Mosheshe and Odeyinka, J Comput Eng Inf Technol 2017, 6:6
DOI: 10.4172/2324-9307.1000185

Journal of Computer Engineering & Information Technology

Research Article

A SCITECHNOL JOURNAL

Developing and Testing a Magazine Website: The Chartist

Jennifer Mosheshe[1*] and Olatunbosun Odeyinka[2]

Abstract

Online magazine websites are one of the major traditional platforms for publishing various magazine contents through various means of public computer networks, such as the bulletin board system, in order to share historical artefacts, online newspapers, blog features, and several forms of political occurrence online. Due to increased competitiveness in the world market, several developers are experiencing numerous challenges, for website development. These challenges can be broadened in diverse editorial and end users issues, requiring an abrupt and effective strategic development and testing to positively address these issues in order to assist developers in creating robust platforms during website development. This paper shows the development and testing process of a political e-commerce magazine website named “The Chartist”, which included the development of the checkout page, download article button, saved article page and gallery page, using dream weaver, flash and ASP.Net during the design phase.

Keywords

Magazine website; Software development; White box testing; ASP. NET; Dream weaver; Robust platforms; Databases; SQL server; PHP/MYSQL

Introduction

This report documents the entire process of developing a magazine website tool for ASP.Net and Html web development using dream weaver as the dynamic and interactive environment for designing the web pages in the website. All the software development stages are been documented in this report which also include interim reports. For the development of the project, the waterfall model was adopted and the system was tested and evaluated. The output of the project is application software which is called “the chartist” political magazine website where readers can read articles online, purchase articles and order magazines online. It retrieves data from the database and tabulates it. And also converts html codes to dream weaver in the design view [1]. The inspiration of this project stemmed from Chartist Magazine Website, which publishes political contents of the democratic left [2].

The website has been created to assist end-users with purchasing and reading articles online as well as Magazines. In this website the end- user will need to log in or register on the account page where he or she will need to purchase any magazine of his or her choice, after purchase and payment has been successful, the end-user will need to select the download article button which will then re direct the end-user to another page called the Saved Articles Page which will contain all the articles that the end-user has previously purchased. Now the end- user can view all the Articles which he has earlier requested for on this saved articles page as required. The website will include a handful of magazines in the gallery page where the user will be able to achieve the ability to view any magazines that he or she will love to purchase after which those magazines can be sent by post to the end-users address if purchased. This report will analyse the different stages of the website what had to be done in order to create a user friendly interactive e- commerce political magazine website [1].

Project scope

The website that has been created has been designed for people who are interested in the political matters, which allow users who will prefer to get political articles online instead of having a bulk full of newspapers which can be littering the whole place they would prefer an electronic document instead and they have the knowledge of browsing the internet. The website will allows users to familiarise themselves with the program. Individual web pages were created on different areas within the dream weaver environment, they were then edited and implemented in to a website which was done using the ASP.Net environment and the use of hyperlinks to connect different pages together [3].

Aims and objectives

The main aim for this project is to provide the public world who is interested in political matters to be able to successfully purchase and view article online. In achieving this goal the objectives are:

1. **Branding:** This talks about underlining messages with the use of copying, using logos and designs in the whole website.
2. **Illustrating message:** This can be used in getting particular messages to end users by the use of images, sound, words and movements to get my message across to the end-users.
3. **Lead generation:** This describes the generation of a consumer interest into products of a business, there will be a qualified lead arriving from my website straight into my existing sales process for end users.
4. **Increased sales:** This is my ability to convert chartist into sales by emphasizing and focusing on the benefits of my articles and the services rendered by me to the end users.
5. **Cost reduction:** This can be achieved with the help of intelligent systems which can help to cut down administration and processing overhead.
6. **Building relationships:** Some people have a wrong idea thinking that gathering business cards alone will help achieve this aim but I realise that to establish stronger bonds with end users I will need constant contact with them through my website [4].

Literature Review

Programming tools help programmers in writing programs by

*Corresponding author: Jennifer Mosheshe, Department of Information Technology, Cardiff Metropolitan University, Cardiff, United Kingdom, Tel: +44 29 2041 6070; E-mail: jennymosh@yahoo.com

Received: June 24, 2017 Accepted: August 11, 2017 Published: August 17, 2017

providing code generation and editing capabilities. A programming tool or software development tool is a program or application that software developers use to create, debug, maintain, or otherwise support other programs and applications. The use of a programming tool can maximize programmer's productivity by reducing the time required to write code, minimize error and debug. The goal of application programming tool is to decrease the drudgery of programming while increasing the efficiency and productivity of programmers. Looking at the open source community, it can be seen to have a high community of developers, where help could be easily found and bugs could be easily fixed. To my best knowledge there is still less development tools for developer as my basic comparison to the commercial application developments such as ASP.NET but I still decided to take a step, by developing an application tool for ASP.NET which would retrieve content from a database and format the content in a table called the GRIDVIEW in ASP.NET in the area of the saved articles page by writing codes and also designing web pages as a developer [5].

Identifying ASP.Net and html development problems relating to the project: I met with ASP.Net and html developers to get additional information about the issue concerning data retrieval and tabulating, and the conversion of HTML script respectively through a friend of mine who introduced me to them and gave me the opportunity to have a conversation with them via phone. Understanding development syntax thoroughly: I made research and reading on html scripting and ASP.Net web development methods, code and syntax to aid a good problem solution [6].

Analysing: I analysed the system. And I analysed problems related and involved in the development of the software, websites and my proposed solutions.

Designing: In designing the system, I thought about defining the architecture, components, modules and interface for the system.

Implementation: Here, I thought about designing a user-friendly Graphical User Interface, to aid simplicity of the tool so that users can move around the environment easily while accessing which articles they need to purchase and also I thought about coding the system.

Testing: I tested the application to make sure it is up to requirement and has fewer / no bugs.

User evaluation: I made Verifications and validations to make sure that the tool works as expected and met the requirements, then I distributed the software to users to the minimum of six (6) for the purpose of evaluation.

Modify the application: I made necessary change on the application regarding the user evaluation test result.

Deployment: Finally I decided to package the software and making it available for use, which is the major aim of my project [6].

On the analysis done on the project, I was able to address some ASP.Net and html developers live and via email on issues that was intended to be addressed in the project. I also made a research on how ASP.Net web application is written. As the result of the analysis, I was able to deduce that the tool will be very useful as most developers do the coding from scratch and it consumes time and effort. I was able to find the different ways in which HTML can be used with dream weaver. Before beginning to create my own e-commerce website, research had to be carried out to look at other websites which offer a similar service. I began my research with Chartist website which is the main model for my website on (www.chartist.org.uk). I found out that this website the website was really interactive and has a lot of articles that users will love. I also found out that the colour of the website was really bright and attractive to users this will in turn attract users around the globe. Also there was no background music, it has a huge font size readers can view easily, and there were no overlapping windows or annoying pop up. The only limitation of the website that I found was that you can't make money out of it as the articles in the website are free and there is no login, register or account page this will in turn cause lack of finances in maintaining the website [2] (Figure 1).

To actually achieve my goal I adopted object oriented web modelling methodology in my project implementation. Also, based my research and analysis I developed a conceptual view of the entire system which shows all objects within the system domain. Additionally I point out sections wish can be further developed to ensure long-term functionality of the website. In this Chapter I am going to be discussing on the steps taking in designing my website that actually helped me in achieving my desired goal.

Method of approach

Project management is the discipline of planning, organizing, securing and managing resources to bring about the successful completion of specific project goals and objectives.

There are different approaches to managing project / project management methodology [7]. Some of the project management methodologies are:

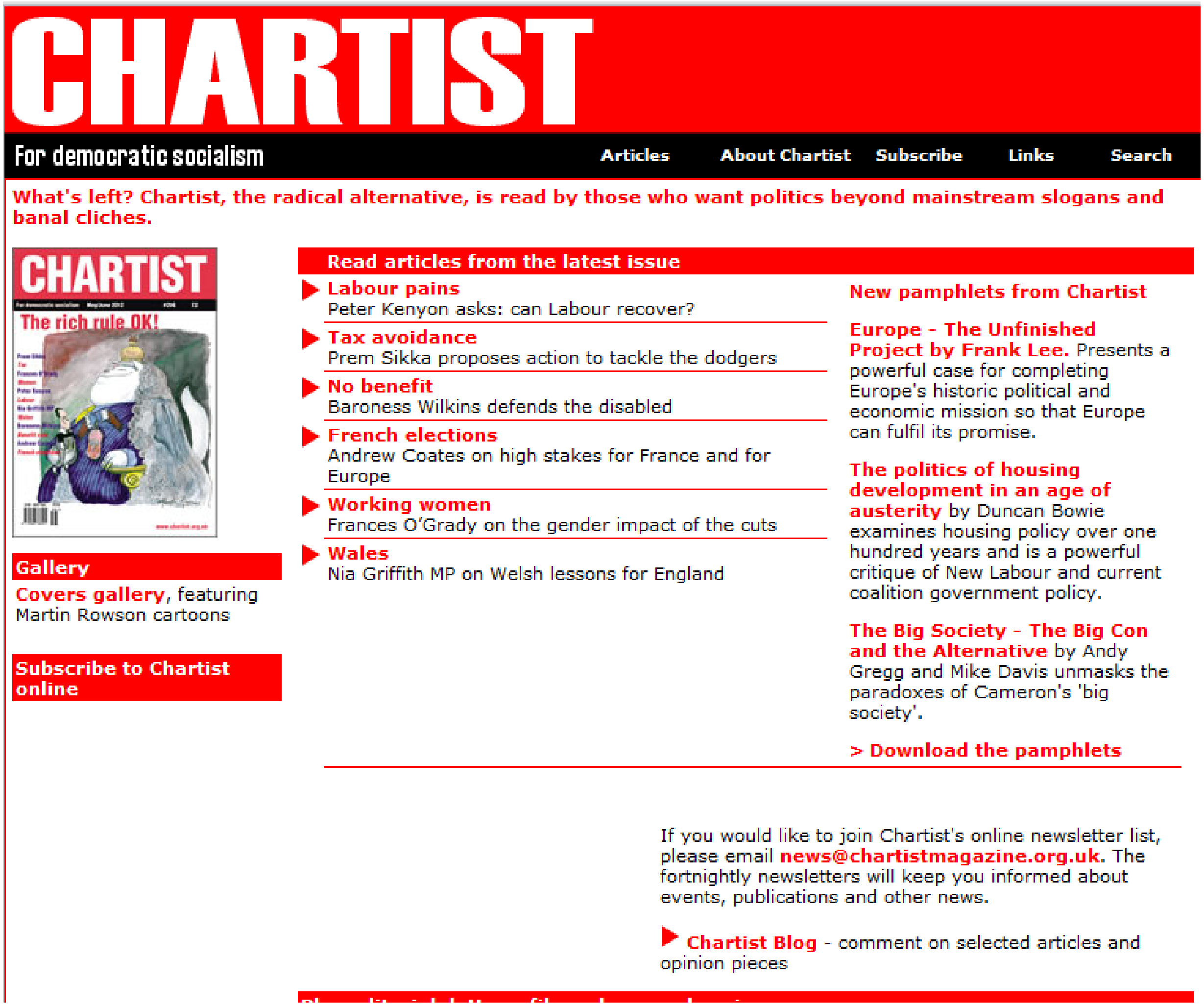
- Agile Methodology
- Waterfall Methodology
- Agile Versus Waterfall
- Change Management
- Risk Management
- Quality Management
- PRINCE2
- Six Sigma [7].

Project management implementation

For this project, I used the PRINCE2 methodology. PRINCE2 (Projects in Controlled Environments) is a process-based method for effective project management. PRINCE2 is a process-based approach for project management, providing an easily tailored and saleable project management methodology for the management of all types of projects.

The key features of PRINCE2 are:

- Its focus on business justification
- A defined organisation structure for the project management team
- Its product-based planning approach
- Its emphasis on dividing the project into manageable and controllable stages
- Its flexibility to be applied at a level appropriate to the project [8].



CHARTIST

For democratic socialism

Articles About Chartist Subscribe Links Search

What's left? Chartist, the radical alternative, is read by those who want politics beyond mainstream slogans and banal cliches.

Gallery

Covers gallery, featuring Martin Rowson cartoons

Subscribe to Chartist online

Read articles from the latest issue

Labour pains
Peter Kenyon asks: can Labour recover?

Tax avoidance
Prem Sikka proposes action to tackle the dodgers

No benefit
Baroness Wilkins defends the disabled

French elections
Andrew Coates on high stakes for France and for Europe

Working women
Frances O'Grady on the gender impact of the cuts

Wales
Nia Griffith MP on Welsh lessons for England

New pamphlets from Chartist

Europe - The Unfinished Project by Frank Lee. Presents a powerful case for completing Europe's historic political and economic mission so that Europe can fulfil its promise.

The politics of housing development in an age of austerity by Duncan Bowie examines housing policy over one hundred years and is a powerful critique of New Labour and current coalition government policy.

The Big Society - The Big Con and the Alternative by Andy Gregg and Mike Davis unmasks the paradoxes of Cameron's 'big society'.

> Download the pamphlets

If you would like to join Chartist's online newsletter list, please email **news@chartistmagazine.org.uk**. The fortnightly newsletters will keep you informed about events, publications and other news.

Chartist Blog - comment on selected articles and opinion pieces

Figure 1: Chartist magazine website source: [2].

Software development methodology

In choosing the right software development methodology for my project, which is as important to the success of the project as the implementation of any project management best practices. A software development methodology is a framework that is used to structure, plan, and control the process of developing an information system, this includes the pre-definition of specific deliverables and artefact's that are created and completed by a project team to develop or maintain an application.

Software development life cycle

Software development life cycle involves phases which generally include the planning, definition, requirement, design, building, implementation and maintenance.

Software development life cycle can be defined as a methodology or models which are used to develop computer systems. Without a software development life cycle a computer system may not certify the requirement if a particular system or client. Or if they do certify, there is a tendency for the project to go over budget. As stated by Yu Gao [9] "For the success of developing software, Software development process model plays a crucial role in order to complete the task of developing software and to get high quality software"

The stages of the software development life cycle include:

1. Project planning
2. System analysis
3. System design
4. Implementation
5. Testing
6. Maintenance

The six stages of the software development life cycle are designed to build on one another, taking the outputs from the previous stage, adding additional effort, and producing results that leverage the previous effort and are directly traceable to the previous stages.

