Vidyalankar Polytechnic

CERTIFICATE

This is to certify that

Rishi Gupta

Shivraj Patil

Abhishek More

Adesh Borchate

Have successfully completed the project titled

STUDENT CASH MANAGER

A partial fulfillment requirement of Third Year Diploma in

INFORMATION TECHNOLOGY ENIGINEERING

Affiliated to Maharashtra state Board of Technical

Education , Mumbai for the Academic Year 2015-2016

Internal examiner **External Examiner**

Project Guide **Head of Department** **Principal**

A PROJECT REPORT ON

Student Cash Manager

Submitted By

Rishi Gupta

Shivraj Patil

Abhishek More

Adesh Borchate

Under the Guidance of

Prof. Sushma Pawar

Head of Department

Prof. Sanjay Shimpy

Department of Information Technology

Vidyalankar Polytechnic

Wadala(E),Mumbai 400037

Maharashtra State Board of Technical Education,Mumbai

2015-2016

Acknowledgement

It has been a sincere desire of every individual to get an opportunity to express his views , skill and talent in which he is proficient to give him satisfaction and confidence in his ability to do or produce something useful for humankind. A project one such avenue through wich an engineer gives vent to his feelings and expressions.

We feel privileged to express our deepest sense of gratitude and sincere thanks to our guide **Mrs. Sushma Pawar** for her excellent guidance throughout the project work

We would like to express our sincere gratitude and thanks towards our Head of Department **Prof. Sanjay Shimpi** and our **Principal Prof. Ashish** Ukidve whose help and inspiration provided beneficial to us.

Their prompt and kind help directed us to competition of the dissertation work.

We would like to thank the lab assistant of computer department for providing us with the requisite facilities to complete the project work

Finally, we would like to acknowledge all our friends those who directly or indirectly helped ud in the completion

Abstract

The general distinction between an interactive web site of any kind and a "web application" is unclear. Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a mobile app. HTML5 introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.

Single-page applications are more application-like because they reject the more typical web paradigm of moving between distinct pages with different URLs. A single-page framework like Sencha Touch might be used to speed development of such a web app for a mobile platform.

In earlier computing models like client–server, the processing load for the application was shared between code on the server and code installed on each client locally. In other words, an application had its own pre-compiled client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server-side code of the application would typically also require an upgrade to the client-side code installed on each user workstation, adding to the support cost and decreasing productivity.

In addition, both the client and server components of the application were usually tightly bound to a particular computer architecture and operating system and porting them to others was often prohibitively expensive for all but the largest applications.

As common approaches to web application building it requires knowledge of ASP, CGI, ColdFusion, Dart, JSP/Java, Node.js, PHP, Python it may be very time consuming.
So we have decided to develop a Web Application. We have taken idea from our day to day activities and develop this web application on Visual Studio on which we have used ASP ton develop this web application.

Despite the dominance of many other Web Application in both consumer and business applications, there is still simple straightforward way of producing them.
As there was no option to generate the code of ASP so only thing that was available was to write code in ASP,CGI and Python.

Table of Contents

Introduction

Introduction

In computing, a **web application** or **web app** is a client–server software application in which the client runs in a web browser.
Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility. Common web applications include webmail, online retail sales, online auctions, wikis, instant messaging services and many other functions.
The general distinction between an interactive web site of any kind and a "web application" is unclear. Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a mobile app. HTML5 introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.
Single-page applications are more application-like because they reject the more typical web paradigm of moving between distinct pages with different URLs. A single-page framework like Sencha Touch might be used to speed development of such a web app for a mobile platform.
There are several ways of targeting mobile devices:

- Responsive web design can be used to make a web application - whether a conventional web site or a single-page application viewable on small screens and work well with touchscreens
- Native apps or "mobile apps" run directly on a mobile device, just as a conventional software application runs directly on a desktop computer, without a web browser
- Hybrid apps embed a mobile web site inside a native app, possibly using a hybrid 1framework like Apache Cordova or React Native. This allows development using web technologies

HTML

Hypertext Markup Language, commonly referred to as **HTML**, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.
HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.
The language is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like `<html>`). Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page.

HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

ASP.NET

ASP.NET is an open-source[3] server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.
It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.
ASP.NET is in the process of being re-implemented as a modern and modular web framework, together with other frameworks like Entity Framework. The new framework will make use of the new open-source .NET Compiler Platform (code-name "Roslyn") and be cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) will merge into a unified MVC 6.[4] The project is called ASP.NET vNext.

PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994 the PHP reference implementation is now produced by The PHP Group.[4] PHP originally stood for *Personal Home Page* but it now stands for the recursive backronym *PHP: Hypertext Pre-processor*.
PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications

Jackson structured programming (JSP)

Jackson structured programming (**JSP**) is a method for structured programming based on correspondences between data stream structure and program structure. JSP structures programs and data in terms of sequences, iterations and selections, and as a consequence it is applied when designing a program's detailed control structure, below the level where object-oriented methods become important.
Michael A. Jackson originally developed JSP in the 1970s. He documented the system in his 1975 book *Principles of Program Design*. Jackson's aim was to make COBOL batch file processing programs easier to modify and maintain, but the method can be used to design programs for any programming language that has structured control constructs, languages such

as C, Java and Perl. Despite its age, JSP is still in use and is supported by diagramming tools such as Microsoft's Visio and CASE tools such as Jackson Workbench.
Jackson Structured Programming was seen by many as related to Warnier structured programming, but the latter method focused almost exclusively on the structure of the output stream. JSP and Warnier's method both structure programs and data using only sequences, iterations and selections, so they essentially create programs that are parsers for regular expressions which simultaneously match the program's input and output data streams.
Because JSP focuses on the existing input and output data streams, designing a program using JSP is claimed to be more straightforward than with other structured programming methods, avoiding the leaps of intuition needed to successfully program using methods such as top-down decomposition.
Another consequence of JSP's focus on data streams is that it creates program designs with a very different structure to the kind created by the stepwise refinement methods of Wirth and Dijkstra. One typical feature of the structure of JSP programs is that they have several input operations distributed throughout the code in contrast to programs designed using stepwise refinement, which tend to have only one input operation. Jackson illustrates this difference in Chapter 3 of *Principles of Program Design*. He presents two versions of a program; one designed using JSP, the other using "traditional" methods.

Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.
Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source-control systems and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle.
It also supports different programming languages and allows the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++ and C++.

Review of Literature

Review of Literature

Need for the new system

Web Application is the preferred way of interacting between the server and the client.

One of the reasons behind this is that a Web Application program that is stored on a remote server and delivered over the Internet through a browser interface. In addition, web application can also handle real-time data and results.

Other aspects to consider are also human factor such as ease of use, real-time, behind the goa of the work presented here. On the other hand advanced web application are expensive tools that a common user or too low to mid-complexity projects might not afford.

So we have decided to develop a Web Application. We have taken idea from our day to day activities and develop this web application on Visual Studio on which we have used ASP ton develop this web application.

Despite the dominance of many other Web Application in both consumer and business applications, there is still simple straightforward way of producing them.

As there was no option to generate the code of ASP so only thing that was available was to write code in ASP, CGI and Python.

With this project we are trying to achieve two factors which will be very important for any software developer

- dividing the work
- making more reliable web application

Presently available system

As such there was no option to generate the code of web application only the thing was available to write code in Visual Basic and to do some trial and error methods so that we can get a proper layout which was time consuming process for a web application developer.

