Tiruchirappalli – 620 017 and has not been included in any other project work submitted for any other degree.

Name of the Candidate : NARAIN KARTHIK J

Register Number : 205915130

Semester : THIRD

Academic Year : 2020 – 2021

Course code : U19IT3F3

**Signature of the Candidate**

**ACKNOWLEDGEMENT**

First of all, I would thank **ALMIGHTY GOD** to give abundant grace. Good health and knowledge to do this Project.

I express my sincere gratitude to **Dr.D.PAUL DHAYABARAN, M.Sc., M.Phil. PGDCA., Ph.D.**, Principal of Bishop Heber College (Autonomous), Tiruchirapalli for his blessing

I am highly indebted to thank **Dr.J.JOHN RAYBIN JOSE, M.Sc.**, **M.C.A,M.Phil, PGDCA,** Associate Professor and Head, Department of Information Technology, Bishop Heber College (Autonomous), Trichy for having grand me permission to proceed with this project.

I wish to place on record my gratitude **Mrs. P. USHA, MCA, M.Phil., SET., NET.,** Assistant Professor and Department of Information Technology, Bishop Heber College (Autonomous), Trichy for his support during this project work.

I record my deep sense of gratitude to my beloved parents and my friends for their encouragement and moral support extended during the period of my project.

**NARAIN KARTHIK J**

**SYNOPSIS**

**E-Banking System** is specially designed for the purpose of adding records in a system. In this system, the user can store records easily according to their categories. The main aim of this project is to carry out the process easily and quickly. There are different types of categories such as View Account, Branch, Deposit Withdrawal and Transactions. It tracks all the information and generates the report, to increase the efficiency of bank system.

he BANK MANAGEMENT SYSTEM undertaken as a project is based on

relevant technologies. The main aim of this project is to develop software for

bank management system. This project is to develop software for bank

management system. This project has been developed to carry out the

processes easily and quickly, which is not possible with the manuals

systems, which are overcome by this software

he BANK MANAGEMENT SYSTEM undertaken as a project is based on

relevant technologies. The main aim of this project is to develop software for

bank management system. This project is to develop software for bank

management system. This project has been developed to carry out the

processes easily and quickly, which is not possible with the manuals

systems, which are overcome by this software

he BANK MANAGEMENT SYSTEM undertaken as a project is based on

relevant technologies. The main aim of this project is to develop software for

bank management system. This project is to develop software for bank

management system. This project has been developed to carry out the

processes easily and quickly, which is not possible with the manuals

systems, which are overcome by this software

**CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.No., TITLE PAGE. NO** | | |
| **1** | **INTRODUCTION** | **1** |
| **2** | **SYSTEM STUDY** | **2** |
|  | 2.1. Project Description | **2** |
|  | 2.1.1. Existing system | **2** |
|  | 2.1.2. Proposed system | **2** |
| **3** | **REQUIREMENT ANALYSIS** | **3** |
|  | 3.1 Hardware Requirement | **3** |
|  | 3.2 Software Requirement | **3** |
| **4** | **SYSTEM DESIGN** | **6** |
|  | 4.1 Logical Design | **6** |
|  | 4.2 Dataflow Diagram | **7** |
|  | 4.3 Database Design | **9** |
|  | 4.4 Program Design | **10** |
| **5** | **SYSTEM DEVELOPMENT** | **12** |
|  | 5.1 Program Development | **12** |
| **6** | **SYSTEM TESTING** | **48** |
|  | 6.1 Unit Testing | **48** |
|  | 6.2 Integration Testing | **49** |
|  | 6.4 Validation Testing |  |
| **7** | **SYSTEM IMPLEMENTATION** |  |
|  | 7.1 Screenshots |  |
| **8** | **CONCLUSION** |  |
|  | **BIBLIOGRAPHY** |  |

# 

# 1. INTRODUCTION

The E-Banking System is one of the systems which help the administration in speeding up the tasks at the same time reducing the complexity. The purpose or objective of this system is to digitalize and create an automated system The E-Banking System is an application that helps the bankers in east maintenance of the record details. E-Banking System is based on a concept of recording customer’s account details.

Access is usually through a secure website using username and passwords. Online banking system is the practice of making bank transactions. The popularity of internet banking system will typically connect to be part of the core banking system.

It manage the information of internet banking. It shows the information and description of the transactions. The E-Banking system is the most efficient system in the world. Here the user can perform all the tasks like creating an account, deposit, withdraw, check balance, view all account holder detail.

**2. SYSTEM STUDY**

System analysis is a process of gathering the facts concerning the system breaking them into elements and relationship between elements; it provides a framework for visualizing the organizational and environmental factors that operate on a system. The quality of work performed by a machine is usually uniform, neat and more reliable when compared to doing the same operations manually.

* 1. **PROJECT DESCRIPTION**

The E-Banking System is an application and maintains a centralized repository of all related information. The project is the developed a system which provides a login system of the Bank which will allow only admins to add Bank details, Create Account, Loan, Branch , Deposit and also view the login details of Bank.

* + 1. **EXISTING SYSTEM**

The information system of the old times totally differs from the today’s. It is automation system of a E-Banking system or any information system. In the previous ages the process of maintaining data and record seemed very difficult. Now a days, computer is very essential component. The implementation if computer-based information system can be depicted as the ease in handling and maintaining the data bank of any size. In the existing system, all the records are not kept perfectly because all the work is done manually, so keeping up to date details of the New User details, Branch details, Loan details is not done. Amount of the overall trips are kept in documents and the calculations done are manually which made lead to huge mistakes. Thus, the existing system is very time consuming and being manual work sometimes lead to a great loss.

* + 1. **PROPOSED SYSTEM**

The propose system is highly automated and makes the recording and Login, Branch, Loan, Deposit activities much easier and flexible. The admins can get the very right information at the very right time. Students can get the knowledge of the easy and reliable and minimum tie needed for the Bank details method which they are going to use their Bank.

**3. REQUIREMENT ANALYSIS**

For efficient use all computer software and hardware components and other software sources in the computer system where it is used. These prerequisites are known as computer or system requirements. The hardware and software requirements of the system are given as below.

* 1. **HARDWARE REQUIREMENT**

|  |  |  |
| --- | --- | --- |
|  | Processor | : AMD PRO A4-4350B R4, 5 COMPUTE CORES 2C+3G 2.50GHz |
|  | RAM | : 4 GB |
|  | Hard disk | : 500 GB |
|  | Keyboard | : Standard keyboard |
|  | Monitor | : 14 inch color monitor |

* 1. **SOFTWARE SPECIFICATION**

|  |  |  |
| --- | --- | --- |
|  | Front End | : Visual Studio |
|  | Back End | : MySQL |
|  | Language | : C# |
|  | Platform | : Windows |

**PROGRAMING LANGUAGE:**

**Microsoft.NET Framework**

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.

To provide a code-execution environment that minimizes software deployment and versioning conflicts.

To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.

To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.

To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.

To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers

Generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

.NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

Console applications.

Scripted or hosted applications.

Windows GUI applications (Windows Forms).

ASP.NET applications.

XML Web services.

Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

Client Application Development

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file

C# Language

C# (pronounced “See Sharp”) is a simple, modern, object-oriented, and type-safe programming language. C# has its roots in the C family of languages and will be immediately familiar to C, C++, and Java programmers. C# is standardized by ECMA International as the ECMA-334 standard and by ISO/IEC as the ISO/IEC 23270 standard. Microsoft’s C# compiler for the .NET Framework is a conforming implementation of both of these standards.

C# is an object-oriented language, but C# further includes support for component-oriented programming. Contemporary software design increasingly relies on software components in the form of self-contained and self-describing packages of functionality. Key to such components is that they present a programming model with properties, methods, and events; they have attributes that provide declarative information about the component; and they incorporate their own documentation. C# provides language constructs to directly support these concepts, making C# a very natural language in which to create and use software components.

Several C# features aid in the construction of robust and durable applications: Garbage collection automatically reclaims memory occupied by unused objects; exception handling provides a structured and extensible approach to error detection and recovery; and the type-safe design of the language makes it impossible to have uninitialized variables, to index arrays beyond their bounds, or to perform unchecked type casts.

.Net Framework provide four basic objects: the Command, Connection, DataReader and DataAdapter. In the remaining sections of this document, we'll walk through each part of the DataSet and the OLE DB/SQL Server .NET Data Providers explaining what they are, and how to program against them.

The following sections will introduce you to some objects that have evolved, and some that are new. These objects are:

Connections. For connection to and managing transactions against a database.

Commands. For issuing SQL commands against a database.

DataReaders. For reading a forward-only stream of data records from a SQL Server data source.

DataSets. For storing, Remoting and programming against flat data, XML data and relational data.

DataAdapters. For pushing data into a DataSet, and reconciling data against a database.

Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

  ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous

What is ASP.NET Web Forms?

      The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

      Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework has been specifically designed to address a number of key deficiencies in the previous model. In particular, it provides:

       ASP.NET Web Forms pages are text files with an .aspx file name extension. They can be deployed throughout an IIS virtual root directory tree. When a browser client requests .aspx resources, the ASP.NET runtime parses and compiles the target file into a .NET Framework class. This class can then be used to dynamically process incoming requests. (Note that the .aspx file is compiled only the first time it is accessed; the compiled type instance is then reused across multiple requests).

       In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the <asp: adrotator> control can be used to dynamically display rotating ads on a page.

       During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest.  If your business needs change over time, you define any additional fields or change the definition of existing field.

**4. SYSTEM DESIGN**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of system theory to product development.

**4.1 LOGICAL DESIGN**

Logical design is an abstract concept in computer programming by which programmers arrange data in a series of logical relationships known as attributes or entities. An entity refers to a chunk of information, whereas an attribute defines the unique properties of an entity.

E-BANKING SYSTEM

ADMIN

**Deposit List**

**Edit User**

**Withdrawal**

**Balance List**

**USER**

**Withdrawal Amount**

**Deposit Amount**

**Balance Amount**

**Transaction**

**Register ATM**

**4.2 PROGRAM DESIGN**

Program design is the activity of progressing from a specification of some required program to a description of the program itself. Most phase models of the software life cycle recognize program design ns one of the phase. The input to this phase specification of what the program is required to do. During the phase the design decisions are made as to how the program will meet these requirements, and the output of the phase is a description of the program in some form that provides a suitable basis for subsequent implementation is the activity of progressing from a specification of some required program to a description of the program itself. Most phase models of the software life cycle recognize program design ns one of the phase. The input to this phase specification of what the program is required to do. During the phase the design decisions are made as to how the program will meet these requirements, and the output of the phase is a description of the program in some form that provides a suitable basis for subsequent implementation

**4.2.1 MODULE DESCRIPTION**

**Module:**

* Admin
* User
* Branch
* Deposit Amount
* Withdrawal
* Transaction
* ATM
* Edit User

**Module Description:**

**ADMIN**

Admin can handle all the records from the system of bank details and also edit .

**USER**

User can login and enter into the details to a bank via username and passwords.

**BRANCH**

In this module we can select the branch and also select details of place, time, city, state.

**DEPOSIT AMOUNT**

It shows all the deposit amount to the account holder.

**WITHDRAWL**

It shows all the withdrawal amount on bank.

**TRANSACTION**

It shows all the transaction amount from the account.

**EDIT USER**

Admin can edit the user details whether it comes wrong

**5. SYSTEM DEVELOPMENT**

Systems development is the process of defining, designing, testing, and implementing a new software application or program. It could include the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software. Written standards and procedures must guide all information systems processing functions. The organization’s management must define and implement standards and adopt an appropriate system development life cycle methodology governing the process of developing, acquiring, implementing, and maintaining computerized information systems and related technology.

**5.1 PROGRAM DEVELOPMENT:**

**Cs Code:**

**LOGIN**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

public partial class Login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

ErrorMsgLabel.Visible = false;

}

protected void LOGINButton\_Click(object sender, EventArgs e)

{

using (SqlConnection sqlCon = new SqlConnection(@"Data Source=LENOVO\SQLEXPRESS;Initial Catalog=BANK SYSTEM;Integrated Security=True"))

{

sqlCon.Open();

string query = "SELeCT COUNT(1) FROM login WHERE userid=@userid AND password=@password";

SqlCommand sqlcmd = new SqlCommand(query, sqlCon);

sqlcmd.Parameters.AddWithValue("@userid", UserIDTextBox.Text.Trim());

sqlcmd.Parameters.AddWithValue("@password", PasswordTextBox.Text.Trim());

int count = Convert.ToInt32(sqlcmd.ExecuteScalar());

if (count == 1)

{

Session["userid"] = UserIDTextBox.Text.Trim();

Response.Redirect("HomeLogin.aspx");

}

else { ErrorMsgLabel.Visible = true; }

}

**}**

**}**

**CREATE ACCOUNT**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Collections;

using System.Configuration;

public partial class CreateAccount : System.Web.UI.Page

{

string constr = ConfigurationManager.ConnectionStrings["BANK SYSTEMConnectionString"].ConnectionString;

SqlConnection cn;

SqlCommand cmd;

SqlDataReader dr;

SqlDataAdapter ada;

ArrayList name = new ArrayList();

DataSet ds = new DataSet();

protected void Page\_Load(object sender, EventArgs e)

{

cn = new SqlConnection(constr);

}

protected void ClearButton\_Click(object sender, EventArgs e)

{

NameTextBox.Text = "";

FathersNameTextBox.Text = "";

MothersNameTextBox.Text = "";

AgeTextBox.Text = "";

DOBTextBox.Text = "";

GenderRadioButtonList.SelectedValue = "";

QualificationTextBox.Text = "";

AddressTextBox.Text = "";

PhonenoTextBox.Text = "";

EmailTextBox.Text = "";

AadhaarTextBox.Text = "";

}

protected void SubmitButton\_Click(object sender, EventArgs e)

{

cn.Open();

cmd = new SqlCommand("insert into createaccount(name,fathername,mothername,age,dob,gender,qualification,address,phoneno,email,aadhaar) values('"+NameTextBox.Text+"','"+FathersNameTextBox.Text+"','"+MothersNameTextBox.Text+"','"+AgeTextBox.Text+"','"+DOBTextBox.Text+"','"+GenderRadioButtonList.SelectedValue+"','"+QualificationTextBox.Text+"','"+AddressTextBox.Text+"','"+PhonenoTextBox.Text+"','"+EmailTextBox.Text+"','"+AadhaarTextBox.Text+"')", cn);

cmd.ExecuteNonQuery();

cn.Close();

ClientScript.RegisterStartupScript(this.GetType(), "ins suc", "alert('Successfully Created Account');", true);

}

}

**BRANCH DETAILS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Collections;

using System.Configuration;

public partial class BranchDetails : System.Web.UI.Page

{

string constr = ConfigurationManager.ConnectionStrings["BANK SYSTEMConnectionString"].ConnectionString;

SqlConnection cn;

SqlCommand cmd;

SqlDataReader dr;

SqlDataAdapter ada;

ArrayList name = new ArrayList();

DataSet ds = new DataSet();

protected void Page\_Load(object sender, EventArgs e)

{

cn = new SqlConnection(constr);

}

protected void FindButton\_Click(object sender, EventArgs e)

{

cn.Open();

SqlCommand cmd = new SqlCommand("select \*from branch where pincode='" + PinCodeTextBox.Text + "'", cn);

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

Label1.Text = dr["branchname"].ToString();

Label2.Text = dr["branchmanager"].ToString();

Label3.Text = dr["address"].ToString();

Label4.Text = dr["district"].ToString();