Evaluation of different software development life cycle

There are different types of software development life cycle. The waterfall model, incremental model and spiral model (which are the common once), will be evaluated.

The waterfall model: It's a framework for software development in which development proceeds sequentially through a series of phases, starting with system requirements analysis and leading up to product release and maintenance.

The states of the waterfall model include requirements analysis, design, implementation, testing, installation and maintenance (Figure 2).

Pros: The waterfall model can aid efficient knowledge transfer when team members are dispersed in different locations.

- Every phase has a defined start and end point; and progress can be conclusively identified (through the use of milestones).
- Enforceddisciplinethroughdocuments.
- No phase is complete until the docs are done & checked by SQA group.
- Concrete evidence or progress
- Testing is inherent in every phase

Cons: No fair division of phases in the life cycle, the following phase should not start until the previous phase has finished. Document driven model as result customers cannot understand these.

- Re-designisproblematic.

The incremental model: Incremental model is an evolution of waterfall model. The product is designed, implemented, integrated and tested as a series of incremental builds.

When an incremental model is used, the first increment is often a core product. The core product is used by the customer or undergoes a detailed review (Figure 3).

Cons: One of the main advantages of the incremental models is their ability to cope with change during the development of the system.

- System is developed and delivered in increments after establishing an overall architecture. Requirements and specifications for each increment may be developed...
- Generates working software quickly and early during the software life cycle.

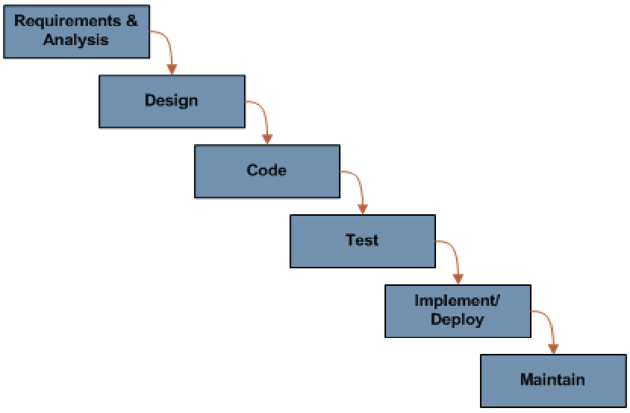


Figure 2: The waterfall model source: [10].

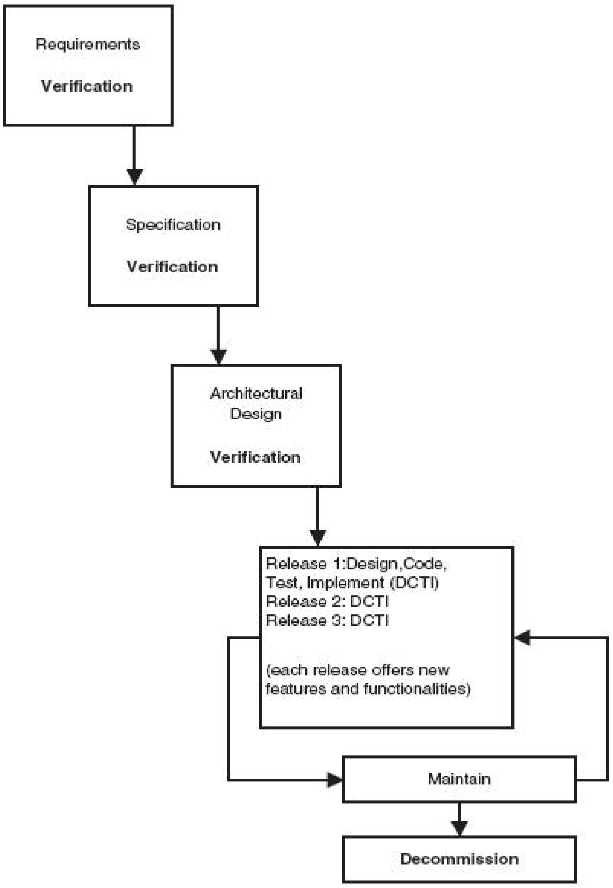


Figure 3: The incremental model source: [10].

- More flexible - less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Easier to manage risk because risky pieces are identified and handled during its iteration.

Pros: Each phase of iteration is rigid and do not overlap each other.

- Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.
- The incremental models can easily degenerate into the build and fix approach.
- Design errors become part of the system and are hard to remove.
- Clients see possibilities and want to change requirements.

The spiral model: The spiral model is similar to the incremental model, with more emphases placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements is gathered and risk is assessed. Each subsequent spiral builds on the baseline spiral [10,11] (Figure 4).

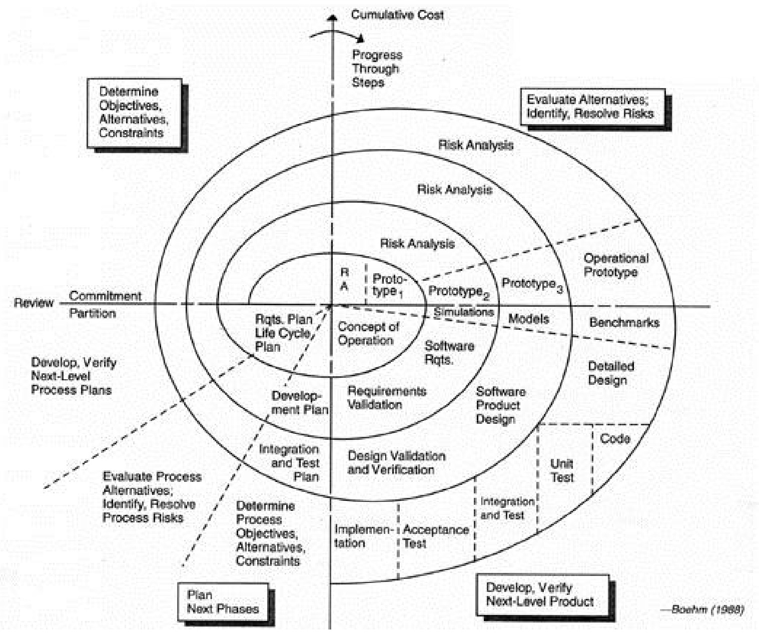


Figure 4: The spiral model source: [11].

Pros: High amount of risk analysis

- Good for large and mission-critical projects.
- Software is produced early in the software life cycle.

Cons: Can be a costly model to use

- Risk analysis requires highly specific expertise.
- Project's success is highly dependent on the risk analysis phase.
- Doesn't work well for smaller projects

Selecting the development life cycle

As the project objectives and requirement are clearly understood, the pure waterfall development life cycle is chosen. The pure waterfall model performs well for products with clearly understood requirements or when working with well understood technical tools, architectures and infrastructures [10]. Using the waterfall model which is easy to understand and implement, the project deliverable and milestones will be easily identified.

Implementation of waterfall model

Before the requirement specification and analysis, the identification of ASP and html web development problems were performed. This was done by asking individuals to air their views concerning the problems via the questionnaires and meeting people whom I knew are in this area of specialisation [9]. *"Getting information about data retrieval and tabulation and Making researches and understanding the source codes/syntax"*

Requirement specification and analysis

This is the first phase in the waterfall software development model. It is in this phase that all the requirements are captured. Analysis of the requirements is carried out to find out the possibility and validity of the requirements, which helps in assessing if the requirements can be incorporated in the system [9]. *"Analyse problems related and involved in the development of the software and the proposed solution"*

Design Phase

In this phase the software to be developed is designed. The specifications of the system are taken into consideration and on the basis of the study of the specification the system design is made. *"Class, begin and the structure of the software is been design"*

Implementation

The software is been developed at this stage. *"The application coding starts at this phase. Writing classes, methods and creating the GUI"*

Testing

This tests helps in identifying the problems created after the entire system has been developed. Two types of testing are conducted at this phase. The system testing and usability testing

Deployment

At this stage, the software application is ready for use. *"Packaging the software and making it available for use"*

Requirement Specification

The main requirement for this project is to essentially provide the end-user with the ability to login or register into the website, purchase articles and view them. The users should be able to go online type the web address in; this will then take the user on to the website. Here the user will be able to click through the different pages. The user will be able to click on to the following pages:

- Home Page
- Article Page
- Magazines Page

- About Chartist Page
- Subscribe Page
- Links Page
- Account Page
- Gallery Page
- Blog Page

On clicking some of these pages will in turn take the user to other hidden pages and buttons such as:

- Check out Page
- Saved Articles Page
- Download Article button

The system should allow the user to flick through the different pages, and then be able to click on any of the articles and view their titles before purchase and view them in full after purchase; finally they should be able to click on the links which are on the Links Page to take the user to external websites.

Design/Coding

Web design is the skill of creating presentations of content (usually hypertext or hypermedia) that are delivered to an end-user through the World Wide Web, using a Web browser or other Web-enabled software while Coding is the process of assigning a code to something for the purposes of classification or identification.

System Design

This section describes how the system works.

The programme was separated, the html converter and the ASP. Net dynamic tabulating of data.

Choice of Language

As I decided to make this application an online desktop application, the programming languages that could be used are java, C# and VBScript in regards to my knowledge on programming languages. Java and C# are the two programming language I have learnt to programme desktop application with. My final decision to use VBScript in the ASP.Net environment was based on my experience on the language. I have been able to work on more than three project on VBScript, unlike java which I just lean at my previous school. Though comparing these languages might seem to me that, no much difference as these programming languages shares a lot of similarities. Comparing data types, reference types, object orientation, methods and properties, statements, expression and operators, exceptions, Arrays and collections, Threading and synchronization, and finally platform support : there are only very little difference. My decision on writing the application in html and VBScript in ASP.Net environment and also doing lots of the web designing with dream weaver would save time and effort on writing and debugging of codes, as I am more experienced and confident in these languages. Also I would be able to spend more time on logical problem rather than code, syntax and bugs.

ASP and SQL Server

ASP is a general-purpose scripting language originally designed for web development to produce dynamic web pages. For this purpose, ASP code is embedded into the HTML source document and interpreted by a web server with an ASP processor module, which generates the web page document. HTML stands for: Hypertext Mark-up Language. It was originally named empty elements. . "The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages". "The browser does not display the HTML tags, but uses the tags to interpret the content of the page". Proprietary products are Microsoft's Active Server Pages, Macromedia's ColdFusion, and Sun's Java Server Pages.ASP is an official module of Apache HTTP Server, the market-leading free Web server that runs about 59.13 percent of the World Wide Web according to the Netcraft Web server survey January 2011. *"Apache remains the dominant force, commanding a 59.13% share and an increase of 10.1M hostnames"* ASP is fully cross-platform; it runs native on several flavours of UNIX, as well as on Windows and on Mac OS X. It is also called the classic scripting language (ASP Classic) due to the fact that "it's the first server side script engine for dynamically generated web pages" used by Microsoft.

SQL is a structured query language used in database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is among the very first commercial languages for Edgar F. Codd's relational model. The SQL phrase which stands for Structured Query Language is used for managing data in different ways. An advantage of using ASP/SQL Server is low cost of development and the fast connection of ASP to SQL. It is also used for easy Implementation and management. Reusability was thoughtfully considered when designing classes and methods. In Table 1, retail figure shows some current retail figures for similar products in the United States".

Due to the fact that this is a group project, I am not in charge of the connecting to database and designing of the whole system area but because this is my final year project and I have been obliged to research and show designs on my software engineering skills as a graduate of software engineering BSc honours have I decided to divert a bit and look into one of my above examples, PHP/MYSQL using them to achieve my goals in designing the whole system for my website (the chartist).

Looking into PHP/MYSQL

PHP dynamic data tabulation

The objective of this section of the programme is, to create my website which will enable me and other PHP Developers to write a programme that retrieves data from the MySQL database and tabulate it which will eventually help end users to select articles from the website and have them uploaded in the saved articles page in a

Table 1: Retail figure.

Item	ASP/SQL Server	ColdFusion MX/SQL Server	JSP/Oracle	PHP/MySQL
Development tool	$0–2499	$599	$0–~2000	$0–249
Server	$999	$2298	$0–~35,000	$0
RDBMS	$4999	$4999	$15,000	$0–220

Source: [12].

tabular form as required (Figure 5).

Design requirements are:

1. Connecting to the MySQL database to retrieving data
2. Tabulating the data
3. A user friendly easy to use GUI

On the creating of new project "PHP dynamic data tabulation", the server folder and setting has to be setup. For the development of this project WAMP server was used.

The connection to the server need to be setup because the application directly communicate to the server (execute query on the server).

- The Folder URL input takes the URL address of the folder on the server.
- The Server Folder sets the folder on the web server which in which the files are to be stored.
- The Test Connection button tests if the server folder and the folder URL correspond.

How the Test Connection was Implemented

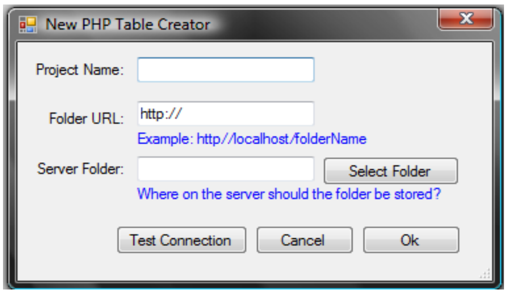
On the click of the Test Connection button, a PHP script file is created on the server folder which has been selected. On the execution of this file, a test successful message is echo. If the Folder URL and Server Folder selected are corresponding and the server is configured correctly, the PHP script file will be executed successfully which returns a message "test successful" else, some errors are thrown which are used to display the type of error occurring to the user. On an attempt to write the Script to the server folder, if the server folder is not found an error message is display to the user. (Figure 6: Flowchart1). The UriFoemat Exception it thrown if the Folder URL could not be located (Figure 7: Flowchart 2). The web exception is thrown if there PHP script could not be executed due to the Folder URL and Server Folder not connecting. (Figure 8: Flowchart 3). The script created is then deleted.

