**ASSIGNMENT 2**

**CODING**

The Coding part comprises of designing ‘html web pages’ and writing appropriate codes to serve the purpose. For each of the designed web pages we have a java servlet file written which include the code to connect with the database and serve the purpose of the web page. The database used here is Oracle. The code snippets of the designed web pages are given as follows.

**MEMBER REGISTRATION**

The member registration page is used to get the details of the members who want to make a membership in the library. The page is used to accept the member details which are the stored in the database.

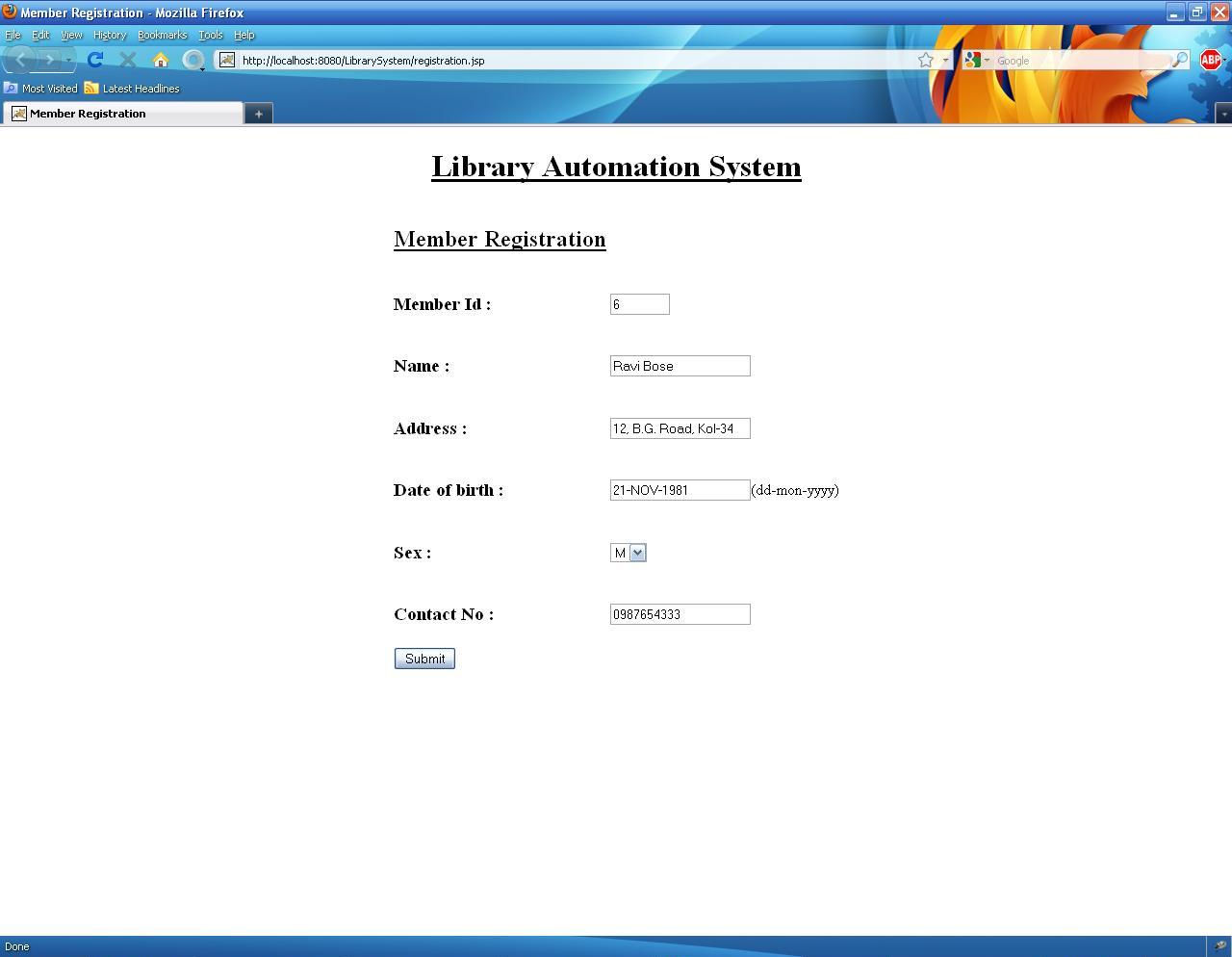
 The member registration page is shown below:

Figure : Member Registration Page

**CODE SNIPPETS:**

As mentioned earlier the code part of the page is contained in a separate java servlet page. Here the connection with the database is made and the values are inserted into the database. The connection with the database is done as follows:

Class.forName (""oracle.jdbc.driver.OracleDriver"");

con = DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:dbname", "username", "password");

Here con is a connection object. The *username* and *password* should be given by the user accordingly. The database name referred here as *dbname* should also be specified. The values mentioned in the textboxes of the webpage are assigned to local variables with similar names to get the values in them as shown:

String memid= request.getParameter("memid").toString();

String name = request.getParameter("memname").toString();

String address = request.getParameter("add").toString();

String dob= request.getParameter("dob").toString();

String sex= request.getParameter("sex").toString();

String contact= request.getParameter("contact").toString();

Then the connection with the database is made and a statement object is created and the query is executed to insert the values into the database. The following code is used to do it:

s1 = con.createStatement();

String s="insert into member values('"+ memid +"','"+name+"','"+address+"','"+dob+"','"+sex+"','"+contact+"')";

n=s1.executeUpdate(s);

Here s1 is the statement object and the string‘s’ is used to create the complete query which is then executed. Here n is a local integer variable. A suitable report is then given to the user to inform whether the record was successfully inserted or an error occurred while executing the statement.