Label5.Text = dr["state"].ToString();

Label6.Text = dr["pincode"].ToString();

Label7.Text = dr["ifsc"].ToString();

Label8.Text = dr["micrcode"].ToString();

Label9.Text = dr["workinghours"].ToString();

Label10.Text = dr["nonworkinghours"].ToString();

Label11.Text = dr["classification"].ToString();

Label12.Text = dr["phonenumber"].ToString();

Label13.Text = dr["emailaddress"].ToString();

}

}

}

**DEPOSIT**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Collections;

using System.Configuration;

public partial class ATM\_and\_Deposit\_DepositBalance : System.Web.UI.Page

{

string constr = ConfigurationManager.ConnectionStrings["BANK SYSTEMConnectionString"].ConnectionString;

SqlConnection cn;

SqlCommand cmd;

SqlDataReader dr;

SqlDataAdapter ada;

ArrayList name = new ArrayList();

DataSet ds = new DataSet();

protected void Page\_Load(object sender, EventArgs e)

{

cn = new SqlConnection(constr);

}

protected void DisplayButton\_Click(object sender, EventArgs e)

{

cn.Open();

SqlCommand cmd = new SqlCommand("select \*from deposit where name='" +TextBox1.Text + "'", cn);

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

Label1.Text = dr["name"].ToString();

Label2.Text = dr["number"].ToString();

Label3.Text = dr["type"].ToString();

Label4.Text = dr["deposit"].ToString();

}

}

}

**TRANSACTION**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Collections;

using System.Configuration;

public partial class Transaction\_Transaction : System.Web.UI.Page

{

string constr = ConfigurationManager.ConnectionStrings["BANK SYSTEMConnectionString"].ConnectionString;

SqlConnection cn;

SqlCommand cmd;

SqlDataReader dr;

SqlDataAdapter ada;

ArrayList name = new ArrayList();

DataSet ds = new DataSet();

protected void Page\_Load(object sender, EventArgs e)

{

cn = new SqlConnection(constr);

}

protected void SubmitButton\_Click(object sender, EventArgs e)

{

cn.Open();

cmd = new SqlCommand("insert into transaction(fromaccount,toaccount,amount,paymentdate,remark) values('" + FromAccountTextBox.Text + "','" + ToAccountTextBox.Text + "','" + AmountTextBox.Text + "','" + PayemntDateTextBox.Text + "','" + RemarkTextBox.Text + "')", cn);

cmd.ExecuteNonQuery();

cn.Close();

ClientScript.RegisterStartupScript(this.GetType(), "ins suc", "alert('Successfully Transacted');", true);

}

}

**LOAN**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Collections;

using System.Configuration;

public partial class Loan : System.Web.UI.Page

{

string constr = ConfigurationManager.ConnectionStrings["BANK SYSTEMConnectionString"].ConnectionString;

SqlConnection cn;

SqlCommand cmd;

SqlDataReader dr;

SqlDataAdapter ada;

ArrayList name = new ArrayList();

DataSet ds = new DataSet();

protected void Page\_Load(object sender, EventArgs e)

{

cn = new SqlConnection(constr);

}

protected void SubmitButton\_Click(object sender, EventArgs e)

{

cn.Open();

cmd = new SqlCommand("insert into loan(name,phoneno,state,city,employementtype,typeofloan,salaryrange,property,emailid) values('"+NameTextBox.Text+"','"+PhonenoTextBox.Text+"','"+StateDropDownList.SelectedValue+"','"+CityDropDownList.SelectedValue+"','"+EmployementypeownList.SelectedValue+"','"+TypeofLoanDropDownList.SelectedValue+"','"+SalaryRangeDropDownList.SelectedValue+"','"+HaveyouidentifiedpropertyDropDownList.SelectedValue+"','"+EmailidTextBox.Text+"')", cn);

cmd.ExecuteNonQuery();

cn.Close();

ClientScript.RegisterStartupScript(this.GetType(), "ins suc", "alert('Successfully Created');", true);

}

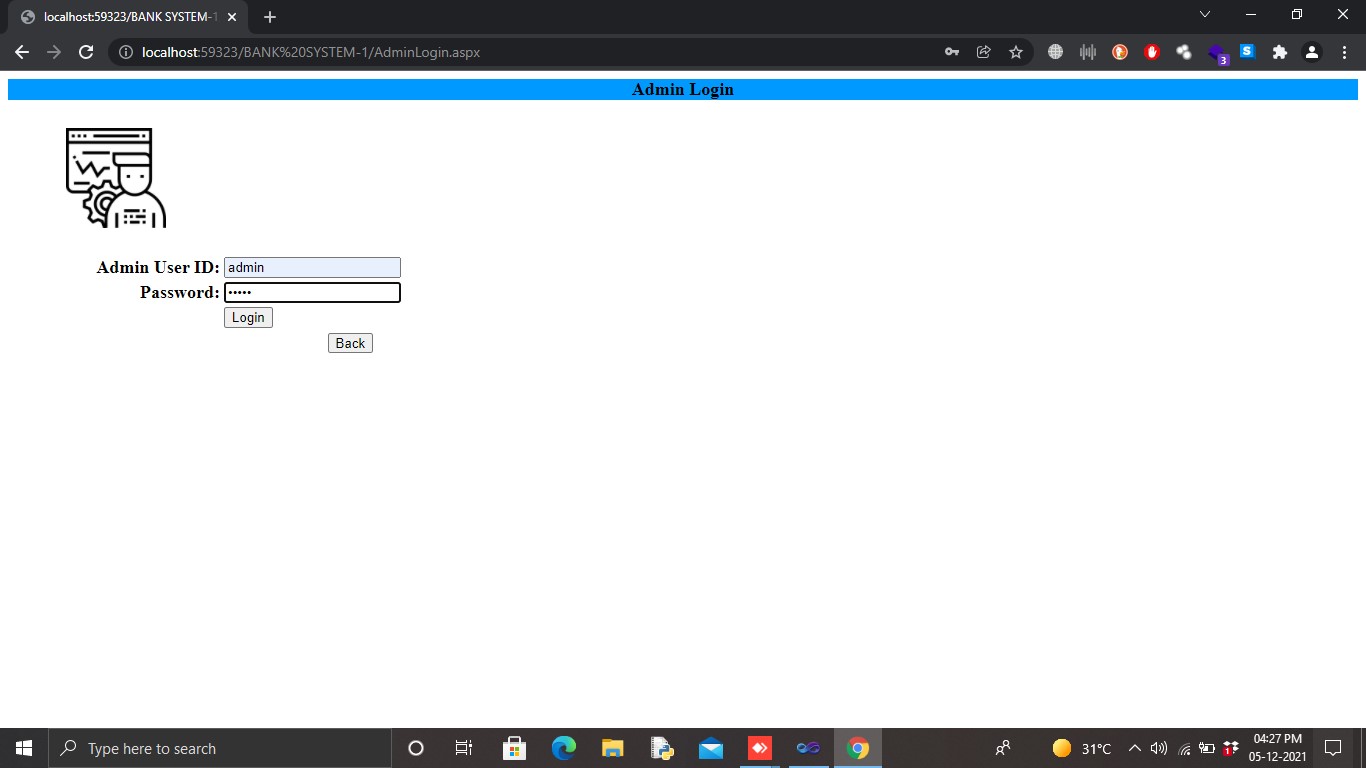
}

**6. SYSTEM TESTING**

System testing is the process of evaluation and software item to detect differences testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

**6.1 UNIT TESTING**

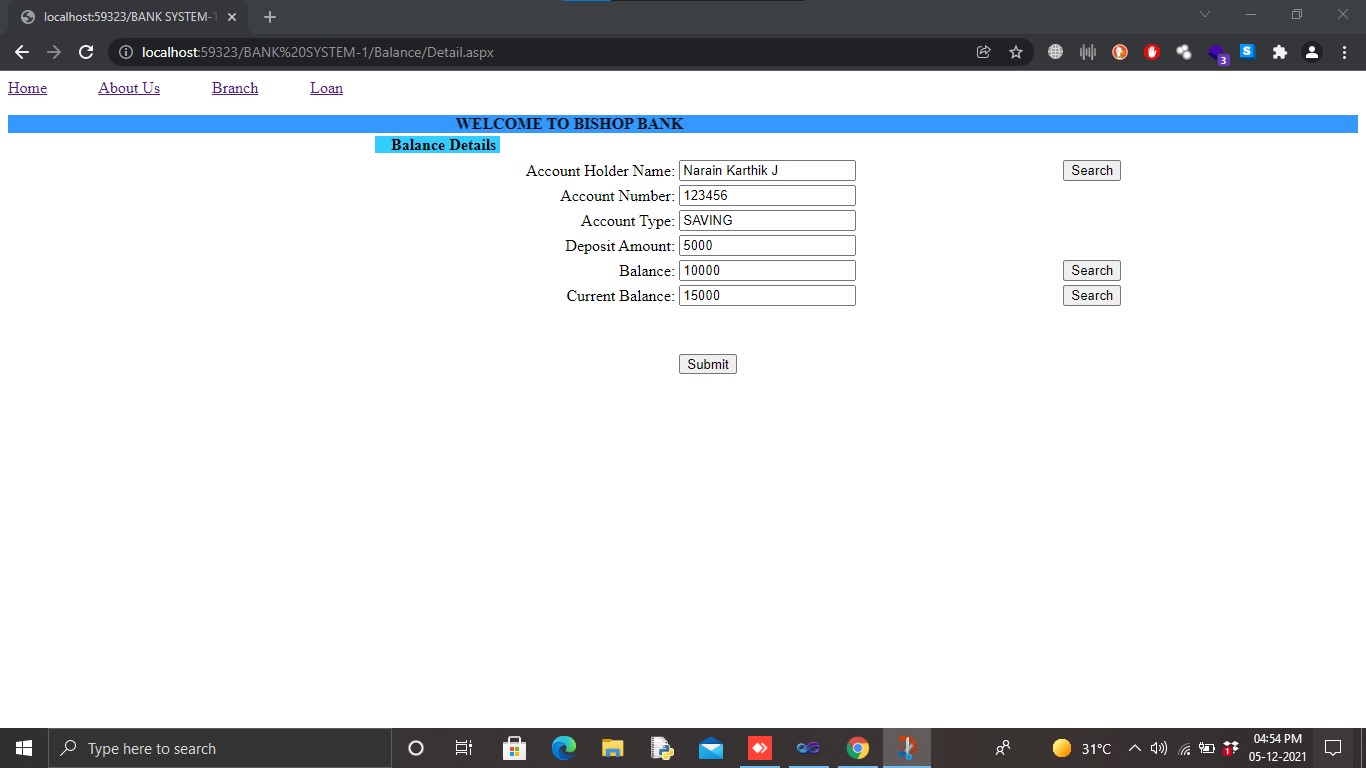
Unit testing verification efforts on the smallest unit of software design, module. This is known as "Module Testing” After testing every field in the modules, the modules of the project is tested separately. Unit testing focuses verification efforts on the smallest unit of software design and field. For example, Username and Password are entered in correct manner and checked.

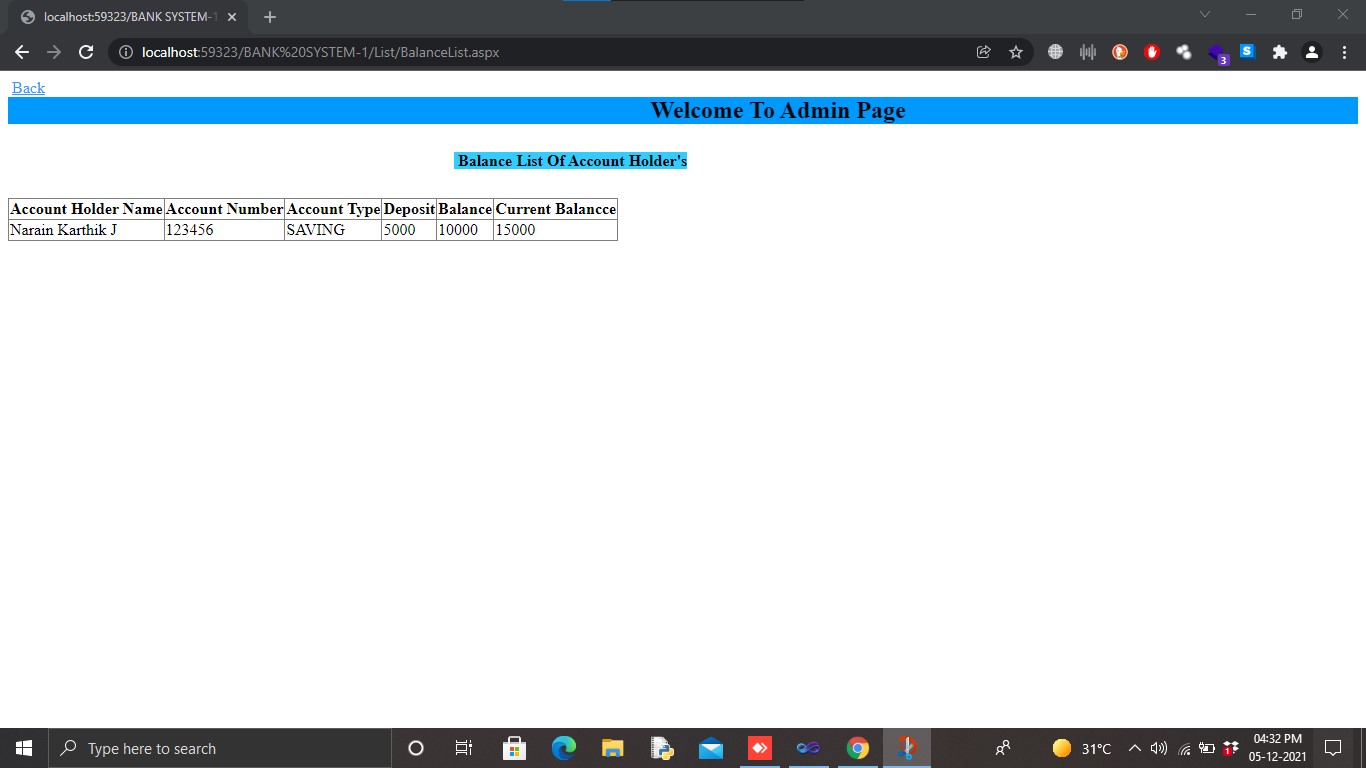


**Fig 6.1 Unit Testing**

**6.2 INTEGRATION TESTING**

Testing of integrate modules to verify the combined functionality after integration. The purpose of integration testing to verify the functionality, performance, and reliability between the modules that are integrated.