After the Test Connection is successful, a new PHP script file is created and the form is displayed. (Image: Dynamic data tabulation form). The script in the .php file created is a HTML script which has the PHP tag in it (Figure 9: PHP script).

Custom Tags Description

<!-- phpST -->: this indicates the start of the PHP tag

//End**: indicate the end of all PHP scripts



New PHP Table Creator

Project Name:

Folder URL: http://

Example: http//localhost/folderName

Server Folder:

Where on the server should the folder be stored?

Figure 5: Setting up the server folder.

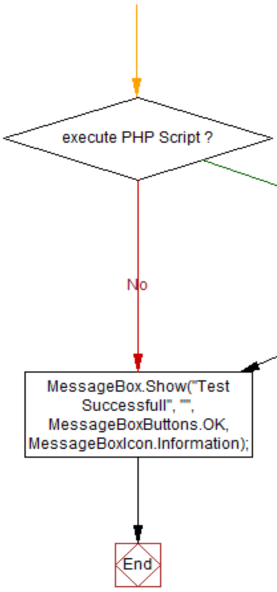

execute PHP Script ?
Yes
No
MessageBox.Show("File not found: "" + sFolderNameTB.Text + "", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
MessageBox.Show("Test Successfull", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
End

Figure 6: Flow Chart 1.

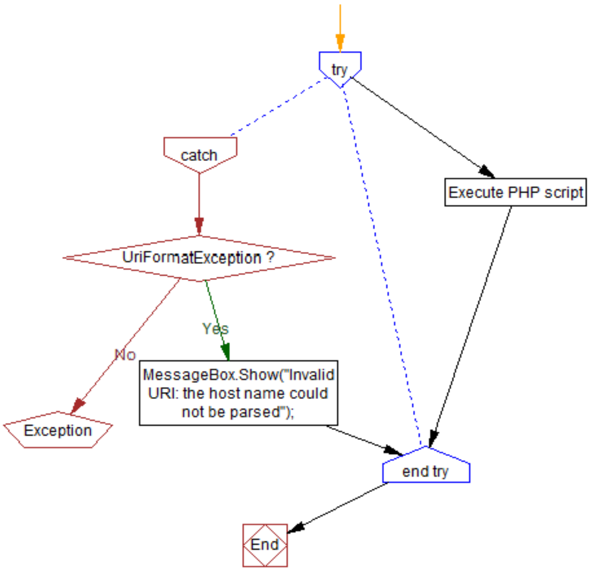


Figure 7: Flow Chart 2.

<!-- phpET -->: indicate the end of the PHP tag

These customs tags are used in the design of the script to indicate some vital lines on the script (Figure 10).

Creating a Table

A table of once desire design need to be created for the tabulation of the data retrieved from the database.

The create table button is used to create a table. On the click of the button a dialog box "Create table" is displayed, which accept inputs to create a table. Inputs accepted are: Number of columns, numbers of rows, width, height, Border, BG Colour, horizontal align, vertical align, cell spacing, cell padding, border colour, and Font Style (Figure 11).

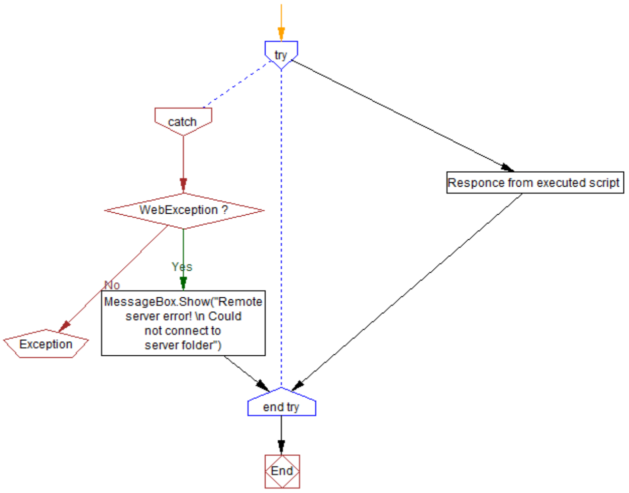

try
catch
WebException ?
Yes
No
MessageBox.Show("Remote server error! \n Could not connect to server folder")
Exception
Responce from executed script
end try
End

Figure 8: Flow Chart 3.

```
<html>
<head>
<meta />
<title> </title>
</head>
<body>
<!-- phpST --> <?php

//End**

?> <!-- phpET -->

</body>

</html>
```

Figure 9: PHP script.

Editing a Table

A table created can as well be edited. Editing shows the same form as creating a new table except that it functions is different.

On editing the table the script is read from the file, edited and written back to the file. If the Table to be edited contains controls (Images, Tests), the controls are first copied to from the table. Otherwise the table is deleted and recreated to the file (Figure 12).

Deleting a Table

All tables created have a unique name which is used for identification for the purpose of editing, deleting or insert of controls. Example of a figure script (Figure 13).

//Table0 [row='3' column='2']: this is a table unique id (Table start tag). (Table0: indicates that this is the first table.

Row='3': indicates that the table has three rows.

Column='2': indicates the table has two rows)

//EndTable: this indicates the end of the table script.

#Controls [img='0' lab='0']: indicate the controls in a table. (img='0': indicates the number of image controls (this case it is zero).

Lab='0': indicate the number of label controls)

While deleting a table, the .PHP file is read to an array list.

The array is searched for the table start tag to be deleted.

The Table is been deleted from the table start tag (//Table0 [row='3' column='2']) to the table end tag (//EndTable) (Figure 14).

After the table has been deleted from the array, it is then written back to the file (It overwrite the file content)

After the table has been deleted from the array, it is then written back to the file (It overwrite the file content)

Insert Controls

There are basically two types of controls that can be added to a table

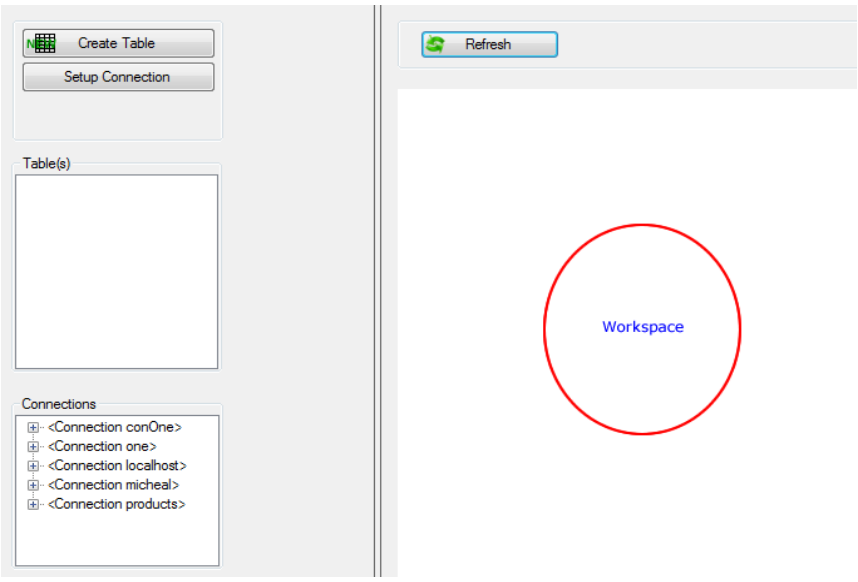
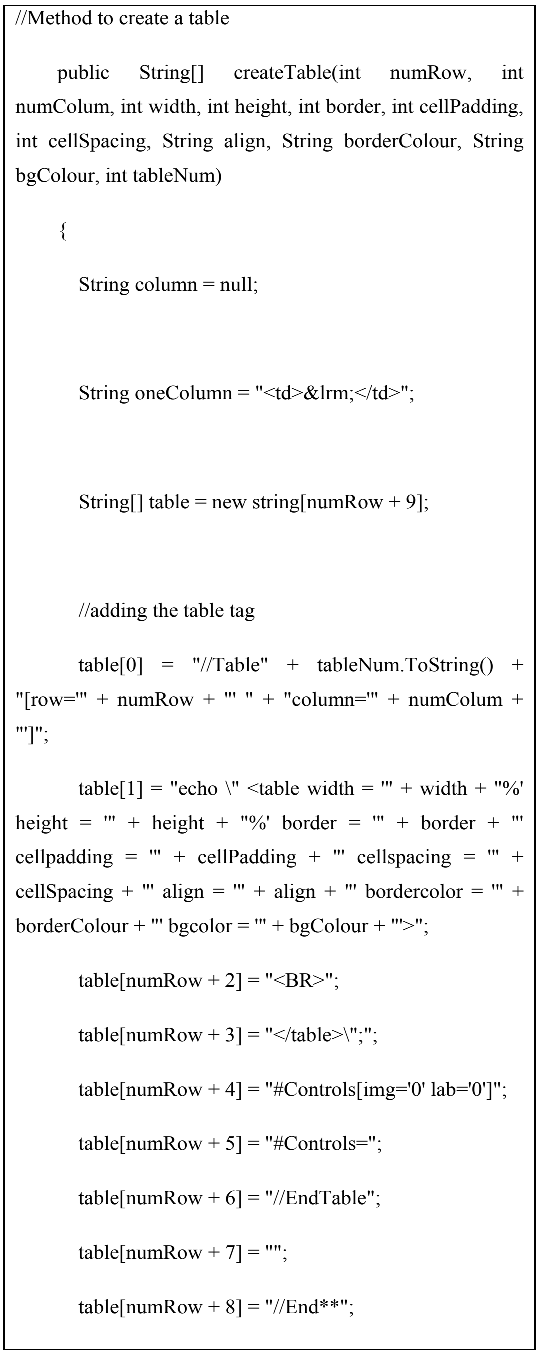


Figure 10: Dynamic data tabulation form.



```
//Method to create a table

    public String[] createTable(int numRow, int numColum, int width, int height, int border, int cellPadding, int cellSpacing, String align, String borderColour, String bgColour, int tableNum)

    {

        String column = null;


        String oneColumn = "<td>&lrm;</td>";


        String[] table = new string[numRow + 9];


        //adding the table tag

        table[0] = "//Table" + tableNum.ToString() + "[row='" + numRow + "' " + "column='" + numColum + "']";

        table[1] = "echo \" <table width = '" + width + "%' height = '" + height + "%' border = '" + border + "' cellpadding = '" + cellPadding + "' cellspacing = '" + cellSpacing + "' align = '" + align + "' bordercolor = '" + borderColour + "' bgcolor = '" + bgColour + "'>";

        table[numRow + 2] = "<BR>";

        table[numRow + 3] = "</table>\";";

        table[numRow + 4] = "#Controls[img='0' lab='0']";

        table[numRow + 5] = "#Controls=";

        table[numRow + 6] = "//EndTable";

        table[numRow + 7] = "";

        table[numRow + 8] = "//End**";
```

Figure 11: Method to create table.

1. Image Control
2. Label Control

Image Control: this adds an image to the table. The image could be just an image or an image button (an image with hyperlink)

Label Control: this adds a text on the table. The label could be an ordinary label or could be a label with a link (Hyperlink)

Controls can be inserted directly (static) or dynamically (from a database).

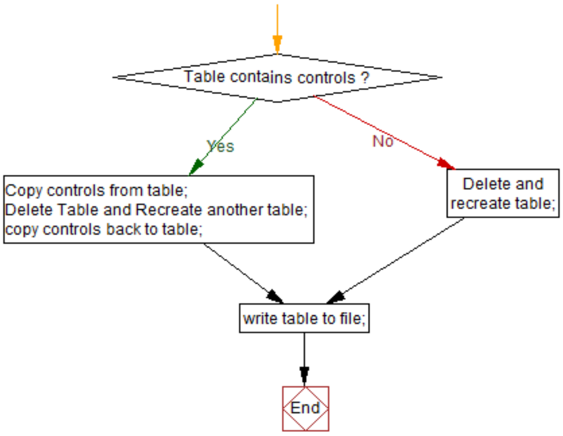
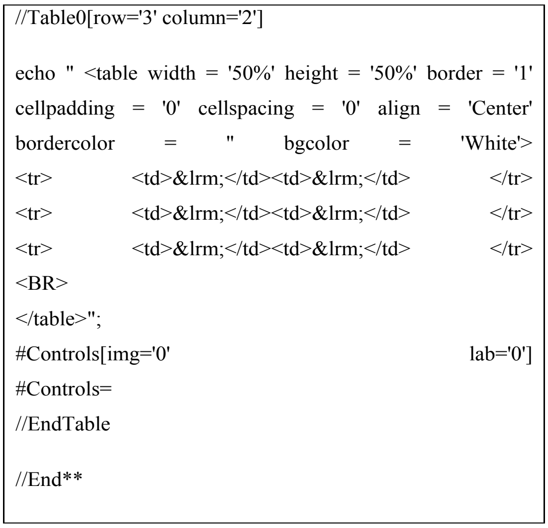


Table contains controls ?
Yes
No
Copy controls from table;
Delete Table and Recreate another table;
copy controls back to table;
Delete and
recreate table;
write table to file;
End

Figure 12: Edit table flowchart.



```
//Table0[row='3' column='2']

echo " <table width = '50%' height = '50%' border = '1' cellpadding = '0' cellspacing = '0' align = 'Center' bordercolor = '' bgcolor = 'White'>
<tr>        <td>&lrm;</td><td>&lrm;</td>        </tr>
<tr>        <td>&lrm;</td><td>&lrm;</td>        </tr>
<tr>        <td>&lrm;</td><td>&lrm;</td>        </tr>
<BR>
</table>";
#Controls[img='0'                    lab='0']
#Controls=
//EndTable

//End**
```

Figure 13: Table script.

Controls can be directly inserted into the table by specifying it path (in the case of an image) or, directly insert text (if is a label control).

Deleting a Control

A control can as well be deleted.