**SERCH RESOURCES AND SELECT THEM**

The search resource page is used by the members of the library to search the resources present in the library. Here the resources are Books or CD or DVD. The members are provided with different search criteria. They can use either one of the search criteria or any combination of them. The search criteria are Book name, Author name and Category.

The search resources page is shown as follows:

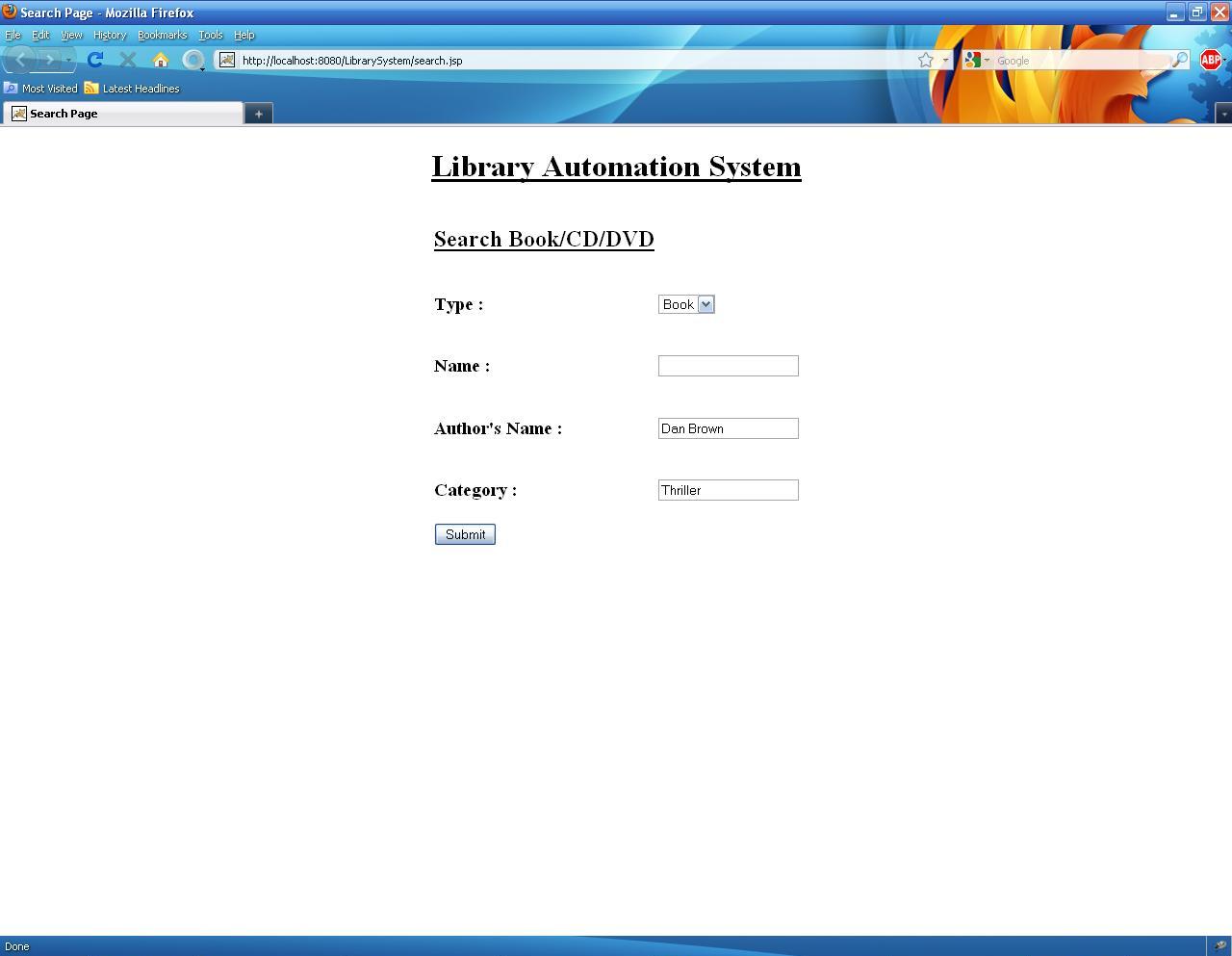


Figure : Search Resources Page

**CODE SNIPPETS:**

In the servlet file corresponding to this page we create a query statement to which the search criteria are appended if they are given to finally create the complete query statement which can be executed to produce the desired result. But first as mentioned earlier we make a connection with the database as :

Class.forName (""oracle.jdbc.driver.OracleDriver"");

con = DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:dbname", "username", "password");

Here con is a connection object. The *username* and *password* should be given by the user accordingly. The database name referred here as *dbname* should also be specified. The values mentioned in the textboxes of the webpage are assigned to local variables with similar names to get the values in them as shown:

String type= request.getParameter("type").toString();

String name = request.getParameter("name").toString();

String author = request.getParameter("author").toString();

String category= request.getParameter("category").toString();