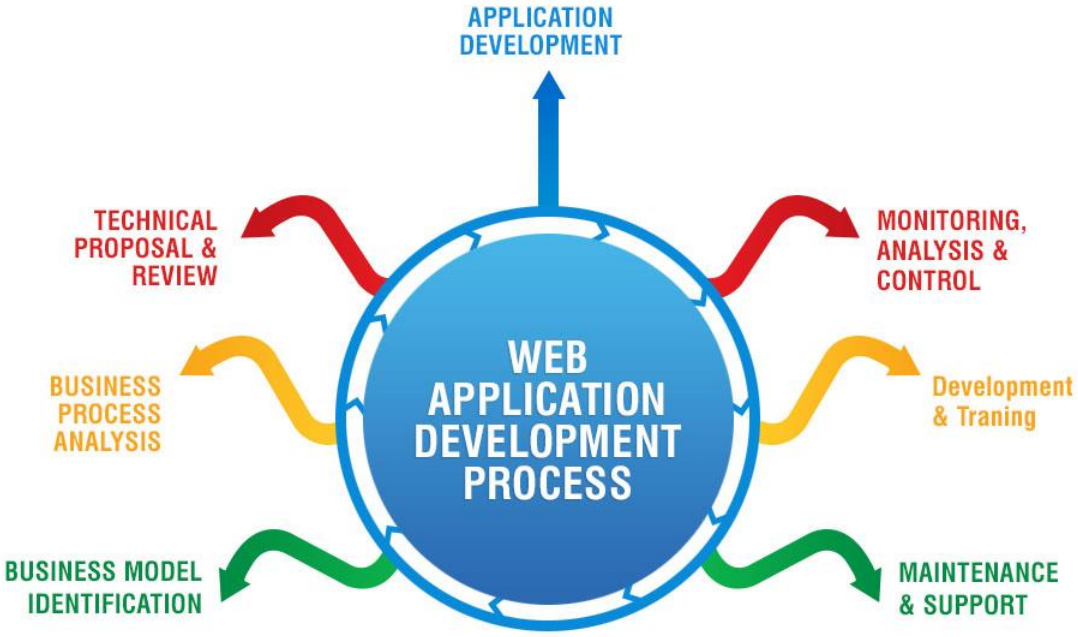
Below is detail description

- Visual Studio: *Visual Studio* is a comprehensive collection of developer tools and services to help you create web application

- Components of Visual Studio

Planning

Planning



Planning is an objective if each and every activity, where we want to discover things that belong to the project . An important task in creating software program is extracting the requirements or requirement analysis .Costumers typically have an abstract idea of what they want as end result , but not what software should do. Skilled and experienced software engineers recognize incomplete , ambiguous , or even contradictory requirement at his point . Frequently demonstrating live code may help reduce the risk that the requirement are incorrect . Once the general requirement are gathered from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope of the project as a function of cost or as a result of unclear requirements at the start of development

Implementation and documenting

Implementing part of the process where software engineers actually program the code for project. Software testing is an integral and important the code for the project . Software testing

is an integral and important phase of the software development process. This part of the process ensure that defects are recognized as soon as possible .

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development .This may also include the writing of an API,be external or internal. The software engineering process chose by the development team will determine how much internal documentation is necessary

Process Models

There are different types of models used by a team to do their work systematic that is step by step. The original process models have certainly given a guideline or roadmap for the whole development process.

The process models are properly structured so that at least structure of the process can be understood. The different process models are:

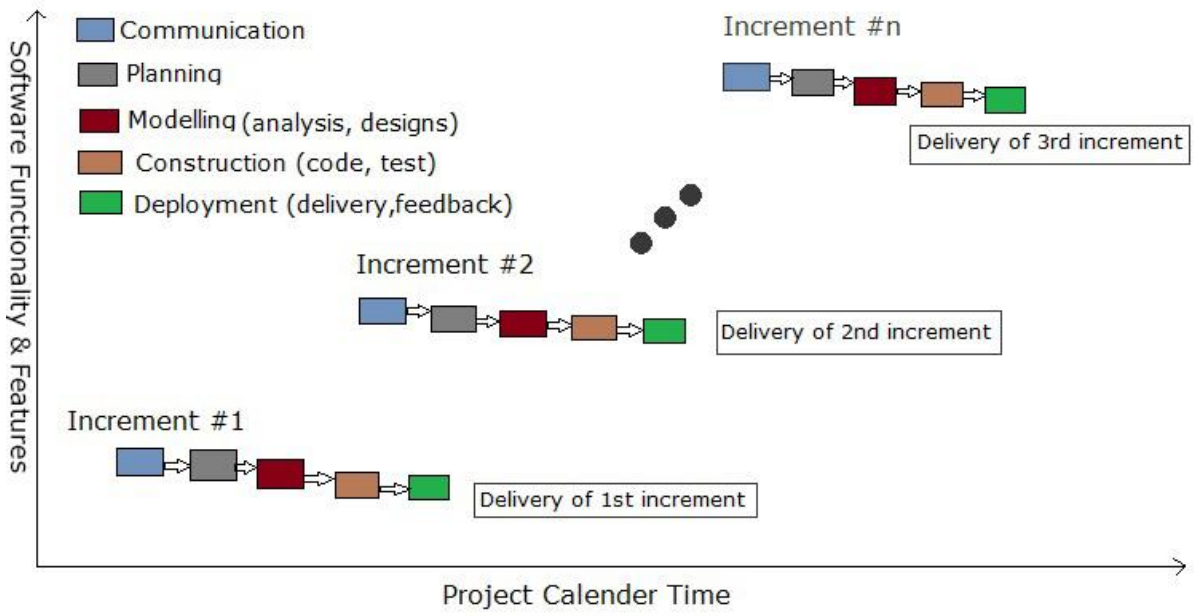
1. Waterfall Model
2. Incremental Model
3. RAD Model
4. Prototype Model
5. Spiral Model

We have chosen Incremental Model for development of our project

Incremental Model

The incremental build model is a method of software development where the product is designed, implemented and tested incrementally until the product is finished. It involves both development and maintenance. In this the progress is seen through the phases of

- Communication
- Planning
- Modelling
- Construction
- Deployment



Communication- In this requirements are collected from the customer via survey, magazine review, competitive product, etc.

Planning- In this phase, scheduling is done so that it helps to track the progress of the project and there is no delay in delivery of the project, and the project cost is also estimated.

Modelling- Here the project requirement are analysed into three domains are formed namely behaviour domain, function domain and information domain.

Construction- In this phase layout ids designed using the selected programming language and then proceeded by through incremental testing

Deployment- This is the final stage and here the application is ready and help is provided in the form of support.

Advantages

- After each iteration, regression testing should be conducted. During this testing, faulty elements of the software can be quickly identified because few changes are made within any single iteration.
- It is generally easier to test and debug than other methods of software development because relatively smaller changes are made at each iteration. This allows for more targeted and rigorous testing of each element within the overall product.
- Initial product delivery is faster and costs lower.

Plan of Work

PLAN OF WORK

The primary objective of this phase are to identify the scopes of the new system, ensures that the project is feasible, develop a schedule, allocate the resources and budget for the remainder of the project.