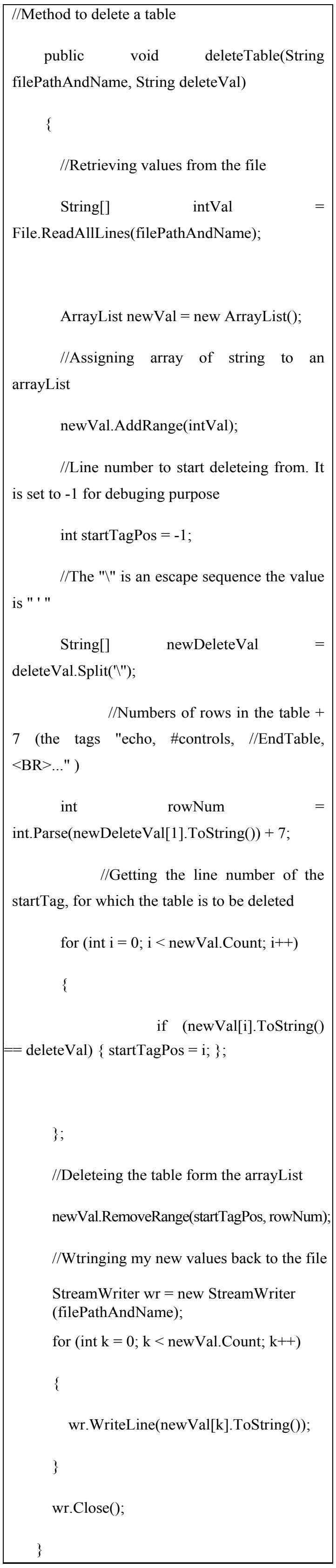
For a control to be deleted, it reads the script from the file to an array list. Then the control is deleted from the Array list and then written back to the file (Figure 15).

Setting up a Connection

For retrieval of data from the Database, a connection to the database has to be setup. This application has been written to make an easy connection to the MySQL Database.

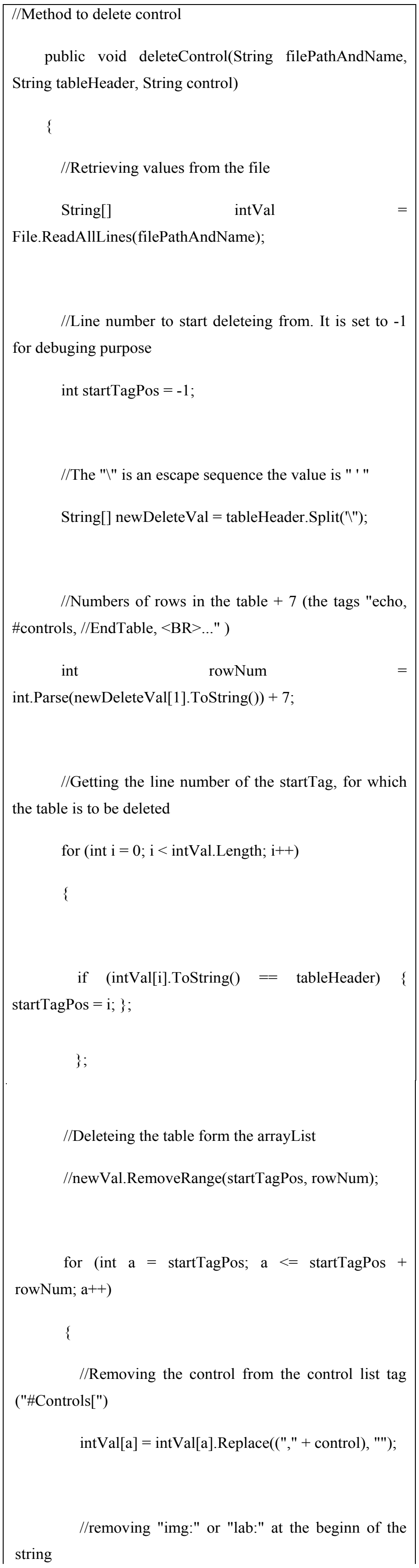
A Class called "Database connection" has been designed to make it easy for a connection to the database. This class does basically four tasks, which are.

doi: 10.4172/2324-9307.1000185



```
//Method to delete a table

    public void deleteTable(String
filePathAndName, String deleteVal)

        {

          //Retrieving values from the file

          String[] intVal =
File.ReadAllLines(filePathAndName);



          ArrayList newVal = new ArrayList();

          //Assigning array of string to an
arrayList

          newVal.AddRange(intVal);

          //Line number to start deleteing from. It
is set to -1 for debuging purpose

          int startTagPos = -1;

          //The "\" is an escape sequence the value
is " ' "

          String[] newDeleteVal =
deleteVal.Split('\'');

                    //Numbers of rows in the table +
7 (the tags "echo, #controls, //EndTable,
<BR>..." )

          int rowNum =
int.Parse(newDeleteVal[1].ToString()) + 7;

                   //Getting the line number of the
startTag, for which the table is to be deleted

          for (int i = 0; i < newVal.Count; i++)

          {

                               if (newVal[i].ToString()
== deleteVal) { startTagPos = i; };



         };

         //Deleteing the table form the arrayList

         newVal.RemoveRange(startTagPos, rowNum);

         //Wtringing my new values back to the file

         StreamWriter wr = new StreamWriter
         (filePathAndName);

         for (int k = 0; k < newVal.Count; k++)

         {

            wr.WriteLine(newVal[k].ToString());

         }

         wr.Close();

      }
```

Figure 14: Method to delete table.



```
//Method to delete control

      public void deleteControl(String filePathAndName,
String tableHeader, String control)

        {

          //Retrieving values from the file

          String[] intVal =
File.ReadAllLines(filePathAndName);



          //Line number to start deleteing from. It is set to -1
for debuging purpose

          int startTagPos = -1;



          //The "\" is an escape sequence the value is " ' "

          String[] newDeleteVal = tableHeader.Split('\'');



          //Numbers of rows in the table + 7 (the tags "echo,
#controls, //EndTable, <BR>..." )

          int rowNum =
int.Parse(newDeleteVal[1].ToString()) + 7;



          //Getting the line number of the startTag, for which
the table is to be deleted

          for (int i = 0; i < intVal.Length; i++)

          {



            if (intVal[i].ToString() == tableHeader) {
startTagPos = i; };


             };



          //Deleteing the table form the arrayList

          //newVal.RemoveRange(startTagPos, rowNum);



          for (int a = startTagPos; a <= startTagPos +
rowNum; a++)

          {

             //Removing the control from the control list tag
("#Controls[")

             intVal[a] = intVal[a].Replace(("," + control), "");



             //removing "img:" or "lab:" at the beginn of the
string
```

```
            //removing "img:" or "lab:" at the beginn of the
string
            intVal[a] = intVal[a].Replace(control.Remove(0,
4), "");


            if (intVal[a].ToString().StartsWith("#Controls["))
            {
              String[] con = intVal[a].Split('\'');


              if (control.StartsWith("img"))
              {   intVal[a]   =   "#Controls[img='"   +
(int.Parse(con[1]) - 1) + "' lab='" + con[3] + "']"; };
//increase the control number
              if (control.StartsWith( "lab"))
              { intVal[a] = "#Controls[img='" + con[1] + "'
lab='" + (int.Parse(con[3]) - 1) + "']"; }; //increase the
control number
            }
          };


          //Wtringing my new values back to the file
          StreamWriter       wr       =       new
StreamWriter(filePathAndName);
          for (int k = 0; k < intVal.Length; k++)
          {
            wr.WriteLine(intVal[k].ToString());
          }


          wr.Close();
        }
```

Figure 15: Method to delete control.

1. Accept the database server details from the user (username, password…)

2. Test the database connection and connect to the database

3. Retrieves the database information (Tables, columns ...)

4. Provides the information required for a connection to the database, to be set up

After the form has been field as required, the connect button can be clicked to connect to the database. On the successful connection to the database, a connection is created and saved to a file named "setting" in the "my Document" folder.

Testing Database Connection and Connecting to the Database (Implementation)

To connect to a database with the information provided by the user in the Database Connection form (Figure 16), a test has to be made to make sure the details provided are correct. A new PHP file is created on the server folder. In this file a script is written in it, to connect to the database server and also get the list of database on the server. (Script: Test Database, Connect to Database) (Figures 17 and 18).

Retrieve Database Information (Implementation)

Database details such as: The entire database present on the server, all tables present on a database and columns present in a table are displayed. With these details, a user can select the fields on the database where data is to be retrieved to the Dynamic table. It also enhances the implementation of the connection made to the database for the retrieval of data to the Dynamic table. Database information is retrieved by executing a query, which gets list of database, tables, and columns from the specified database server. After this information has been retrieved, it is stored in a file called "setting.txt". By storing the data in the setting.txt file, it makes the data available for future use and creating a connection to the database for the purpose of data retrieval (Figure 19-21).

SETTING.TXT File

The setting.txt file is created in a directory "php_perl_enhancer" of "my document" directory on any machine the application is installed on (Figure 22).

The "setting.txt" file stores all the connection that has been made on the application. (Example of a setting.txt file)

Example of a setting.txt file: this file has two setting in it. "Connection jennifer" and "Connection protesting"

<Connection protesting>: A start tag of a connection and also states the name of the setting "protesting"

Server=: server name of the database

User=: database user name

password=: database password

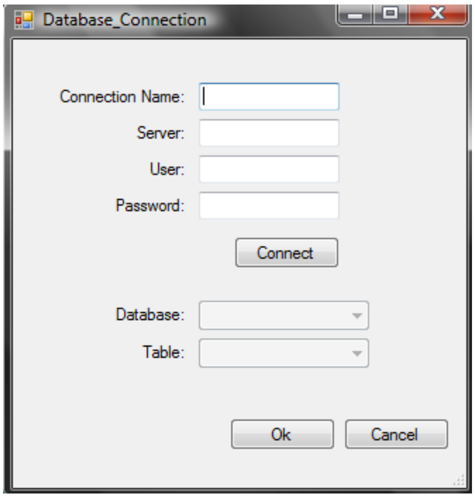
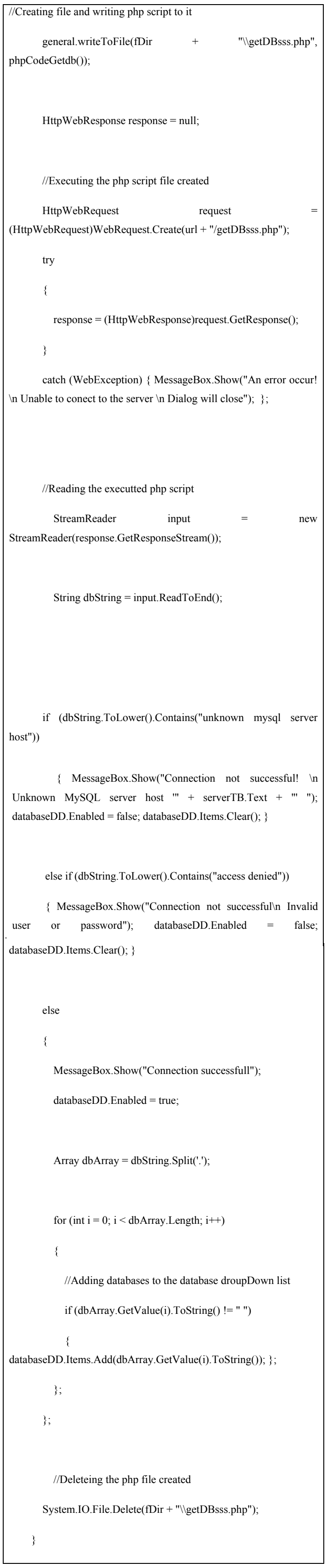


Figure 16: Data connection form.



```
//Creating file and writing php script to it

        general.writeToFile(fDir + "\\getDBsss.php",
phpCodeGetdb());


        HttpWebResponse response = null;


        //Executing the php script file created

        HttpWebRequest request =
(HttpWebRequest)WebRequest.Create(url + "/getDBsss.php");

        try

        {

           response = (HttpWebResponse)request.GetResponse();

        }

        catch (WebException) { MessageBox.Show("An error occur!
\n Unable to conect to the server \n Dialog will close"); };



        //Reading the executted php script

           StreamReader input = new
StreamReader(response.GetResponseStream());


           String dbString = input.ReadToEnd();




        if (dbString.ToLower().Contains("unknown mysql server
host"))

            { MessageBox.Show("Connection not successful! \n
 Unknown MySQL server host '" + serverTB.Text + '" ");
 databaseDD.Enabled = false; databaseDD.Items.Clear(); }


         else if (dbString.ToLower().Contains("access denied"))

         { MessageBox.Show("Connection not successful\n Invalid
 user or password"); databaseDD.Enabled = false;
databaseDD.Items.Clear(); }


        else

        {

           MessageBox.Show("Connection successfull");

           databaseDD.Enabled = true;


           Array dbArray = dbString.Split('.');


           for (int i = 0; i < dbArray.Length; i++)

           {

              //Adding databases to the database droupDown list

              if (dbArray.GetValue(i).ToString() != " ")

              {
databaseDD.Items.Add(dbArray.GetValue(i).ToString()); };

           };

        };


           //Deleteing the php file created

        System.IO.File.Delete(fDir + "\\getDBsss.php");

     }
```

Figure 17: Test database, connect database script.

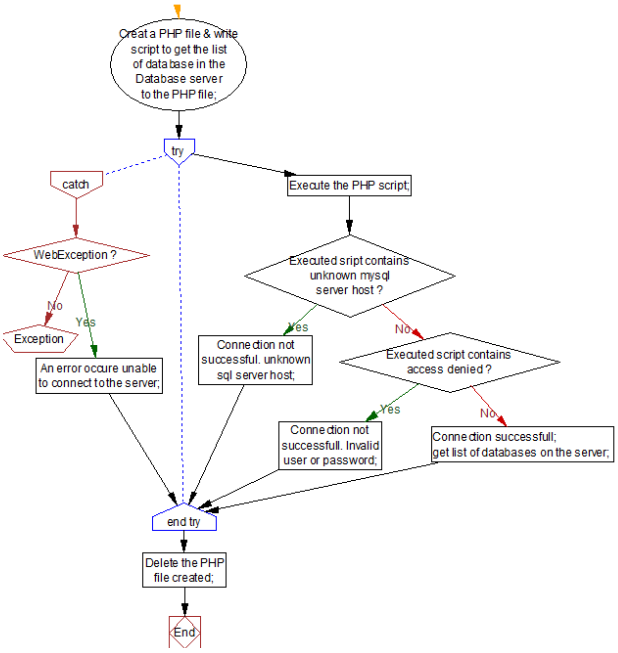

Creat a PHP file & write script to get the list of database in the Database server to the PHP file;
try
catch
Execute the PHP script;
WebException ?
No
Yes
Exception
An error occure unable to connect to the server;
Executed sript contains unknown mysql server host ?
Yes
No
Connection not successful. unknown sql server host;
Executed script contains access denied ?
Yes
No
Connection not successfull. Invalid user or password;
Connection successfull; get list of databases on the server;
end try
Delete the PHP file created;
End

Figure 18: Test, connect database flow chart.