Then the connection with the database is made and a query statement is made by checking whether at least one of the search criteria is given or not. The code for this is shown below:

**s1 = con.createStatement();**

String s=”select \* from resource where type=’”+type+”’”;

if(!name.isempty())

{

s=s+”and name=’”+name+”’”;

}

if(!author.isempty())

{

s=s+”and authorname=’”+author+”’”;

}

if(!category.isempty())

{

s=s+”and category=’”+category+”’”;

}

Resultset records;

records =s1.executeQuery(s);

Here s1 is the statement object and the string‘s’ is used to create the complete query which is then executed. Here ‘records’ is a Resultset Object used to display the records. The records are displayed by creating an html page in which the records are displayed in proper format by placing them as done while creating a normal html page. While displaying the resources we check if it is available. If it is available then only it is displayed. Else a suitable message is given to the user in case of an unsuccessful search. In that web page we also provide an option to select the resource which the user wants to check out. An entry corresponding to that check out is made in the database. The resource selection page is shown as follows.

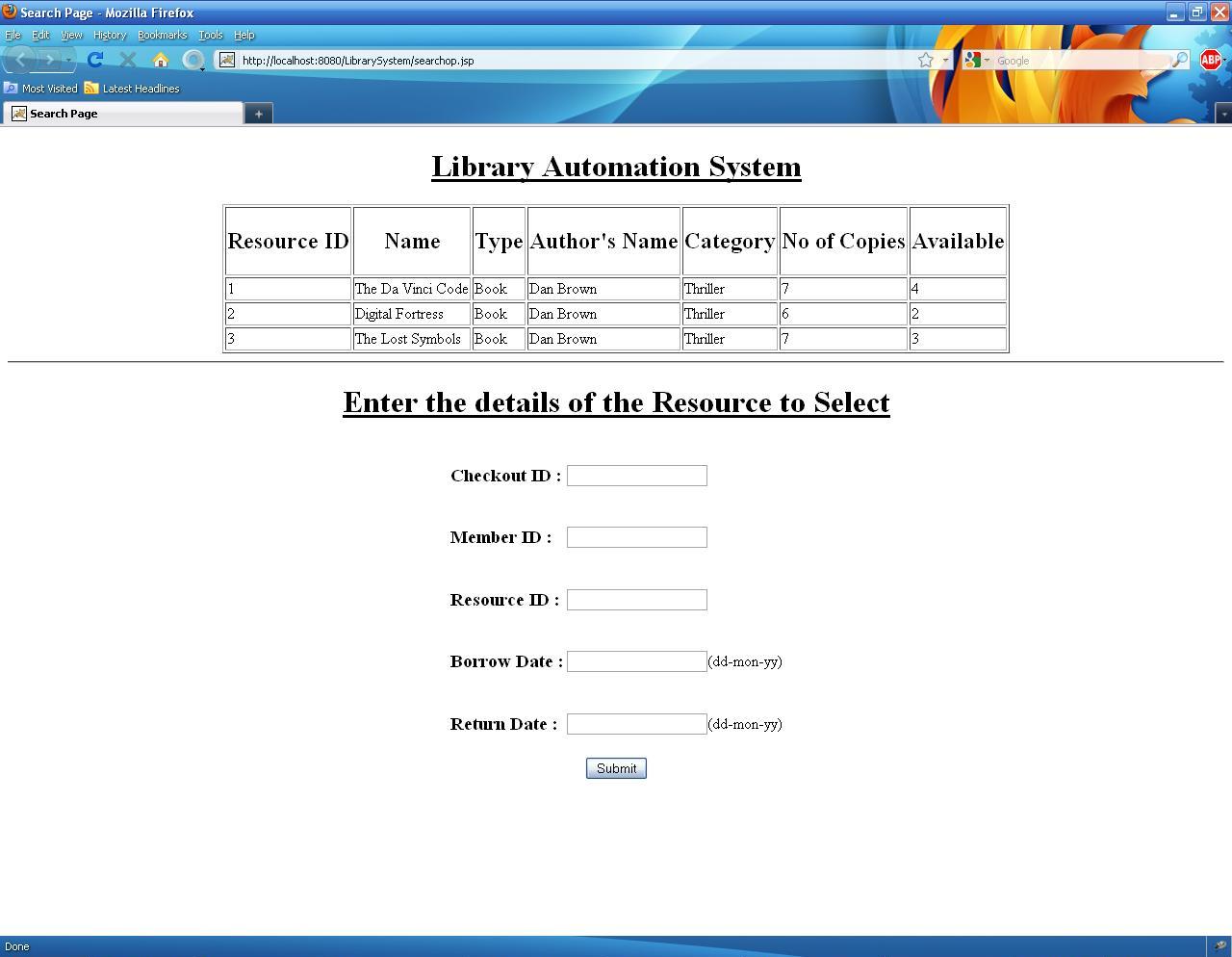


Figure : The Resource Selection Page

In the Resource selection page we select a resource and a record of it is made in the database. The following code gives an idea of how this is carried out.

**CODE SNIPPETS:**

In the servlet corresponding to the Resource Selection Page we make an entry in the database corresponding to the resource that the user has selected. As mentioned earlier in this case as well we make a connection with the database as shown.

