

LIBRARY MANAGEMENT SYSTEM

**A Project Report Submitted in Partial
Fulfilment of the Requirements for the
Degree of**

BACHELOR OF TECHNOLOGY in Computer Science and Engineering



**FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW, LUCKNOW.
2020-21**

Under the Guidance of

Dr. Jasvant Kumar

by

Sidhant Rajpoot(Univ.Roll.No 190013135086)

Sakshi Prajapati(Univ.Roll.No 190013135075)

Rose Gupta (Univ.Roll.No 190013145071)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person or material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher education, except where due acknowledgement has been made in the text.

Signature

Sakshi Prajapati

(Unvi.Roll.No. 190013135075)

Date:

Signature

Sidhnat Rajpoot

(Unvi.Roll.No. 190013135086)

Date:

Signature

Rose Gupta

(Unvi.Roll.No. 190013135071)

Date:

CERTIFICATE

Certified that **Sakshi Prajapati (190013135075)** has carried out the project work presented in this project report entitled "**Library Management System**" for the award of **Bachelor of Technology** (Computer Science and Engineering) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the student himself/herself (print only that is applicable) and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature

(Name of Guide: Dr. Jasvant Kumar)

CERTIFICATE

Certified that **Sidhant Rajpoot (190013135086)** has carried out the project work presented in this project report entitled "**Library Management System**" for the award of **Bachelor of Technology** (Computer Science and Engineering) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the student himself/herself (print only that is applicable) and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature

(Name of Guide: Dr. Jasvant Kumar)

CERTIFICATE

Certified that **Rose Gupta (190013135071)** has carried out the project work presented in this project report entitled "**Library Management System** " for the award of **Bachelor of Technology** (Computer Science and Engineering) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the student himself/herself (print only that is applicable) and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature

(Name of Guide: Dr. Jasvant Kumar)

TABLE OF CONTENTS

	PAGE NO.
ACKNOWLEDGEMENT	i
LIST OF TABLES	ii
CHAPTER 1: INTRODUCTION	
1.1 MODULE DESCRIPTION	1-2
CHAPTER 2: SYSTEM ANALYSIS	
2.1 SOFTWARE REQUIRMENTS	3
2.2 HARDWARE REQUIRMENTS	3
CHAPTER 3 : TECHNOLOGY	
3.1 ASP.NET	5
3.2 ADO.NET	5
CHAPTER 4: PROGRAMMING LANGUAGE	
4.1 OVERVIEW OF C#.NET	7
4.2 JAVA	7
CHAPTER 5: DATABASE MANAGEMENT SYSTEM	
5.1 DATABASE(SQL)	9
CHAPTER 6: CODING SECTION	11
CHAPTER 7: PROJECT SCREENSHOTS	33
CHAPTER 8: CONCLUSION	35
REFERENCES	36

ER DIGRAM OF LMS

ER Diagram is known as Entity-Relationship Diagram, it is used to analyze the structure of the Database. It shows relationships between entities and their attributes. An ER Model provides a means of communication.

The Library Management System database keeps track of readers with the following considerations –

- The system keeps track of the staff with a single point authentication system comprising login Id and password.
- Staff maintains the book catalog with its ISBN, Book title, price(in INR), category(novel, general, story), edition, author Number and details.
- A publisher has publisher Id, Year when the book was published, and name of the book.
- Readers are registered with their user_id, email, name (first name, last name), Phone no (multiple entries allowed), communication address. The staff keeps track of readers.
- Readers can return/reserve books that stamps with issue date and return date. If not returned within the prescribed time period, it may have a due date too.
- Staff also generate reports that has readers id, registration no of report, book no and return/issue info.

This Library ER diagram illustrates key information about the Library, including entities such as staff, readers, books, publishers, reports, and authentication system. It allows for understanding the relationships between entities.

Entities and their Attributes –

- Book Entity : It has authno, isbn number, title, edition, category, price. ISBN is the Primary Key for Book Entity.
- Reader Entity : It has UserId, Email, address, phone no, name. Name is composite attribute of firstname and lastname. Phone no is multi valued attribute. UserId is the Primary Key for Readers entity.
- Publisher Entity : It has PublisherId, Year of publication, name. PublisherID is the Primary Key.
- Authentication System Entity : It has LoginId and password with LoginID as Primary Key.
- Reports Entity : It has UserId, Reg_no, Book_no, Issue/Return date. Reg_no is the Primary Key of reports entity.
- Staff Entity : It has name and staff_id with staff_id as Primary Key.
- Reserve/Return Relationship Set : It has three attributes: Reserve date, Due date, Return date.

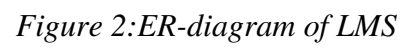


Figure 2:ER-diagram of LMS

CHAPTER -1

INTRODUCTION

A library management system is software that is designed to manage all the functions of a library. It helps librarian to maintain the database of new books and the books that are borrowed by members along with their due dates.

This system completely automates all your library's activities. The best way to maintain, organize, and handle countless books systematically is to implement a library management system software.

A library management system is used to maintain library records. It tracks the records of the number of books in the library, how many books are issued, or how many books have been returned or renewed or late fine charges, etc.

You can find books in an instant, issue/reissue books quickly, and manage all the data efficiently and orderly using this system. The purpose of a library management system is to provide instant and accurate data regarding any type of book, thereby saving a lot of time and effort.

The acquisition & cataloguing module of the library management system enables the librarian to select & buy books, journals, and other resources and create a database of the same for easy book search.

The serial control module of the library software enables the librarians to handle or control processes such as subscription, renewals of books or their cancellations and generate accurate MIS reports.

The circulation module enables the librarian to create and manage borrower types along with keeping a tab on their book issue date, return date, dues, and fines. It enables a smooth circulation of books in the library.

Management information systems or MIS reports related to the library allow the librarians to extract crucial data & information of all the library transactions at a few clicks. Accurate MIS reports fostering better decision-making.

Online Public Access Catalogue or OPAC is a digital catalogue that enables the users to search for books, journals, or any other material by entering keywords such as the name of the book, its title, author's name, etc.