Table=: table selected on the database

Columns=: columns on the table

</Connection>: the end of a connection

Custom Tags

This are some tags used in the system for identification of some significant point / line in the generated file. The tags are described as follows:

System Requirement

A web server, to run the server script (php). Mysql server. For the PHP Dynamic Data Tabulation to work, I web server has to be provided. The application has to set up a connection to a web server before the work space will be displayed [12,13].

For the retrieval of data from a database, a connection to MySQL server needs to be set up on the application.

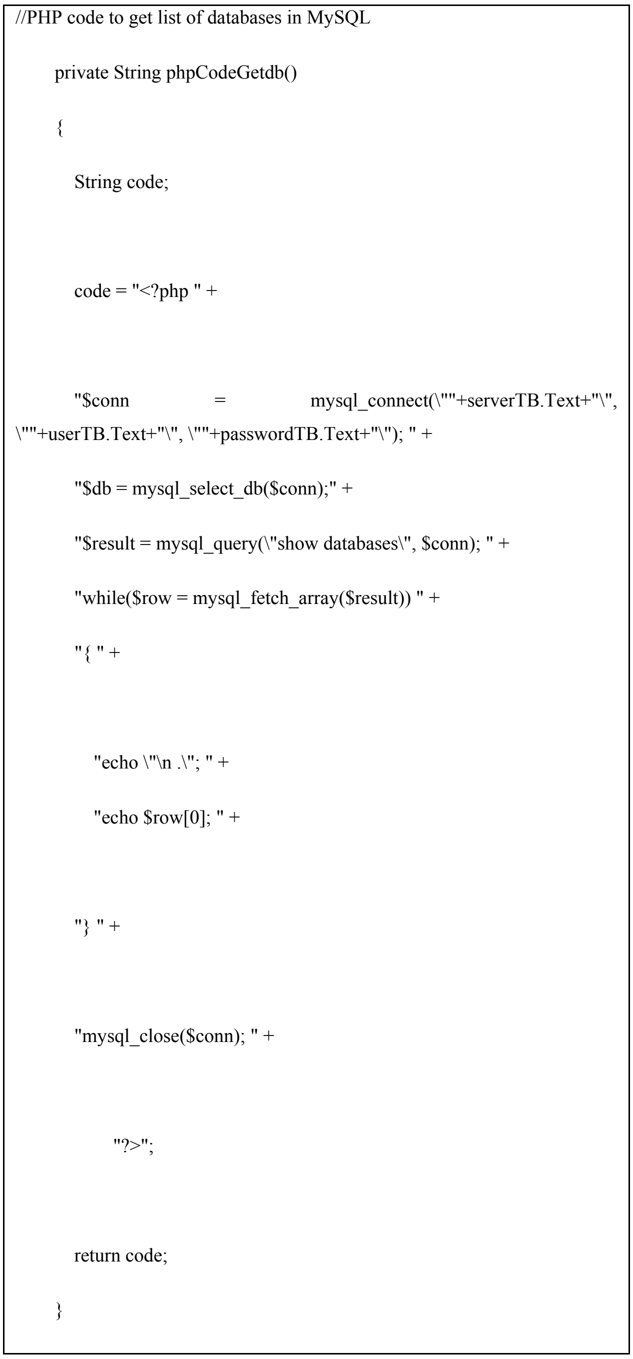
To run the programme the followings are required:

1. Microsoft Window OS (Tested on XP, Vista and Win7)
2. A web server
3. MySQL server

Checkout Page

Firstly I designed two check out pages which is an online processing service aimed at simplifying the process of paying for online purchases is used to notify the user whether or not their purchase has been successful and helping them in confirming their order that has been purchased on the account page. I designed these pages using dream weaver (Figure 23).

This check out page notifies the end-user that their order has been sent to the address in which the user provided in the address

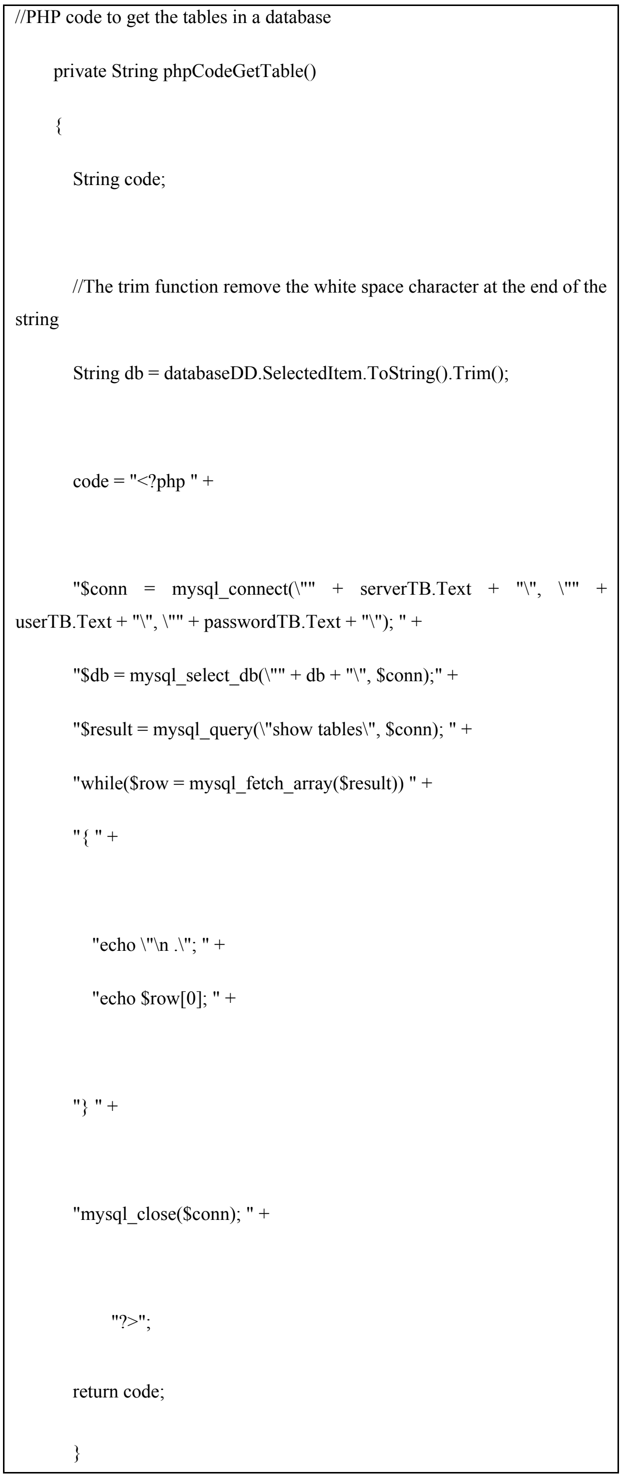


```
//PHP code to get list of databases in MySQL

    private String phpCodeGetdb()

    {

        String code;


        code = "<?php " +


        "$conn = mysql_connect(\""+serverTB.Text+"\", \""+userTB.Text+"\", \""+passwordTB.Text+"\"); " +

        "$db = mysql_select_db($conn);" +

        "$result = mysql_query(\"show databases\", $conn); " +

        "while($row = mysql_fetch_array($result)) " +

        "{ " +


            "echo \"\n .\"; " +

            "echo $row[0]; " +


        "} " +


        "mysql_close($conn); " +


                "?>";


        return code;

    }
```

Figure 19: Method to list database on MYSQL.

section and also charges the user for the purchase through the credit card information provided by the user in the credit card information section. Here is the code for the checkout page:

```
<%
‘ Retrieve Registration Information
sqlString = “SELECT * FROM users “ &_
“WHERE user_id=” & userID
SET RS = Con.Execute( sqlString )
```

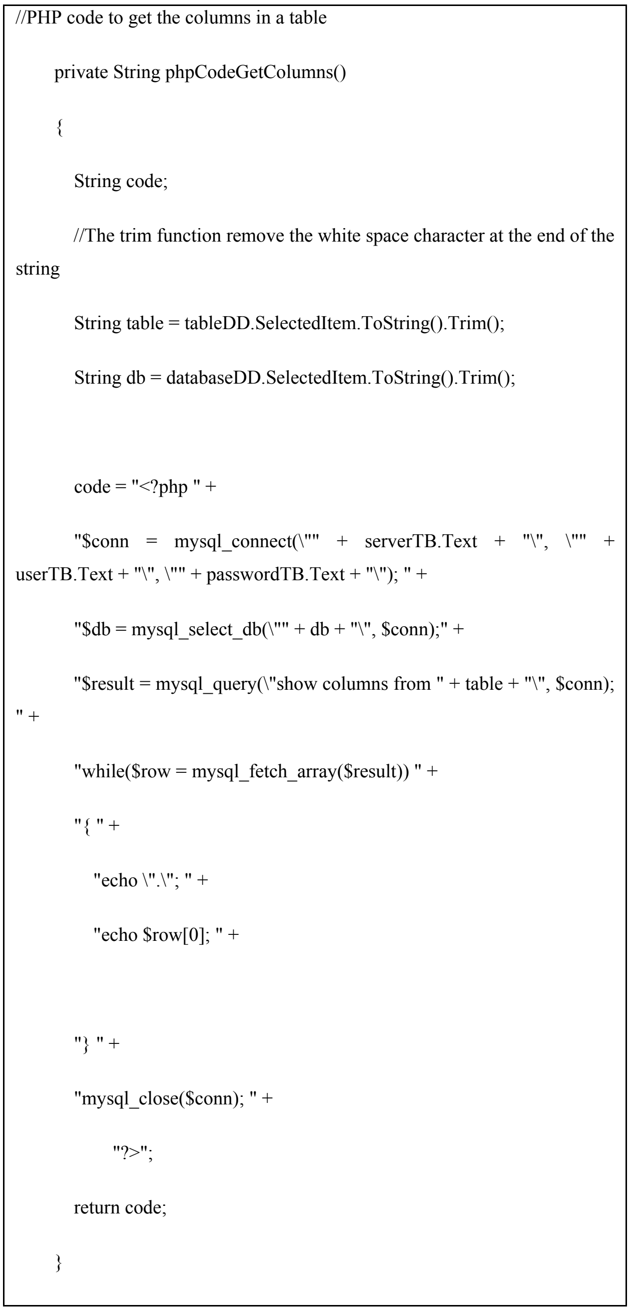


```
//PHP code to get the tables in a database

    private String phpCodeGetTable()

    {

        String code;


        //The trim function remove the white space character at the end of the string

        String db = databaseDD.SelectedItem.ToString().Trim();


        code = "<?php " +


        "$conn = mysql_connect(\"" + serverTB.Text + "\", \"" + userTB.Text + "\", \"" + passwordTB.Text + "\"); " +

        "$db = mysql_select_db(\"" + db + "\", $conn);" +

        "$result = mysql_query(\"show tables\", $conn); " +

        "while($row = mysql_fetch_array($result)) " +

        "{ " +


            "echo \"\n .\"; " +

            "echo $row[0]; " +


        "} " +


        "mysql_close($conn); " +


                "?>";


        return code;

        }
```

Figure 20: Methods to get list of tables in a database.

```
IF NOT RS.EOF THEN
street = RS( “user_street” )
street2 = RS( “user_street2” )
street3 = RS( “user_street3” )
city = RS( “user_city” )
```

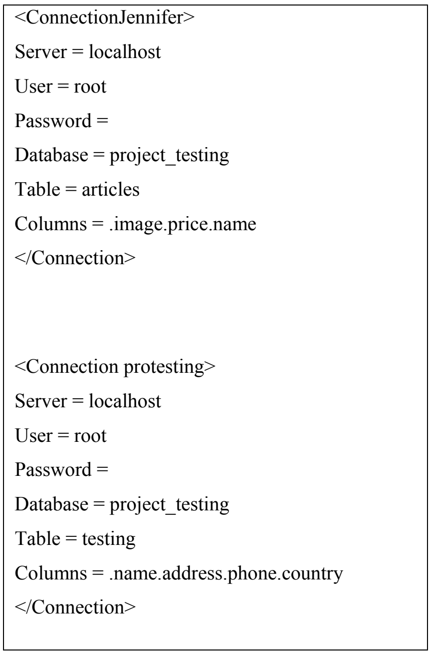
doi: 10.4172/2324-9307.1000185



```
//PHP code to get the columns in a table

      private String phpCodeGetColumns()

      {

         String code;

         //The trim function remove the white space character at the end of the
string

         String table = tableDD.SelectedItem.ToString().Trim();

         String db = databaseDD.SelectedItem.ToString().Trim();



         code = "<?php " +

         "$conn = mysql_connect(\"" + serverTB.Text + "\", \"" +
userTB.Text + "\", \"" + passwordTB.Text + "\"); " +

         "$db = mysql_select_db(\"" + db + "\", $conn);" +

         "$result = mysql_query(\"show columns from " + table + "\", $conn);
" +

         "while($row = mysql_fetch_array($result)) " +

         "{ " +

            "echo \".\"; " +

            "echo $row[0]; " +



         "} " +

         "mysql_close($conn); " +

              "?>";

         return code;

      }
```

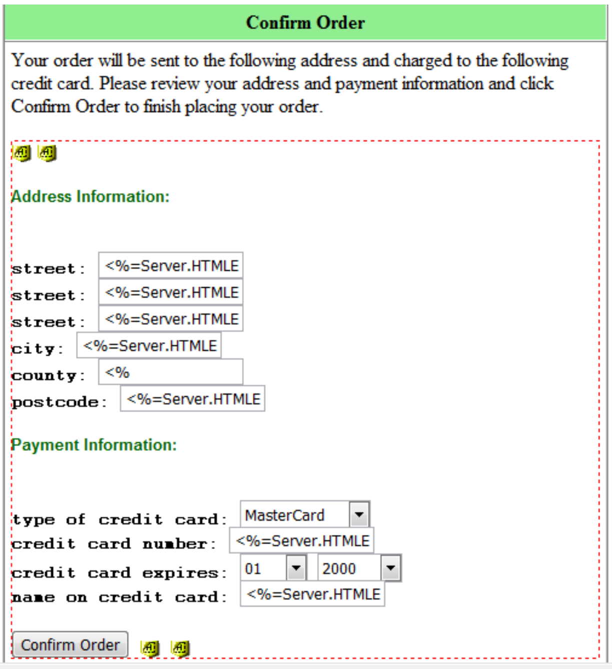
Figure 21: Method to get column on a table.

```
county = RS( "user_county" )
postcode = RS( "user_postcode" )
END IF
%>
<html>
<head><title>Checkout</title></head>
<body>
```



```
<ConnectionJennifer>
Server = localhost
User = root
Password =
Database = project_testing
Table = articles
Columns = .image.price.name
</Connection>



<Connection protesting>
Server = localhost
User = root
Password =
Database = project_testing
Table = testing
Columns = .name.address.phone.country
</Connection>
```

Figure 22: Setting.txt file example.