The plan for the project was mad in month of August with some amount of research for the components and similar systems present in the market. We schedule the work according to our academic calendar in a way that it would not affect our studies

After the analysis phase we started our work directly in December with designing and modelling phase

In the modelling phase we made the flow charts, use case diagrams, and also discussed the incremental model on which our project was based on.

Coding phase was started in last week of December and went for weeks. In the 3rd week of coding we began with the 1st part f testing phase where we tested the basic features provided and decided to add some features which would add more value to the system.

Flow Diagram

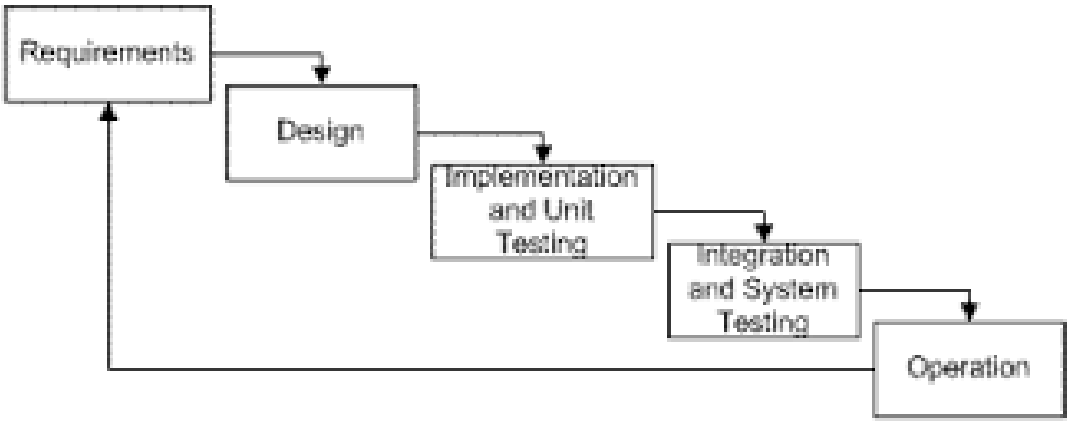
Flow Diagram

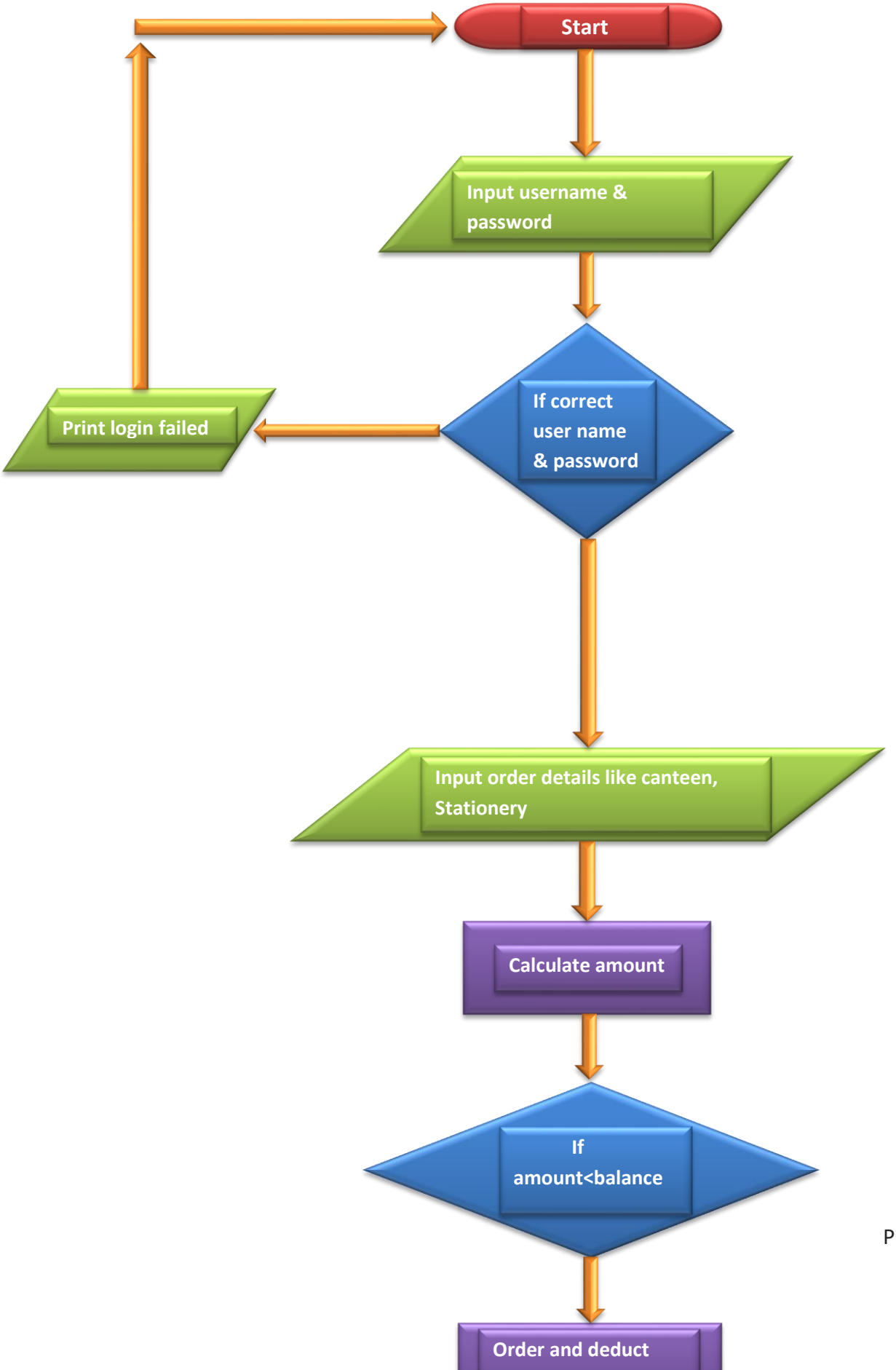
Flow diagram is a collective term for a diagram representing a flow or set of dynamic relationships in a system. The term flow diagram is also used as synonym of the flowchart, and sometimes as counterpart of the flowchart.