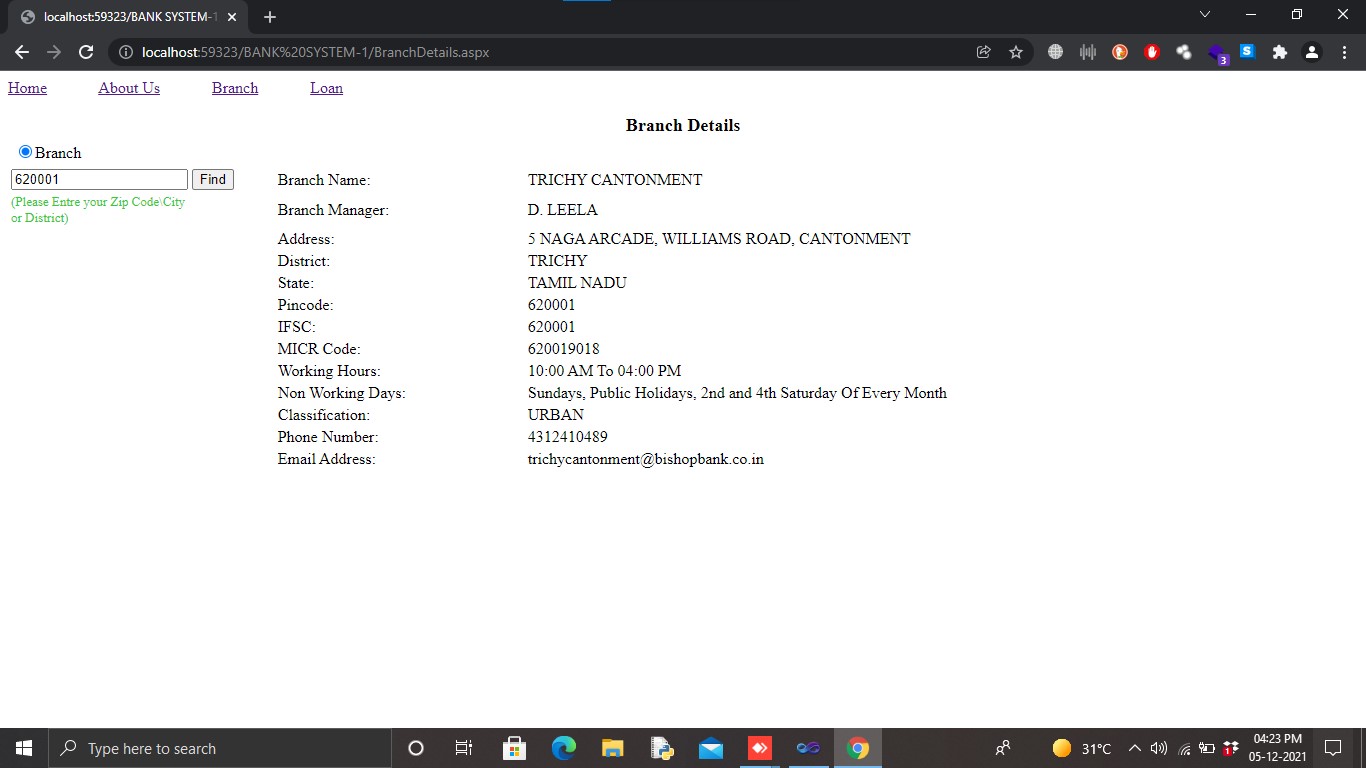




**Fig 6.2 Integration Testing**

**6.3 VALIDATION TESTNG**

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation testing ensures that the product actually meets the client’s needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

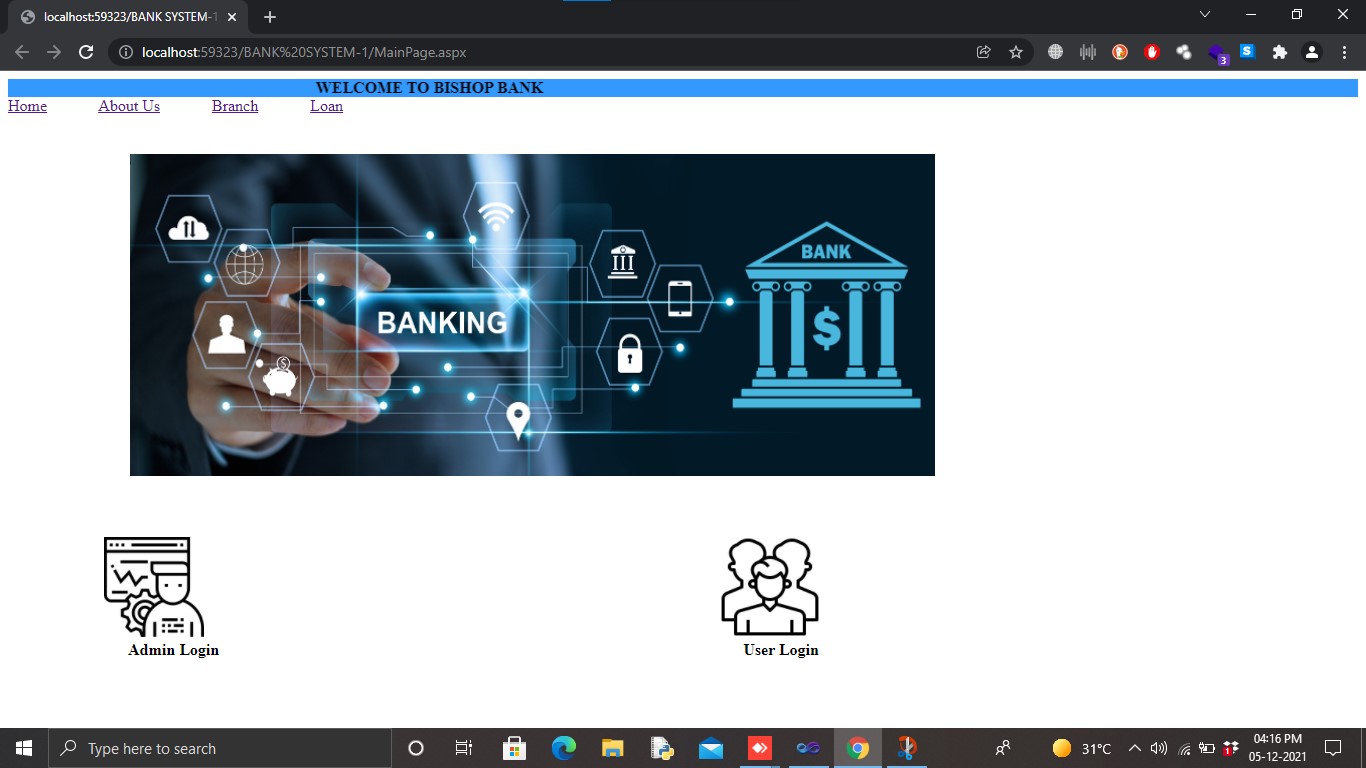


**Fig 6.3 Validation Testing**

1. **SYSTEM IMPLEMENTATION**

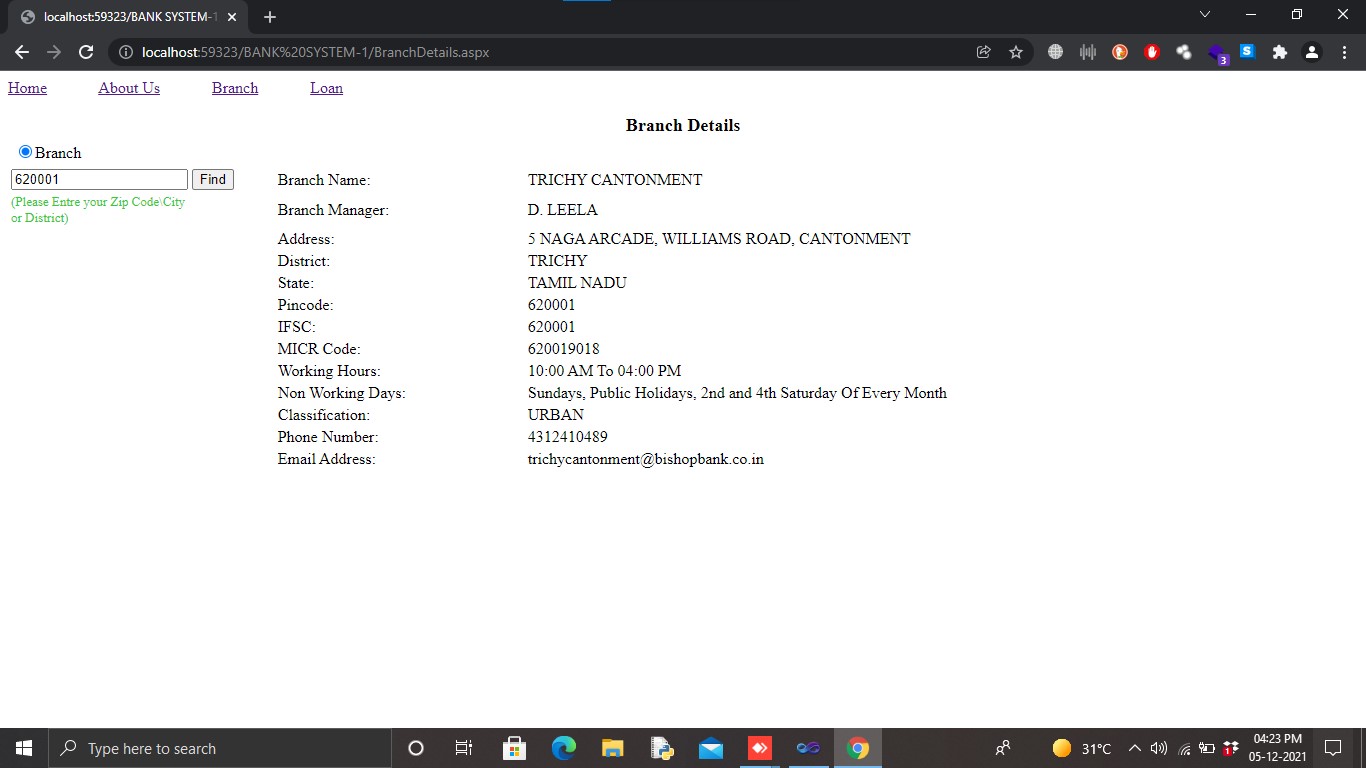
Implementation is the stage of the project when the theoretical design is turned out working system. Thus, it can be consider being the most critical stage m achieving successful new system and in giving the user, confidence that the new system will week and he effective The main objective of this user manual is to introduce the user manual is to introduce he user with the available facilities in the Download Manager It provides a conceptual overview of the functioning of the system and the detailed discussion and reports.