Confirm Order

Your order will be sent to the following address and charged to the following credit card. Please review your address and payment information and click Confirm Order to finish placing your order.

Address Information:

street : <%=Server.HTMLE
street : <%=Server.HTMLE
street : <%=Server.HTMLE
city : <%=Server.HTMLE
county : <%
postcode : <%=Server.HTMLE

Payment Information:

type of credit card : MasterCard
credit card number : <%=Server.HTMLE
credit card expires : 01 2000
name on credit card : <%=Server.HTMLE

Figure 23: Confirm order.

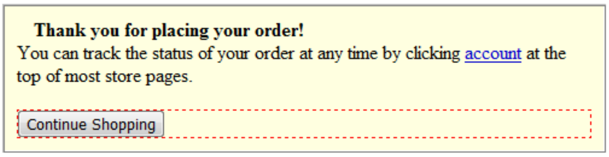
```
<center>
<table border=1 width=500
 cellpadding=5 cellspacing=0>
<tr>
 <td align="center" bgcolor="lightgreen">
 <b>Confirm Order</b>
 </td>
</tr>
<tr>
 <td>
```

Your order will be sent to the following address and charged to the following credit card. Please review your address and payment information and click Confirm Order to finish placing your order.

```
 <form method=”post” action=”checkout2.asp”>
 <input name=”username” type=”hidden”
  value=”<%=username%>”>
 <input name=”password” type=”hidden”
  value=”<%=password%>”>
  <p><font face=”Arial” size=”2” color=”darkgreen”><b>Address Information:</b>
     </font>
    <p><font face=»Courier» size=»2»> <br>
     <b>street:</b>
        <input name=”Char_BillTo_Postal_Street_Line1” size=20 maxlength=50
  value=”<%=Server.HTMLEncode( street )%>”>
     <br>
     <b>street:</b>
        <input name=”Char_BillTo_Postal_Street_Line2” size=20 maxlength=50
<% IF street2 <> “” THEN %>
value=”<%=Server.HTMLEncode( street2 )%>”
 <% END IF %>
  >
     <br>
     <b>street:</b>
        <input name=”Char_BillTo_Postal_Street_Line3” size=20 maxlength=50
 <% IF street3 <> “” THEN %>
value=”<%=Server.HTMLEncode( street3 )%>”
 <% END IF %>
  >
     <br>
     <b>city:</b>
           <input name=”Char_BillTo_Postal_City” size=20 maxlength=50
  value=”<%=Server.HTMLEncode( city )%>”>
     <br>
     <b></b><b>county:</b>
          <input name=”Char_BillTo_Postal_County” size=20 maxlength=2
  value=”<%=Server.HTMLEncode( county )%>”>
     <br>
     <b>postcode:</b>
         <input name=”Char_BillTo_Postal_Postcode” size=20 maxlength=20
  value=»<%=Server.HTMLEncode( postcode )%>»>
     </font>
    <p>
  <font face=”Arial” size=”2” color=”darkgreen”><b>Payment Information:</b>
 </font>
 <font face=»Courier» size=»2»>
 <br><b>type of credit card:</b>
 <select name=”Char_Payment_Card_Type”>
 <option value=”VISA”
  <%=SELECTED( cctype, “VISA” )%> > VISA
 <option value=”MAST”
  <%=SELECTED( cctype, “MAST” )%> >MasterCard
 </select>
 <br><b>credit card number:</b>
     <input name=”Char_Payment_Card_Number” size=20 maxlength=20
  value=”<%=Server.HTMLEncode( ccnumber )%>”>
 <br><b>credit card expires:</b>
 <select name=”Char_Payment_Card_ExpDate_Month”>
 <option value=1 <%=SELECTED( ccexpiremonth, 1)%>>01
 <option value=2 <%=SELECTED( ccexpiremonth, 2)%>>02
 <option value=3 <%=SELECTED( ccexpiremonth, 3)%>>03
 <option value=4 <%=SELECTED( ccexpiremonth, 4)%>>04
 <option value=5 <%=SELECTED( ccexpiremonth, 5)%>>05
 <option value=6 <%=SELECTED( ccexpiremonth, 6)%>>06
 <option value=7 <%=SELECTED( ccexpiremonth, 7)%>>07
 <option value=8 <%=SELECTED( ccexpiremonth, 8)%>>08
 <option value=9 <%=SELECTED( ccexpiremonth, 9)%>>09
 <option value=10 <%=SELECTED( ccexpiremonth, 10)%>>10
 <option value=11 <%=SELECTED( ccexpiremonth, 11)%>>11
 <option value=12 <%=SELECTED( ccexpiremonth, 12)%>>12
 </select>
```

```
<select name="Char_Payment_Card_ExpDate_Year">
    <option value=2000 <%=SELECTED(ccexpiremonth, 2000)%>>2000
    <option value=2001 <%=SELECTED(ccexpiremonth, 2001)%>>2001
    <option value=2002 <%=SELECTED(ccexpiremonth, 2002)%>>2002
    <option value=2003 <%=SELECTED(ccexpiremonth, 2003)%>>2003
    <option value=2004 <%=SELECTED(ccexpiremonth, 2004)%>>2004
    <option value=2005 <%=SELECTED(ccexpiremonth, 2005)%>>2005
</select>
<br><b>name on credit card:</b>
    <input name="Char_Payment_Card_Name" size=20 maxlength=20
 value="<%=Server.HTMLEncode( ccname )%>">
<p><input type="submit" value="Confirm Order">
   <input type="hidden" name="Ecom_SchemaVersion" value="http://www.ecml.org/version/1.0">
<input type="hidden" name="Ecom_TransactionComplete">
</font>
</form>
</tr>
</table>
<p>
</center>
</body>
</html>
```

The second check out page opens up when the purchase is completed it gives a message to the end-user to thank them for placing an order and to inform them that they can track the status of their order at any time by going to the account page. There is a button labelled continue shopping which is telling the end-user to continue shopping if they want to and it will take them back to the Article page (Figure 24).



Thank you for placing your order!
You can track the status of your order at any time by clicking account at the top of most store pages.

Figure 24: Continue shopping button.

Here is the code for the second check out page:

```
<%
' Add new address & payment info
updateUser
' Begin A Transaction
Con.BeginTrans
' Transfer cart to orders table
sqlString = "INSERT INTO orders (" &_
 "order_id, " &_
 "order_productID, " &_
 "order_quantity, " &_
 "order_userID, " &_
 "order_entrydate, " &_
 "order_status " &_
 ") SELECT " &_
 «cart_id, « &_
 «cart_productID, « &_
 «cart_quantity, « &_
 "cart_userID, " &_
 "NOW(), " &_
 "0 " &_
 "FROM cart WHERE " &_
 "cart_userID =" & userID
Con.Execute sqlString
' Empty shopping cart
sqlString = "DELETE FROM cart " &_
 "WHERE cart_userID=" & userID
Con.Execute sqlString
' End the transaction
Con.CommitTrans
%>
<html>
<head><title>Order Placed</title></head>
<body>
<center>
<table border=1 width=500 bgcolor="lightyellow"
 cellpadding=10 cellspacing=0>
<tr>
 <td>
 <font face="Arial" size="3">
 <b>Thank you for placing your order!</b>
 <br>You can track the status of your order
 at any time by clicking
 <a href=”account.asp”>account</a> at the
 top of most store pages.
 <br>
 <form action=”Articlepage.asp”>
 <input type=”submit” value=”Continue Shopping”>
 </form>
 </td>
</tr>
</table>
</center>
</body>
</html>
```

Download Article Button

Next is the Download Article button this button is actually located on the account page it is connected to the Saved Articles Page which is a hidden page. When the user have completed his or her purchase and now want to view all the articles in which he or she has selected then the end- user will click on the Download Article Button which will then take the user to the Saved Articles page and the articles which has been selected by the users will then be displayed on the page [13].

Here is the code for the Download Article button: (Figure 25)

```
<button type=”button” onClick=”location.href(‘Saved Articles.asp’)”>DOWNLOAD ARTICLES</button></br></br>
 <%
sqlString = “SELECT role FROM users WHERE users.user_id=” & Session(“userID”) & “”
        SET RS = Con.Execute( sqlString )

        IF RS.EOF THEN

        ELSE
        WHILE NOT RS.EOF

                IF RS(“role”) = “admin” THEN
                %>
```

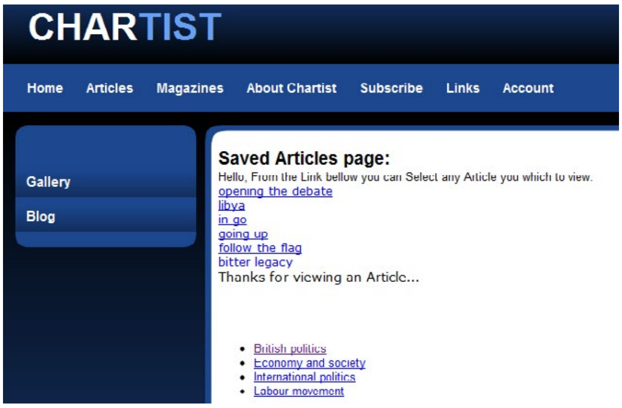
Save Article Page

This is the Saved Articles page, here there is a message telling the end-user that he or she can select any of the article he or she wishes to view one at a time, and another message thanking the End-user for viewing the articles. I saved all the articles in a Microsoft word file so that it can be viewed easily anywhere in the world and I created a link for them each on the webpage.

Figure 25: Download article button.

Here is the code for the Saved Articles page: (Figure 26)

```
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Transitional//EN” “http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd”>
<html xmlns=”http://www.w3.org/1999/xhtml”>
<head>
<meta http-equiv=”Content-Type” content=”text/html; charset=utf-8” />
<link rel=”stylesheet” type=”text/css” href=”style.css” />
<title>SavedArticles</title>
<style type=”text/css”>
<!--
.style2 {font-family: “Times New Roman”, Times, serif}
-->
</style>
</head>

<body>
<!-- #INCLUDE FILE=”adovbs.inc” -->
<%
‘ Get Current Category
cat = TRIM( Request( “cat” ) )
IF cat = “” THEN cat = “Home”

%>

<!-- #INCLUDE FILE=”DBconnection.inc” -->
```



Saved Articles page:

Hello, From the Link bellow you can Select any Article you which to view.
opening the debate
libya
in go
going up
follow the flag
bitter legacy
Thanks for viewing an Article...

- British politics
- Economy and society
- International politics
- Labour movement

Figure 26: Saved article page.

```
   <html>
   <div id=”container”>
                 <div id=”header”>
       <h1>CHAR<span class=”off”>TIST</span></h1>
     <h2>For Democratic Socialism</h2>
                     <h2>For Democratic Socialism <br> Welcome <%Response.Write(Session(“username”))%></h2>
     </div>

     <div id=”menu”>
       <ul>
       <li class=”menuitem”><a href=”default.asp”>Home</a></li>
      <li class=”menuitem”><a href=”Articlepage.asp”>Articles</a></li>
       <li class=”menuitem”><a href=”Magazines.asp”>Magazines</a></li>
      <li class=”menuitem”><a href=”About Chartist.asp”>About Chartist</a></li>
      <li class=”menuitem”><a href=”Subscribe.asp”>Subscribe</a></li>
      <li class=”menuitem”><a href=”Links.asp”>Links</a></li>
        <li class=”menuitem”><a href=”account.asp”>Account</a></li>
      <li class=”menuitem”></li>
     </ul>
     </div>

     <div id=”leftmenu”>
     <div id=”leftmenu_top”></div>

       <div id=”leftmenu_main”>

      <h3>&nbsp;</h3>

                           <ul>
       <li><a href=”Gallery.asp”>Gallery</a></li>
       <li><a href=”Blog.asp”>Blog</a></li>
      </ul>
   </div>

      <div id=”leftmenu_bottom”></div>
```

```
   </div>



   <div id=”content”>



   <div id=”content_top”></div>
   <div id=”content_main”>
     <h2>Saved Articles page:</h2>
     <p>Hello, From the Link bellow you can Select any Article you which to view. </p>
     <p><font face=”Verdana” size=”2” color=”#000000”><a href=”opening the debate.docx”></a><a href=”opening the debate.docx”></a><a href=”opening the debate.docx”></a><a href=”opening the debate.docx”>opening the debate</a></font></p>
     <p><font face=”Verdana” size=”2” color=”#000000”><a href=”libya.docx”>libya</a></font></p>
     <p><font face=”Verdana” size=”2” color=”#000000”><a href=”in go.docx”>in go</a></font></p>
     <p><font face=”Verdana” size=”2” color=”#000000”><a href=”going up.docx”>going up</a></font></p>
     <p><font face=”Verdana” size=”2” color=”#000000”><a href=”follow the flag.docx”>follow the flag</a></font></p>
     <p><font face=”Verdana” size=”2” color=”#000000”><a href=”bitter legacy.docx”>bitter legacy</a><br />
    </font> </p>
     <p class=”style2”><font face=”Verdana” size=”2” color=”#000000”>
    <label style=”font-size:15px;”>Thanks for viewing an Article...<br />
    </label>
    </font><font face=»Verdana» size=»2» color=»#000000»><br />
    </font> </p>
     <font face=»Verdana» size=»2» color=»#000000»>
    <label style=»font-size:15px;»></label>
    </font>
     <p>&nbsp;</p>
     <p>&nbsp;</p>
                <p>&nbsp;</p>
                <p><b>
    <% If cat = “Home” and searchfor=”” THEN %>
    <% ELSE %>
```

doi: 10.4172/2324-9307.1000185

```
   </b></p>
                    <p>&nbsp;</p>
                    <p><b>
   <% END IF %>
   </b></p>


   <form name="jump">
   <p>
    <!-- #INCLUDE FILE="select.asp" -->
   </p>
   <p></p>
   </form>
    <p>&nbsp;</p>
              <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>


     </div>
<div id="content_bottom"></div>


      <p>
       <% IF cat="Home" THEN %>
```

