**MAJOR PROJECT ON:**

**( NETBANKING SYSTEM )**

**Major Project Submitted in Partial Fulfillment of the Requirements**

**for**

**the degree of**

**Bachelor of Computer Application**

**By:**

| **NAME** | **ROLL NO.** |
| --- | --- |
| **CHIRANJIT DAS** | **30901218081** |

**Under the guidance of**

**ARINDAM SENGUPTA**

****

**Techno India, Salt Lake**

**EM 4/ Salt Lake City, Sector V**

**KOLKATA – 700091**

**<2021>**

**Techno India, Salt Lake**

**(Maulana Abul Kalam Azad University of technology (WBUT) )**

**FACULTY OF BCA DEPARTMENT**

**Certificate of Recommendation**

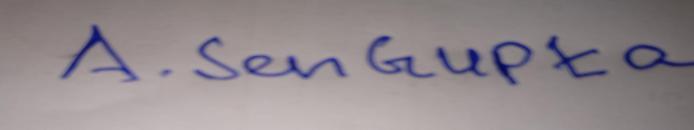
**This is to certify that Anik Raj Paul, Amrita Chakraborty, Koushik Das, Chiranjit Das, Deepon Mukhuty and Abhradip Chakraborty has completed his project work**

**titled “Majorproject on: Netbanking System”, under the direct supervision and guidance**

**of Arindam Sengupta. We are satisfied with his work, which is being presented for**

**the partial fulfillment of the degree of Bachelor of Computer Application**

**(BCA), West Bengal University of technology (WBUT), Kolkata– 700032.**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(Arindam Sengupta)**

**Date: 9th July 2021**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**H.O.D <Dr.Monalisa Banerjee>**

**Techno India Kolkata**

**Date: 9th July 2021**

**Maulana Abul Kalam Azad University of technology (WBUT),**

**FACULTY OF BCA DEPARTMENT**

**Certificate of Approval \***

**The foregoing Minor project is hereby approved as a creditable study of**

**Bachelor of Computer Application (BCA) and presented in a manner**

**satisfactory to warrant its acceptance as a pre-requisite to the degree for**

**which it has been submitted. It is understood that by this approval the**

**undersigned do not necessarily endorse or any statement made, opinion**

**expressed or conclusion therein but approve this Minor project only for**

**the purpose for which it is submitted.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature of the Examiners**

**PREFACE**

**Goal of the Minor project:**

**This project is aimed at developing an Online Banking for customer. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided.**

**The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of MySQL and all the user interfaces have been designed using the JAVA. The database connectivity is planned using the “Database” methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.**

**The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as MySQL.The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. The MySQL was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using the HTML 5. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations. The database connectivity was planned using the latest “ Database connection” technology provided by MySQL. The authentication and authorization was crosschecked at all the relevant stages. The user level accessibility has been restricted into two zones namely.**

**ACKNOWLEDGEMENT**

**I would like to express my special thanks of gratitude to our respected**

**teacher prof. Arindam Sengupta who gave us the golden opportunity to do**

**this wonderful project on the topic NETBANKING SYSTEM, which also helped us in doing a lot of research and we came**

**to know about so many new things I am really thankful to them.**

**Secondly, we would also like to thank our friends who helped us a lot in**

**finalizing this project within the limited time frame.**

**Table of Contents**

1. **INTRODUCTION**
   1. Overview
   2. About the Project
   3. Bank Profile
2. **SOFTWARE REQUIREMENT SPECIFICATIONS**
3. **SYSTEM DESIGN**
   1. ER-Diagrams
   2. DFDs
   3. Database Table
4. **CODING**
5. **OUTPUT SCREENS**
6. **SYSTEM TESTING** 
   1. Introduction
   2. Strategic Approach OF Software Testing
   3. Unit Testing
   4. Test
7. **CONCLUSION & FUTURE IMPROVMENT & BIBLIOGRAPHY**

**CHAPTER- 1**

**Introduction**

**1.1 Overview**

Internet Banking is all about knowing our customer need and provide them with the right service at the right time through right channel 24\*7 day a week.

Being “electronic”, it not only provides its customers with faster and better facilities, it even reduces the manual overhead of accounts maintenance.

**1.2 ABOUT THE PROJECT**

UCUCI BANK C.P. is one of the most prestigious BANKs in India. Founded as a Public BANK in 1972 in New Delhi, it is a private institution run by the Delhi Public BANK Society.

UCUCI BANK, C.P. is affiliated to the Central Board of Bank (CBB), which is the largest educational board in the country. It is recognized by the Department of Education, Govt. of NCT Delhi and the Ministry of HRD, Govt. of India. Over 5000 BANKs in India, with over 80,000 students, are members of the Board.

The BANK is also affiliated to the Indian Public BANKs' Conference (IPSC), and the National Progressive BANKs' Conference (NPSC). The members of these organizations include some of the premier BANKs in the country.

Life at DPSRKP centers on a shared commitment to academic excellence, intellectual growth, art, athletics, high standards of ethical awareness, sportsmanship, and community service. The BANK's traditions and accessibility to a broad curriculum add depth to each student’s life.  
The BANK upholds the founders' commitment to excellence in all fields, with emphasis on its motto Service Before Self.

**1.3 BANK PROFILE:**

UCUCI BANK, C.P. is a co-educational day-cum-boarding BANK, with approximately 9,500 customer on its rolls. These children, in the Junior and Senior branches, study in the three different campuses at East of Kailash, Vasant Vihar and C.P.

The BANK is among the most distinguished members of the Ravi Public BANK, C.P.. It is a path breaker in the pursuit of excellence. Its endeavor of integrating quality with quantity is reflected in the pivotal role it has played in the setting up of DPS Vasant Kunj, DPS Faridabad and DPS Manali at the national level. It has also promoted three BANKs abroad in Kuwait, Nepal and Indonesia. As their Linking BANK it also co-ordinates their activities.

The BANK has also extended its expertise further and in collaboration with the Government of Haryana, has taken up 3 BANKs in the under-privileged area of Mewat, to augment and enhance their standards and make them more conducive to teaming.

The BANK considers education to be a life-long process which should have a strong foundation. The goal of the BANK is to inculcate in the customer a love for learning and a desire to excel at every level. The BANK also aims at equipping the customer with the intellectual and practical skills that are necessary to meet the challenges in the future.

To sum up, the mission of UCUCI BANK, C.P. “to open doors and open minds” and prepare the ground for the future of the nation.

**CHAPTER- 2**

**Software Requirement & Specification**

**Software Required:**

The project is implemented in Core Java as it provides the implementation of Socket and Server Socket classes that are used to connect distinct applications, hence the software’s required in the creation and execution of the project are j2sdk1.8 or Eclipse .As we know JAVA is a platform independent language so this software runs with JRE environment on any desired platform i.e. Linux ,Mac, Windows 10 or any operating system.

**Hardware Required:**

As the project does not involve any database, its hardware requirements are minimal. Any System with Pentium P2 or above processor, 32MB RAM, 1GB Hard Disk, a LAN Card, and a CDROM is sufficient. Its network based software so computers connected with any kind of mode (wireless, LAN connected etc) will suit its requirements. . . . It can also be run on a single machine for its demo use.