Mobile - Online Public Access Catalogue or M-OPAC is a mobile-based smartphone app that enables users to search for any book via their smartphones by entering keywords such as author's name, title, etc.

Benefits Of Library Automation System ,User-Friendly System ,Simple And Easy To Use, Online And Offline Storage Of Data, Automatically Updates And Backups Data.

Flexible And Can Be Fully Configurable ,Increased Member Engagement, Easily Accessible From Anywhere And Any Time. Easy Access From Smartphones And Tablets, Reliable And Secure Operations, Review And Tracking Of The Library Functions, Cost Effective ,Eliminate The Need For Extensive Paperwork, Maintenance Overheads And Operation Costs Are Reduced ,Eliminates The Need For Manual Entries, Makes The Database Error-Free And Accurate

1.1Module Description:

Admin Login:

- 1) User registers the site.
- 2) Application will be showed.
- 3) If user selected the record will show.
- 4) User selected record is send to the database .
- 5) If user wants to delete, update the data they can also do.

Student login:

- 1) send books details
- 2) send payment verification
- 3) books buying detail
- 4) Store line items using array and order.

Books order :

- 1) Store all the user details.
- 2) Order detail

CHAPTER – 2

SYSTEM ANALYSIS

Requirement Analysis Existing System: In an Existing we address these challenges and present an approach to efficient, incremental consolidation of data-intensive flows. Following common practice, our method iterates over information requirements to create the final design. we show how to efficiently accommodate a new information requirement to an existing design and also, how to update a design in lieu of an evolving information requirement. The final design satisfying all requirements comprises a multi-flow. As ‘coal’ is formed after the process and extreme compaction of layers of partially decomposed materials¹, Co AI processes individual data flows and incrementally consolidates them into a unified multi-flow. Proposed System Following the previously proposed set of flow transformations in the context of ETL processes in Co AI we extend this set considering also the associative property of n-array operations (e.g., Join) and thus rely on the following four flow transformations used for reordering the operations. Swap Applied to a pair of adjacent unary operations, it interchanges the order of these operations. Distribute/Factorize. Applied on a unary operation over an adjacent n-array operation, it respectively distributes the unary operation over the adjacent nary operation or factorizes several unary operations over the adjacent narray operation. Merge/Split. Applied on a set of adjacent unary operations, it respectively merges several operations into a single unary operation or splits a unary operation into several unary operations. Re-associate. Applied on a pair of mutually associative n-array operations, it interchanges the order in which these operations are executed.

SOFTWARE REQUIREMENTS SPECIFICATIONS

System configurations The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

2.1 Software Requirements:

- Operating system : Windows 7 Ultimate or Window 8 - 11 .
- Coding Language : MVC 4 Razor , C# , HTML, CSS, Visual Basic.
- Front-End : Visual Studio 2019 Professional.
- Data Base : SQL Server 2019.

2.2 Hardware Requirement:

- System : intel I3 9-Gen processor 2.4 GHz.
- Hard Disk : 1TB.
- Ram : 8GB.

CHAPTER - 3

TECHNOLOGY

3.1 ASP.NET

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices. ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .NET platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .NET framework. These codes can use the entire hierarchy of classes in .NET framework. ASP.NET web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response. All client side user activities are forwarded to the server for tasteful processing. The server processes the output of the client actions and triggers the reactions. Now, HTTP is a stateless protocol. ASP.NET framework helps in storing the information regarding the state of the application, which consists of:

- Page state
- Session state

The page state is the client state, i.e., the content of various input fields in the web form. The session state is the collective information obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take an example of a shopping cart. User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown on a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.NET session state and server side infrastructure keeps track of the information collected globally over a session.

6 The ASP.NET runtime carries the page state to and from the server across page requests while generating ASP.NET runtime codes, and incorporates the state of the server side components in hidden fields. This way, the server becomes aware of the overall application state and operates in a two-tiered connected way. The ASP.NET component model provides various building blocks of ASP.NET pages. Basically it is an object model, which describes:

ASP.NET is a technology, which works on the .NET framework that contains all web-related functionalities. The .Net framework is made of an object-oriented hierarchy. An ASP.NET web applications is made of pages. When a user requests an ASP.NET page, the IIS delegates the processing of the page to the ASP.NET runtime system. The ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework. Therefore, each ASP.NET page is an object and all its components i.e., the server-side controls are also objects.

3.2 ADO.NET

As you develop applications using ADO.NET, you will have different requirements for working with data. In some cases, you might simply want to display data on a form. In other cases, you might need to devise a way to share information with another company. No matter what you do with data, there are certain fundamental concepts that you should understand about the data approach in ADO.NET. You might never need to know some of the details of data handling- for example, you might never need to directly edit an XML file containing data- but it is very useful to understand the data architecture in ADO.NET, what the major data components are, and how the pieces fit together. This introduction presents a high-level over view of these most important concepts. The topic deliberately skips over many details- for example, there is much more to data sets than what is mentioned here- in favour of simply introducing you to ideas behind the data integration in ADO.NET. ADO.Net does not continuously live connections. In traditional client/server applications, components establish a connection to a data base and kept it open while the application is running. For a variety of reasons, this approach is impractical in many applications. Open database connections take up valuable system resources. In most cases, databases can maintain only a small number of concurrent connections. The overhead of maintaining these connections detracts from overall application performance. 7 Similarly, applications that require an open database connection are extremely difficult to scale up. An application that does not scale up well might perform acceptable with four users but will likely not do so with hundreds. ASP.NET Web applications in particular need to be easily scalable, because traffic to a website can go up by orders of magnitude in a short period. A model based on always connected data can make a difficult and impractical to exchange data across application and organizational boundaries using a connected architecture. If two components need to share the same data, both have to be connected, and a way must be devised for the components to pass data back and forth. For all the reasons, data accessed with ADO.NET is designed around an architecture that uses connections sparingly. Applications are connected to the database only long enough to fetch or update the data. Because the database is not holding onto connections that are largely idle, it can service many more users.

CHAPTER – 4

PROGRAMMING LANGUAGE

4.1 Overview of C#.Net

C# is a simple, modern, object oriented, and type –safe programming language derived from C and C++.