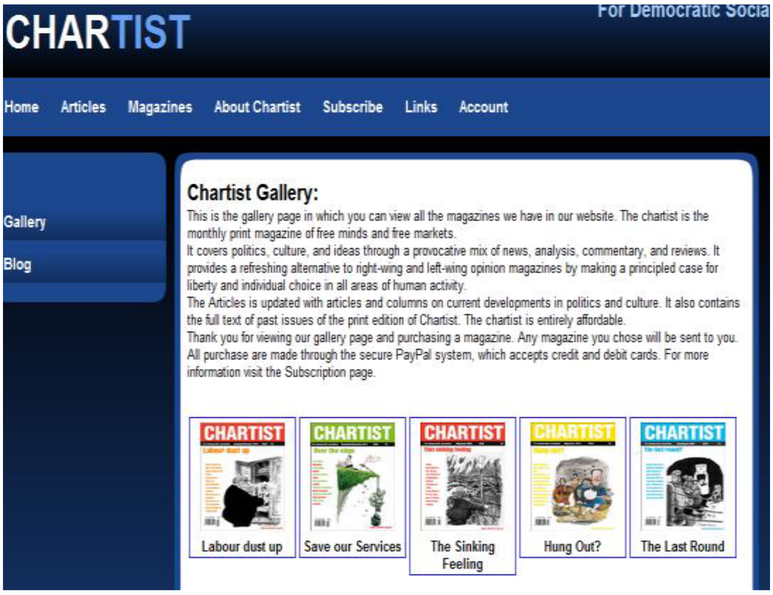
```
  <span style="text-align: center"></span></p>
  <p>
  <% ELSE %>
  </p>
     <!-- #INCLUDE FILE="MPProductList.asp" -->
<% END IF %>


     <h3>Copyright 2012</a></h3>




</div>


</body>
</html>
```

Gallery Page

Finally the last page I created is the gallery page unlike the other pages the gallery page is not a hidden page it can be viewed by anybody who visits the site after successful login or registration. It comprises of online magazines which are the magazine version of the articles in the website. The difference between the article and the magazine is that the magazines cannot be downloaded they can only be sent by post after you make an order. In the gallery page you can click on each magazine template which are links to the full image of each magazine covers respectively. Below is a view of the gallery page [13].

Here are the source files of the gallery page: (Figure 27)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-



Chartist Gallery:

This is the gallery page in which you can view all the magazines we have in our website. The chartist is the monthly print magazine of free minds and free markets.
It covers politics, culture, and ideas through a provocative mix of news, analysis, commentary, and reviews. It provides a refreshing alternative to right-wing and left-wing opinion magazines by making a principled case for liberty and individual choice in all areas of human activity.
The Articles is updated with articles and columns on current developments in politics and culture. It also contains the full text of past issues of the print edition of Chartist. The chartist is entirely affordable.
Thank you for viewing our gallery page and purchasing a magazine. Any magazine you chose will be sent to you. All purchase are made through the secure PayPal system, which accepts credit and debit cards. For more information visit the Subscription page.

Figure 27: Gallery page.

```
transitional.dtd">
   <html xmlns="http://www.w3.org/1999/xhtml">
   <head>
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
   <link rel="stylesheet" type="text/css" href="style.css" />
   <title>Gallery</title>
   <style type="text/css">
   <!--
   #Layer1 {
        position:absolute;
        left:5px;
        top:724px;
        width:1081px;
        height:23px;
        z-index:1;
   }
   -->
   </style>
   </head>

   <body>
   <div id="container">
   <div id="header">
        <h1>CHAR<span class="off">TIST</span></h1>
      <h2>For Democratic Socialism</h2>
   </div>

     <div id="menu">
        <ul>
        <li class="menuitem"><a href="default.asp">Home</a></li>
       <li class="menuitem"><a href="Articlepage.asp">Articles</a></li>
        <li class="menuitem"><a href="Magazines.asp">Magazines</a></li>
       <li class="menuitem"><a href="About Chartist.asp">About Chartist</a></li>
       <li class="menuitem"><a href="Subscribe.asp">Subscribe</a></li>
       <li class="menuitem"><a href="Links.asp">Links</a></li>
         <li class="menuitem"><a href="account.asp">Account</a></li>
        <li class="menuitem"></li>
       </ul>
      </div>

      <div id="leftmenu">

      <div id="leftmenu_top"></div>

                                       <div id="leftmenu_main">

        <h3>&nbsp;</h3>

                                       <ul>
         <li><a href="Gallery.asp">Gallery</a></li>
         <li><a href="Blog.asp">Blog</a></li>
         <li></li>
        </ul>
    </div>

        <div id="leftmenu_bottom"></div>
      </div>

                       <div id="content">

      <div id="content_top"></div>
      <div id="content_main">
           <h2>Chartist Gallery:</h2>
    <p> This is the gallery page in which you can view all the magazines we have in our website. The chartist is the monthly print magazine of free minds and free markets. <br />
    It covers politics, culture, and ideas through a provocative mix of news, analysis, commentary, and reviews. It provides a refreshing alternative to right-wing and left-wing opinion magazines by making a principled case for liberty and individual choice in all areas of human activity. </p>
    <p>The Articles is updated with articles and columns on current developments in politics and culture. It also contains the full text of past issues of the print edition of Chartist. The chartist is entirely affordable.</p>
    <p> Thank you for viewing our gallery page and purchasing a magazine. Any magazine you chose will be sent to you. All purchase is made through the secure PayPal system, which accepts credit and
```

```
debit cards. For more information visit the Subscription page. </p>
    <p>&nbsp;</p>
     <p>&nbsp; </p>
     <p>
      <style type=”text/css”>
    div.img
     {
     margin:2px;
     border:1px solid #0000ff;
     height:auto;
     width:auto;
     float:left;
     text-align:center;
     }
    div.img img
     {
     display:inline;
     margin:3px;
     border:1px solid #ffffff;
     }
    div.img a:hover img
     {
     border:1px solid #0000ff;
     }
    div.desc
     {
     text-align:center;
     font-weight:normal;
     width:120px;
     margin:2px;
     }
        </style>
     </p>
     <div class=”img”>
      <a href=”Labour dust up.gif”><img src=”Labour dust up.gif” alt=”view” name=”Labourdustup” width=”100” height=”100” border=”0” align=”middle” lowsrc=”Labour dust up.gif” /></a>
     <div class=”desc”><strong>Labour dust up </strong></div>
    </div>
    <div class=”img”>
      <a href=”Over the Edge.gif”><img src=”Over the Edge.gif” alt=”view” name=”OvertheEdge” width=”100” height=”100” border=”0” align=”middle” lowsrc=”Over the Edge.gif” /></a>
     <div class=”desc”><strong>Save our Services </strong></div>
    </div>
     <div class=”img”>
       <a href=”That Sinking Feeling.gif”><img src=”That Sinking Feeling.gif” alt=”view” name=”TheSinkingFeeling” width=”100” height=”100” border=”0” align=”middle” lowsrc=”That Sinking Feeling.gif” /></a>
      <div class=”desc”><strong>The Sinking Feeling </strong></div>
    </div>
     <div class=”img”>
      <a href=”Hung Out.gif”><img src=”Hung Out.gif” alt=”view” name=”HungOut” width=”100” height=”100” border=”0” align=”middle” lowsrc=”Hung Out.gif” /></a>


     <div class=”desc”><strong>Hung Out? </strong></div>
    </div>
     <div class=”img”>
      <a href=”The Last Round.gif”><img src=”The Last Round.gif” alt=”view” name=”TheLastRound” width=”100” height=”100” border=”0” align=”middle” lowsrc=”The Last Round.gif” /></a>
    <div class=”desc”><strong>The Last Round </strong></div>
    </div>
    </body>
    </html>


    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>


    </div>
      <div id=”content_bottom”></div>
```

```
        <div id=”footer”>
         <h3>Copyright 2012</a></a></h3>
    </div>
      </div>
     </div>
    </body>
    </html>
```

Testing and Analysis

A test case is a set of test inputs, execution and expected results developed for a particular objective. In order to make sure that the different buttons, links, images and pages in my website work I have had to test and re test my work throughout the project. Every time a new code for a button or for a link was added testing had to be done to make sure it will meet the user requirement for the chartist website. So I have decided to use the white box testing to test my Download Article button to make sure that it works properly. It is recommended for any software project to be tested and documented regardless of its size and outcomes. Many question and compare software projects with building project. Truthfully, they will never always be the same because requirement changes during software project cycle while the models used in building projects are fixed. Even so, to achieve higher success rate, learn the practices from building projects – where investigations were done for each building project failure [14].

Note that from the principle of software testing exhaustive testing is not possible this means that it will not be possible for me to test all my webpages at once this is why I tested each component separately and I have chosen the Download Article button for the White box testing. Another principle of software testing says any software that is free of bugs was not properly tested meaning that no software can be bug free. So even though the website has been tested it does not guaranty that it is 100% perfect [15].

Software Testing Process

Software Testing is the process of executing a program or system with the intent of finding errors. Or, it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well [16].

Type of Software Testing

Unit testing

This type of testing tests the individual application objects or methods in an isolated environment. It verifies the smallest unit of the application to ensure the correct structure and the defined operations. Unit testing it the most efficient and effective means to detect defects or bugs [13].

Integration testing

This testing it to evaluate proper functioning of the integrated model that makes up a system. This is also the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. The focus of integration testing is on cross-functional test [17].

System testing

System testing should be executed as soon as an integrated set of modules has been assembled to form the application. System testing verifies the product by testing the application in the integrated system environment [18].

Regression tes

Regression testing ensures that the code modification, bug correction and any postproduction activity has not introduced any additional bug [17].

Usability testing

Usability testing ensures that the presentation, data flow, and general algorithm of the application meet the intended requirement of the user.

Stress testing

Stress testing ensures that the features of the software and hardware continue to function correctly under redesigned set and volume of test scenario. The purpose of stress testing is to ensure that the system can hold and operate efficiently under different load operation.

Performance testing

Performance testing measure the response time of the system to complete a task and the efficient of the algorithm under varied condition.

Testing Implementation

For this project, I am used System testing, white box testing, Usability testing and Regression.

System testing

After the completion of the development of the project a system test was performed. This test verifies the software by testing the application in the integrated system environment. Prior to the testing a test case was design. This test case was use as the guide line for the testing. For each test case, a test was done more than 3 times. The software was tested on three different Microsoft Windows OS (XP, Vista and Windows 7). ASP code generated was tested on three web browsers (Internet Explorer, Google chrome and Mozilla Firefox).

Usability testing

The software was distributed to people to use for the purpose of user evaluation. Replay was gotten from these users on how they feel using the application.

Regression testing

After all correcting and bugs ware fixed, a system testing was conducted. This was to make sure new bugs or defect ware not introduced to the system and the system still meets it specification.

This defect was caused due to the reason that; on retrieving data from a table, the script was coded to retrieve data and even at the end of the data, it still try’s to retrieve more data(articles).

The warning message is only displayed when the application tries to retrieve data from the database and there is no more data to be retrieved.

There are two approaches to the solution of the defect:

1. Disable notice and warning in the ASP .asp file
2. Turn off notice and warning during the execution of a script.

The most efficient and appropriate solution here, is to turn of warning message during the execution of a script [18].

End User Evaluation

The application was tested by some selected users and ware asked for their opinion about it.

There are basically two categories of users that tested the application.

1. Novice
2. Expert

The novices are testers who are not so familiar with ASP or web applications but may be doing programming with other languages / technologies. While the experts are the once who are familiar with ASP web development.

I intended to choose these categories of users to evaluate the system, to know how easy the system is easy to use and understand.

From the result gotten from the evaluation it is found that the expert finds the system easy to use and understand, while the novice also finds it easy to use but some did not find it easy to understand the generated codes but can easily browse on the website as it has been designed as a user friendly device [19].

Bugs and Defects

The known bugs in the system have a proposed solution. The misinterpretation of colours: some colours are miss-displayed due to the different in colour code format of window and HTML. The application could have been design not to directly write the generated script to file every time a script is generated. As this causes some performance issue, also once an action has been performed it is not possible to reverse it (undo). This is not a good design as a user can make a mistake of deleting a control and in that case will cause a rework [20].

Critical Evaluation Process

The objectives and the goal of the project have not been changed, and the objectives of the project ware fully achieved. The risk of "Automatic Application configuration for Web server (Apache)" stated in the risk and contingency plan was really taken note off and made sure it did not occur though a fall back plane/action was set. Comparing the final project output with the project objectives, I am so proud that all objectives have been met. The application has been designed to for ease of use. Connecting and retrieving data from the database was designed with end user consideration of possible issues that may be encountered. This issues are been displayed to the user to let know their error on connection. This programmer really requires writing algorithms to solve the user problems and as well programme and scripting problems [17].