Best suited in laboratory where we can run its server on any machine and many clients can use it simultaneously.

**Technologies and Requirement**

**IDE:**

Eclipse IDE for Enterprise Java and Web Developers

**Front End:**

JSP, JDBC, HTML,CSS

**Programming Language:**

JAVA

**Back End:**

MySQL

**CHAPTER- 3**

**System Design**

**5.1 E – R DIAGRAMS:**

* + The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
  + The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.
  + The set of primary components that are identified by the ERD are

◆ Data object ◆ Relationships

◆ Attributes ◆ Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.

**5.2 DATA FLOW DIAGRAMS:**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

****

**5.3 DATABASE TABLE:**

| **Column name** | **Data type** | **Nullable** | **Primary key** |
| --- | --- | --- | --- |
| USERNAME | VARCHAR | No | Yes |
| PASSWORD | VARCHAR | Yes | No |
| AMOUNT | VARCHAR | Yes | No |
| ADDRESS | VARCHAR | Yes | No |
| PHONE | VARCHAR | Yes | No |
| AADHAAR | VARCHAR | Yes | No |
| ACCOUNT TYPE | VARCHAR | Yes | No |
| GENDER | VARCHAR | Yes | No |

**CHAPTER- 4**

**CODING**

**loginBean.java**

package bean;

import java.sql.\*;

import connect.dbConnect;

public class loginBean

{

private String username;

private String password;

public String getPassword() {

return password;

}

public String getUsername() {

return username;

}

public void setPassword(String password) {

this.password = password;

}

public void setUsername(String username) {

this.username = username;

}

public String validate()

{

try

{

Connection con=dbConnect.getConnect();

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from record where username='"+username+"' and password='"+password+"'");

if(rs.next())

{

return "home.jsp";

}

else

{

return "index.jsp";

}

}catch(Exception e)

{

return "error.jsp";

}

}

}

**registerBean.java**

package bean;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import connect.dbConnect;

public class registerBean

{

private String username;

private String password;

private String amount;

private String address;

private String phone;

private String aadhaar;

private String accountType;

private String gender;

public void setAddress(String address) {

this.address = address;

}

public void setAmount(String amount) {

this.amount = amount;

}

public void setPassword(String password) {

this.password = password;

}

public void setPhone(String phone) {

this.phone = phone;

}

public void setUsername(String username) {

this.username = username;

}

public void setAadhaar(String aadhaar) {

this.aadhaar = aadhaar;

}

public void setAccountType(String accountType) {

this.accountType = accountType;

}

public void setGender(String gender) {

this.gender = gender;

}

public String getAddress() {

return address;

}

public String getAmount() {

return amount;

}

public String getPassword() {

return password;

}

public String getPhone() {

return phone;

}

public String getUsername() {

return username;

}

public String getAadhaar() {

return aadhaar;

}

public String insert()

{

try

{

Connection con=dbConnect.getConnect();

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from record where username='"+username+"'");

if(rs.next())

{

return "error";

}

int x=st.executeUpdate("insert into record values('"+username+"','"+password+"',"+amount+",'"+address+"','"+phone+"','"+aadhaar+"','"+accountType+"','"+gender+"')");

if(x>0)

{

return "register";

}

else

return "error.jsp";

}catch(Exception e)

{

return "error.jsp";

}

}

public String checkAccountNO(String username)

{

try

{

Connection con=dbConnect.getConnect();

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select username from record('"+username+"')");

while(rs.next())

{

username=rs.getString(1);

}

}

catch(Exception e)

{

e.printStackTrace();

}

return username;

}

public ResultSet getData(String username)

{

String sql="select \* from record where username='"+username+"'";

ResultSet rs = null;

try

{

Connection con=dbConnect.getConnect();

Statement st=con.createStatement();

rs=st.executeQuery(sql);

}

catch(SQLException e)

{

e.printStackTrace();

}

return rs;

}

}

**connect.dbConnect**

package connect;

import java.sql.Connection;

import java.sql.DriverManager;

public class dbConnect

{

public static Connection getConnect()

{

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bank\_db","root","CC@12Anik");

return con;

}catch(Exception e)

{

return null;

}

}

}

**register.jsp**

<%String name=(String)session.getAttribute("username");

String s=(String)session.getAttribute("register\_status");

if(name==null)

{

response.sendRedirect("index.jsp");

}

else

{

if(name.equals("admin"))

{

%>

<script type="text/javascript">

function check()

{

if(document.f1.username.value=="")

{

alert("plz enter username!");

return false;

}

if(document.f1.password.value=="")

{

alert("plz enter password!");

return false;

}

if(document.f1.password.value!=document.f1.repassword.value)

{

alert("password does not match!");

return false;

}

if(document.f1.amount.value=="")

{

alert("plz enter amount!");

return false;

}

if(document.f1.address.value=="")

{

alert("plz enter address!");

return false;

}

if(document.f1.phone.value=="")

{

alert("plz enter phone no.!");

return false;

}

if(document.f1.aadhaar.value=="")

{

alert("plz enter aadhaar no.!");

return false;

}

}

</script>

<div id="container">

<link href="css.css" rel="stylesheet" type="text/css"/>

<div id="header">

<img src="images/banklogo.jpg" width="500" height="150"/>

</div>

<div id="navigation">

<ul>

<li><a href="index.jsp">Home</a></li>

<li><a href="allaccounts.jsp">Show All Accounts</a></li>

<li><a href="register.jsp">Open Account</a></li>

<li><a href="deposit.jsp">Deposit</a></li>

<li><a href="delete.jsp">Close Account</a></li>

<li> <a href="logout.jsp">Logout</a></li>

</ul>

</div>

<div id="content-container">

<div id="content">

<center>

<%if(s.equals("register"))

{%>

<h2 style="color: red;">Account Opened!!</h2><br/>

<%session.setAttribute("register\_status","no\_status");

}

if(s.equals("error"))

{%><h2 style="color: red;">Account Already Exists!!</h2><br/><%

session.setAttribute("register\_status","no\_status");}%>

<br/><h2>Open Account</h2><br/><br/>

<form name="f1" action="registerBean.jsp" onsubmit="return check();">

<b>

Username: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="username"/><br/><br/>

Password: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="password" name="password"/><br/><br/>

Re-Password: &nbsp;<input type="password" name="repassword"/><br/><br/>

Amount: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="amount"/><br/><br/>

Address: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="address"/><br/><br/>

Phone No: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="phone"/><br/><br/><b>

Aadhaar No: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="aadhaar"/><br/><br/><b>

Account Type: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="radio" name="accountType" value="savings" checked> Savings

<input type="radio" name="accountType" value="current"> Current<br><br>

Gender: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type="radio" name="gender" value="male" checked> Male

<input type="radio" name="gender" value="female"> Female<br/><br/><b>

</b>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type="submit" value="Submit"/>

&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="Clear"/>

</form></center>

</div>

<!-- <div id="aside">

<p>\* Online banking is the practice of making bank transactions or paying bills via the Internet.