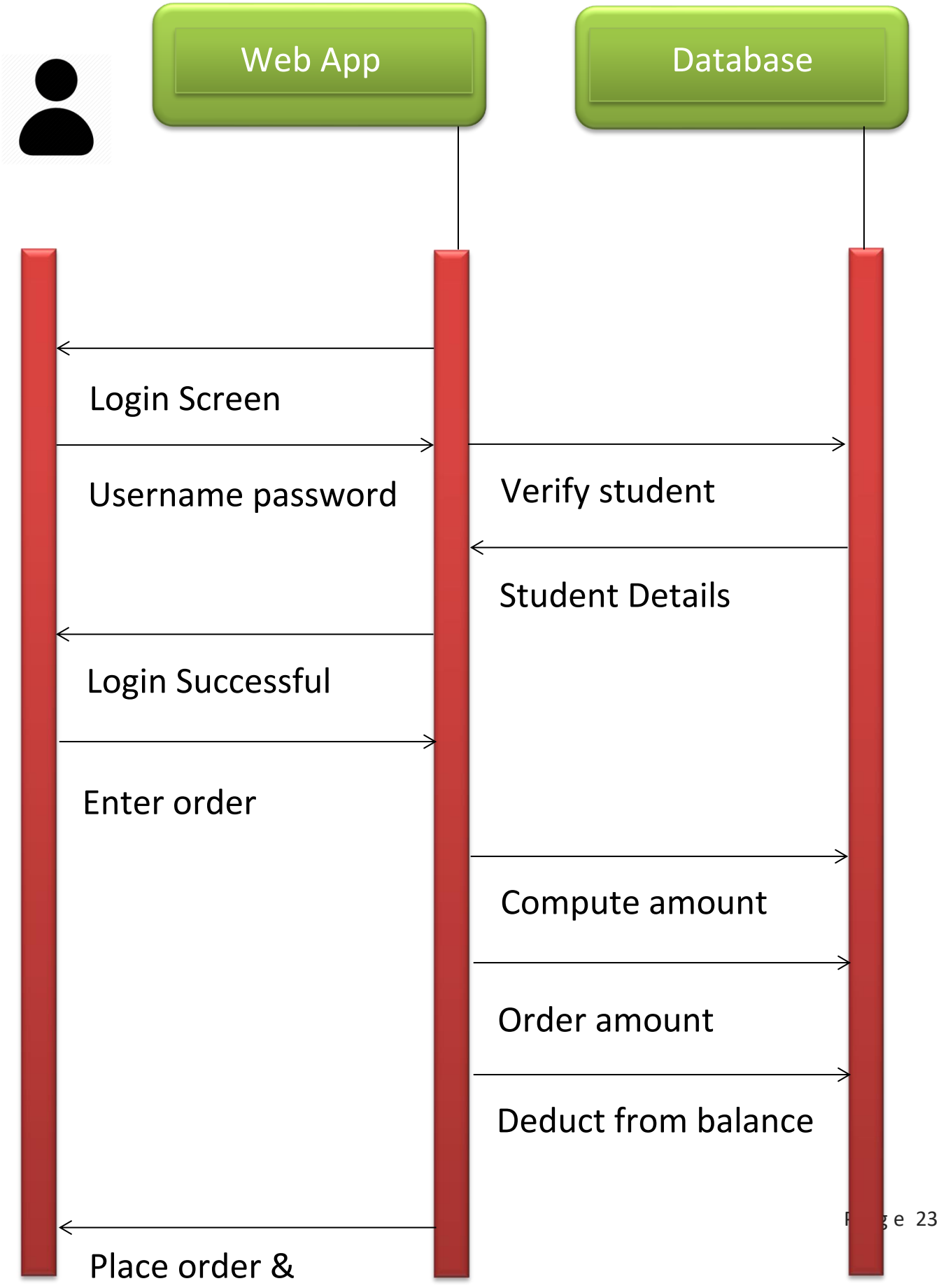
Flow diagrams are used to structure and order a complex system, or to reveal the underlying structure of the elements and their interaction

Flow chart or flow diagram... is a diagram that visually displays interrelated information such as events, steps in a process, functions, etc., in an organized fashion, such as sequentially or chronologically.

Flow diagram a graphic representation of the physical route or flow of people, materials, paper works, vehicles, or communication associated with a process, procedure plan, or investigation.




Start
Input username & password
If correct user name & password
Print login failed
Input order details like canteen, Stationery
Calculate amount
If amount<balance
Order and deduct


Web App
Database
Login Screen
Username password
Verify student
Student Details
Login Successful
Enter order
Compute amount
Order amount
Deduct from balance
Place order &
Logout