- It will immediately be familiar to C and C++ programmers.
- C# aims to combine the high productivity of visual basic and the raw power of C++ → Visual C#.NET is Microsoft's C# development tool. → It includes an interactive development environment, visual designers for building windows and web applications, a compiler and a debugger. → Visual C#.NET is part of a suite of products, called Visual Studio .NET, that also includes Visual Basics .NET, Visual C++.NET and the Jscript scripting language. → The .NET frame work defines a “Common Language Specification” (CLS), a short of lingua franca that ensures seamless interoperability between CLS-complaint languages and class libraries. → For C# developers this means even though C# is a new language, it has complete access to the same rich class libraries that are used by seasoned tools such as Visual Basic.NET and Visual C++.NET.

4.2 JAVA

JavaScript is an interpreter, client-side, event-based, object oriented scripting language that you can use to add dynamic interactivity to your web pages. JavaScript scripts are written in plain text, like HTML, XML, Java, PHP and just about any other modern computer code. In this code, we will use Windows Note Pad to create and edit our JavaScript code, but there are a large number of alternatives available. Note Pad is chosen to demonstrate JavaScript's immediacy and simplicity. You can use JavaScript to achieve any of the following: 8

- Create special effects with images that give the impression that a button is either highlighted or depressed whenever the mouse pointer is hovered over it.
- Validate information that users enter into your web forms
- Open pages in new windows, and customise the appearance of those new windows.
- Detect the capabilities of the user's browser and alter your page's content appropriately.
- Create custom pages “on the fly” without the need for a server-side language like PHP. JavaScript is not Java, though if you come from a Java background, you will notice that both languages look similar when written. Java is a full featured and comprehensive programming language similar to C or C++, and although JavaScript can interact with Java web applications, the two should not be confused. Different web browsers will run your JavaScript in different, sometimes incompatible ways. In order to work around this, it is often necessary to use JavaScript itself to detect the capabilities of the browser in which it finds itself, and alter its operation depending on the result. To revisit the original definition in this chapter, note the following points:
 - Interpreted refers to the fact that JavaScript code is executed (acted on) as it is loaded into the browser. This is a change of pace from compiled languages like Java, which check your program thoroughly before running a single line of code, and can have many implications that can catch you out if you are from a non-interpreted programming background.
 - Client-side has been defined already in the previous chapter.
 - Event-based refers to JavaScript's ability to run certain bits of code only when a specified event occurs. An event could be the page being loaded, a form being submitted, a link being clicked, or an image being pointed at by a mouse pointer.
 - Object-oriented signals that JavaScript's power to exert control over an HTML page is based on manipulating objects within that page.
 - If you are familiar with object-oriented programming, you will be aware of some of the power that this can bring to the coding environment.

CHAPTER – 5

DATABASE MANAGEMENT SYSTEM

5.1 DATABASE

About Microsoft SQL Server 2008 Microsoft SQL server is a Structured Query Language (SQL) base, client/server relational database. Each of these terms describes a fundamental part of the architecture of SQL server. 9 A database is similar to a data file in that it is storage place for data. Like a data file, a database does not present information directly to a user, the user runs an application that accesses data from the database and presents it to the users in an untreatable format. A database typically ha two components: the files holding the physical database access data. The DBMS is responsible for enforcing database structure, including:

- Maintaining the relationships between data in the database.
 - Ensuring that data is stored correctly, and the rules that defining data relationships are not violated.
 - Recovering all data to a point of known consistency in case of system failures.
- Relational Database There are different ways to organize data in a database but relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data. In a relational database is collected into tables called relations in relation theory. When organizing data into tables, you can usually find many different ways to define tables. Relational database theory defines a process, normalization, which ensures that the set of tables you define will organize our data effectively. Client/Server In Client/Server system the server is a relatively large computer in a central location that manages a resource used by many people. When individuals need to use the resource, they connect over the network from their computers, or clients, to the server. Examples of servers are: In Client/Server database architecture, the database files and DBMS software resides on a server. A communications component is provided so applications can run on separate clients and communicate to the database server over a network. The SQL server communication component also allows communication

between an application running on the server and SQL server. Server applications are usually capable of working with several clients at the same time. SQL server can work with thousands of client applications simultaneously. The server has features to prevent the logical problems that occur if a user. While SQL server is design to work as a server in a Client/Server network, it is also capable of working as a stand-alone database directly on the client. The scalability and ease of use features of SQL server allows it to work efficiently on a client without consuming too many resource.

Structured Query Language (SQL)

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational database; the most common is SQL. Both the American national standards institute (ANSI) and the International Standards Organization (ISO) has defined standards for SQL.

1. Data is all around us, we used to store data on paper in big filling cabinets, but eventually we store them online in what we call databases.
2. How do we easily pull the data we want to look at it? That's what SQL is for.
3. It's a language that communicate with databases.
4. SQL : stand for structure query language. People call it SQL or Sequel.

In SQL query be like :

“select name from table student”

TABLE = **student**

name	age	output
Shivam	22	Shivam
Ayushi	21	Ayushi
Siddhant	20	Siddhant
Sakshi	20	Sakshi

CHAPTER – 6

COADING SECTOIN

6.1 Std_regrestration.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Std_regrestration.aspx.cs"
Inherits="LMS_software.Std_regrestration" %>
```

```
<%@ Register src="user_control/Header.ascx" tagname="Header" tagprefix="uc1" %>
<%@ Register src="user_control/Footer.ascx" tagname="Footer" tagprefix="uc2" %>
```