<br/><br/>\* Banking online allows a customer to make deposits, withdrawals, transfers with the click of a mouse.

<br/><br/>\* Online banking also eliminates paper waste, which is a plus not only for those who have to handle all the paper work, but also for the environment.

<br/><br/>\* Security is always an issue with Internet transactions. Although information is encrypted , and the chances of an account being hacked are slim, it happen.

</p>

</div> -->

</div>

<jsp:include page="footer.jsp"></jsp:include>

</div>

<%}

else

response.sendRedirect("index.jsp");

}

%>

**login.jsp**

<center>

<form name="f2" action="loginBean.jsp" onsubmit="return check();"><br>

<b>Username:</b> <input type="text" name="username"/><br><br>

<b>Password: </b>&nbsp;<input type="password" name="password"/><br><br>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" value="Submit"/>

&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="Clear"/>

</form></center>

<jsp:useBean id="t1" class="bean.loginBean" scope="session"/>

<jsp:setProperty property="\*" name="t1"></jsp:setProperty>

<% String s=t1.validate();

session.setAttribute("transfer\_status","null");

session.setAttribute("withdraw\_status","null");

session.setAttribute("delete\_status","null");

session.setAttribute("register\_status","null");

session.setAttribute("deposit\_status","null");

if(s.equals("home.jsp"))

{

session.setAttribute("username",t1.getUsername());

session.setAttribute("password",t1.getPassword());

response.sendRedirect("home.jsp");

}

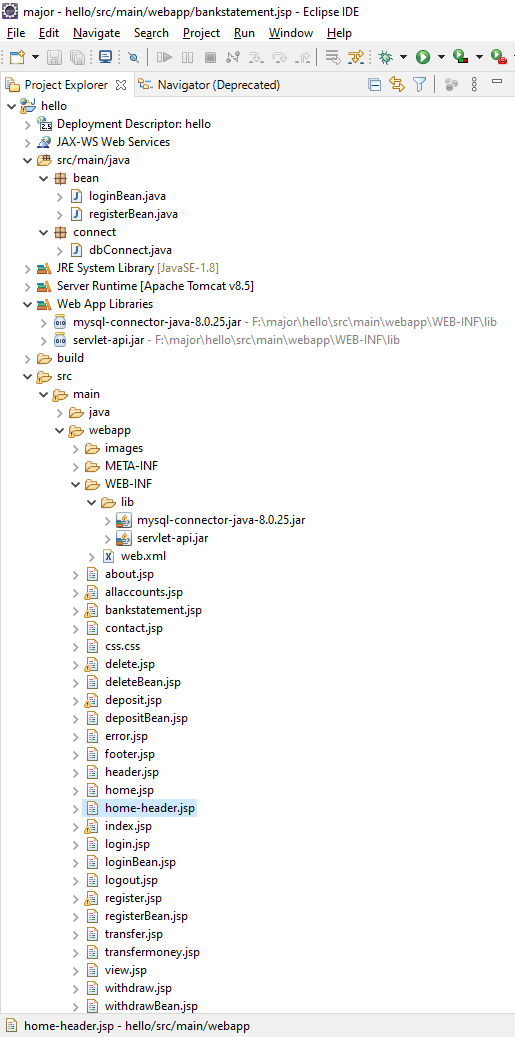
if(s.equals("error.jsp"))

response.sendRedirect("error.jsp");

if(s.equals("index.jsp"))

response.sendRedirect("index.jsp");