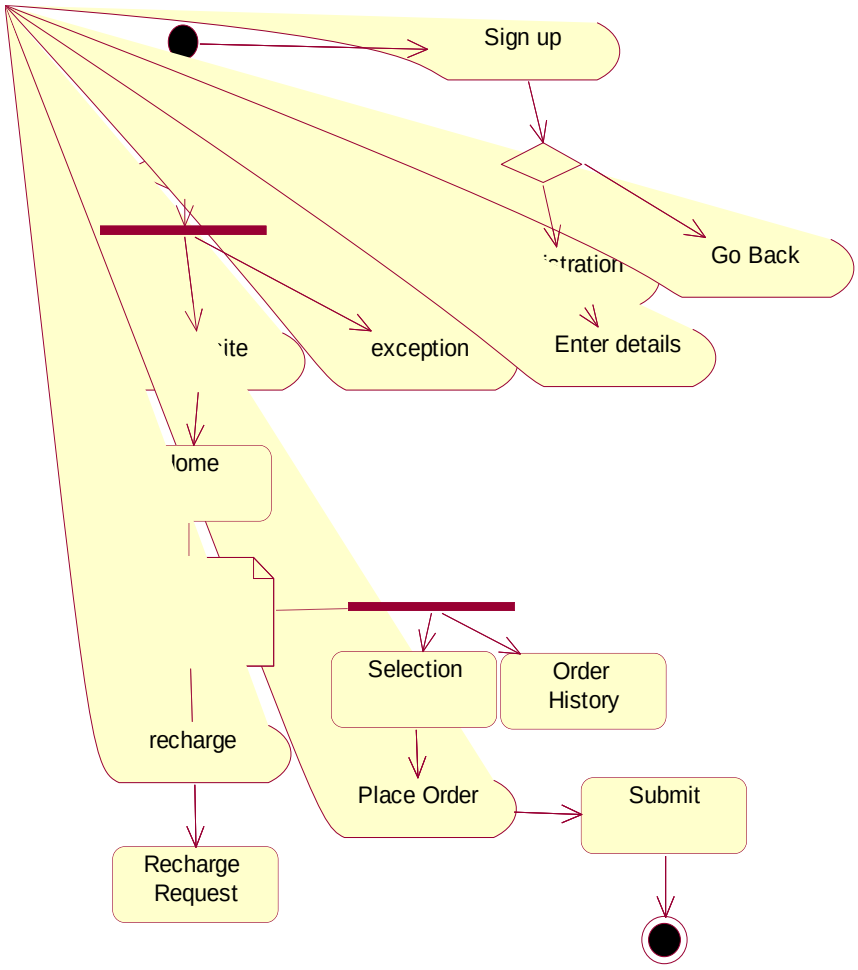
Activity Diagram

Activity Diagram

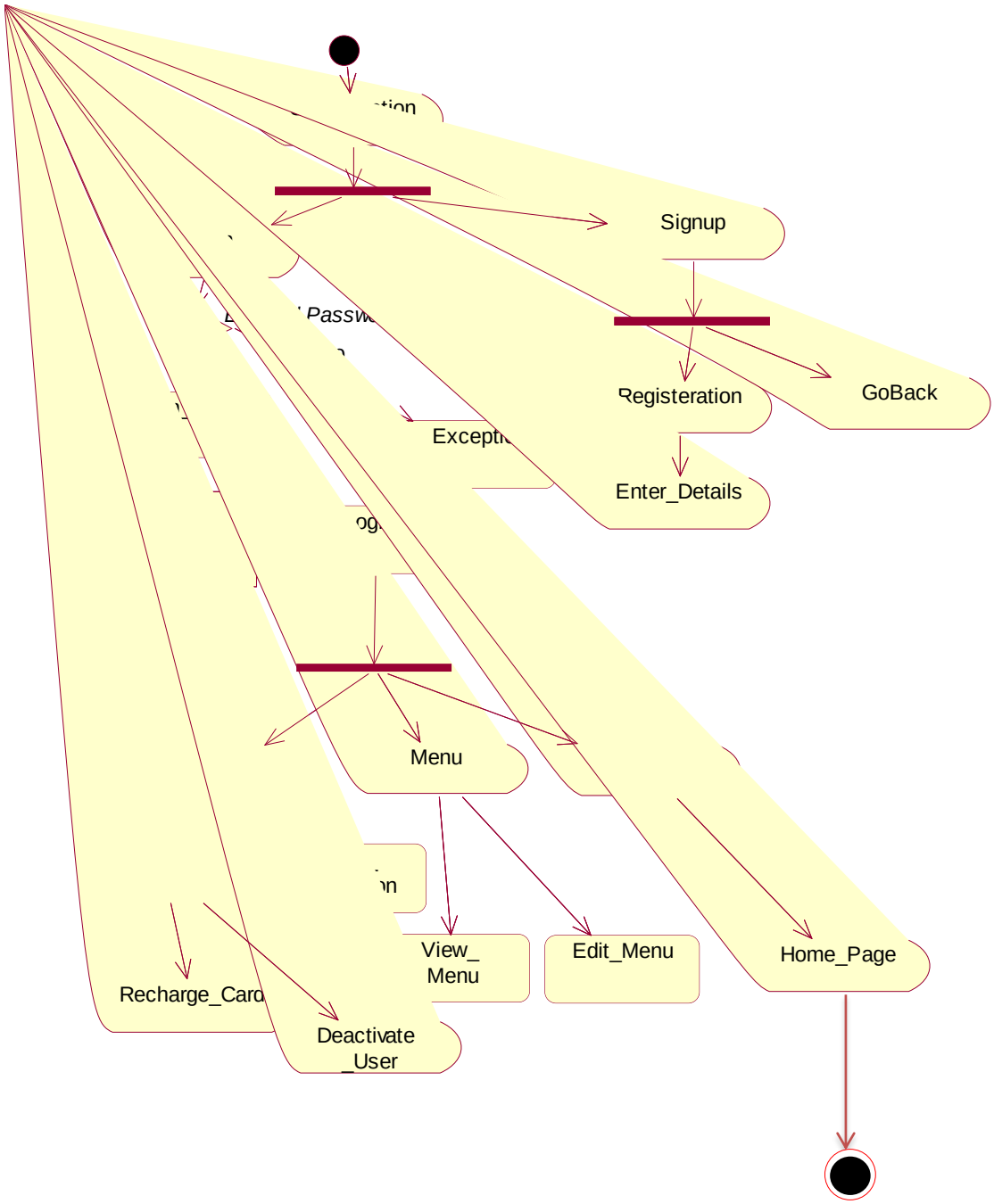
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes . Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows.[4] The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activity .
- **Student**



Admin


Signup
Registeration
GoBack
Enter_Details
Menu
View_
Menu
Edit_Menu
Home_Page
Deactivate
_User

Project Implementation

Project implementation

Why use ASP.NET?

Microsoft ASP.NET is more than the next generation of Active Server Pages. It provides an entirely new programming model for creating network applications that take advantages of the internet.

Improved performance and Scalability

Compiled execution ASP.NET is much faster than classic ASP, while preserving the “just hit save” update model of ASP. No explicit compile step is required ASP.NET automatically detects any change, dynamically compiles files if needed, and stores the compiled results to reuse for subsequent requests. Dynamic compilation ensures that your application is always up to date, and compiled execution makes it faster. Most applications migrates from classic ASP.NET

Rich Output Caching: ASP.NET output caching can dramatically improve the performance and scalability of your application. When output caching is enabled on a page, ASP.NET executes the page once the saved the results in memory before sending to the user. When another user requests the same page, ASP.NET serves the cached result from memory without re-executing the page. Output caching is configuring, and it can be used to cache individual region or an entire page.

Web Farm Session State: ASP.NET session state lets you share session data across all machines in Web farm. Now a user can hit different servers in the web farm over multiple requests and still have full access to session data.

Enhanced Reliability

Memory Leak, Dead lock and Crash Protection: ASP.NET automatically detects and recovers from errors such as dead locks and memory leaks to ensure that your application is always available. For example, when memory leak is detected, ASP.Net automatically starts up a new copy of the ASP.NET worker process and directs all new requests to the process. After the old process has finished processing pending requests, it is gracefully disposed of the leaked memory is release finished processing pending request , it is gracefully disposed of and the leaked memory is released.

Easy Deployment

“No Touch” Application Deployment : With .Net you can deploy an entire application by copting it to the server. Configuration settings are stored in an XML file within the application

Dynamic update of running application: .net lets you update compiled components without restarting the web servers. Unlike classic COM components that required the web server to be manually restarted when an update was deployed .Net automatically detects the change and starts using the new code.

Easy Migration Path: .Net runs side by side on IIs with classic Asp

Application on Microsoft Windows 2000 and on members of the Windows Server 2003 family. You can migrate one application at a time or even single page

New Application Models

.XML Web Services: XML Web services allow application to communicate and share data over the internet, regardless of operating system or programming language. Asp.net makes exposing and calling XML Web services simple.

Mobile Web Device Support ASP.Net mobile control let you target over 80

Mobile Web Device Support ASP.NET. You write the application once, and the mobile controls automatically generate pages for the requesting devices.

Developer Productivity

Easy Programming Model Asp.Net makes building real-world Web application dramatically easier with server control that let you build great pages with far less code than classic Asp.

Flexible Language Options. ASP.NET supports not only Microsoft Visual Basic Scripting Editing and Microsoft Jscript but also more than 25.NET Languages, including built-in support for Visual Basic .NET , Microsoft c#, and Jscript.NET.

Rich Class Framework The .Net Framework class library offers over 4,500

Classes that encapsulate rich functionality such as XML, date access, file upload, regular expression image generation , performance monitoring and logging transactions , message queuing , and SMTP mail.

Advantages of ASP.NET:

- ASP.NET drastically reduces the amount of code required to build large applications.
- With built-in Windows authentication and per-application configuration, your applications are safe and secured.
- It provides better performance by taking advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box.
- The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server

controls, and automatic deployment are just a few of the features this powerful tool provides.

- Provides simplicity as ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration.
- The source code and HTML are together therefore ASP.NET pages are easy to maintain and write. Also the source code is executed on the server. This provides a lot of power and flexibility to the web pages.
- All the processes are closely monitored and managed by the ASP.NET runtime, so that if process is dead, a new process can be created in its place, which helps keep your application constantly available to handle requests.
- It is purely server-side technology so, ASP.NET code executes on the server before it is sent to the browser.
- Being language-independent, it allows you to choose the language that best applies to your application or partition your application across many languages.

Navigation Sequence of ASP.NET:

IIS starts the ASP.NET worker process that it turns, loads the assembly.
The assembly composes of a response to the user
IIS returns the response to the user in the form of HTML.

ASP.NET web form components:

In ASP.NET Web Forms pages, user interface programming is divided into two pieces: the visual component and the logic. If you have worked with tools like Visual Basic and Visual C++ in the past, you will recognize this division between the visible portion of a page and the code that interacts with it.

The visual element consists of a file containing static mark-up such as HTML or ASP.NET server controls or both. The ASP.NET Web Forms page works as a container for the static text and controls you want to display.