```
<!DOCTYPE html>
<!-- Created By Shivam Rajput -->
<head runat="server">
    <title>Student Regestration</title>
    <link rel="stylesheet" href="StyleSheet1.css"/>
    <style type="text/css">
        .auto-style1 {
            width: 100%;
        }
        .auto-style2 {
            text-align: center;
        }
        .auto-style3 {
            width: 60%;
            height: 70%;
            border: 2px solid #808080;
            background-color: #013358;
        }
        .auto-style4 {
            background-color: #009999;
        }
    </style>
</head>
```

[illegible]

<code>runat="server"></asp:TextBox></code>	<code><asp:TextBox</code>	<code>ID="TextBox1"</code>
	<code></td></code>	
	<code></tr></code>	
	<code><tr></code>	
	<code><td class="auto-style5">Roll No.</td></code>	
<code>runat="server"></asp:TextBox></code>	<code><td class="auto-style6"></code> <code><asp:TextBox</code>	<code>ID="TextBox2"</code>
	<code></td></code>	
	<code></tr></code>	
	<code><tr></code>	
	<code><td class="auto-style5">Branch</td></code>	
	<code><td class="auto-style6"></code>	
<code>Height="21px" Width="169px"></code>	<code><asp:DropDownList ID="DropDownList1" runat="server"</code> <code></asp:DropDownList></code>	
	<code></td></code>	
	<code></tr></code>	
	<code><tr></code>	
	<code><td class="auto-style5">Father Name</td></code>	
	<code><td class="auto-style6"></code>	
<code>runat="server"></asp:TextBox></code>	<code><asp:TextBox</code>	<code>ID="TextBox3"</code>
	<code></td></code>	
	<code></tr></code>	
	<code><tr></code>	
	<code><td class="auto-style5">Address</td></code>	
	<code><td class="auto-style6"></code>	
<code>runat="server"></asp:TextBox></code>	<code><asp:TextBox</code>	<code>ID="TextBox4"</code>
	<code></td></code>	
	<code></tr></code>	
	<code><tr></code>	
	<code><td class="auto-style5">Mobile No.</td></code>	
	<code><td class="auto-style6"></code>	
<code>runat="server"></asp:TextBox></code>	<code><asp:TextBox</code>	<code>ID="TextBox5"</code>
	<code></td></code>	
	<code></tr></code>	
	<code><tr></code>	
	<code><td class="auto-style5">Email</td></code>	
	<code><td class="auto-style6"></code>	
<code>runat="server"></asp:TextBox></code>	<code><asp:TextBox</code>	<code>ID="TextBox6"</code>
	<code></td></code>	

```

        </tr>
        <tr>
            <td class="auto-style5">DOB</td>
            <td class="auto-style6">
                <table class="auto-style1">
                    <tr>
                        <td>
                            <asp:DropDownList
                                runat="server" Height="23px" Width="136px">
                                </asp:DropDownList>
                                ID="DropDownList2"
                            </td>
                        <td>
                            <asp:DropDownList
                                runat="server" Height="24px" Width="133px">
                                </asp:DropDownList>
                                ID="DropDownList3"
                            </td>
                        <td>
                            <asp:DropDownList
                                runat="server" Height="30px" Width="121px">
                                </asp:DropDownList>
                                ID="DropDownList4"
                            </td>
                    </tr>
                </table>
            </td>
        </tr>
        <tr>
            <td class="auto-style5">&nbsp;</td>
            <td class="auto-style6">&nbsp;</td>
        </tr>
        <tr>
            <td class="auto-style5">&nbsp;</td>
            <td class="auto-style6">&nbsp;</td>
        </tr>
    </table>
</td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
</table>
</td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
</table>
</td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
</table>

```

```

        </td>
    </tr>
    <tr>
        <td>
            <uc2:Footer ID="Footer1" runat="server" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

6.2 LMS_Software.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="LMS_Software.aspx.cs"
Inherits="Portal_System.LMS_Software" %>

<!DOCTYPE html>
<!-- Created By Shivam Rajput -->
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>LMS_Software</title>
    <link rel="stylesheet" href="StyleSheet_LMS_Software.css"/>

</head>
<body>
    <form runat="server" action="" method="get">
        <nav class="navbar">
            <div class="max-width">
                <div class="logo"><a href="#">LMS<span>
Software</span></a></div>
                <ul class="menu">
                    <li><a class="menu-btn">
                        <asp:LinkButton ID="Btn_home" runat="server"
OnClick="Btn_home_Click">Home</asp:LinkButton></a></li>
                    <li><a class="menu-btn">
                        <asp:LinkButton ID="Btn_logout" runat="server"
OnClick="Btn_logout_Click">Logout</asp:LinkButton></a></li>
                </ul>
            </div>
        </nav>
    </form>

```

```

        <div class="menu-btn">
            <i class="fas fa-bars"></i>
        </div>
    </div>
</nav>
<!-- services section start -->
<section class="services" id="services">
    <div class="max-width">
        <h2 class="title">My services</h2>
        <div class="serv-content">
            <div class="card">
                <div class="box">
                    <i class="fas fa-paint-brush"></i>
                    <div class="text">STUDENT PORTAL</div>
                    <p><asp:LinkButton ID="Btn_Std_Portal"
ForeColor="White" runat="server">User friendly student portal with
intergrated features like attendance , assignment , forms ,
etc.</asp:LinkButton></p>
                </div>
            </div>
            <div class="card">
                <div class="box">
                    <i class="fas fa-chart-line"></i>
                    <div class="text">Library Portal</div>
                    <p><asp:LinkButton ID="Btn_Lib_Portal"
ForeColor="White" runat="server" OnClick="Btn_Lib_Portal_Click">This
Library Management module with Integrated feature like book management
, attendance , books records , etc.</asp:LinkButton></p>
                </div>
            </div>
            <div class="card">
                <div class="box">
                    <i class="fas fa-code"></i>
                    <div class="text">REGISTRATION</div>
                    <p><asp:LinkButton ID="Btn_Reg"
ForeColor="White" runat="server">Here you can regester or enroll new
entries or sutdent in records directly and check the record at
instant.</asp:LinkButton></p>
                </div>
            </div>
        </div>
    </div>
</section>

<!-- footer section start -->
<footer>

```



```

        <span>-----| Created By <a
href="https://www.facebook.com/shiva.alogic">Shivam Rajput</a> | <span
class="far fa-copyright"></span> 2020 All rights reserved. |-----
</span>
    </footer>