Class.forName (""oracle.jdbc.driver.OracleDriver"");

con = DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:dbname", "username", "password");

Here con is a connection object. The *username* and *password* should be given by the user accordingly. The database name referred here as *dbname* should also be specified. The values mentioned in the textboxes of the webpage are assigned to local variables with similar names to get the values in them as shown:

String cid= request.getParameter("cid").toString();

String member = request.getParameter("member").toString();

String resource = request.getParameter("resource").toString();

String borrow\_dt= request.getParameter("borrow\_dt").toString();

String return\_dt= request.getParameter("return\_dt").toString();

Then the connection with the database is made and a statement object is created and the query is executed to insert the values into the database. The following code is used to do it:

s1 = con.createStatement();

String s="insert into checkout values('"+ cid +"','"+member+"','"+resource+"','"+borrow\_dt+"','"+return\_dt+"')";

n=s1.executeUpdate(s);

In the above code s1 is the statement object and the string‘s’ is used to create the complete query which is then executed. Here n is a local integer variable. A suitable report is then given to the user to inform whether the record was successfully inserted or an error occurred while executing the statement. At the same time when a record is made for a particular check out an update query is executed to update the availability status of the recently selected resource in the resource database. The code for it is shown as follows.

s1 = con.createStatement();

String s2="select available from resource where resource\_id=’”+resource+”’”;

Resultset r1=s1.executeQuery(s2);

int n2;

String s3=”update resource set available=”+(r1.getInt("available")-1)+”where resource\_id=’”+resource+”’”;

n2=s1.executeUpdate(s3);

In the above code s1is a statement object and the string‘s2’ and’s3’ is used to create the complete query which is then executed. Here n2 is a local integer variable.

**VIEW STATUS**

The view status page is used to view the status of a member which means the number of resources he or she had taken and their corresponding return dates. It also includes any fine which they are supposed to pay in case they do not return the resources on time. The page is shown below whose output is given in a different page.

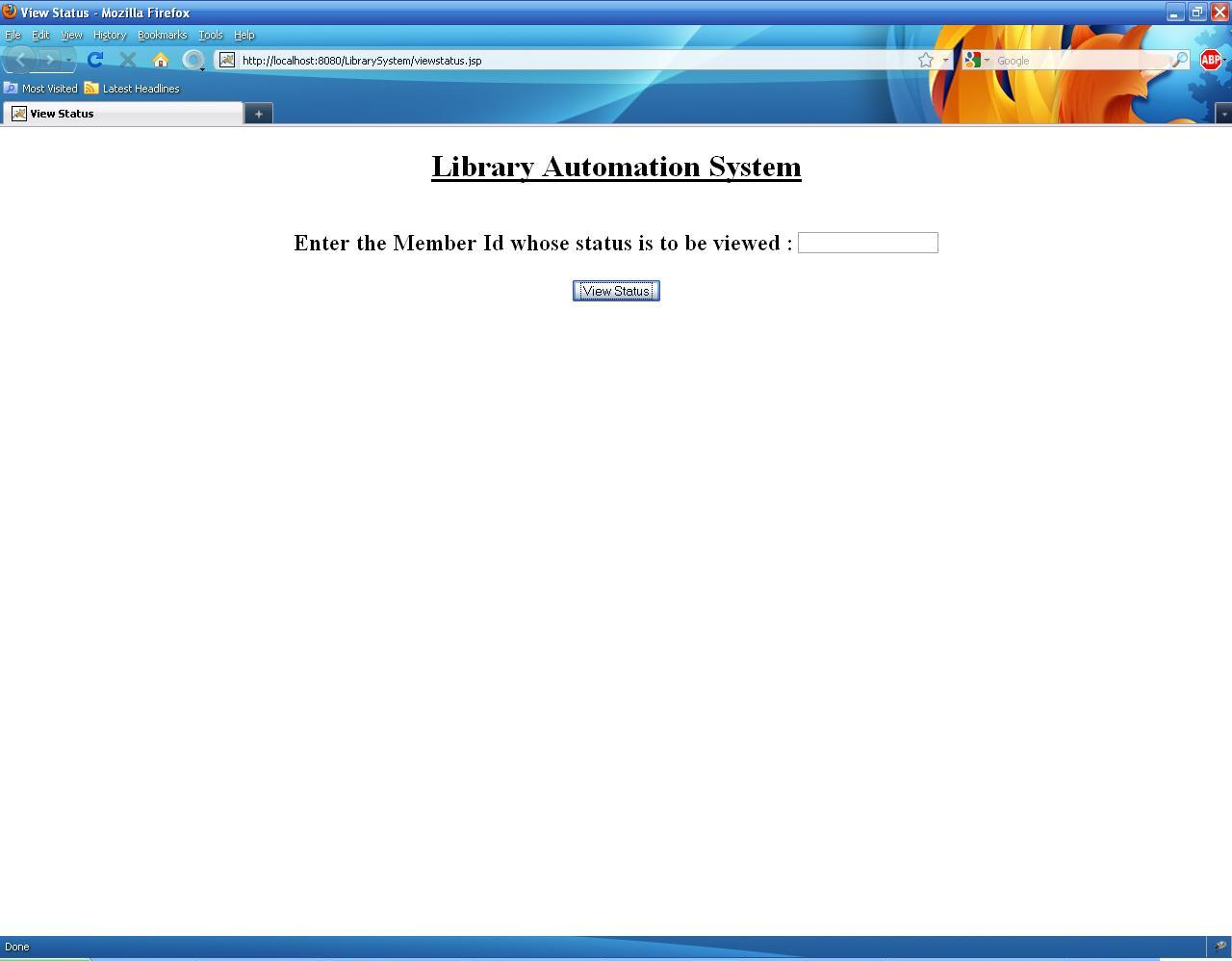


Figure : View Status

**CODE SNIPPETS:**

In the servlet of the view status page the list of resources borrowed by the member is shown. If the member has not borrowed anything then an appropriate message is shown. For this purpose all the entries of the checked out resources is seen and the list is generated to inform the member. Here also we make a connection with the database as shown.