The logic for the Web Forms page consists of code that you create to interact with the page. The code can reside either in a script block in the page or in a separate class. If the code is in a separate class file, this file is referred to as the code-behind file. The code in the code-behind file can be written in Visual Basic, C#, or any other .NET Framework language. For more information about how ASP.NET Web Forms pages are constructed, see ASP.NET Web Forms Page Code Model.

For ASP.NET Web Forms site projects, you deploy page source code to a Web server and the pages are compiled automatically the first time a user browses to any page in the site. For ASP.NET Web application projects, you must compile the Web Forms pages before deployment and deploy one or more assemblies. For more information about differences between how the two project types compile Web Forms pages, see Web Application Projects versus Web Site Projects in Visual Studio

SQL Server 2014:

SQL Server 2014 was released to manufacturing on March 18, 2014, and released to the general public on April 1, 2014. Until November 2013 there were two CTP revisions, CTP1 and CTP2. SQL Server 2014 provides a new in-memory capability for tables that can fit entirely in memory. Whilst small tables may be entirely resident in memory in all versions of SQL Server, they also may reside on disk, so work is involved in reserving RAM, writing evicted pages to disk, loading new pages from disk, locking the pages in RAM while they are being operated on, and many other tasks. By treating a table as guaranteed to be entirely resident in memory much of the 'plumbing' of disk-based databases can be avoided.

For disk-based SQL Server applications, it also provides the SSD Buffer Pool Extension, which can improve performance by cache between DRAM and spinning media. SQL Server 2014 also enhances the “Always On” solution by increasing the readable secondary’s count and sustaining read operations upon secondary-primary disconnections, and it provides new hybrid disaster recovery and backup solutions with Microsoft Azure, enabling customers to use existing skills with the on-premises version of SQL Server to take advantage of Microsoft's global datacentres . In addition, it takes advantage of new Windows Server 2012 and Windows Server 2012 R2 capabilities for database application scalability in a physical or virtual environment.

Pros & Cons

PROS & CONS

The project has been keeping in mind that it should give maximum solution in the user and make learning easier or make learning a good experience.

Advantages

- Easy to operate.
- No requirement of keeping hard cash.
- Students would be able to get all the available items without standing in queue.
- Provide an alternative to current system
- Students can track history and would be able to control their account just like Vprint.

Disadvantages

- English interface only.
- Requires internet connection to the database.
- Requires computer with net connectivity to access the application service.

Application

Application

Our web application can be applied to any organizations, as each organization has services applied by our WEB APPLICATION.

This application is initially developed for VIDYALANKAR POLYTECHNIC. But this application can be used for similar schools, colleges and various business organization

0

FUTURE DEVELOPMENT

FUTURE DEVELOPMENT

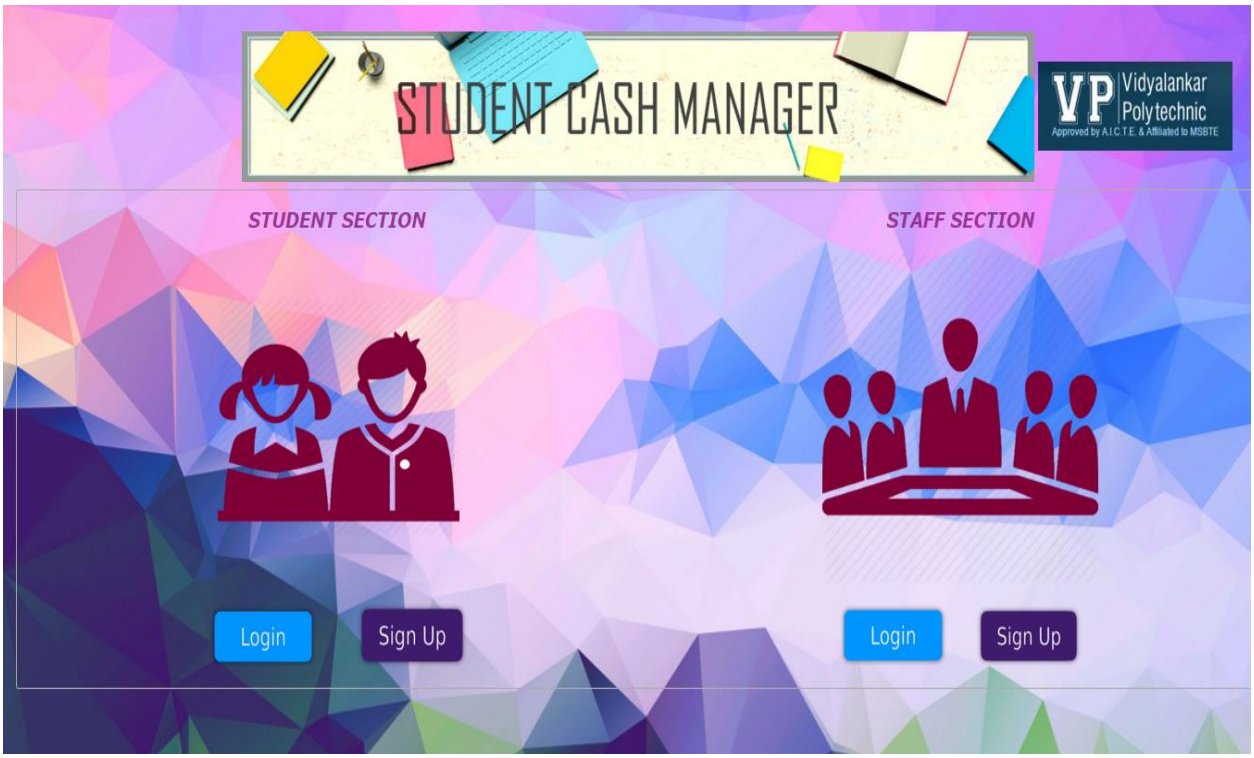
Anything that is made can always be improved in better and in accordance with the technologies and requirements at that time.

Considering the wide scope for improving the application, in the future the following functionalities can be provide to the application

- Application in regional languages
- Provision of the items or services in the application
- Controlling the total account of student
- Can link to Vprint.

Code & Screenshot


Canteen
Home Recharge Card Canteen Stationery Library My Transactions Change Password Logout
You are logged in as : 132020015 Sign-Out
Your Current Available Balance | 130.00 Rs.
OUR MENU
Already Feeling Hungry??... Go Ahead... Order something for you...
NAME
ITEM NAME
ORDER QUANTITY
ITEM UNIT PRICE
Veg Burger Maharaja
50.00
dhokla
20.00
MIsal Pav
30.00
Veg Pizza
60.00
Samosa
12.00
PlaceOrder


STUDENT CASH MANAGER
VP Vidyalankar Polytechnic
Approved by A.I.C.T.E. & Affiliated to MSBTE
STUDENT SECTION
STAFF SECTION
Login
Sign Up
Login
Sign Up

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="PlaceOrder.aspx.cs"
Inherits="CMS.Student.PlaceOrder" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>:: Place Order ::</title>
<link rel="Stylesheet" href="~/CSS/PageStyler.css" />
<style type="text/css">
.auto-style1 {
width: 615px;
height: 199px;
}
.auto-style2 {
width: 344px;
height: 137px;
}
.auto-style3 {
height: 139px;
width: 332px;
}
.auto-style4 {
height: 137px;
width: 359px;
}
.auto-style7 {
border-collapse: collapse;
width: 100%;
background-color: #8C85CB;
color: black;
}
</style>
</head>
<body class="body">
<form id="frmPlaceOrder" runat="server">
<br />
<table width="100%">
<tr>
<td align="center">
<img src="../Images/42.png" class="auto-style1" />
</td>
<td align="right">
```