</form>
</body>
</html>

```

6.3 Default.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>

<%@ Register src="user_control/Header.ascx" tagname="Header"
tagprefix="uc1" %>
<%@ Register src="user_control/Footer.ascx" tagname="Footer"
tagprefix="uc2" %>

<!-- Created By Shivam Rajput -->

<head runat="server">
    <title></title>
    <link href="StyleSheet.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        .style1
        {
            width: 100px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div id="head">

            <uc1:Header ID="Header1" runat="server" />

        </div>

```

```

<div id="main"><div id="img">
    <table class="style1">
        <tr>
            <td>
                &nbsp;   </td>
            <td>
                <asp:Image ID="Image2" runat="server"
ImageUrl="~/img/5294.jpg" Height="310px"
                style="margin-left: 54px" Width="560px" />
            </td>
        </tr>
    </table>
</div>
<div id="login">
    <table class="tbl">
        <tr>
            <td class="tblhead" colspan="2">
                Login Area</td>
        </tr>
        <tr>
            <td>
                &nbsp;   </td>
            <td>
                <asp:Label ID="lbl" runat="server" Font-
Size="11px" ForeColor="Red"></asp:Label>
            </td>
        </tr>
        <tr>
            <td class="lbl">
                UserName :</td>
            <td>
                <asp:TextBox ID="txtuname" runat="server"
CssClass="txt" Width="175px"></asp:TextBox>
                <asp:RequiredFieldValidator
ID="RequiredFieldValidator1" runat="server"
                ControlToValidate="txtuname"
ErrorMessage="!!!" ForeColor="Red"
SetFocusOnError="True"></asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td class="lbl">
                Password :
            </td>
            <td>

```

```

        <asp:TextBox ID="txtupass" runat="server"
CssClass="txt" TextMode="Password" Width="175px"></asp:TextBox>
        <asp:RequiredFieldValidator
ID="RequiredFieldValidator2" runat="server"
ControlToValidate="txtupass"
ErrorMessage="!!!" ForeColor="Red"
SetFocusOnError="True"></asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>
        &nbsp;&nbsp;&nbsp;</td>
    <td>
        <asp:RadioButton ID="rdolibrary" runat="server"
Checked="True"
        ForeColor="black" GroupName="a"
Text="Librarian" />
        &nbsp;<asp:RadioButton ID="rdosudent" runat="server" ForeColor="black"
GroupName="a"
        Text="Student" />
    </td>
</tr>
<tr>
    <td>
        &nbsp;&nbsp;&nbsp;</td>
    <td>
        <asp:Button ID="Button1" runat="server"
CssClass="btn" Text="Login"
        Width="80px" Font-Size="10pt"
onclick="Button1_Click" />
    </td>
</tr>
</table>
</div>

</div>
    <uc2:Footer ID="Footer1" runat="server" />
</form>
</body>
</html>

```

6.4 Home.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Home.aspx.cs" Inherits="Home" %>
<!-- Created By Shivam Rajput -->