Class.forName (""oracle.jdbc.driver.OracleDriver"");

con = DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:dbname", "username", "password");

Here con is a connection object. The *username* and *password* should be given by the user accordingly. The database name referred here as *dbname* should also be specified. The value mentioned in the textbox of the webpage is assigned to local variables with similar names to get the value in them as shown:

String memid= request.getParameter("memid").toString();

Then the connection is made with the database and a query statement is build and executed to serve the purpose. The code for it is shown :

s1 = con.createStatement();

String s="select \* from checkout where member=’"+memid”’”;

records =s1.executeQuery(s);

Here s1 is the statement object and the string‘s’ is used to create the complete query which is then executed. Here n is a local integer variable. Here ‘records’ is a Resultset object to display the records. In this case the records are displayed in a separate web page.

In this webpage we also mention if any fine is to be paid by that member. The code for it is shown as follows:

s1 = con.createStatement();

String s="select return\_dt,type from checkout,resource where checkout.resource=resource.resource\_id and member=’"+memid”’”;

records =s1.executeQuery(s);

String s2=”select to\_date(sysdate)-to\_date(‘records.getDate(“return\_dt”)’) as datediff from dual”;

records2=s1. executeQuery(s2);

int fine;

if(records2.getInt(“datediff”)>0)

{

String type= records.getString(“type”);

if( type.matches(“Book”))

fine= records2.getInt(“datediff”)\*1;

if(type.matches(“CD”))

fine= records2.getInt(“datediff”)\*2;

if(type.matches(“DVD”))

fine= records2.getInt(“datediff”)\*3;

}

else

fine=0;

Then we display the fine in a text box as shown later. Here records and records2 are Resultset objects and s1 is a statement object. The resultant web page is shown as:

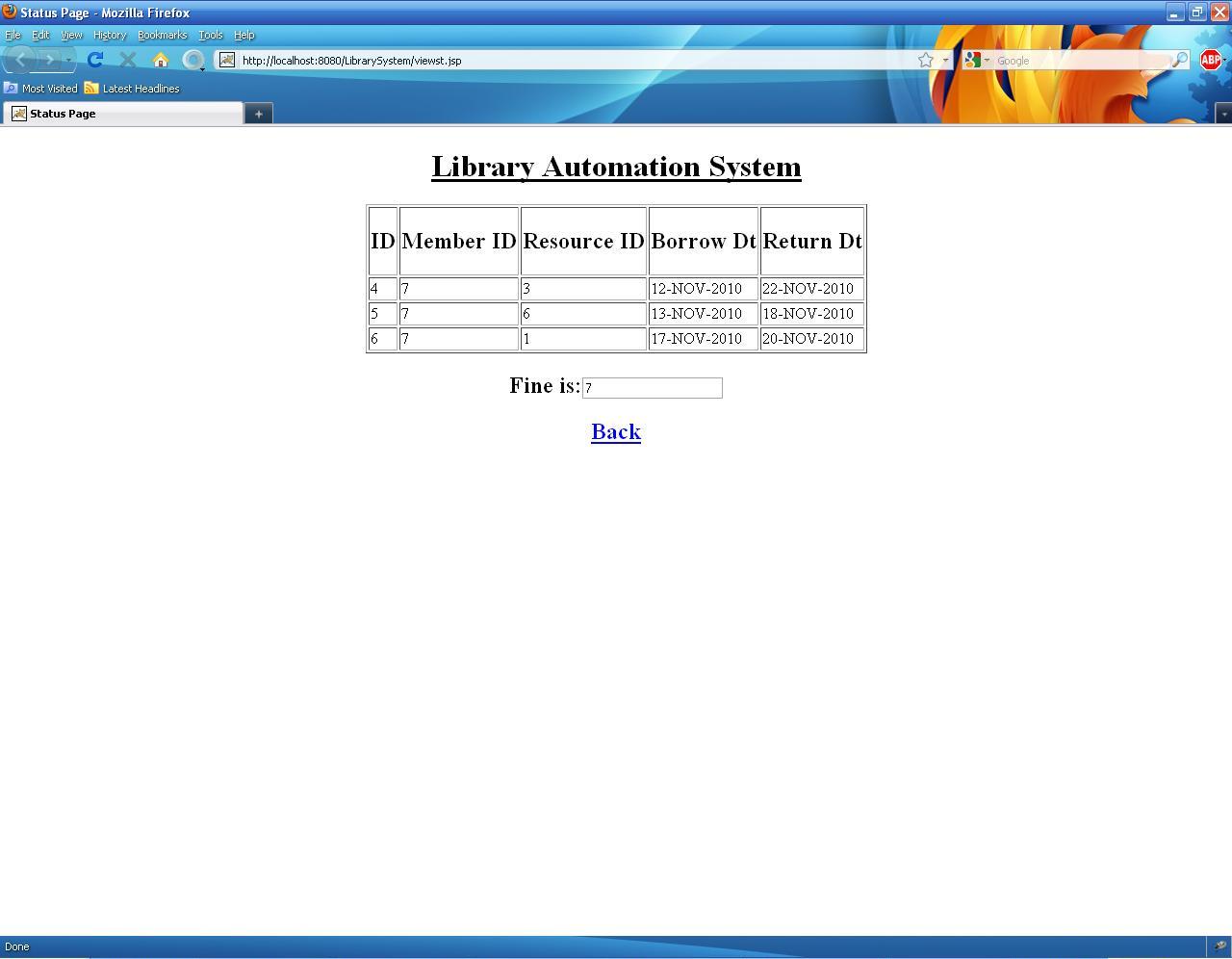


Figure : View Status Result Page

Thus the status of a particular member is shown.

**ADD RESOURCE**

In the add resource page the library administrator adds new resources to the library. This is done by inserting a record into the database corresponding to the new item that is added to the library. The add resource page is shown as follows.

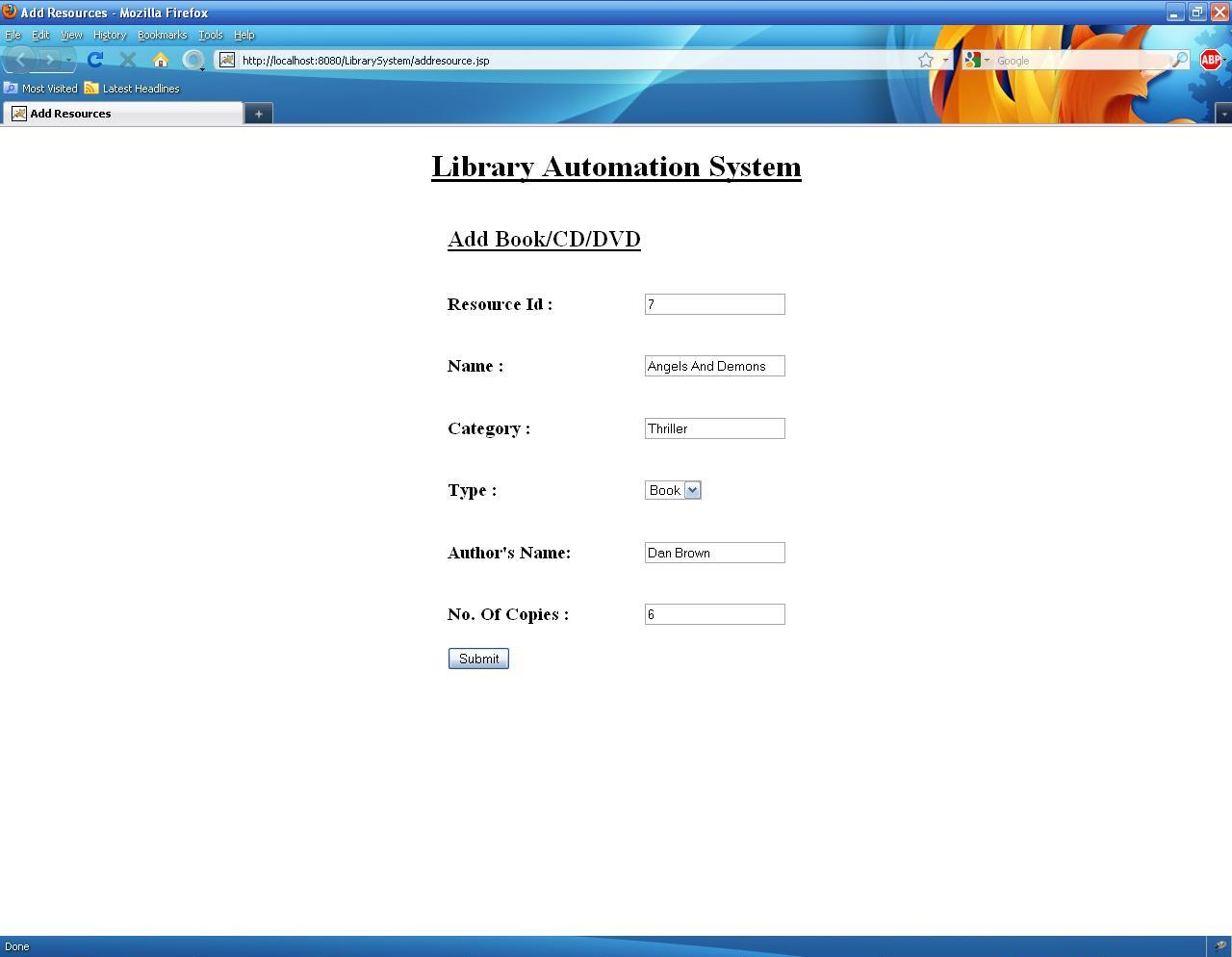


Figure : Add Resource Page

**CODE SNIPPETS:**

As mentioned earlier the code part of the page is contained in a separate java servlet page. Here the connection with the database is made and the values are inserted into the database. The connection with the database is done as follows:

Class.forName (""oracle.jdbc.driver.OracleDriver"");

con = DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:dbname", "username", "password");

Here con is a connection object. The *username* and *password* should be given by the user accordingly. The database name referred here as *dbname* should also be specified. The values mentioned in the textboxes of the webpage are assigned to local variables with similar names to get the values in them as shown:

String resourceid= request.getParameter("resourceid").toString();

String name = request.getParameter("resourcename").toString();

String category = request.getParameter("category").toString();

String type= request.getParameter("type").toString();

String author= request.getParameter("author").toString();

String no\_of\_copies= request.getParameter("no\_of\_copies").toString();

Then the connection with the database is made and a statement object is created and the query is executed to insert the values into the database. The following code is used to do it:

s1 = con.createStatement();

String s="insert into resource values('"+ resourceid +"','"+name+"','"+category+"','"+type+"','"+author+"','"+no\_of\_copies+"')";

n=s1.executeUpdate(s);

Here s1 is the statement object and the string‘s’ is used to create the complete query which is then executed. Here n is a local integer variable. A suitable report is then given to the user to inform whether the record was successfully inserted or an error occurred while executing the statement.

**CHECK RECORDS**

The ‘check records’ page is used by the library administrator to check the number of books borrowed or returned on a given date. The check records page is shown as follows:

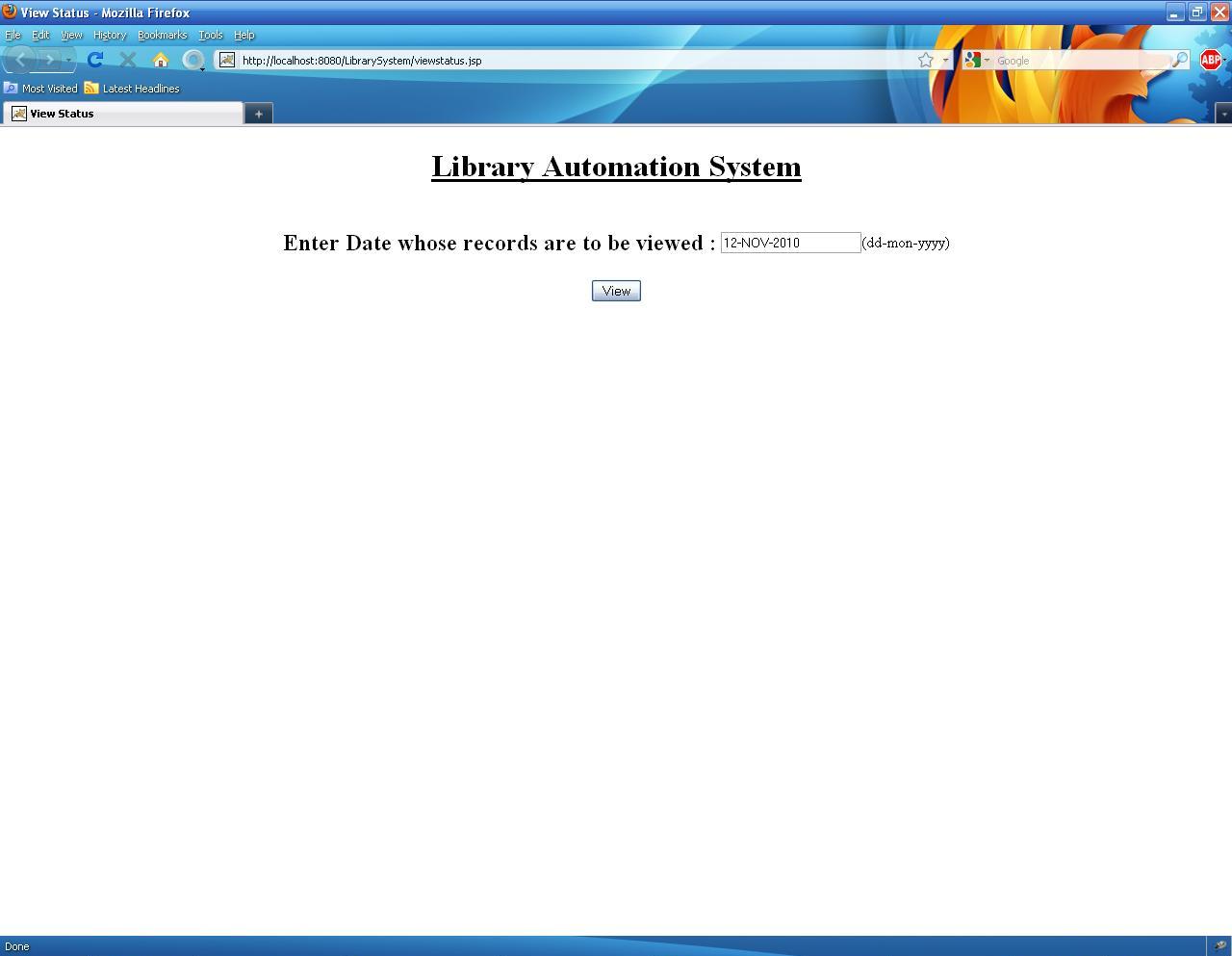


Figure : Check Records

**CODE SNIPPETS:**

As mentioned earlier the code part of the page is contained in a separate java servlet page. Here the connection with the database is made and the values are selected from the database. The connection with the database is done as follows:

Class.forName (""oracle.jdbc.driver.OracleDriver"");

con = DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:dbname", "username", "password");

Here con is a connection object. The *username* and *password* should be given by the user accordingly. The database name referred here as *dbname* should also be specified. The values mentioned in the textboxe of the webpage are assigned to local variable with similar names to get the values in them as shown:

String dt= request.getParameter("dt").toString();

Then the connection with the database is made and a statement object is created and the query is executed to insert the values into the database. The following code is used to do it:

s1 = con.createStatement();

String s="select \* from checkout where borrow\_dt=’”+dt”’”;

Resultset records;

records=s1.executeQuery(s);

Here s1 is the statement object and the string‘s’ is used to create the complete query which is then executed. Here ‘record’ is a Resultset object. This code is used to display the resources checked out on the given date. For the resources which are returned on the same date a similar code is used to display them. The code is as follows:

String s2="select \* from checkout where borrow\_dt=’”+dt”’”;

Resultset records2;

Records2=s1.executeQuery(s2);

Here the string‘s2’ is used to create the complete query which is then executed. Here ‘record2’ is a Resultset object. The output of this page is shown in a separate page as shown:

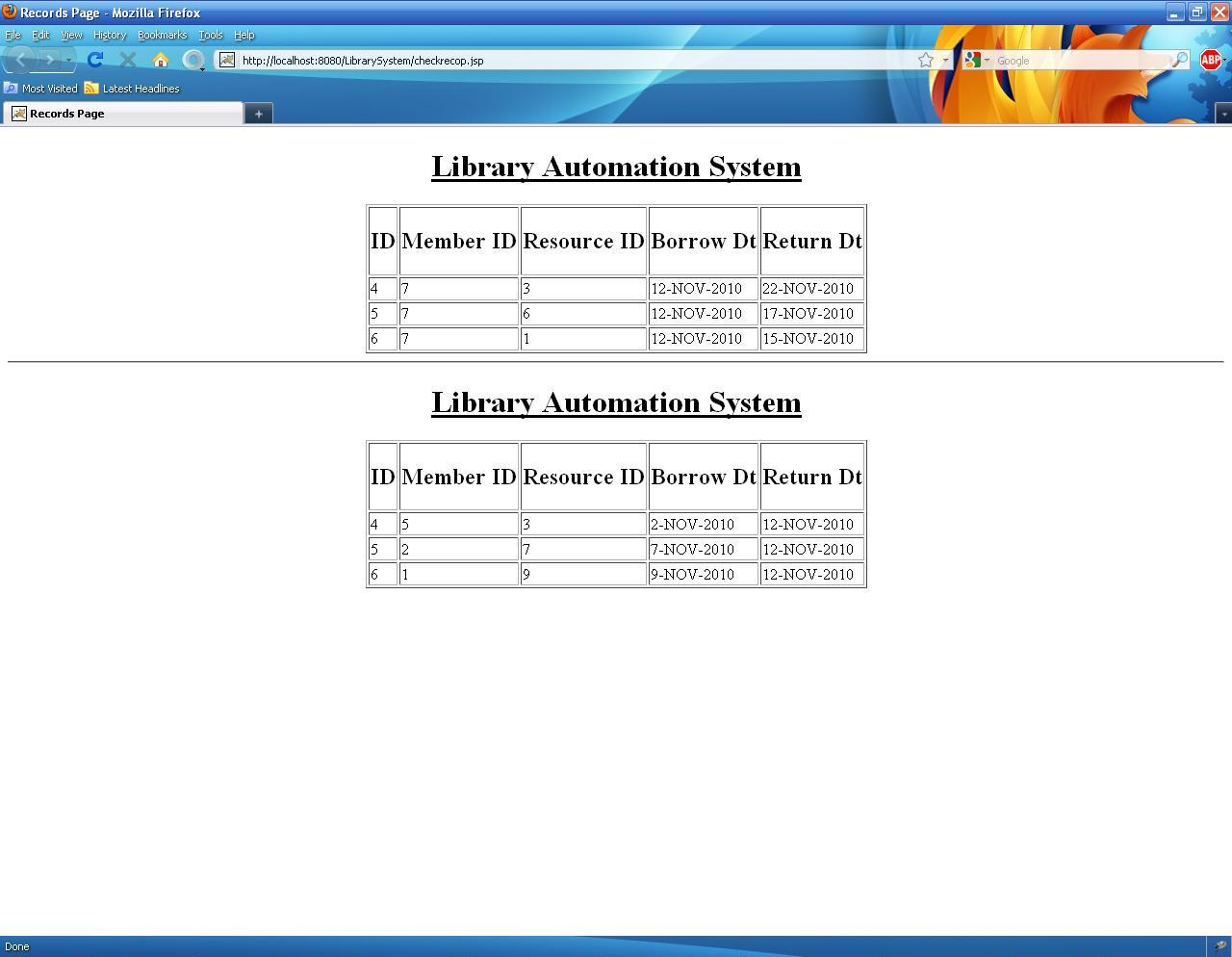


Figure : Output of Check Records

Thus the library administrator can view the resources checked out and returned on a given date.