**7.1 SCREENSHOTS**

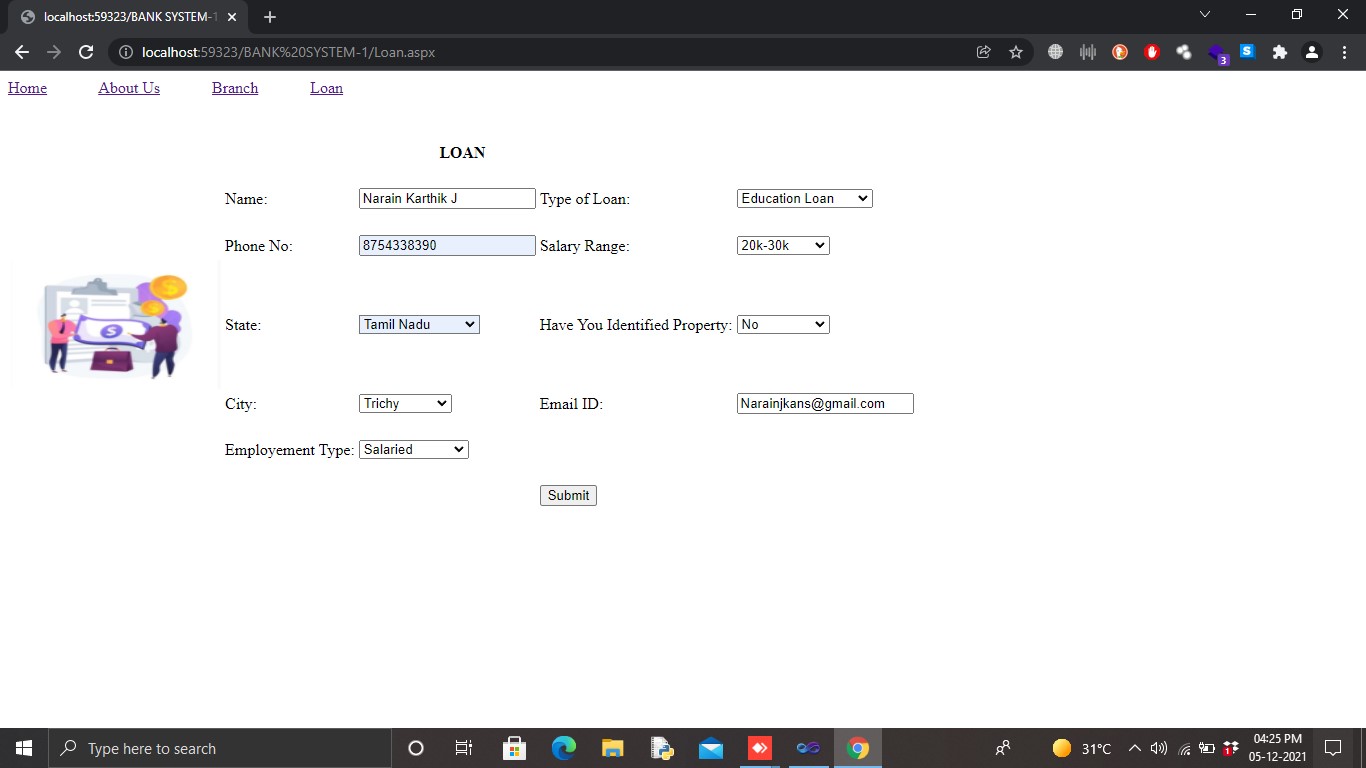


**Fig 7.1.1 Main Page**

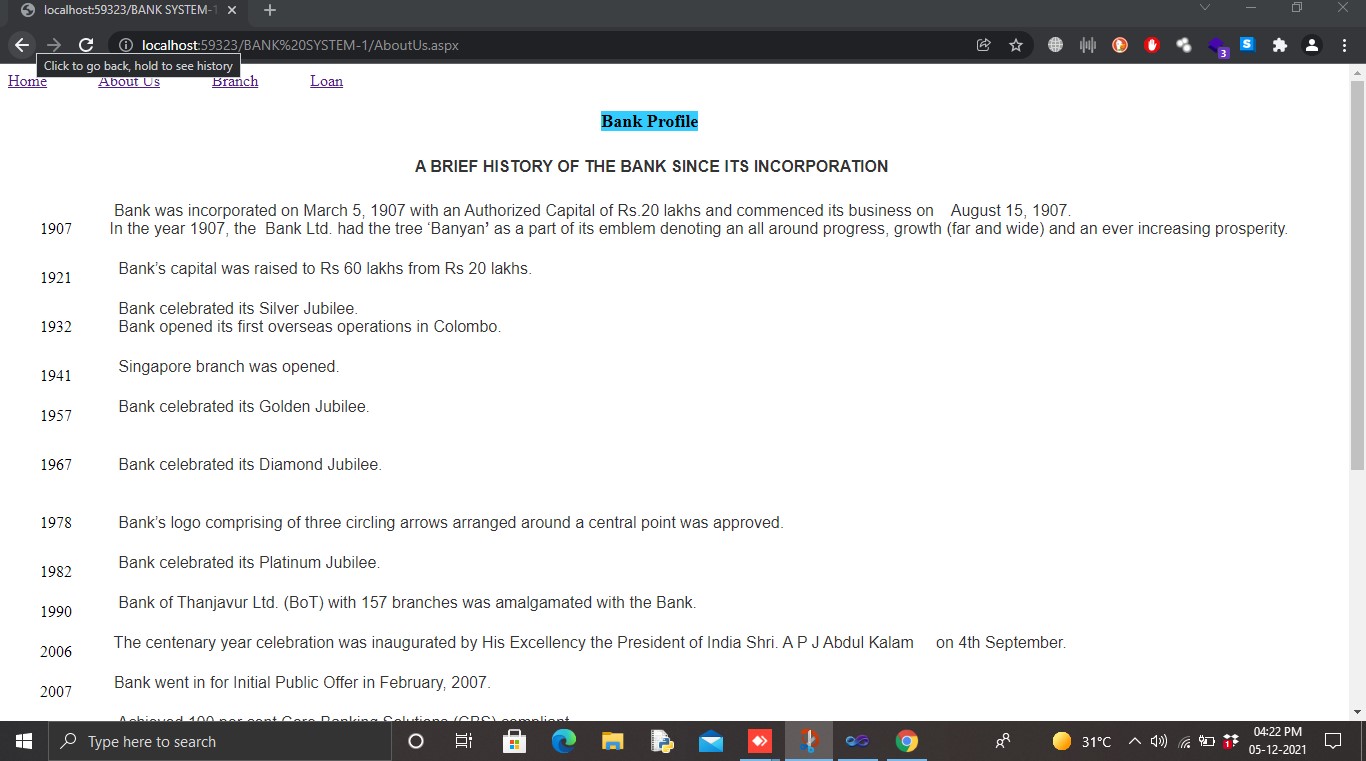
**MENU ITEMS:**



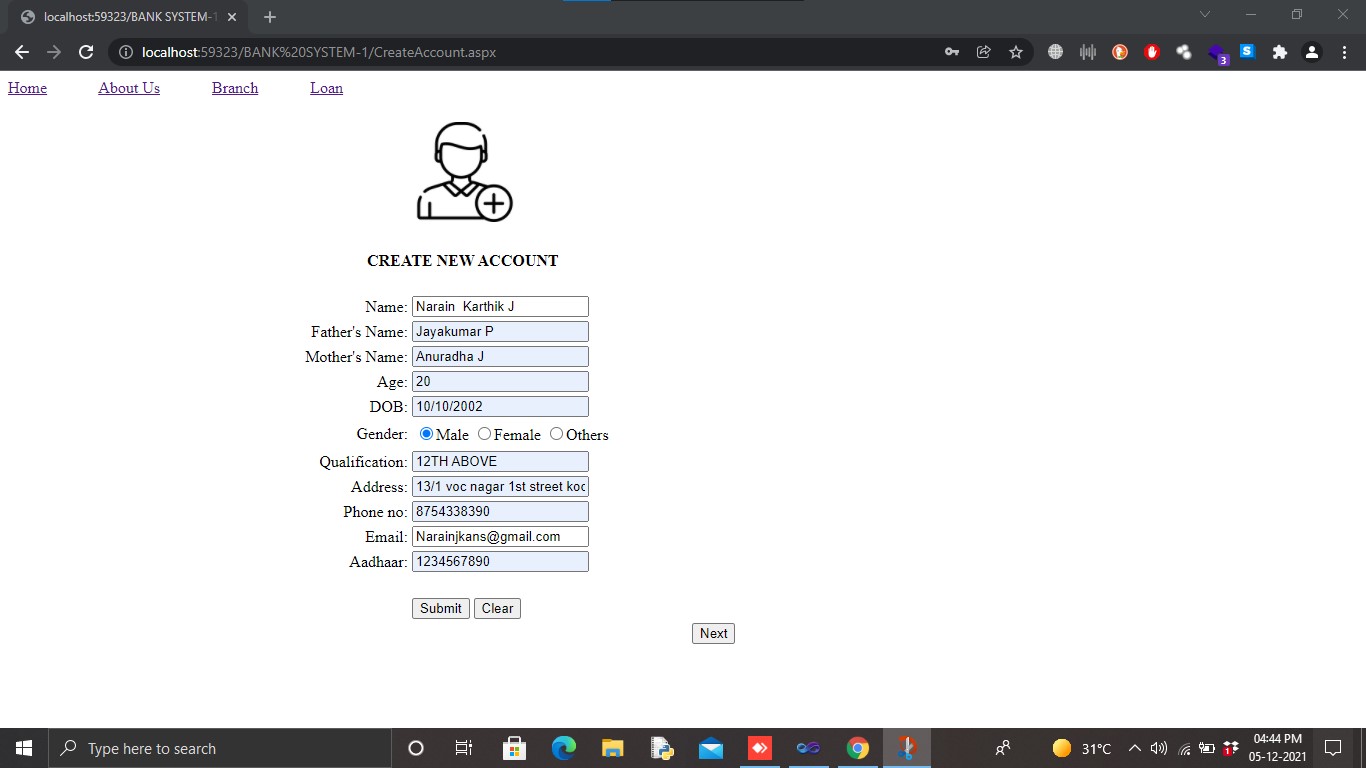
**Fig 7.1.2 Branch**



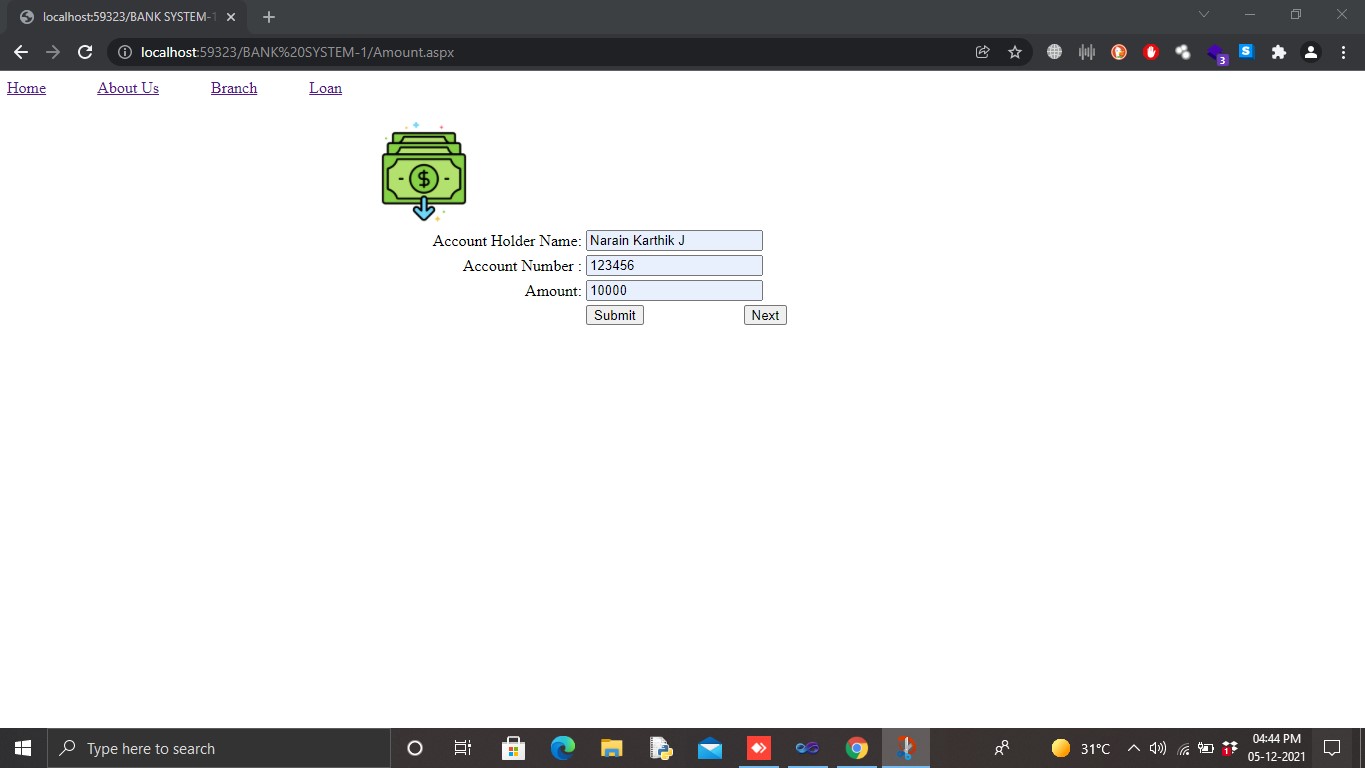
**Fig 7.1.3 Loan**



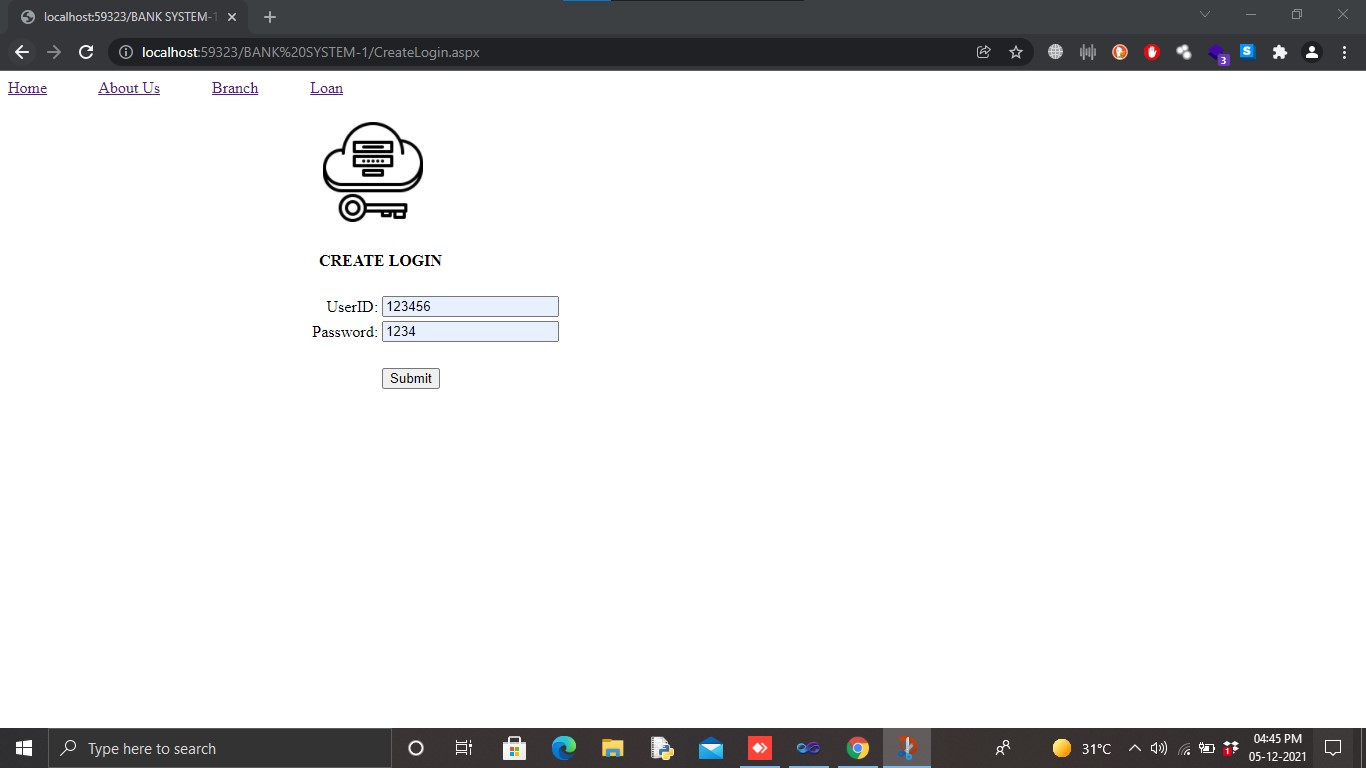
**Fig 7.1.4 About Us**



**Fig 7.1.5 Create Account**

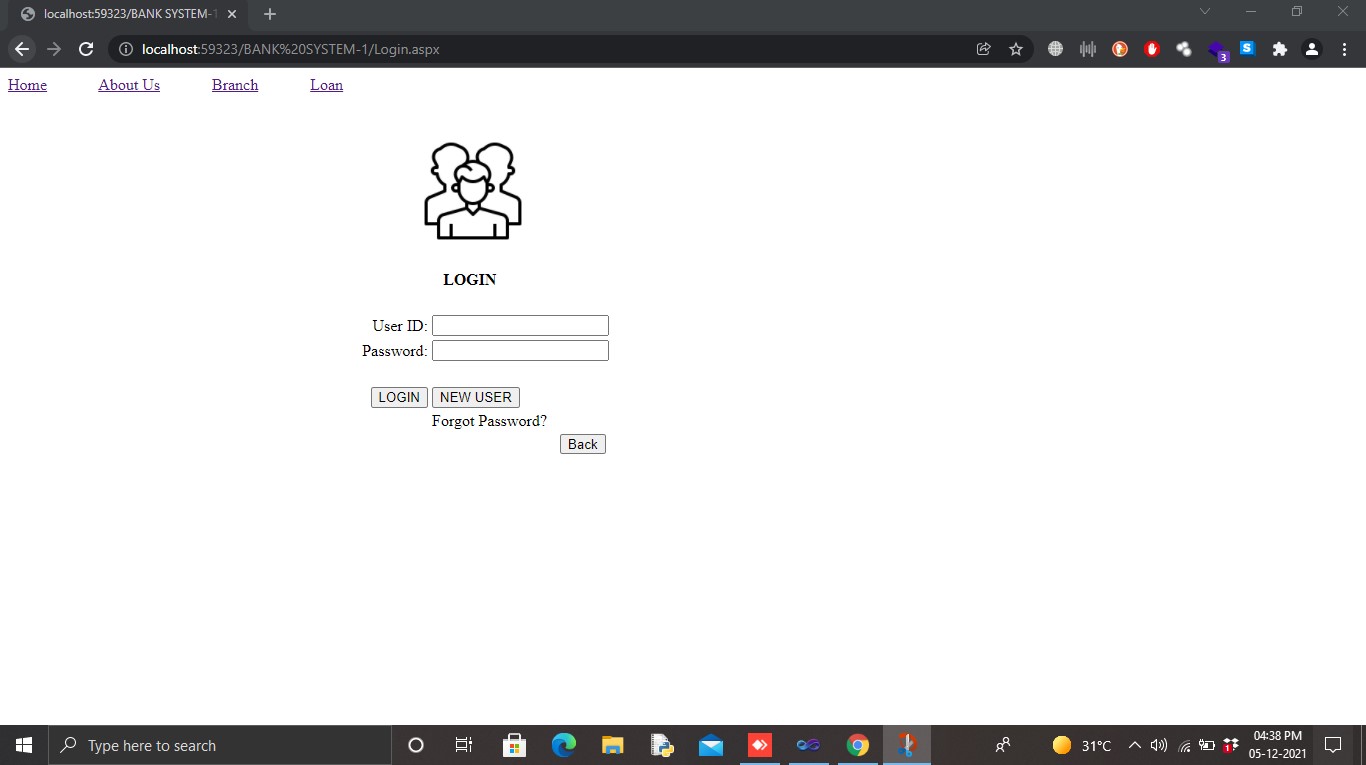


**Fig 7.1.6 Account Holder**

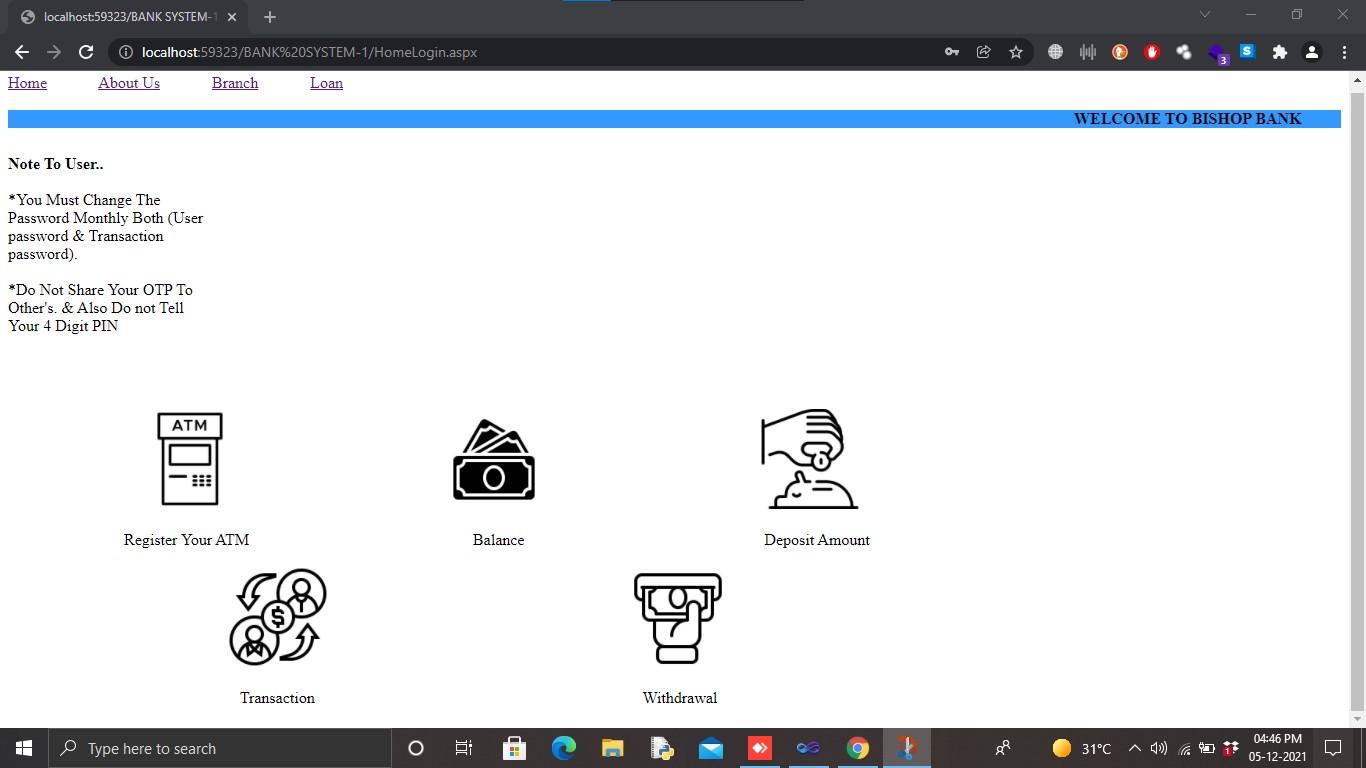


**Fig 7.1.7 Create Login**

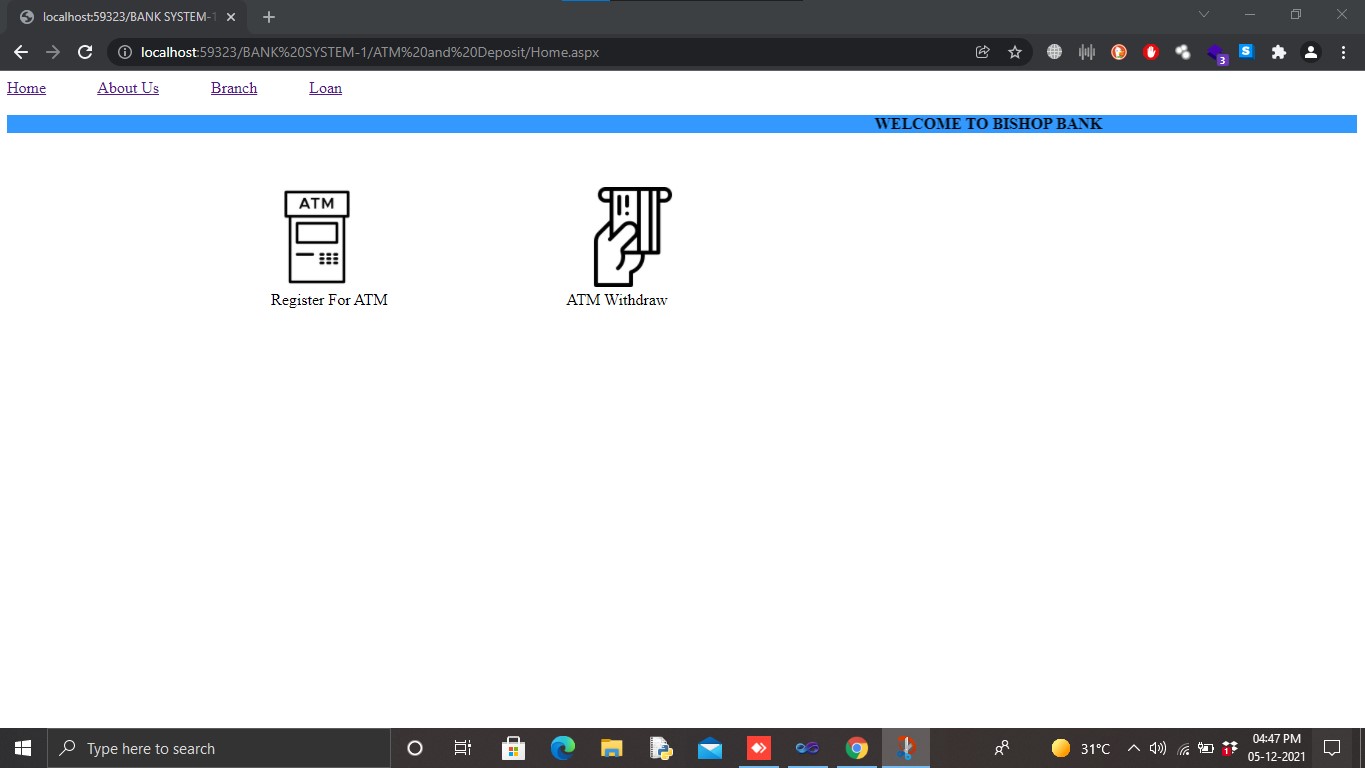