```
 </td>
</tr>
</table>
<table width="100%" border="1px" bordercolor="#3366FF" class="blueNavigationbar">
<tr>
<td class="regulartext">
<asp:Menu ID="menuBar" runat="server" Orientation="Horizontal" BackColor="#F7F6F3"
DynamicHorizontalOffset="2"
Font-Names="Verdana" StaticSubMenuIndent="10px">
<StaticSelectedStyle BackColor="#5D7B9D" />
<StaticMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />
<DynamicHoverStyle BackColor="#7C6F57" ForeColor="White" />
<DynamicMenuStyle BackColor="#F7F6F3" />
<DynamicSelectedStyle BackColor="#5D7B9D" />
<DynamicMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />
<StaticHoverStyle BackColor="#7C6F57" ForeColor="White" />
<Items>
<asp:MenuItem Text="Home" NavigateUrl="~/Student/StudentHome.aspx"
Enabled="true"></asp:MenuItem>
<asp:MenuItem Text="Recharge Card">
<asp:MenuItem Text="Recharge Request"
NavigateUrl="~/Student/RechargeRequest.aspx"></asp:MenuItem>
<asp:MenuItem Text="Recharge History"
NavigateUrl="~/Student/RechargeHistory.aspx"></asp:MenuItem>
</asp:MenuItem>
<asp:MenuItem Text="Canteen">
<asp:MenuItem Text="Place Order" NavigateUrl="~/Student/PlaceOrder.aspx"></asp:MenuItem>
<asp:MenuItem Text="Order History" NavigateUrl="~/Student/OrderHistory.aspx"></asp:MenuItem>
</asp:MenuItem>
<asp:MenuItem Text="Stationery">
<asp:MenuItem Text="Place Order"
NavigateUrl="~/Student/PlaceStationeryOrder.aspx"></asp:MenuItem>
</asp:MenuItem>
<asp:MenuItem Text="Library">
<asp:MenuItem Text="Browse Library" NavigateUrl="~/Student/BrowseLibrary.aspx"></asp:MenuItem>
<asp:MenuItem Text="Submit Request"
NavigateUrl="~/Student/LibraryBookIssueRequest.aspx"></asp:MenuItem>
<asp:MenuItem Text="My Library Subscription Tracker"
NavigateUrl="~/Student/MyLibrarySubscriptionTracker.aspx"></asp:MenuItem>
</asp:MenuItem>
<asp:MenuItem Text="My Transactions">
```

```
<asp:MenuItem Text="View Transaction History"
NavigateUrl="~/Student/TransactionHistory.aspx"></asp:MenuItem>
</asp:MenuItem>
<asp:MenuItem Text="Change Password" NavigateUrl="~/ChangePassword.aspx"></asp:MenuItem>
<asp:MenuItem Text="Logout" NavigateUrl="~/Logout.aspx"></asp:MenuItem>
</Items>
</asp:Menu>
</td>
<td class="regulartext" align="right">You are logged in as :
<asp:Label ID="lblUserName" runat="server" ForeColor="ActiveCaptionText"></asp:Label>
<asp:LinkButton ID="lnkSignOut" runat="server" Text="Sign-Out"
PostBackUrl="~/Logout.aspx"></asp:LinkButton>
</td>
</tr>
</table>
<table align="right" width="15%">
<tr>
<td class="regulartext">Your Current Available Balance :
<asp:Label ID="lblAvailableBalance" runat="server"></asp:Label>
 Rs.
</td>
</tr>
</table>
<div>
<table align="left" width="100%" class="tableStyle">
<tr>
<td>
<img src="../Images/vada-pav.jpg" class="auto-style2" />
<img src="../Images/BurgerFrechFries.jpg" class="auto-style4" /> 
<img src="../Images/Samosa.jpg" class="auto-style3" /> 
</td>
</tr>
<tr>
<td class="blueNavigationbar"></td>
</tr>
<tr>
<td class="pagetitle">OUR MENU</td>
</tr>
<tr>
<td class="regulartext" align="center">
<b>Already Feeling Hungry??... Go Ahead... Order something for you...</b>
```

```
</td>
</tr>
<tr>
<td class="alerts-green">
<asp:Label ID="lblMessage" runat="server" Visible="false"></asp:Label>
</td>
</tr>
<tr>
<td>
<strong>
<asp:GridView ID="gvMenu" runat="server"
DataKeyNames="ItemID,ItemCode,ItemImage,ItemName,ItemCategory,ItemUnitPrice"
AutoGenerateColumns="False" CssClass="auto-style7" HeaderStyle-BackColor="#61A6F8"
ShowFooter="True" HeaderStyle-Font-Bold="true" HeaderStyle-ForeColor="White"
AllowPaging="false" SelectedIndex="1">
<Columns>
<asp:TemplateField>
<ItemTemplate>
<asp:CheckBox ID="chkRow" runat="server" />
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField Visible="false">
<ItemTemplate>
<asp:Label ID="lblItemID" runat="server" Text='<%#Eval("ItemID") %>' />
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField Visible="false">
<ItemTemplate>
<asp:Label ID="lblItemCode" runat="server" Text='<%#Eval("ItemCode") %>' />
</ItemTemplate>
</asp:TemplateField>
<asp:ImageField DataImageUrlField="ItemImage" DataImageUrlFormatString="~/Images/{0}"
HeaderText="NAME" ControlStyle-Width="90px" ControlStyle-Height="90px" ItemStyle-Wrap="true"
ItemStyle-HorizontalAlign="Center">
</asp:ImageField>
<asp:TemplateField HeaderText="ITEM NAME" ItemStyle-HorizontalAlign="Center">
<ItemTemplate>
<asp:Label ID="lblItemName" runat="server" Text='<%#Eval("ItemName") %>' />
</ItemTemplate>
<ItemStyle HorizontalAlign="Center"></ItemStyle>
</asp:TemplateField>
```

```
<asp:TemplateField HeaderText="ORDER QUANTITY" ItemStyle-HorizontalAlign="Center">
<ItemTemplate>
<asp:DropDownList ID="ddlOrderQty" runat="server" CssClass="dropdownStyle">
<asp:ListItem Selected="True" Value="1" Text="1"></asp:ListItem>
<asp:ListItem Value="2" Text="2"></asp:ListItem>
<asp:ListItem Value="3" Text="3"></asp:ListItem>
<asp:ListItem Value="4" Text="4"></asp:ListItem>
<asp:ListItem Value="5" Text="5"></asp:ListItem>
</asp:DropDownList>
</ItemTemplate>
<ItemStyle HorizontalAlign="Center"></ItemStyle>
</asp:TemplateField>
<asp:TemplateField HeaderText="ITEM UNIT PRICE" ItemStyle-HorizontalAlign="Center">
<ItemTemplate>
<asp:Label ID="lblItemUnitPrice" runat="server" Text='<%#Eval("ItemUnitPrice") %>' />
</ItemTemplate>
<ItemStyle HorizontalAlign="Center"></ItemStyle>
</asp:TemplateField>
</Columns>
<HeaderStyle BackColor="#61A6F8" Font-Bold="True" ForeColor="White"></HeaderStyle>
</asp:GridView>
</strong>
<asp:Button ID="btnPlaceOrder"
runat="server" Text="PlaceOrder" OnClick="btnPlaceOrder_Click" />
</td>
</tr>
<tr>
<td class="blueNavigationbar"></td>
<td id="tdTitle" runat="server" class="pagetitle">YOUR ORDER SELECTION</td>
</tr>
<tr>
<td id="tdSubtitle" runat="server" class="regulartext" align="center">
<b>Wow!! Yummy!! It smells Nice.... Can't wait...</b>
<tr>
<td>
<strong>
<asp:GridView ID="gvSelected" runat="server" CssClass="auto-style7" HeaderStyle-
BackColor="#61A6F8" ShowFooter="True" HeaderStyle-Font-Bold="true" HeaderStyle-
ForeColor="White"
AutoGenerateColumns="false">
<Columns>
```