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <style type="text/css">
        .style1
        {
            width: 100%;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <table class="style1">
        <tr>
            <td class="tblhead">
                Welcome to Digital Library System</td>
            </tr>
            <tr>
                <td>
                    &nbsp;</td>
            </tr>
            <tr>
                <td style="text-align: center">
                    <asp:Image ID="Image2" runat="server"
ImageUrl="~/img/8364.jpg" Height="480px"
                    style="margin-left: 54px" Width="660px" />
                </td>
            </tr>
        </table>
</asp:Content>
```

6.4 StyleSheet.css

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    text-decoration: none;
}
```

```
html {  
    scroll-behavior: smooth;  
}  
  
/* custom scroll bar */  
::-webkit-scrollbar {  
    width: 10px;  
}  
  
::-webkit-scrollbar-track {  
    background: #f1f1f1;  
}  
  
::-webkit-scrollbar-thumb {  
    background: #888;  
}  
  
::-webkit-scrollbar-thumb:hover {  
    background: #555;  
}  
  
/* all similar content styling codes */  
section {  
    padding: 100px 0;  
}  
  
.max-width {  
    max-width: 1300px;
```

```
padding: 0 80px;
margin: auto;
}
```

```
.about, .services, .skills, .teams, .contact, footer {
  font-family: 'Poppins', sans-serif;
}
```

```
.about .about-content,
.services .serv-content,
.skills .skills-content,
.contact .contact-content {
  display: flex;
  flex-wrap: wrap;
  align-items: center;
  justify-content: space-between;
}
```

```
section .title {
  position: relative;
  text-align: center;
  font-size: 40px;
  font-weight: 500;
  margin-bottom: 60px;
  padding-bottom: 20px;
  font-family: 'Ubuntu', sans-serif;
}
```

```
section .title::before {  
    content: "";  
    position: absolute;  
    bottom: 0px;  
    left: 50%;  
    width: 180px;  
    height: 3px;  
    background: #111;  
    transform: translateX(-50%);  
}
```

```
section .title::after {  
    position: absolute;  
    bottom: -8px;  
    left: 50%;  
    font-size: 20px;  
    color: crimson;  
    padding: 0 5px;  
    background: #fff;  
    transform: translateX(-50%);  
}
```

```
/* navbar styling */
```

```
.navbar {  
    position: fixed;  
    width: 100%;  
    z-index: 999;  
    padding: 30px 0;
```

```
font-family: 'Ubuntu', sans-serif;  
transition: all 0.3s ease;  
}
```

```
.navbar.sticky {  
    padding: 15px 0;  
    background: crimson;  
}
```

```
.navbar .max-width {  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
}
```

```
.navbar .logo a {  
    color: #fff;  
    font-size: 35px;  
    font-weight: 600;  
}
```

```
.navbar .logo a span {  
    color: crimson;  
    transition: all 0.3s ease;  
}
```

```
.navbar.sticky .logo a span {  
    color: #fff;
```



```
}
```

```
.navbar .menu li {  
  list-style: none;  
  display: inline-block;  
}
```

```
.navbar .menu li a {  
  display: block;  
  color: #fff;  
  font-size: 18px;  
  font-weight: 500;  
  margin-left: 25px;  
  transition: color 0.3s ease;  
}
```

```
.navbar .menu li a:hover {  
  color: crimson;  
}
```

```
.navbar.sticky .menu li a:hover {  
  color: #fff;  
}
```

```
/* menu btn styling */
```

```
.menu-btn {  
  color: #fff;  
  font-size: 23px;
```

```
    cursor: pointer;
    display: none;
}
```

```
.scroll-up-btn {
    position: fixed;
    height: 45px;
    width: 42px;
    background: crimson;
    right: 30px;
    bottom: 10px;
    text-align: center;
    line-height: 45px;
    color: #fff;
    z-index: 9999;
    font-size: 30px;
    border-radius: 6px;
    border-bottom-width: 2px;
    cursor: pointer;
    opacity: 0;
    pointer-events: none;
    transition: all 0.3s ease;
}
```

```
.scroll-up-btn.show {
    bottom: 30px;
    opacity: 1;
    pointer-events: auto;
```

```
}
```

```
.scroll-up-btn:hover {  
    filter: brightness(90%);  
}
```

```
/* home section styling */
```

```
.home {  
    display: flex;  
    background: url("images/profile-14.jpg") no-repeat center;  
    height: 100vh;  
    color: #fff;  
    min-height: 500px;  
    background-size: cover;  
    background-attachment: fixed;  
    font-family: 'Ubuntu', sans-serif;  
}
```

```
.home .max-width {  
    margin: auto 0 auto 30px;  
}
```

```
.home .home-content .text-1 {  
    font-size: 27px;  
}
```

```
.home .home-content .text-2 {
```

```
font-size: 75px;  
font-weight: 600;  
margin-left: -3px;  
}
```

```
.home .home-content .text-3 {  
    font-size: 40px;  
    margin: 5px 0;  
}
```

```
.home .home-content .text-3 span {  
    color: crimson;  
    font-weight: 500;  
}
```

```
.home .home-content a {  
    display: inline-block;  
    background: crimson;  
    color: #fff;  
    font-size: 25px;  
    padding: 12px 36px;  
    margin-top: 20px;  
    font-weight: 400;  
    border-radius: 6px;  
    border: 2px solid crimson;  
    transition: all 0.3s ease;  
}
```

```
.home .home-content a:hover {  
    color: crimson;  
    background: none;  
}
```

```
/* about section styling */
```

```
.about .title::after {  
    content: "who i am";  
}
```

```
.about .about-content .left {  
    width: 45%;  
}
```

```
.about .about-content .left img {  
    height: 400px;  
    width: 400px;  
    object-fit: cover;  
    border-radius: 6px;  
}
```

```
.about .about-content .right {  
    width: 55%;  
}
```

```
.about .about-content .right .text {  
    font-size: 25px;  
    font-weight: 600;
```

```
margin-bottom: 10px;  
}
```

```
.about .about-content .right .text span {  
    color: crimson;  
}
```

```
.about .about-content .right p {  
    text-align: justify;  
}
```

```
.about .about-content .right a {  
    display: inline-block;  
    background: crimson;  
    color: #fff;  
    font-size: 20px;  
    font-weight: 500;  
    padding: 10px 30px;  
    margin-top: 20px;  
    border-radius: 6px;  
    border: 2px solid crimson;  
    transition: all 0.3s ease;  
}
```

```
.about .about-content .right a:hover {  
    color: crimson;  
    background: none;  
}
```

```
/* services section styling */
```

```
.services, .teams {  
    color: #fff;  
    background: #111;  
}
```

```
.services .title::before,  
.teams .title::before {  
    background: #fff;  
}
```

```
.services .title::after,  
.teams .title::after {  
    background: #111;  
    content: "what i provide";  
}
```

```
.services .serv-content .card {  
    width: calc(33% - 20px);  
    background: #222;  
    text-align: center;  
    border-radius: 6px;  
    padding: 20px 25px;  
    cursor: pointer;  
    transition: all 0.3s ease;  
}
```

```
.services .serv-content .card:hover {  
    background: crimson;  
}
```

```
.services .serv-content .card .box {  
    transition: all 0.3s ease;  
}
```

```
.services .serv-content .card:hover .box {  
    transform: scale(1.05);  
}
```

```
.services .serv-content .card i {  
    font-size: 50px;  
    color: crimson;  
    transition: color 0.3s ease;  
}
```