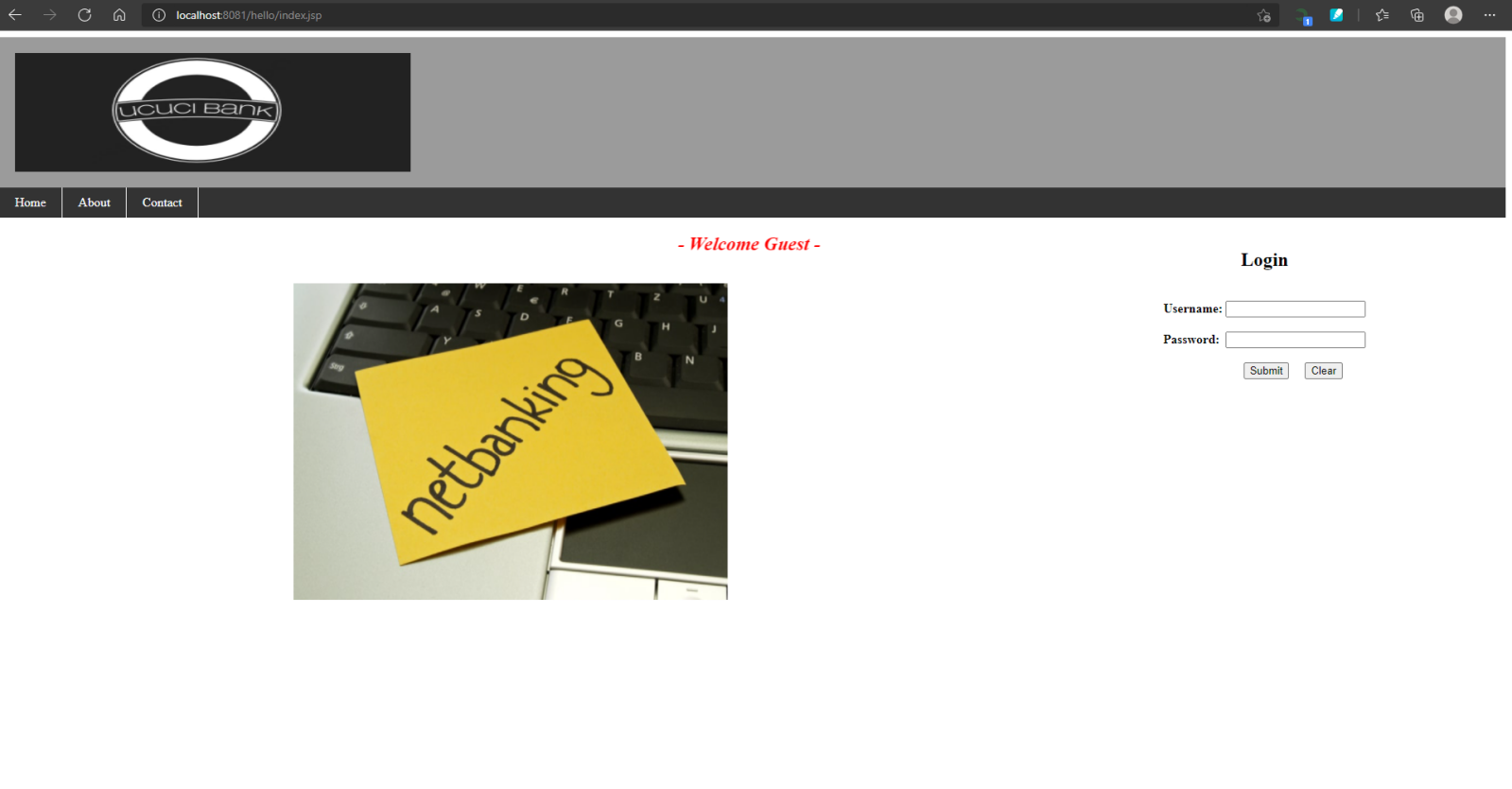
%>



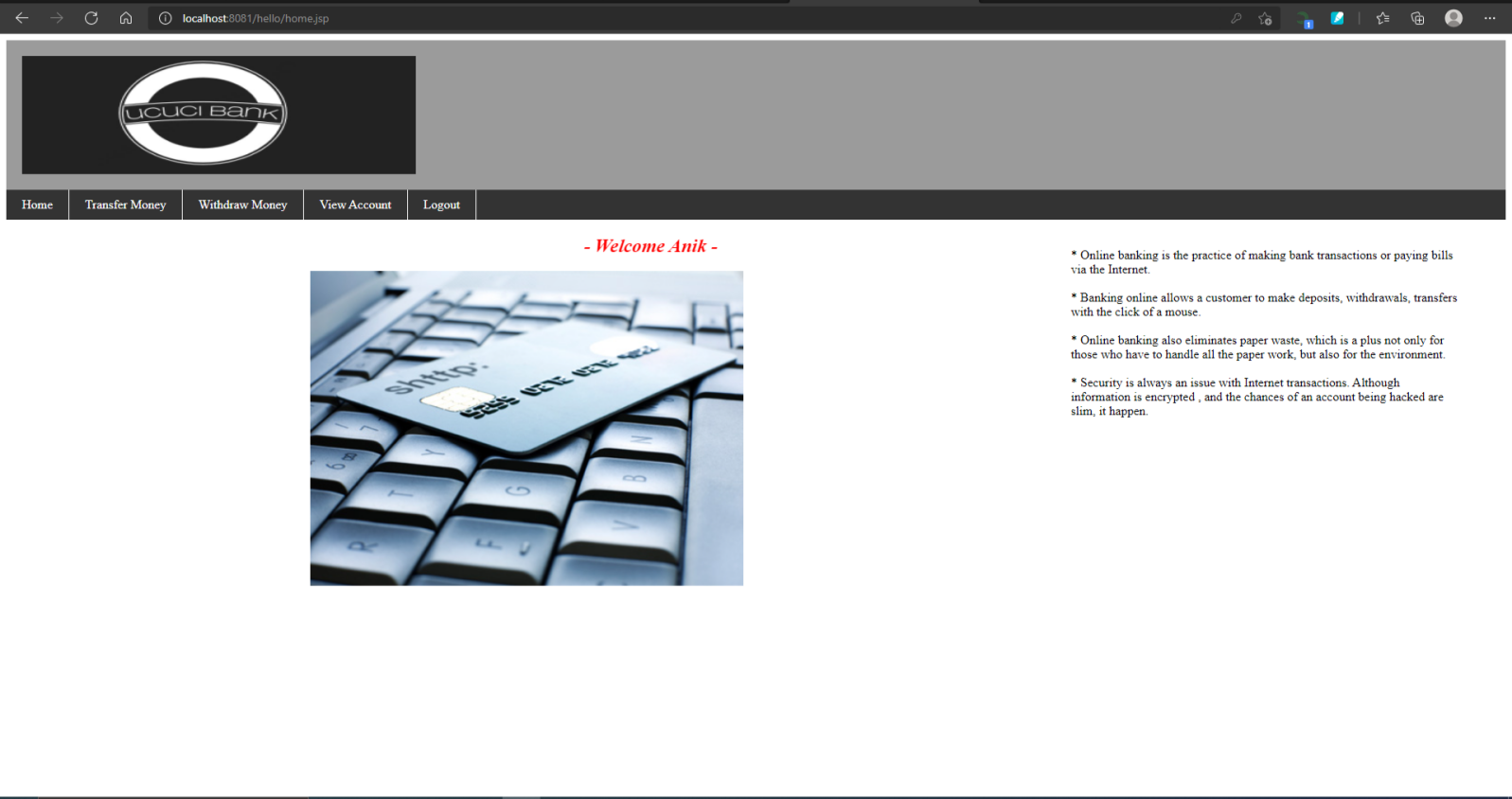
**CHAPTER- 5**

**OUTPUT SCREENS**

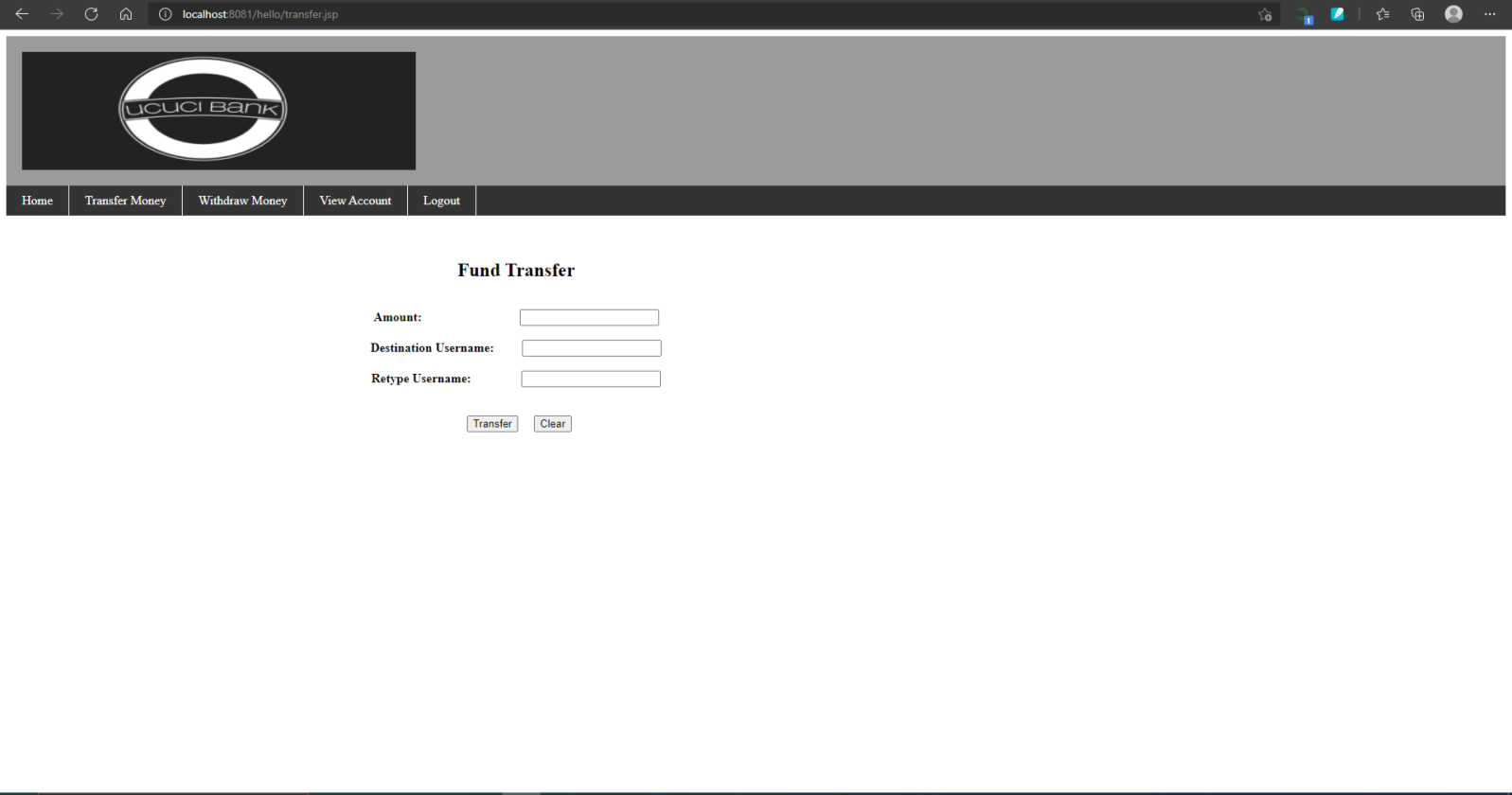
**WELCOME PAGE**



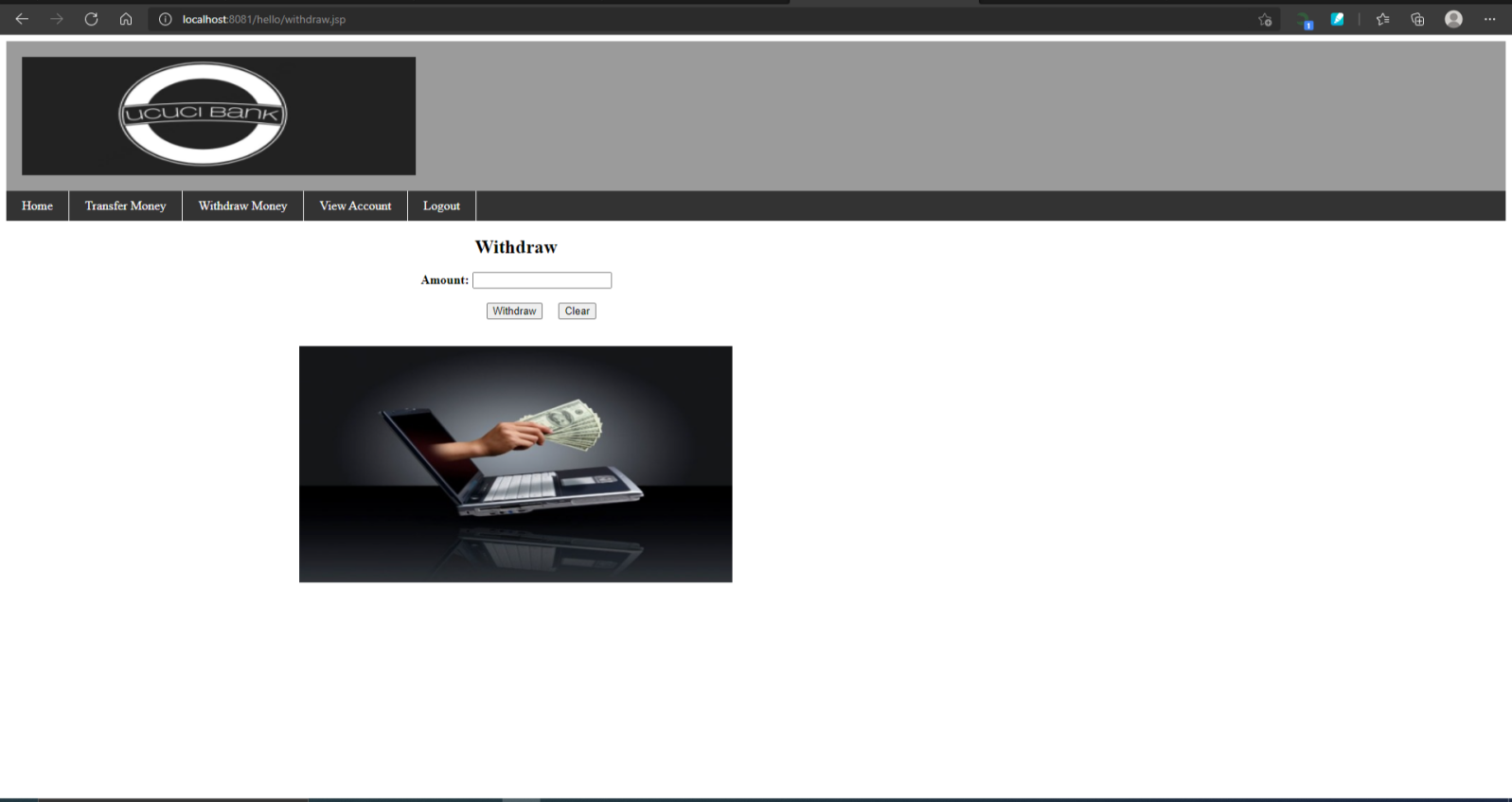
**USER LOGIN**

****

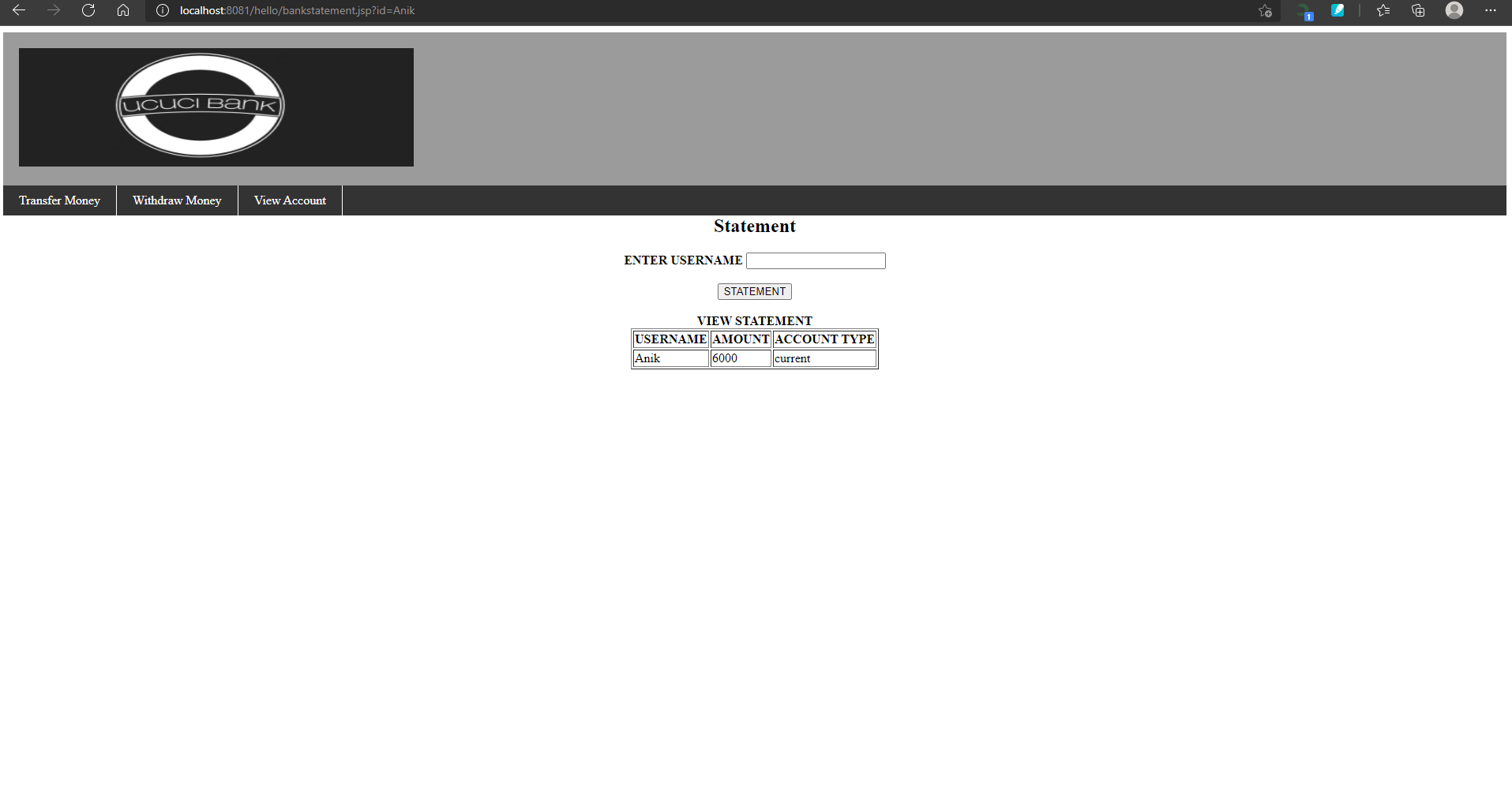
**FUND TRANSFER**

****

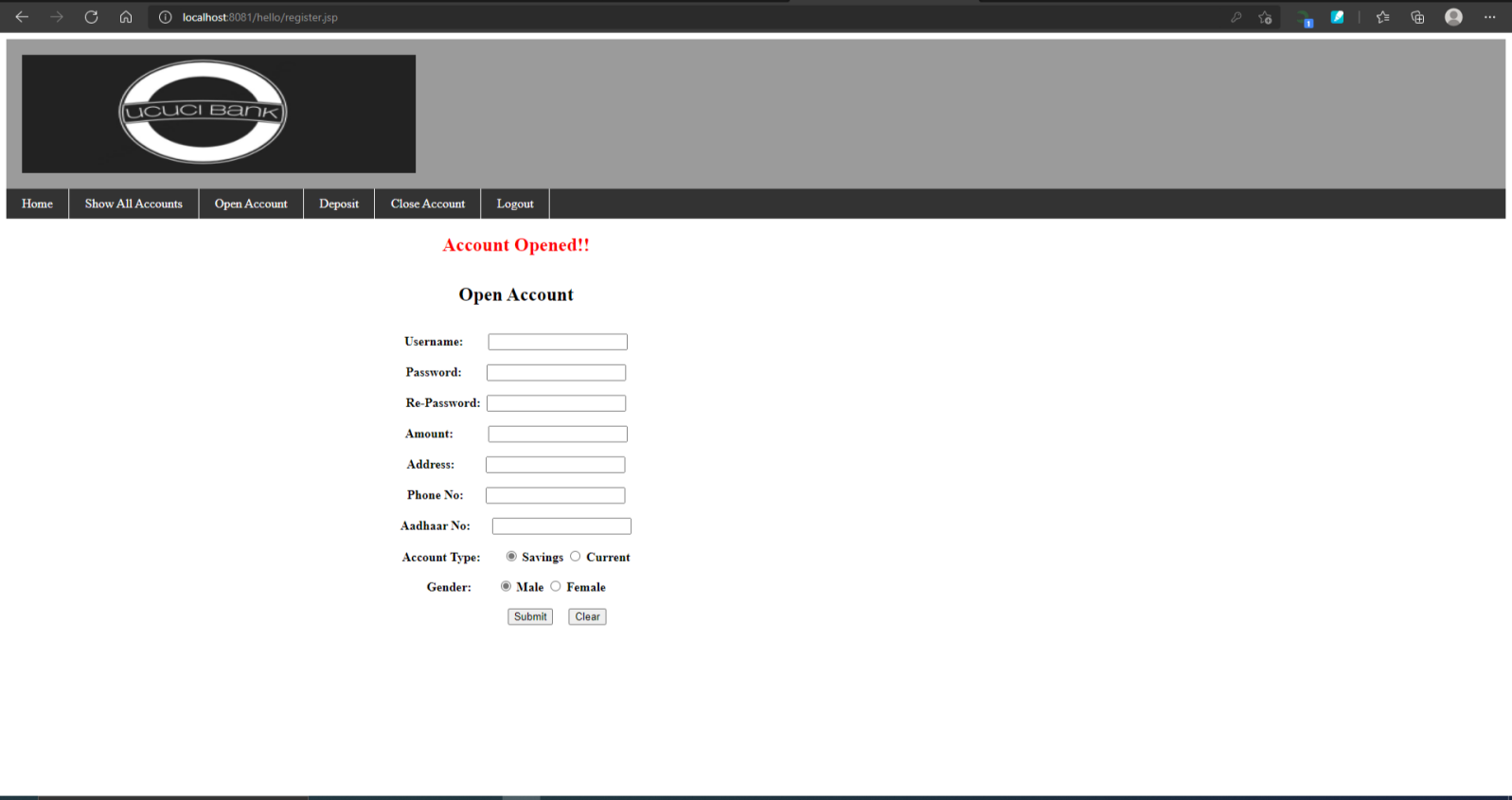
**WITHDRAW FUNDS**

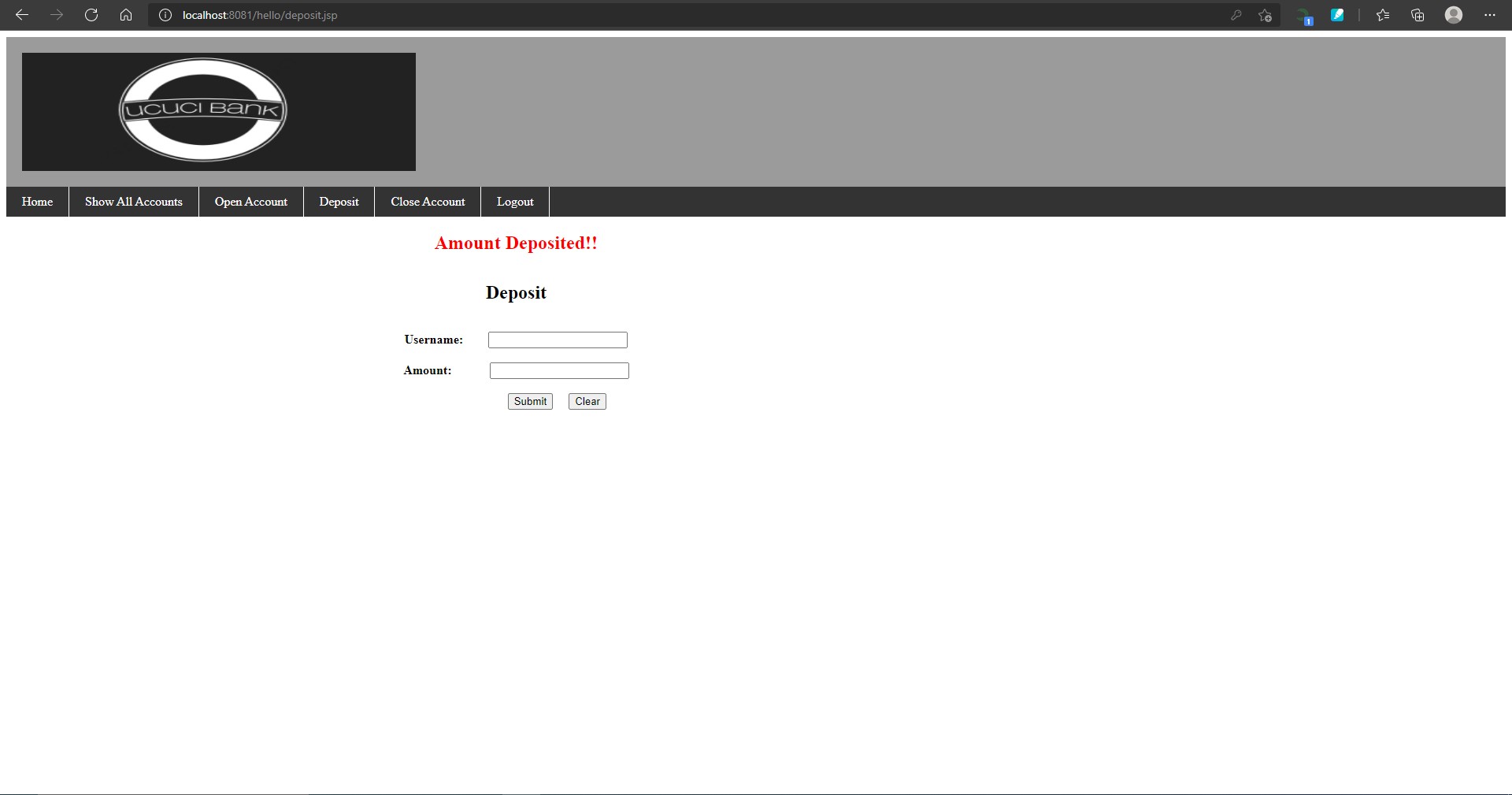
****

**STATEMENT(BOTH USER AND ADMIN CAN ACCESS)**

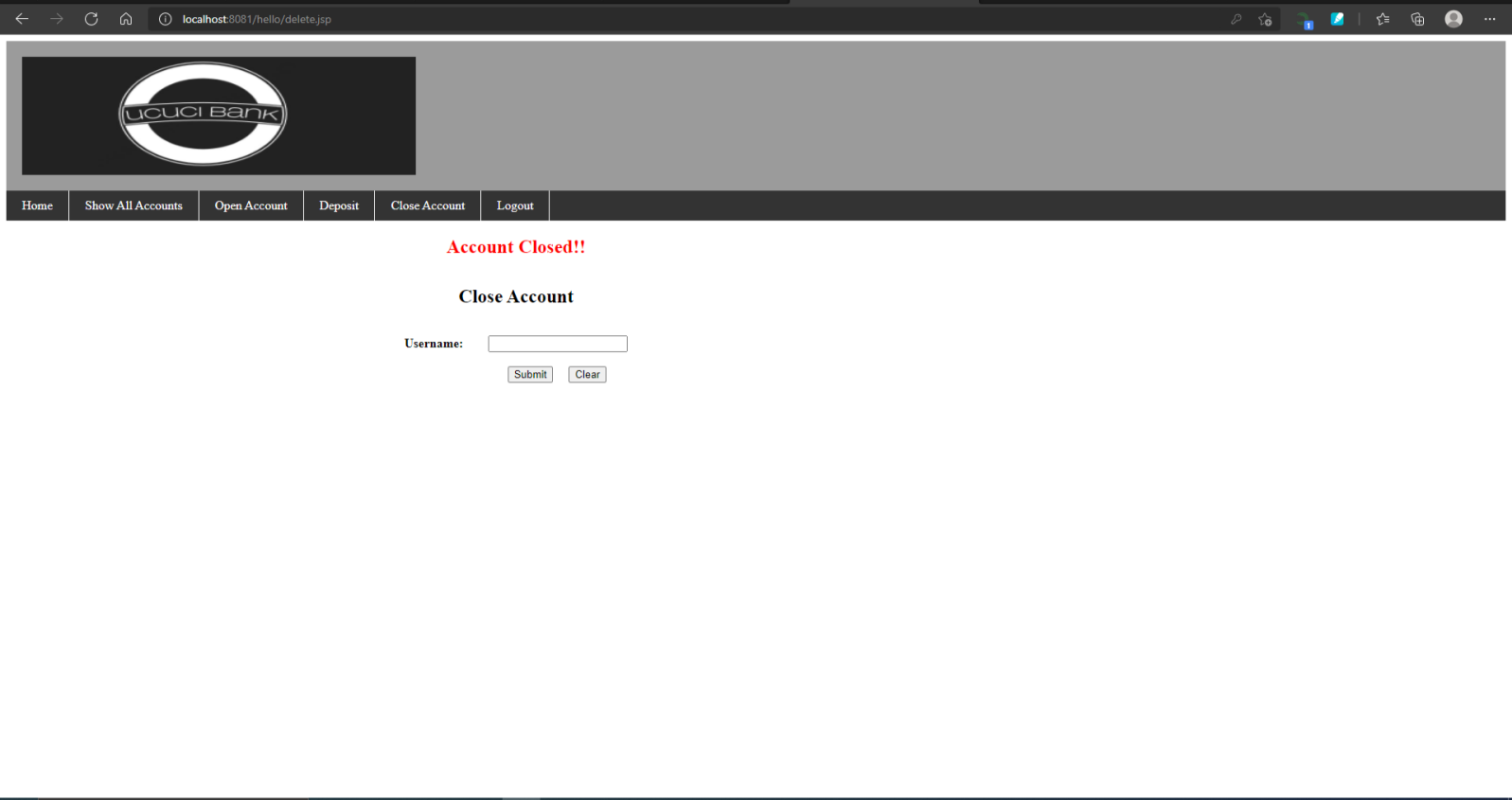
****

**ADMIN LOGIN(OPEN ACCOUNT)**

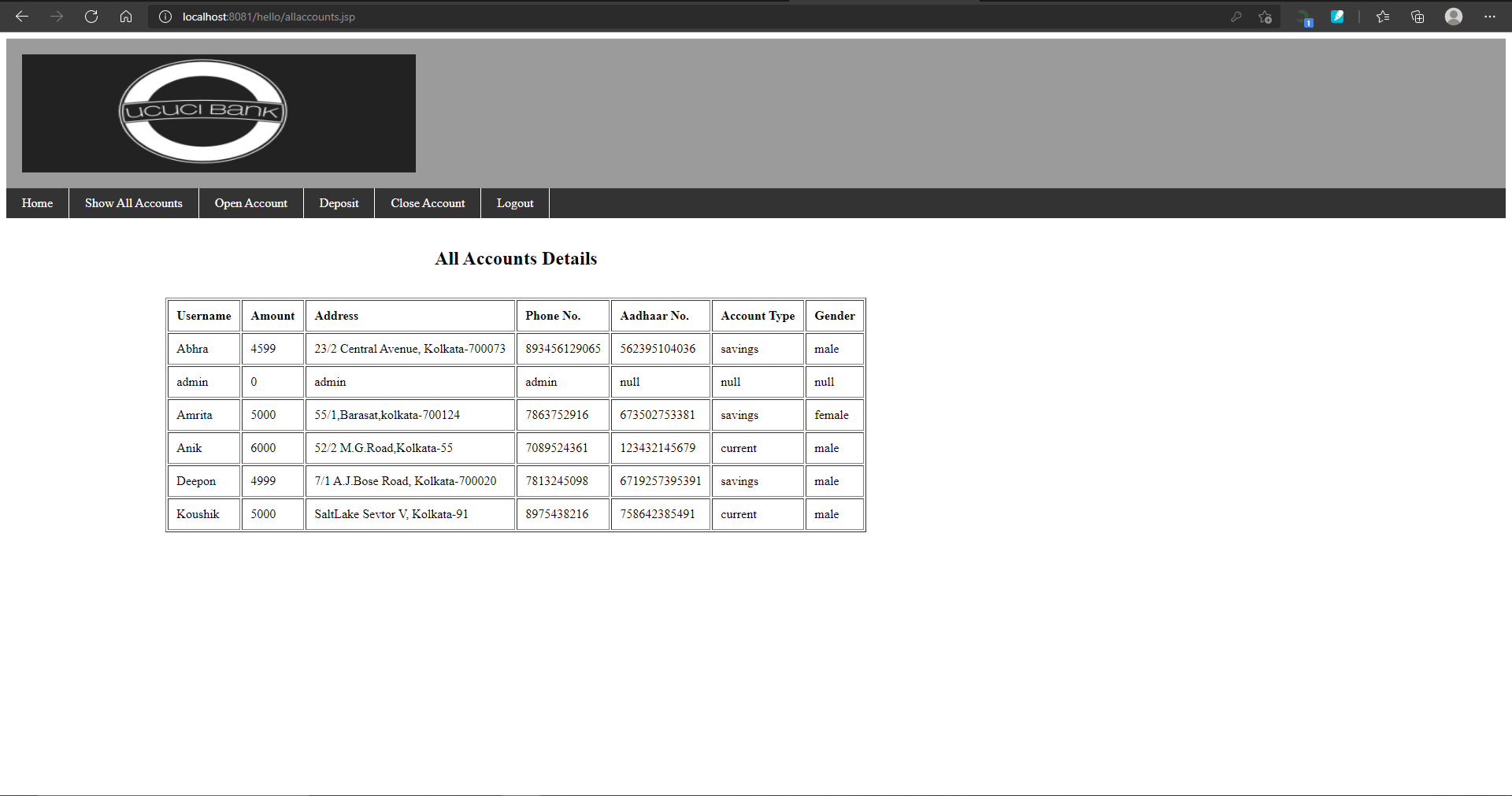
****

**DEPOSIT AMOUNT(ONLY ADMIN)**

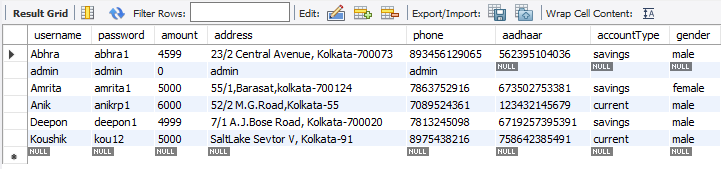