```
<asp:BoundField DataField="ItemID" Visible="false" />
<asp:BoundField DataField="ItemCode" Visible="false" />
<asp:BoundField DataField="ItemName" HeaderText="Name" ItemStyle-Width="150" />
<asp:BoundField DataField="ItemUnitPrice" HeaderText="Price" ItemStyle-Width="150" />
<asp:BoundField DataField="Quantity" HeaderText="Quantity" ItemStyle-Width="150" />
<asp:BoundField DataField="Total" HeaderText="Total" ItemStyle-Width="150" />
</Columns>
</asp:GridView>
</strong>
</td>
</tr>
<tr>
<td class="blueNavigationbar" align="center">ORDER SUMMARY</td>
</tr>
<tr>
<td class="pagetitle">Order ID:
<asp:Label ID="lblOrderID_Display" runat="server"></asp:Label>
</td>
</tr>
<tr>
<td class="pagetitle">Total Order Amount:
<asp:Label ID="lblOrderTotalAmt" runat="server"></asp:Label>
</td>
</tr>
<tr>
<td class="blueNavigationbar"></td>
</tr>
<tr>
<td>
<asp:Button ID="btnConfirm" runat="server" Text="Proceed" OnClick="btnConfirm_Click" />
<asp:Button ID="btnBack" runat="server" Text="Let me Add something more" OnClick="btnBack_Click"
/>
<asp:Label ID="lblOrderID" runat="server" Visible="false"></asp:Label>
<tr>
<td class="blueNavigationbar"></td>
</div>
</form>
</body>
</html>
```

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>::Landing Page::</title>
<link rel="Stylesheet" href="CSS/PageStyler.css" />
<style type="text/css">
.auto-style1 {
width: 397px;
height: 370px;
}
.auto-style2 {
height: 263px;
width: 348px;
margin-top: 0px;
}
.auto-style3 {
margin-left: 53px;
}
.auto-style7 {
margin-left: 36px;
}
.auto-style8 {
width: 1142px;
height: 167px;
margin-left: 337px;
margin-top: 0px;
}
.auto-style12 {
font: normal 15px Verdana, Helvetica, sans-serif;
border: 1px #aeadad solid;
border-radius: 3px;
color: blue;
flex-align: center;
margin-left: 19px;
width: 98%;
```

```
height: 554px;
margin-top: 3px;
}
.auto-style13 {
width: 935px;
}
.auto-style16 {
width: 1359px;
height: 112px;
}
.auto-style17 {
width: 1359px;
}
.auto-style19 {
width: 1239px;
height: 112px;
}
.auto-style20 {
width: 1239px;
}
.auto-style22 {
width: 71px;
}
.auto-style23 {
width: 38px;
}
.auto-style24 {
width: 1028px;
font-size: x-large;
color: #993399;
}
.auto-style25 {
width: 1239px;
font-size: x-large;
color: #993399;
}
.auto-style26 {
width: 97%;
height: 131px;
}
</style>
```

```
</head>
<body class="body" style="height: 1187px; margin-left: 0px; margin-bottom: 0px; width: 1837px;">
<form id="frmLandingPage" runat="server">
<br />
<table class="auto-style26">
<tr>
<td align="center" class="auto-style23">
</td>
<td class="auto-style13">

<img src="Images/BookCover/books-desk-laptop-whiteboard-next-to-few-pencils-66835731.png"
class="auto-style8" />
</td>
<td align="right">
<img alt="Vidyalankar Polytechnic Logo" src="Images/VidyalankarPolytechnic.jpg" />
</td>
<td align="center" class="auto-style22">
</td>
</tr>
</table>
<table class="auto-style12">
<tr>
<td class="auto-style24" valign="middle" align="center"><strong><em>STUDENT SECTION
</em></strong>
</td>
<td class="auto-style25" valign="middle" align="center"><strong><em>STAFF SECTION
</em></strong>
</td>
</tr>
<tr>
<td align="center" class="auto-style16">
 <img src="Images/MYADDS/Students-512.png" class="auto-style2" /></td>
<td align="center" class="auto-style19">
<img src="Images/MYADDS/Meeting-512.png" class="auto-style1" />
</td>
</tr>
<tr>
<td align="center" class="auto-style17">
 <asp:ImageButton ID="imgLoginButton" runat="server" ImageUrl="~/Images/BookCover/thin-
cyan login-button-md.png" OnClick="imgLoginButton_Click" Width="154px" Height="70px" />
```

```
<asp:ImageButton ID="imgRegisterButton" runat="server" ImageUrl="~/Images/MYADDS/signup.png"
OnClick="imgRegisterButton_Click" CssClass="auto-style3" Height="73px" Width="160px" />
</td>
<td align="center" class="auto-style20">
<asp:ImageButton ID="imgStaffLoginButton" runat="server" ImageUrl="~/Images/BookCover/thin-cyan
login-button-md.png" Height="70px" OnClick="imgStaffLoginButton_Click" />
<asp:ImageButton ID="imgStaffSignUpButton" runat="server"
ImageUrl="~/Images/MYADDS/signup.png" OnClick="imgStaffSignUpButton_Click" CssClass="auto-
style7" Width="153px" />
</td>
</tr>
</table>
<div>
</div>
</form>
</body>
</html>
```

BIBLOGRAPHY/REFERENCES

Biblography/References

- ASP.NET
- .NET Frame Work Essentials
- www.wikipedia.org/wiki/VisualBasic.NET
- www.draw.io
- MacDonald, Matthew; Mario (2005). Pro ASP.NET 2.0 in C# 2005 (1st Ed.). Après. *ISBN 1-59059-496-7*.

- Anne Boehm: Marchs ASP.NET 3.5 Web Programming with VB 2008, July 21, 2008, Mike Murach and Associates, ISBN 978-1-890774-47-9
- Stephen Walther: ASP.NET 3.5 Unleashed, December 28, 2007, Sam's Publishing, ISBN 0-672-33011-3 ISBN 0-672-33011-3
- Stephen Walther: Data Access in the ASP.NET 2.0 Framework (Video Training), September 26, 2007, Sam's Publishing, ISBN 0-672-32952-2
- Israel B. Cobia: Mastering VB.NET and C#. 7th Edition. October 22, 2004. by Cyberocbina© Cafe. A .NET Developers Edition.