**USER:**



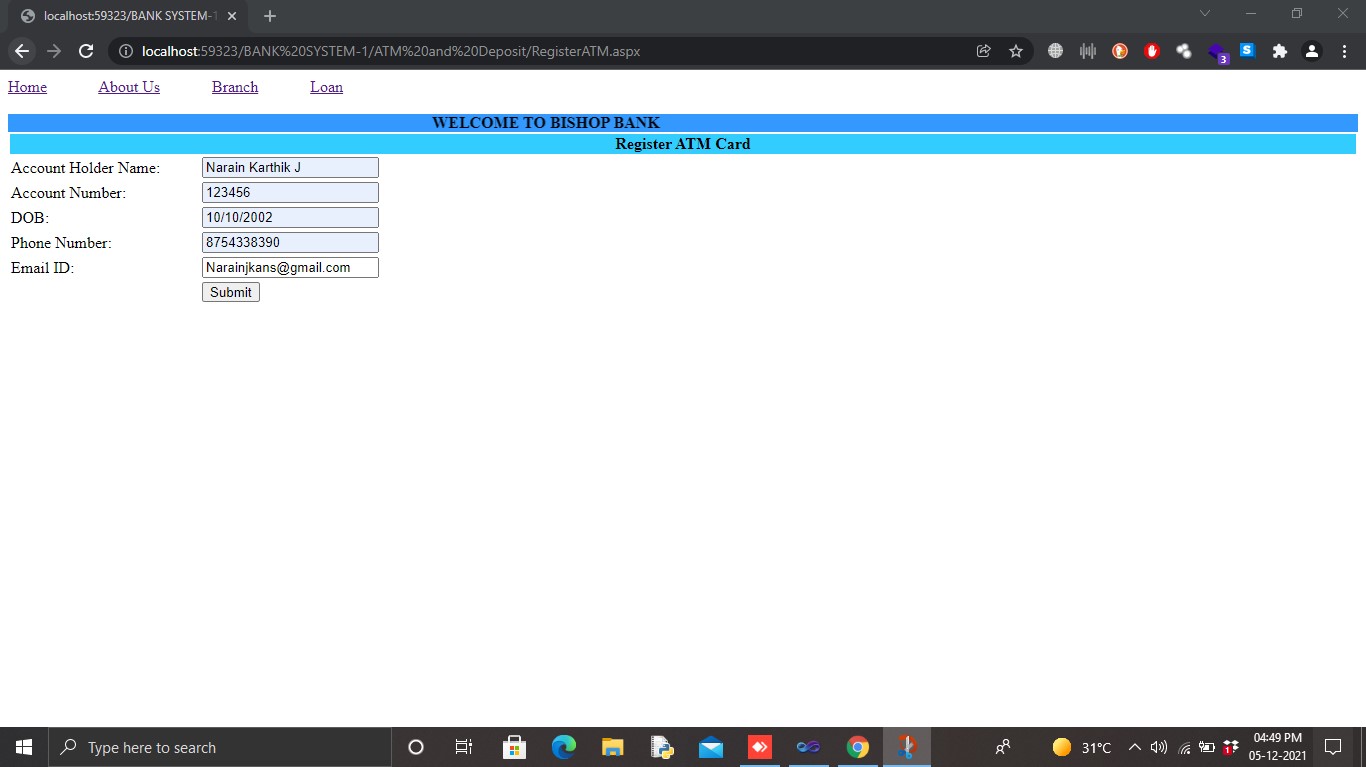
**Fig 7.1.8 User Login**



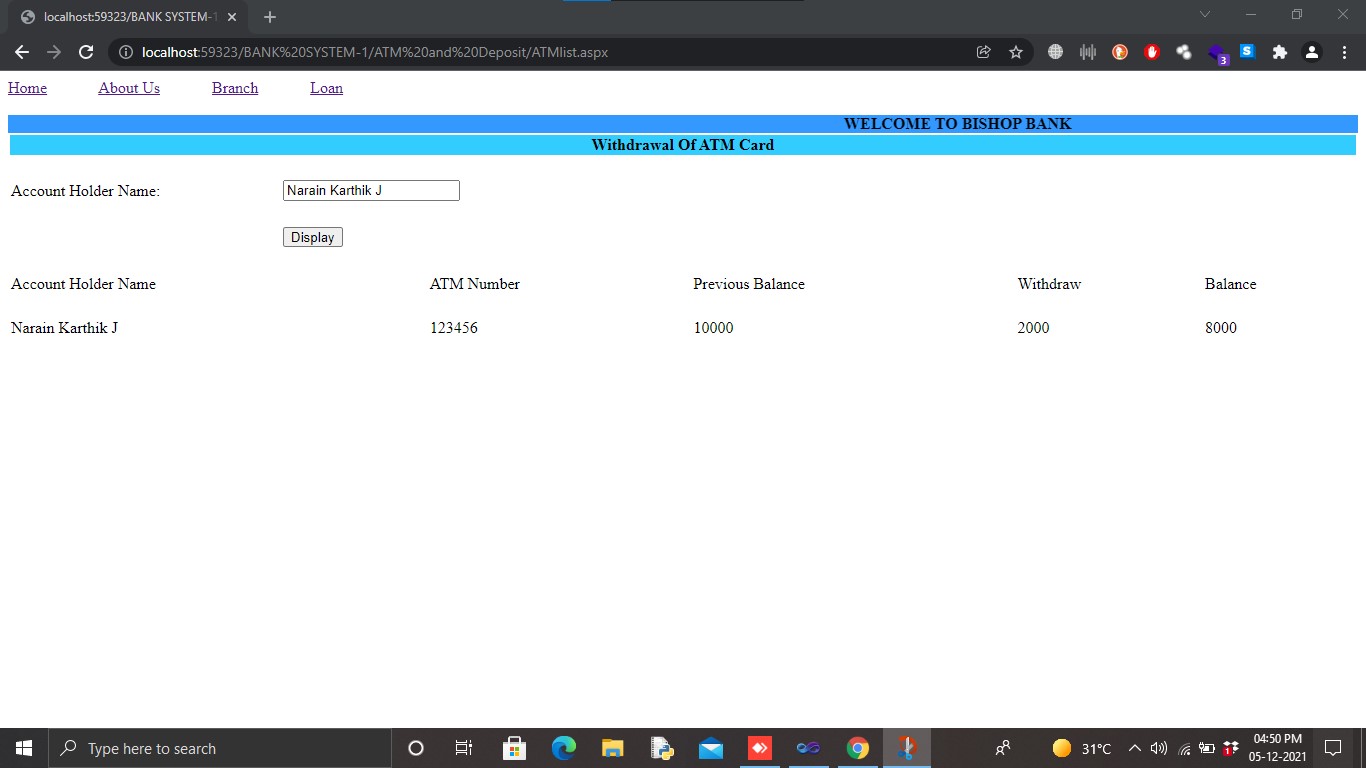
**Fig 7.1.9 User Homepage**



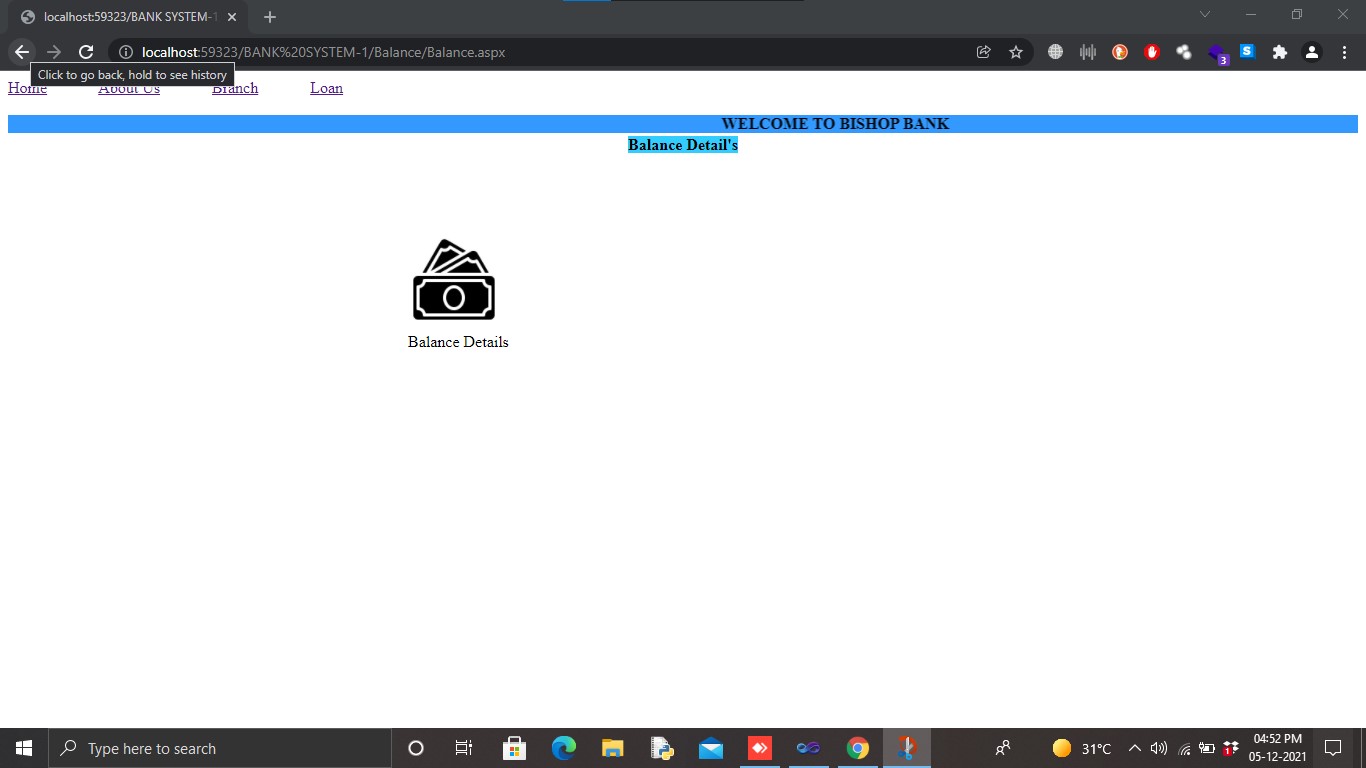
**Fig 7.1.10 ATM Home**

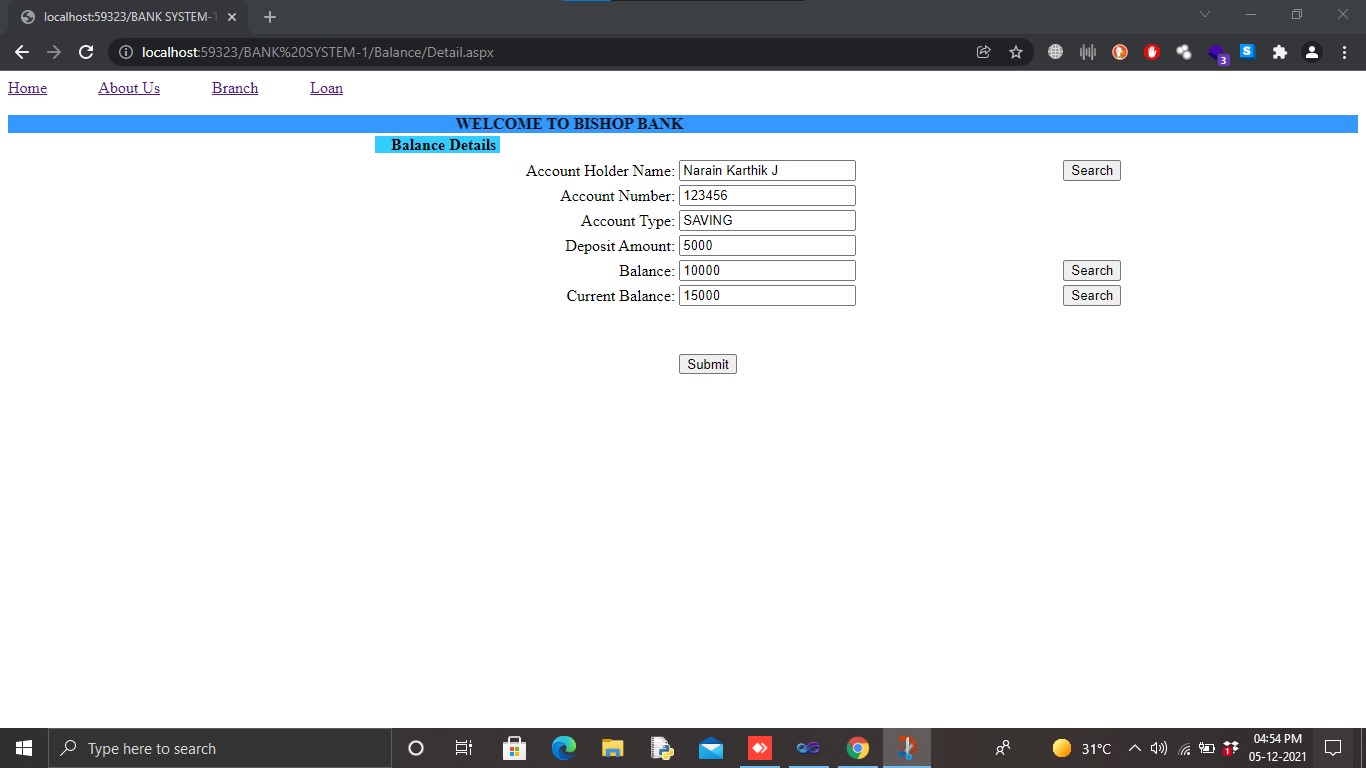


**Fig 7.1.11 Register ATM Card**

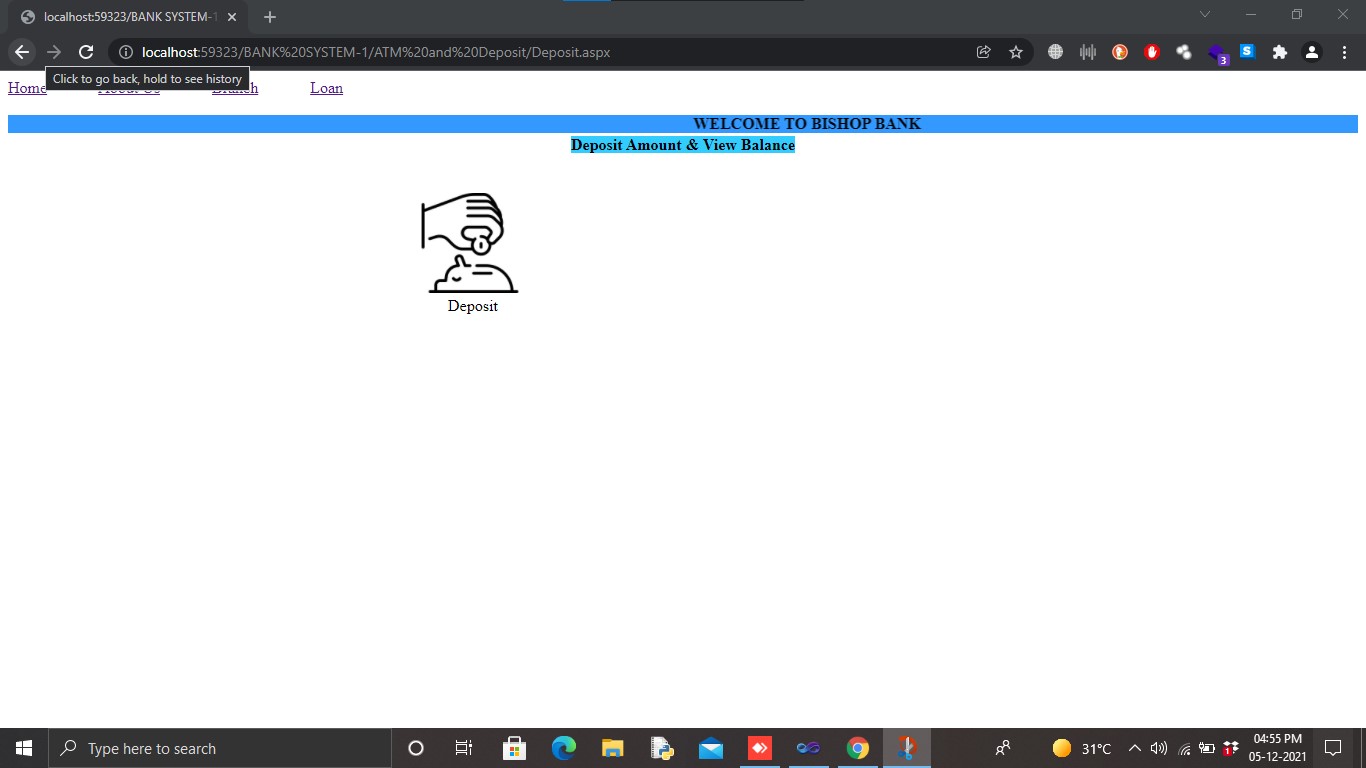


**Fig 7.1.12 Withdrawal of ATM Card**

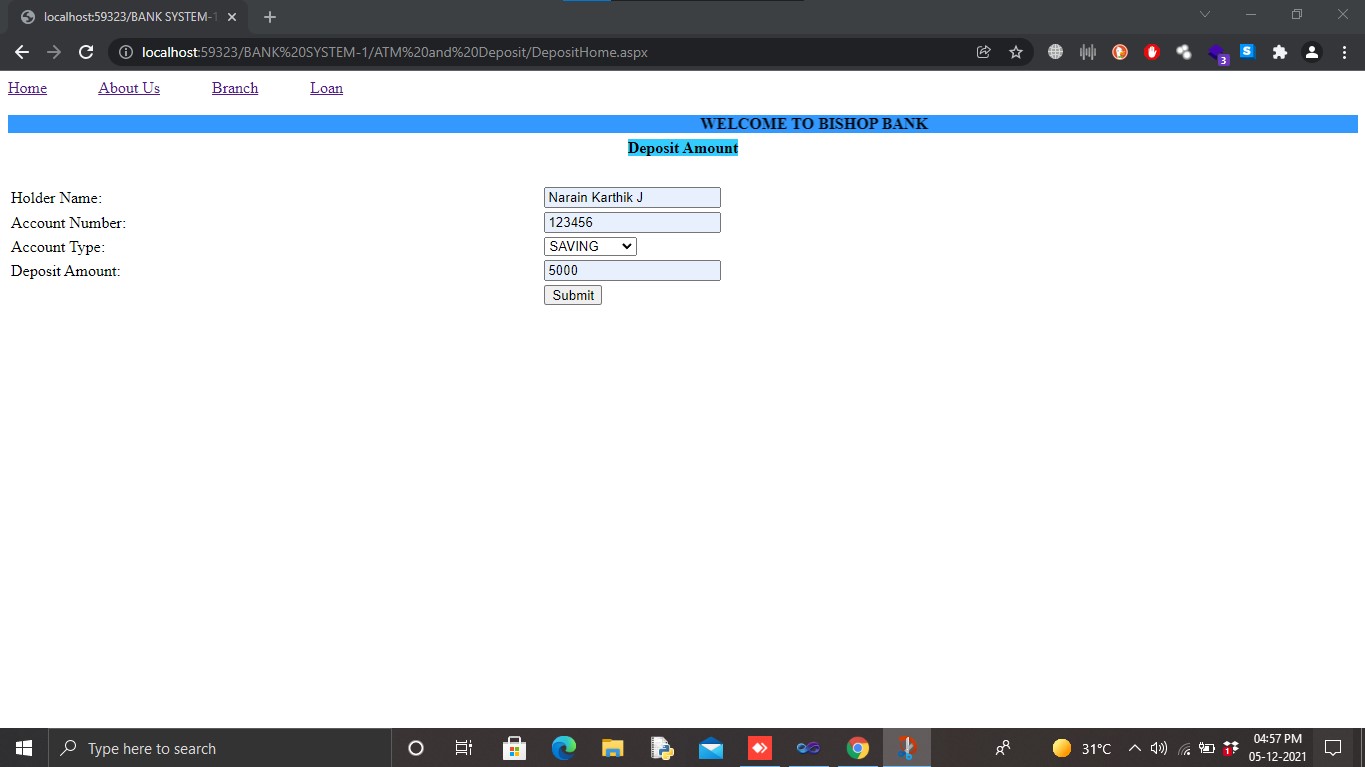
 **Fig 7.1.13 Balance Detail Page**



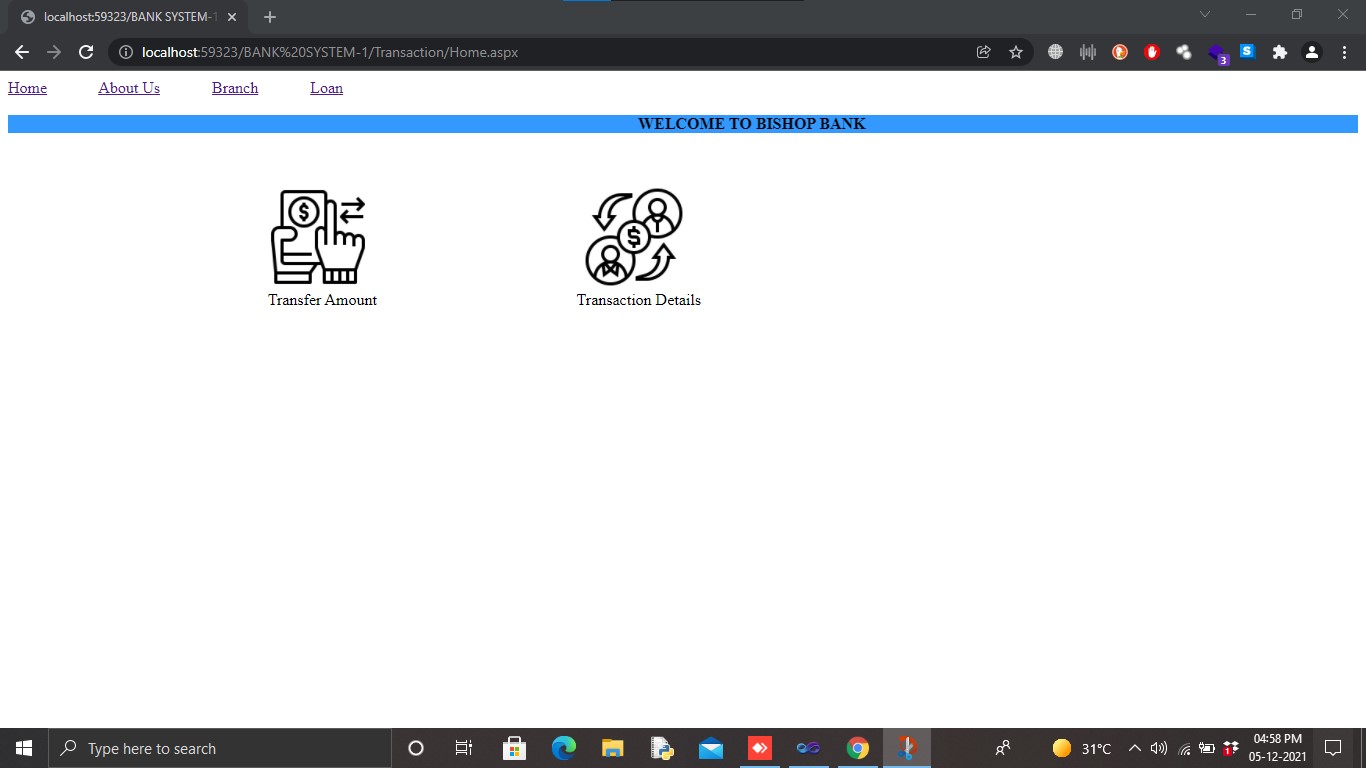
**Fig 7.1.14 Balance Details**



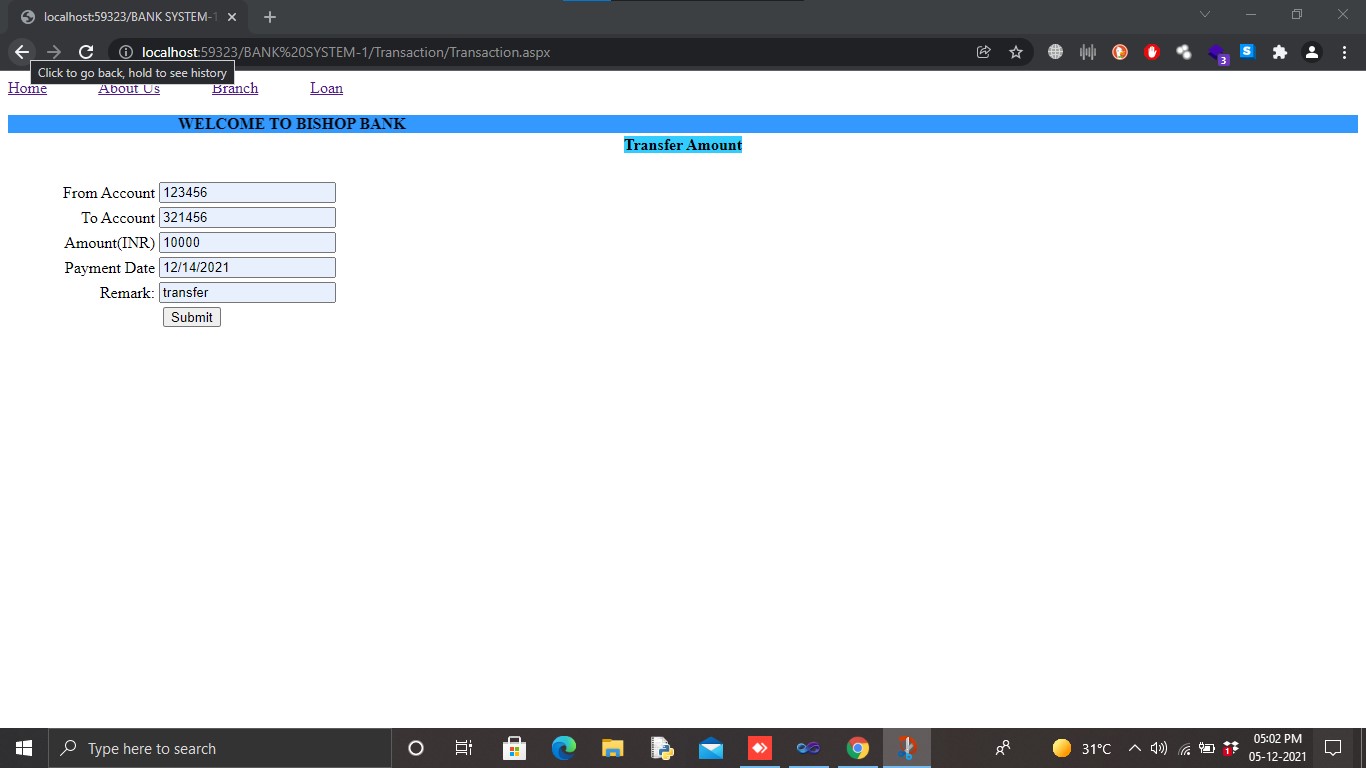