**DELETE ACCOUNT(ONLY ADMIN)**

****

**SHOW ALL ACCOUNT DETAILS**

****

**DATABASE(MySQL)**

****

**CHAPTER- 6**

**SYSTEM TESTING**

**8.1. INTRODUCTION**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

**8.2. STRATEGIC APPROACH TO SOFTWARE TESTING**

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.



UNIT TESTING

MODULE TESTING

SUB-SYSTEM TESING

SYSTEM TESTING

ACCEPTANCE TESTING

**8.3. Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

**1. WHITE BOX TESTING**

This type of testing ensures that

* All independent paths have been exercised at least once
* All logical decisions have been exercised on their true and false sides
* All loops are executed at their boundaries and within their operational bounds
* All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

**2. BASIC PATH TESTING**

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number Of Regions

Where V(G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

**3. CONDITIONAL TESTING**

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

**4. DATA FLOW TESTING**

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

**5. LOOP TESTING**

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

* All the loops were tested at their limits, just above them and just below them.
* All the loops were skipped at least once.
* For nested loops test the inner most loop first and then work outwards.
* For concatenated loops the values of dependent loops were set with the help of connected loop.
* Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

**CHAPTER- 7**

**CONCLUSION & SCOPE FOR FUTRURE DEVLOPMENT**

**CONCLUSION**

This project developed, incorporated all the activities involved in the browsing centre.

It provides all necessary information to the management as well as the customer with the use of this system; the user can simply sit in front of the system and monitor all the activities without any physical movement of the file. Management can service the customers request best in time.

The system provides quickly and valuable information. These modules have been integrated for effective use of the management for future forecasting and for the current need.

**SCOPE FOR FURTHER DEVELOPMENT**

The system can be designed for further enhancement .This could also be developed according to the growing needs of the customer.

**BIBLIOGRAPHY**

* **For Java installation**
* [Download Java for Windows](https://www.java.com/en/download/)
* **For MySQL installation**
* [MySQL :: Download MySQL Workbench](https://dev.mysql.com/downloads/workbench/)
* **For Server Instalation**
* [Apache Tomcat® - Apache Tomcat 8 Software Downloads](https://tomcat.apache.org/download-80.cgi)
* **Reference websites**
* [Java Tutorial (w3schools.com)](https://www.w3schools.com/java/)
* [HTML Tutorial (w3schools.com)](https://www.w3schools.com/html/)
* PREFERED IDE DOWNLOAD LINK(ECLIPSE FOR WEB DEVELOPMENT)
* [Eclipse IDE for Enterprise Java and Web Developers | Eclipse Packages](https://www.eclipse.org/downloads/packages/release/2021-06/r/eclipse-ide-enterprise-java-and-web-developers)