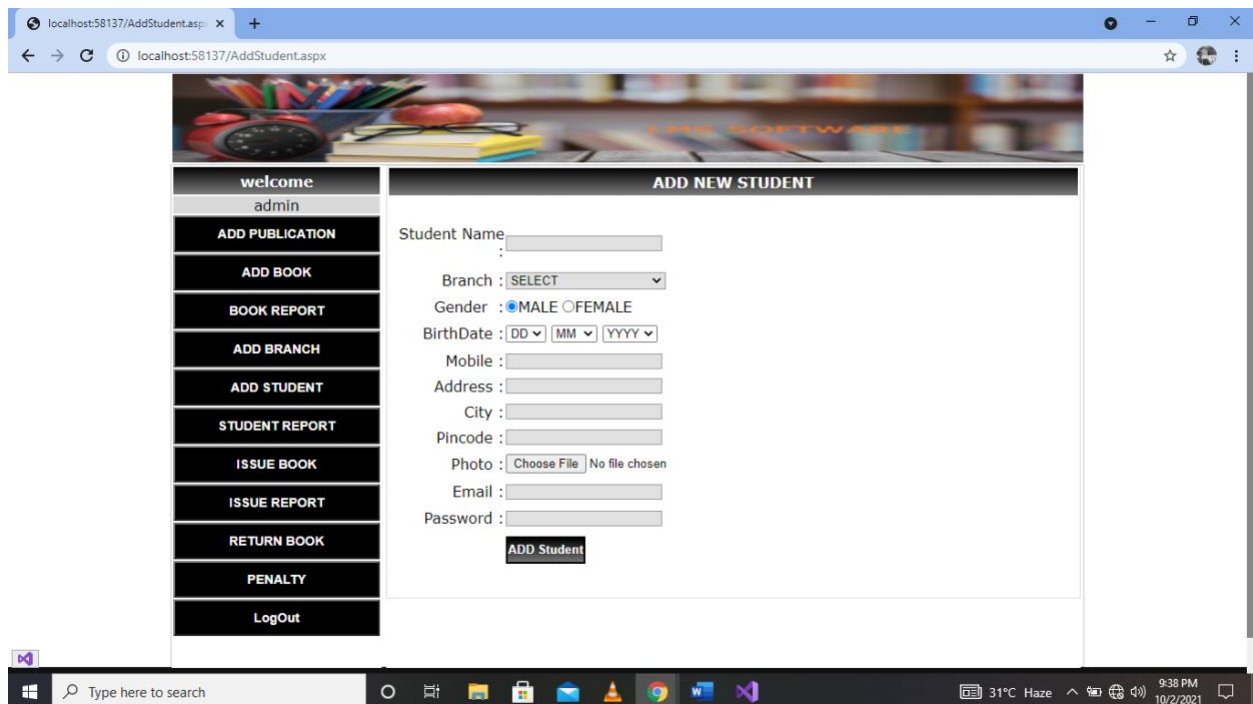
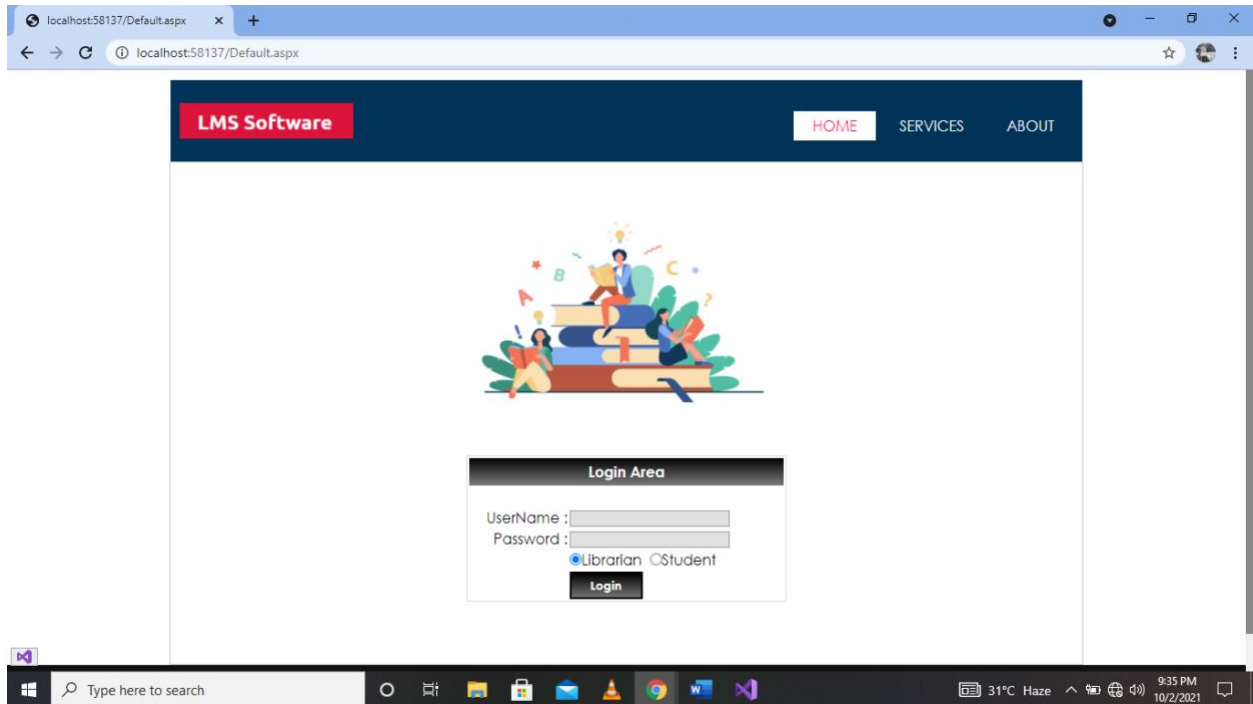
```
.services .serv-content .card:hover i {  
    color: #fff;  
}
```