**Fig 7.1.15 Deposit Amount Page**



**Fig 7.1.16 Deposit Amount User**

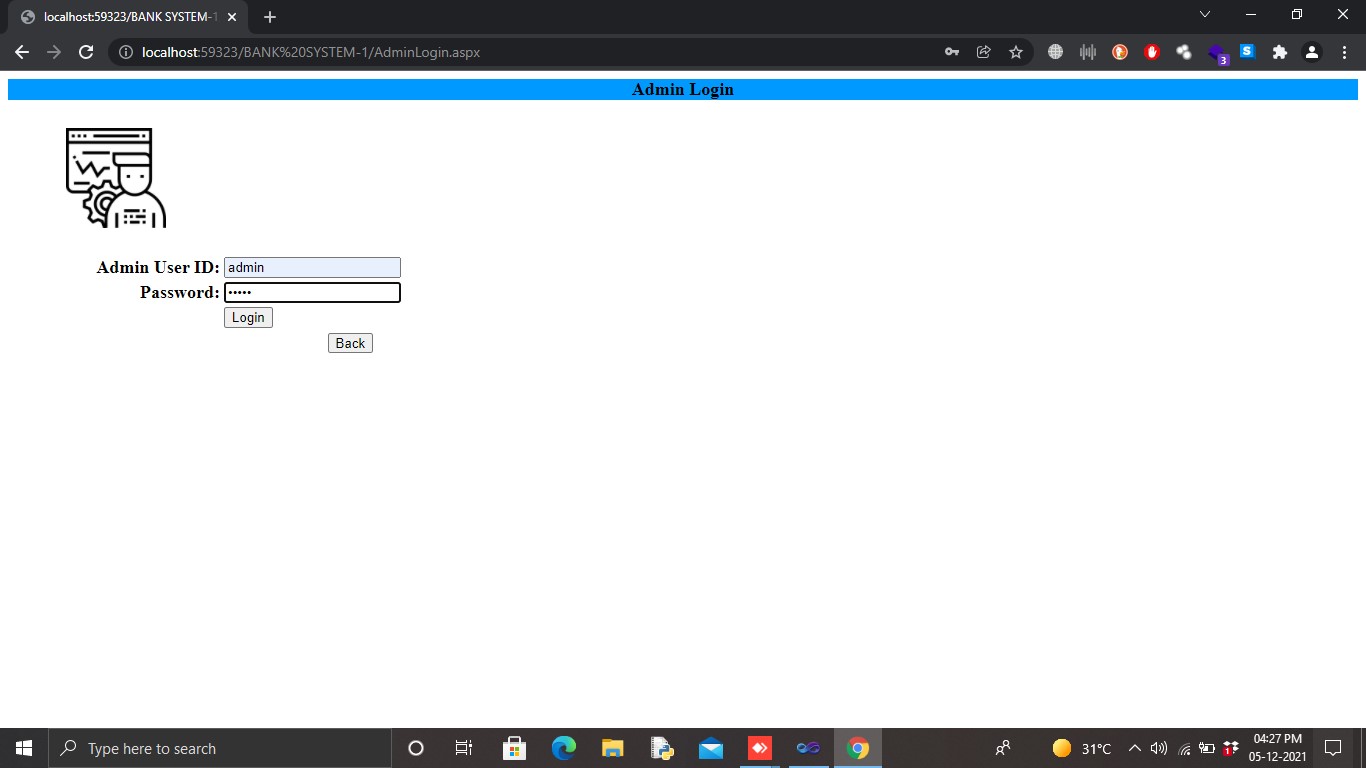


**Fig 7.1.17 Transaction Home**

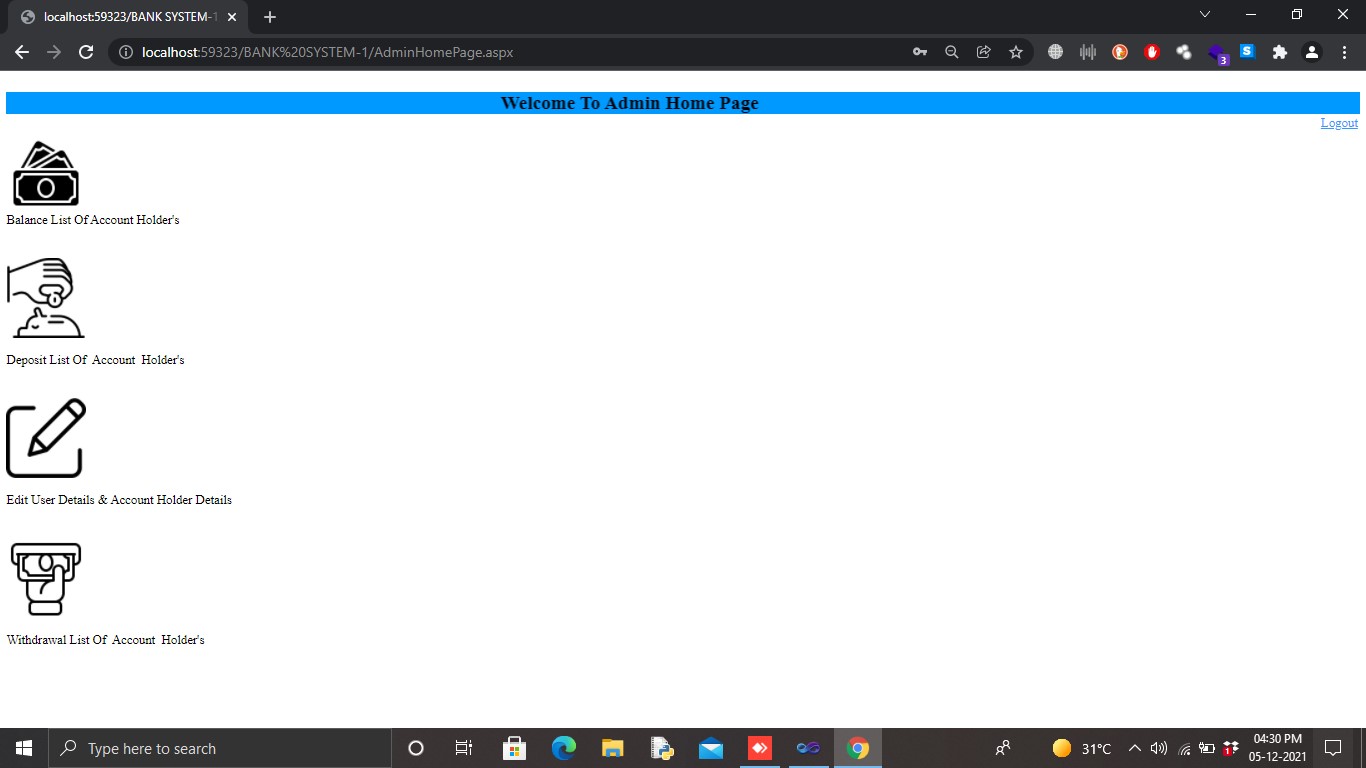


**Fig 7.1.18 Transaction Amount**

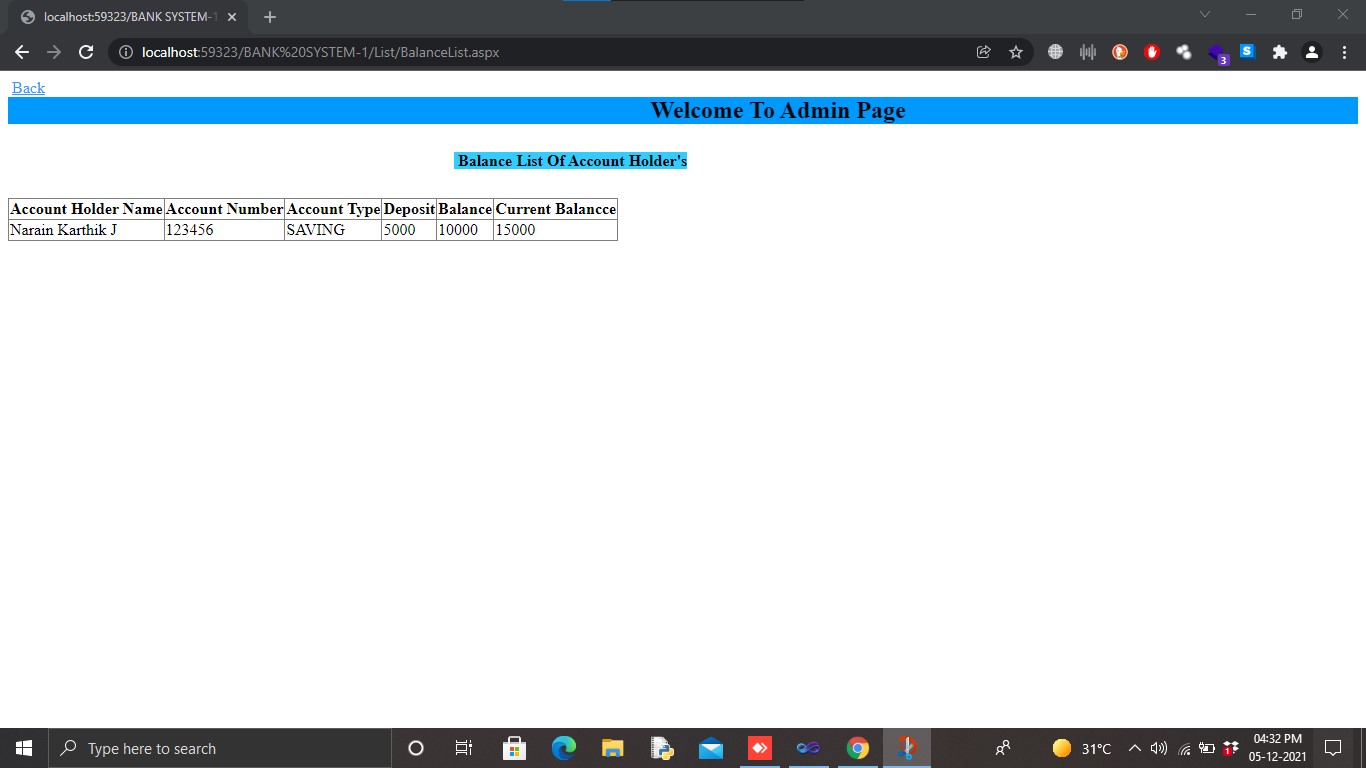
**ADMIN:**



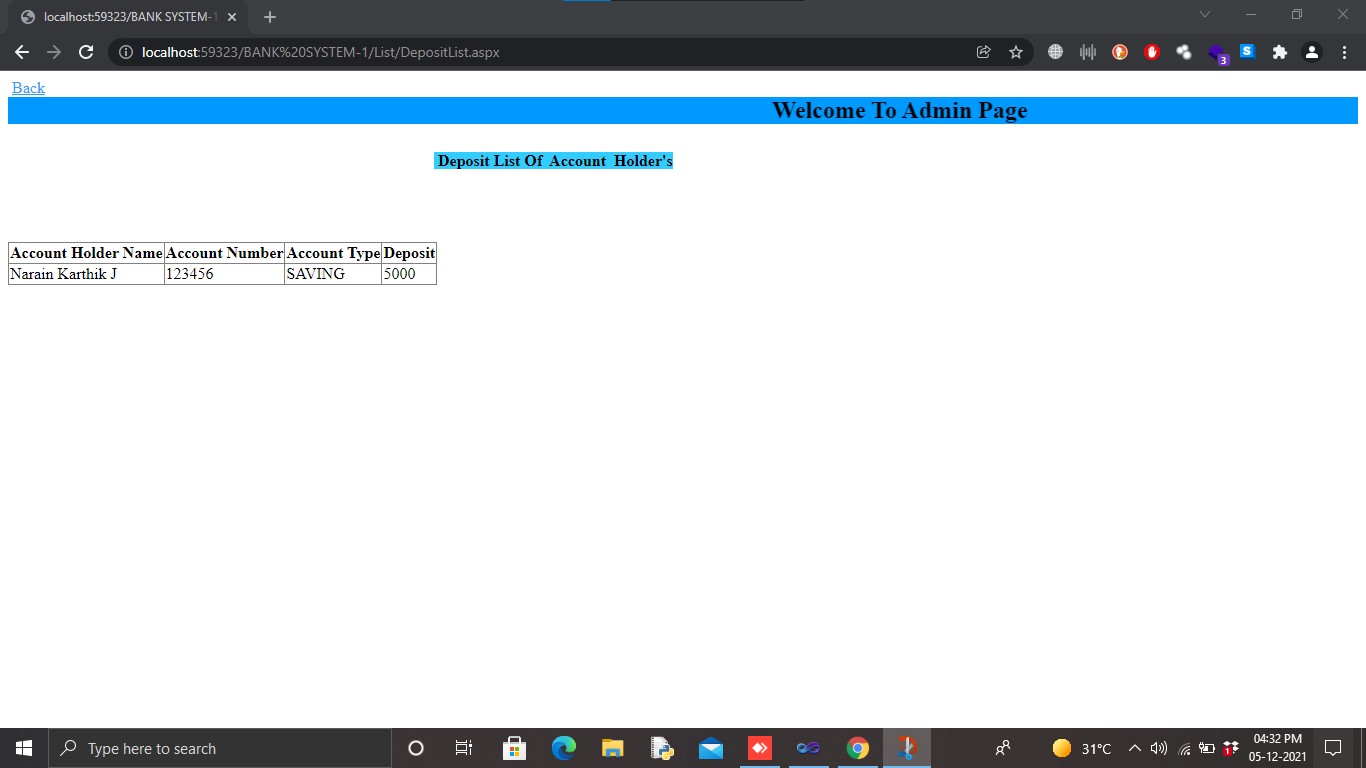
**Fig 7.1.19 Admin Login**



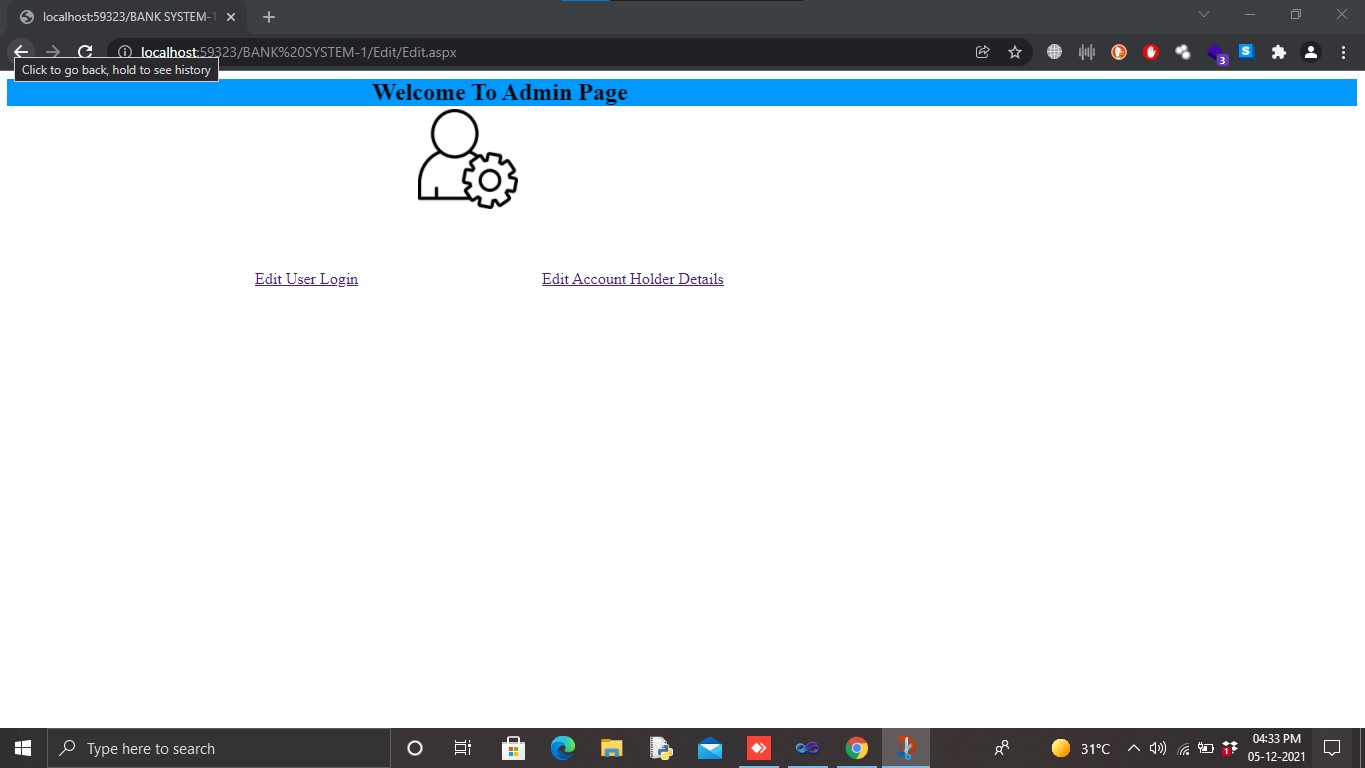
**Fig 7.1.20 Admin Home Page**



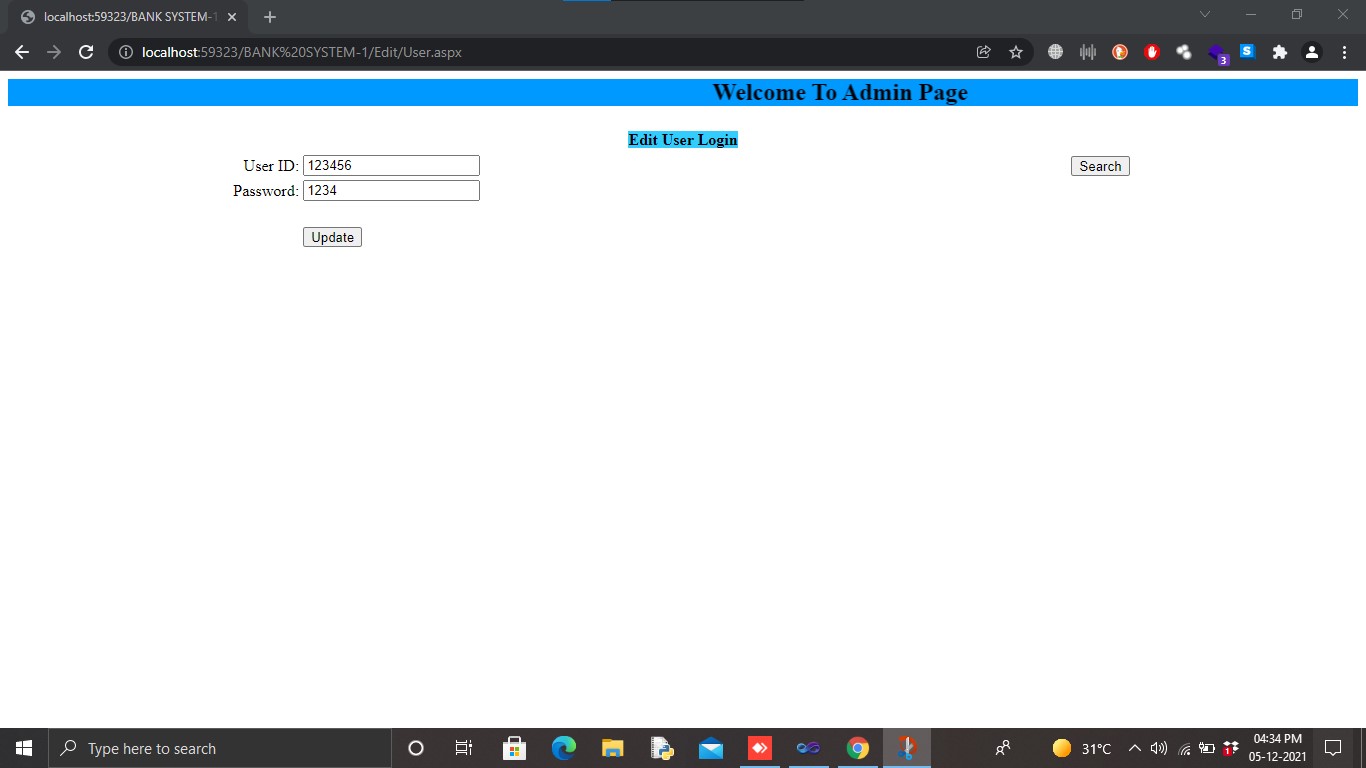
**Fig 7.1.21 Balance List**



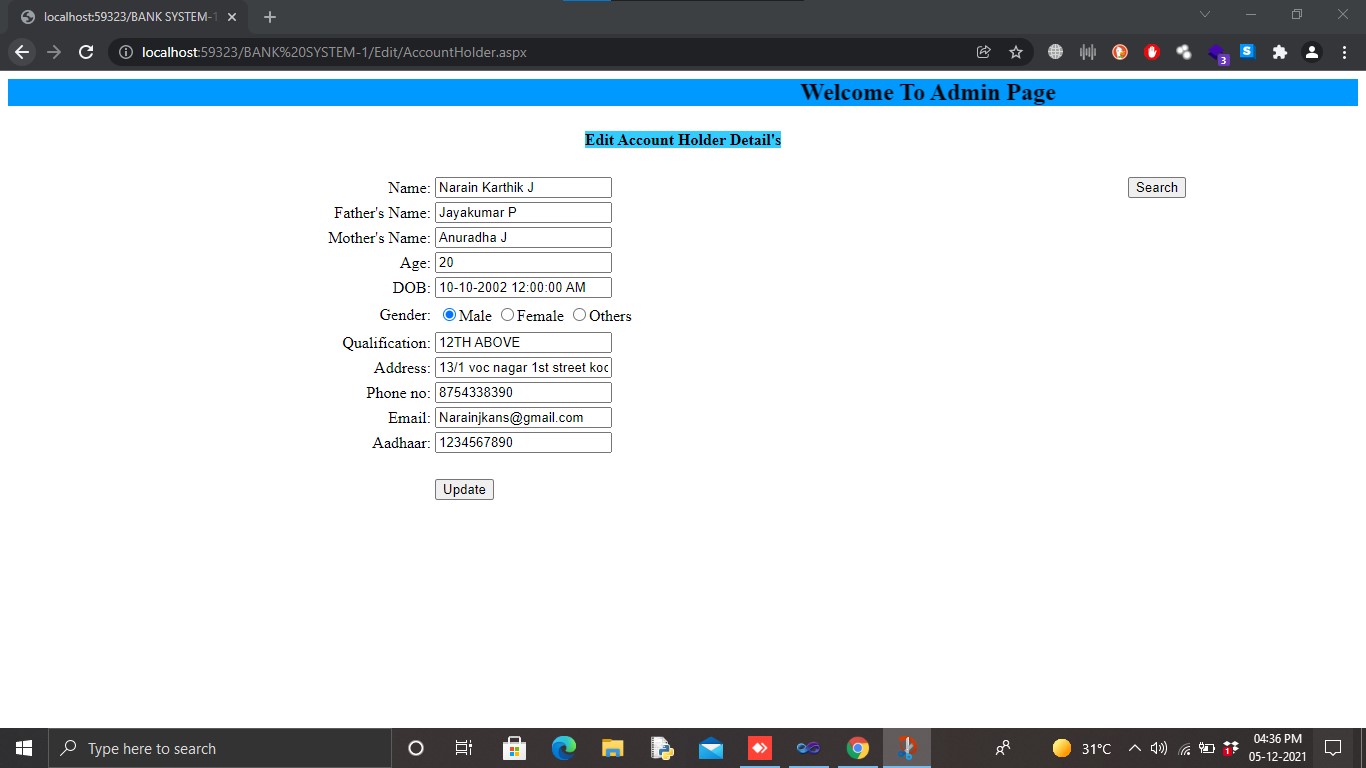