The generated script can be used modified by programmers to increate performance and add functions to it. Some tags (customs tags) ware designed to make it easy for modifications and by users. The application is designed to allow multiple searches to be done on it; this was implemented by using the MDI Form. This also makes it easy for arranging and organising once work. Most part of the design where made using dream weaver and implemented with ASP, HTML, CSS as planned [17].

Test Cases Analysis (White Box Testing)

Test case 1

Purpose: To verify that the download article button works properly

How to test: Run the application, login into the chartist. Select the purchase article link, select the download article button there the downloaded articles will be saved. Click on any articles previously downloaded. Open and view previously selected articles.

Expected result: A message box with the message "Download successful" should be displayed. The page should automatically transfer you to the saved articles page. The page should run on any OS without any errors and of the expected output [18].

Actual result: Test1: User == Successful Purchase, Results- Go to Saved Articles Page

Test2: User <> Successful Purchase, Results- End (Tables 2 and 3)

Below is the test specification for "white box" download article button: (Figure 28)

Test case 2

Purpose: Verity that the created / design is shown and interpreted appropriately and as expected on three different web browser.

How to test: View the .ASP file of your design on Internet Explorer, Google Chrome and Mozilla Firefox.

Table 2: Source code tags.

Tag	Description
<!--phpST-->	Denote the start of a PHP SCRIPT <?php
<!—phpET-->	Denote the end of a PHP SCRIPT ?>
#databaseConnection[database=' ', table=' ', column='']	Denotes the start of a database connection script also hold some details of the database (database name, table name and column name)
#databaseConnectionEnd	Denote the end of the database connection script
//Table0[row=' ', column=' ']	The start of a table script holds the number of rows and columns in the table. Denote the number of table (//Table0, //Table1 etc)
//EndTable	Denote the end of a table script.
#controls[img=' ' lab= ' ']	Holds the numbers of controls on the table. (img signifies image controls while lab signifies label controls)
#controls=	This contains all the controls in a table. Controls are separated by a comer.
//End**	Denote the end of all scripts

Table 3: Test outcome.

Decision	Outcome	Test 1	Test 2	Results
If User Purchase Successful	True	Yes		Go to Saved Articles Page to view downloaded articles that has been previously selected during purchase
	False		Yes	End of File

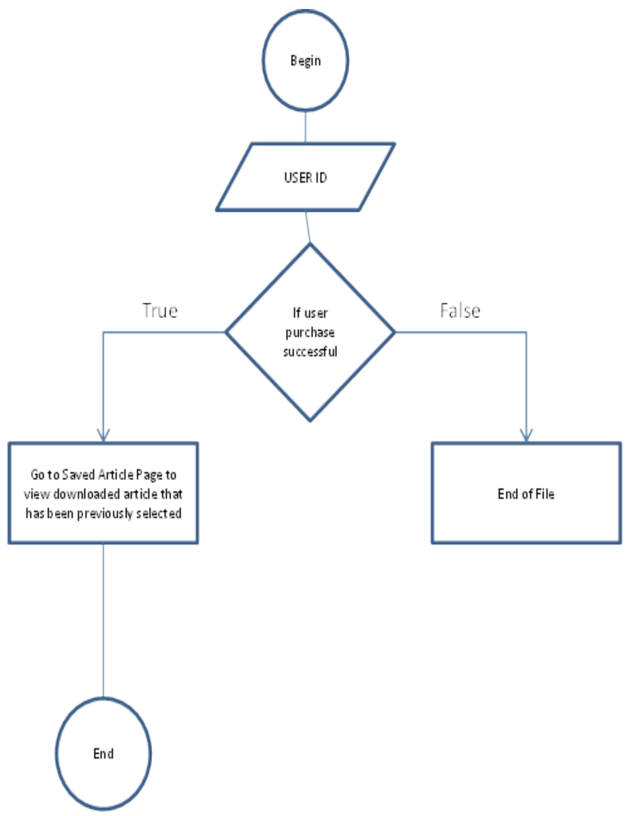

Begin
USER ID
If user purchase successful
True
False
Go to Saved Article Page to view downloaded article that has been previously selected
End of File
End

Figure 28: Flowchart for White Box Testing.

Expected result: The browsers should display the chartist website and each web page in the write format and manner as design and expected.

Actual result: The defect found from Test Case 2 was the issue with the Mozilla Firefox, which display the cell border in a specific colour (Hash) irrespective of the colour selected and the log in page did not work appropriately using the Mozilla Firefox. On an analysis done on this issue, I found that firefox does not display border colour correctly. It is stated that "Some version of fire fox does not display the table border colour directly" [21]. My experience with Firefox shows that it does not support border colour yet. Internet Explorer 4.x and 5.x colors all of the borders of the table including the cell borders [18]. A solution suggested by many author, is to use a CSS to fix the problem.

Implementing CSS in my design will cause some redesign, and which will cause extra work and time, also may cause slippage. For these reasons, the solution has been considered done in the future.

Test case 3

Purpose: To verify that the drop down box (input data) for the checkout page is display as designed.

How to test: Create a drop down menu choice.

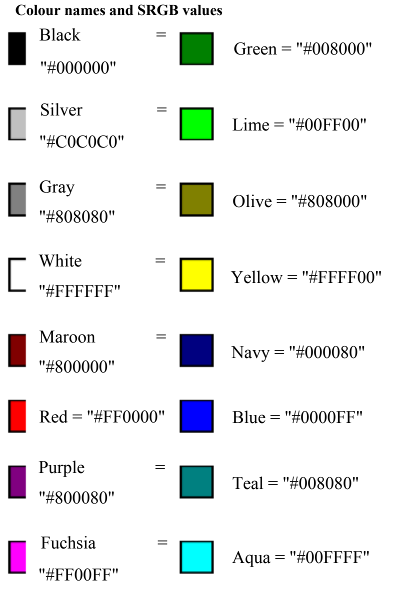
Expected result: the expiry date displayed in the drop down menu should be in accordance with the input value of creating it.

```
" <select name="Char_Payment_Card_ExpDate_Month">
<option value=1 <%=SELECTED( ccexpiremonth, 1)%>>01
<option value=2 <%=SELECTED( ccexpiremonth, 2)%>>02
<option value=3 <%=SELECTED( ccexpiremonth, 3)%>>03
<option value=4 <%=SELECTED( ccexpiremonth, 4)%>>04
<option value=5 <%=SELECTED( ccexpiremonth, 5)%>>05
<option value=6 <%=SELECTED( ccexpiremonth, 6)%>>06
<option value=7 <%=SELECTED( ccexpiremonth, 7)%>>07
<option value=8 <%=SELECTED( ccexpiremonth, 8)%>>08
<option value=9 <%=SELECTED( ccexpiremonth, 9)%>>09
<option value=10 <%=SELECTED( ccexpiremonth, 10)%>>10
<option value=11 <%=SELECTED( ccexpiremonth, 11)%>>11
<option value=12 <%=SELECTED( ccexpiremonth, 12)%>>12
</select>"
```

Actual result: Colours like blue, red green white, black [SRGB] are displayed correctly because when chosen a colour from the application it adds the name of the colour to the HTML code. (e.g. bordercolor = 'White'). Unlike the other colours [SRGB], other colours like, light yellow, dark brown and other colours flavour are displayed incorrectly. This is because windows do not display the colour names and the format in which windows use to represent colour is different from the HTML format. HTML could only interpret the SRBG colours. (e.g. bordercolor = ' ff400080') (Figure 29).

In HTML a colour value may either be a hexadecimal number (prefix by a hash mark) or one of the following sixteen colour names. The colour names are case-insensitive

"Adapted from W3 website" And also due to the fact that the checkout page was not uploaded correctly on the website the ASP drop down list was unable to appear in its correct format as programmed earlier [18].



Colour names and SRGB values

⬛	Black = "#000000"	🟩	Green = "#008000"
▨	Silver = "#C0C0C0"	🟩	Lime = "#00FF00"
▨	Gray = "#808080"	▨	Olive = "#808000"
⬜	White = "#FFFFFF"	🟨	Yellow = "#FFFF00"
🟥	Maroon = "#800000"	🟦	Navy = "#000080"
🟥	Red = "#FF0000"	🟦	Blue = "#0000FF"
🟪	Purple = "#800080"	▨	Teal = "#008080"
🟪	Fuchsia = "#FF00FF"	▨	Aqua = "#00FFFF"

Figure 29: Colour names and SRGB values Source: [22].

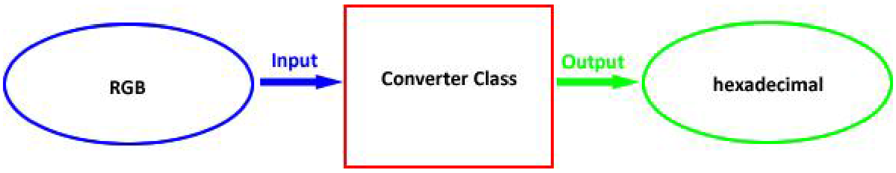


Figure 30: Class Converter [23].

Bug Mitigation

The bug found in test case four it not fixed, but mitigated. It has been mitigated by disabling windows to display more flavours of colour. Now more of the SRGB colours are been displayed but the few non SRGB are still misinterpreted.

Future Solution

A solution has been designed, which will be implemented in the future.

A class/function have to be designed to convert RGB colour to a Hexadecimal format of the HTML (Figure 30).

Conclusion

This report describes the design and development of a political ecommerce magazine website which provides detailed up-to-date information on political articles and also stores old articles in the Saved Articles Page on the website. Using Object Oriented Web Modelling approach (ASP.Net) and dream weaver I was able to model the system requirement specification to generate different pages such as check out page, gallery page and so on, which is used as a blue print for the website development. In particular more attention was given to the ability of new visitors and registered users to login and view political articles and magazines and to purchase any at will. In turn the developed website allows various restaurant, hotels and tourist travelling agent to publish detail information about their services to people by placing adverts on parts of the magazine pages which enhance their productivity as different users from all over the world order for these magazines online [20].

Furthermore, to implement the system design, ASP.Net framework environment and dream weaver was used to develop the website. Also Microsoft word was used in creating the articles. Although the developed website system implements the major system requirements further research work is needed for more people to know what is happening in the political world and in turn better our nation. The work done on the chartist magazine website thus far has been both creative and useful. Much more work needs to be done on website before it can be submitted for consideration as a standard, but it has genuine potential as a future language of the web [22-24]. Based on how clearly the website has now been developed, structured and designed, it is believed that this project has been able to develop a better and more robust e-commerce magazine website system that is utilised by users from all over the world, and will enable end users to become better citizens in the society at large.

References

1. Coad P, Yourdon E (1991) Object Oriented Design. Prentice Hall, Englewood Cliffs, NJ, USA.
2. Chartist (2012), UK.
3. Rumbaugh J, Lorensen W, Blaha M, Premerlani W, Eddy F (1991) Object-oriented Modeling and Design. Prentice Hall International, Englewood Cliffs/ NJ, USA.
4. Jackson R, MacDonald L, Freeman K (1994) Computer Generated Color: A Practical Guide to Presentation and Display. John and Wiley sons, Hoboken, New Jersey, United States.
5. Ian S (2000) Object-oriented-design. Software engineering (8th edtn).
6. Frank U (1994) An object-oriented methodology for analysing, designing, and Prototyping office procedures. Proceedingsof the 27th Annual Hawaii International Conference on System Sciences, IEEE Computer Society Press 663-672.
7. Gottesdiener E (1998) OO Methodologies: Process & Products Patterns. EBG Consulting, Inc. SIGS Publications, Newyork, USA 1: 1-19.
8. William RD (1996) A Guide to the Project Management Body of Knowledge. PMI Standards Committee. Project Management Institute, South State Road, Upper Darby, USA.
9. Yu Gao (2010) Research on the Rule of Evolution of Software Development Process Model. ICIME 2nd IEEE International Conference on Information Management and Engineering, Chengdu, China
10. Munassar Nabil Mohammed A, Govardhan A (2010) A Comparison between Five Models of Software Engineering. IJCSI 7: 1694-0814.
11. Boehm Barry W (1988) A Spiral Model of Software Development and Enhancement. TRW Defense Systems Group 61:72.
12. Converse T, Park J, Morgan C (2004) PHP5 and MySQL Bible Companion website 3-1025.
13. McManus J, Wood-Harper T (2004) A study in project failures
14. Murray James D, William VR (1994) Encyclopedia of graphics files formats. O'Reilly & Associates, Sebastopol, California, United States.
15. Sprigg Graham (1995) The Unsolved Problem of Image Compression. Image Processing 7: 38-40.
16. Charles S, Robert S, Thomas G P (1994) A comparison of JPEG and FIF compression of color medical images for dermatology. Comput Med Imaging Graph 18: 339-342.
17. Peters L, Schultz R (1992) The Application of Petri-Nets in Object-Oriented Enterprise Simulations. Proceedings of the 26th Hawaii International Conference Syst Sci
18. Hitmill (2010) HTML Table Tutorial Lesson 2d BORDERCOLOR Attribute of the TABLE Element
19. Nelson M (1992) The Data Compression Book. IDG Books Worldwide, Inc New York,USA.
20. WeisField M (2013) The object-oriented thought process (Fourth Edtn) Addison-Wesley, Boston, Massachusetts, United States.
21. Kelly H, Mitch G (2008)New Perspectives on Dreamweaver Cs3, Comprehensive. Cengage Learning, Boston, Massachusetts, United States.
22. Rose MG (1996) Computer Generated A Practical Guide to Presentation and Display, Glasgow, Scotland: John Wiley & Sons, Hoboken, New Jersey, United States.
23. Ross MG (1996) Color: A Practical Guide to Presentation and Display, Glasgow, Scotland: John Wiley & Sons, Hoboken, New Jersey, United States.
24. Frank U, Klein S (1992) Three integrated tools for designing and prototyping object-oriented enterprise models. Society for Mathematics and Data Processing, GMD-Research Report, no. 689, Sankt Augustin, Germany.

Author Affiliation

Top

[1]*Department of Software Engineering, London Metropolitan University, London, UK*

[2]*Department of Computing, University of Portsmouth, Portsmouth, UK*