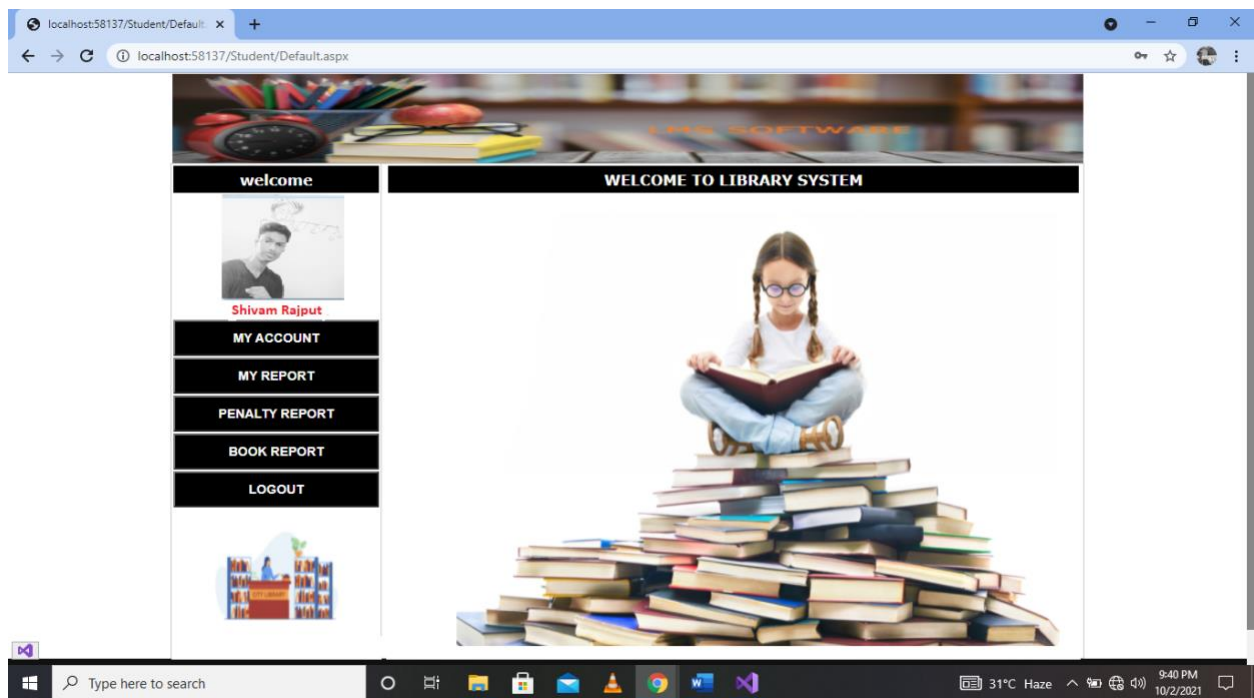
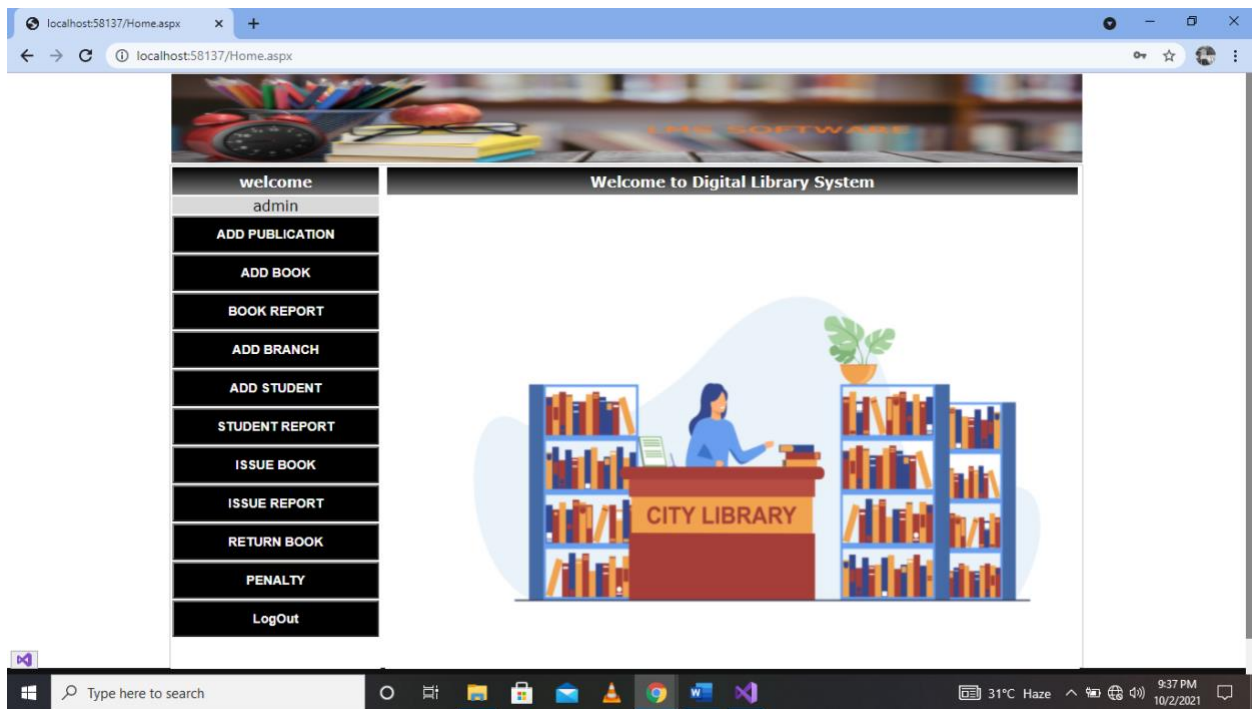
```
.services .serv-content .card .text {  
    font-size: 25px;  
    font-weight: 500;  
    margin: 10px 0 7px 0;  
}
```


CHAPTER – 7

SOFTWARE SCREENSHOTS

7.1 Screenshot:





CHAPTER - 8

CONCLUSION

I have presented Asp.net , our approach to facilitate the incremental consolidation of data-intensive Flows. ASP.NET is a server-side technology used for developing dynamic web applications.it was released in 2002 and had an extension of .aspx . It produces interactive , data driven web applications on the internet.

for example : twitter , Instagram , Facebook , etc.

➤ *Benefits of ASP .NET :-*

- a) High performance*
- b) Secure*
- c) Multiple development mode*
- d) Language independent*
- e) Globalization and localization*

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices. ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .NET platform ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present

Co Asp.net starts from data Flows that satisfy single information requirements. Iteratively, Co Asp.net Identifies different possibilities for integrating new data Flows into the existing multi-Flow, focusing on the maximal data Flow reuse. Finally, Co AI suggests a unified data Flow design evaluating it with the user-specified cost model. We have developed a prototype that implements the complete functionality of Co AI. We used it to evaluate the efficiency, scalability, and the quality of the output solutions of our approach, reporting the improvement of the overall execution time as well as other benefits of integrated multi- Flows. The final goal of our overall work is to provide an end-to-end platform for self-managing the complete lifecycle of BI solutions, from information requirements to deployment and execution of data-intensive Flow

REFERENCES

The following books are referred during the analysis and execution phase of the project 1. M. Lenzerini, “Data integration: A theoretical perspective,” in PODS, 2002, pp. 233– 246. 2. D. Caruso, “Bringing Agility to Business Intelligence,” February 2011, Information Management,[http://www.information-management.com/infodirect/2009191/business intelligence metadata analytics ETL data management-10019747-1.html](http://www.information-management.com/infodirect/2009191/business%20intelligence%20metadata%20analytics%20ETL%20data%20management-10019747-1.html). 3. R. Hughes, Agile Data Warehousing: Delivering world-class business intelligence systems using Scrum and XP. IUniverse, 2008. 4. Y. Chen, S. Alspaugh, and R. Katz, “Interactive analytical processing in big data systems: A cross-industry study of map reduce workloads,” Proceedings of the VLDB Endowment, vol. 5, no. 12, pp. 1802–1813, 2012.

WEBLINKS: 1. www.c#tutorial.com - covering all the most important C# concepts. This tutorial is primarily for new users. 2 .www.DotnetSpider.com