**Fig 7.1.22 Deposit List**



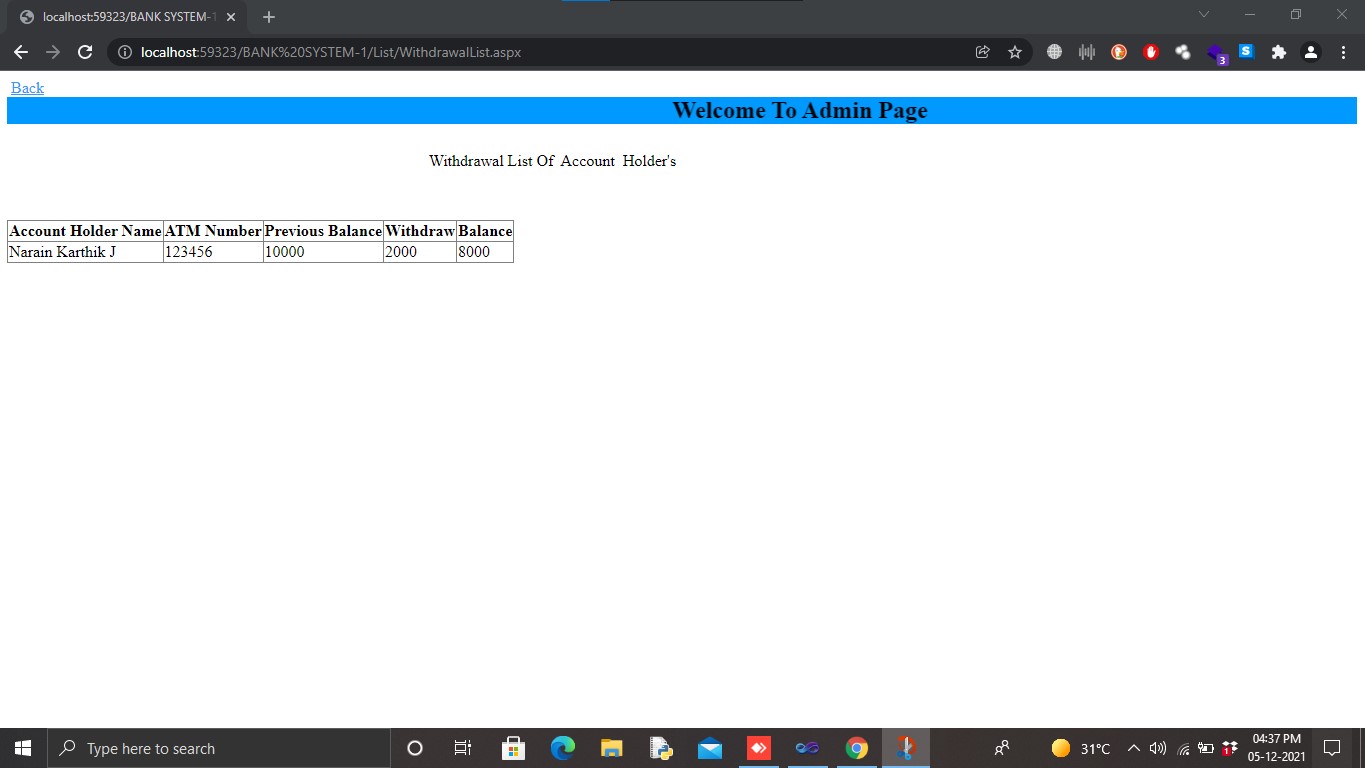
**Fig 7.1.23 Admin Page**



**Fig 7.1.24 Edit User Login**



**Fig 7.1.25 Edit Account Holder Details**



**Fig 7.1.26 Withdrawal of Account Holder**

1. **CONCLUSION**

The **“E-BANKING SYSTEM”** give all information in the bank management system. The whole process of banking is showed in the coding. The project is developed in the need of user in a banking sector by embedded all the tasks of transactions taking place in a bank. The main focus is project is to save the customer time which have the banking system. The maintenance of this record is stored in this data and we are providing this system, user will be satisfied with our service. It reduces time taken by the user to save the money and to reduce the manual work.

**BIBLIOGRAPHY**

**BOOK REFERENCES**

[1]. Beginning ASP.NET 3.5 Author - Matthew MacDonald

Publisher- Raybond Printers c# Year – 2008

[2]. Professional ASP.NET2.0 Author - Devin Rader, Scott Hansel c#

Publisher – Shivthik Year - 2006

**WEBREFERENCES:**

1. <https://www.javatpoint.com/asp-net-tutorial>
2. <https://stackify.com/learn-asp-net-tutorials/>