

**A**

**PROJECT REPORT ON**

# **E-NEWSPAPER.COM**

**UNDER SUPERVISION OF:**

**SUBMITTED BY**

**NAME**

:

**ENROLLMENT NO**

:

**DATE OF SUBMISSION**



**INSTITUTE OF MANAGEMENT TECHNOLOGY  
CENTRE FOR DISTANCE LEARNING  
GHAZIABAD**

# E-NEWSPAPER.COM

**Under Supervision of : G.S.BALUJA**

**Submitted By:**

**Name :**

**Programme Code :**

**Enrollment No. :**

Projecthelpline.in

**CERTIFICATE**

This is to certify that Ms Shagun Warsi, a student of IMT – CDL Ghaziabad has completed project work on “Status and Prospects of Food Processing industry in India” under my guidance and supervision.

I certify that this is an original work and has not been copied from any source.

Signature of Guide -----

Name of Project Guide -----

Date -----

Projecthelpline.in

## **ACKNOWLEDGEMENT**

With Candor and Pleasure I take opportunity to express my sincere thanks and obligation to my esteemed guide **Mr.G.S.Baluja**. It is because of his able and mature guidance and co-operation without which it would not have been possible for me to complete my project.

It is my pleasant duty to thank all the staff member of the computer center who never hesitated me from time during the project.

Finally, I gratefully acknowledge the support, encouragement & patience of my family, and as always, nothing in my life would be possible without God, Thank You!

**(Student name)**

## **DECLARATION**

I hereby declare that this project work titled “**E-NEWSPAPER.COM**” is my original work and no part of it has been submitted for any other degree purpose or published in any other from till date.

**(Student name)**

Projecthelpline.in

## TABLE OF CONTENT

S.No.	TITLE	PAGE NO.
	Certificate.....	3
	Acknowledgement.....	4
	Declaration.....	5
<b>1.</b>	<b>INTRODUCTION AND OBJECTIVES OF THE PROJECT</b>	
	1.1 Introduction.....	8
	1.2 Definition of the Problem .....	10
	1.3 Drawbacks Of Current Manual- System .....	11
	1.4 Established the need of new System.....	12
	1.5 Proposed system.....	13
	1.6 Software and hardware requirements.....	14
<b>2.</b>	<b>SYSTEM ANALYSIS</b>	
	2.1 Principles of the system analysis.....	25
	2.2 Feasibility study.....	27
	2.3 Cost And Benefit Analysis.....	28
	2.4 Technical Feasibility.....	29
	2.5 Operational Feasibility.....	30
	2.6 System overview.....	31
	2.7 System Life Cycle.....	32
	2.8 Identification Of Need.....	34
	2.9 Fact Finding Techniques.....	35
<b>3.</b>	<b>ER DIAGRAM</b>	
	3.1 ER-Diagrams.....	40
<b>4.</b>	<b>DATA FLOW DIAGRAM</b>	

4.1	Context level DFD.....	44
<b>5.</b>	<b>SYSTEM DESIGN</b>	
5.1	System Design.....	51
5.2	System Development Life Cycle.....	55
5.3	Modularization Details.....	56
5.4	Process logic for each module.....	58
5.5	Number Of Functions Used In This Project.....	59
<b>6.</b>	<b>DATABASE DESIGN</b>	
6.1	Table Structure.....	62
<b>7.</b>	<b>DATA DICTIONARY</b>	
7.1	Data Dictionary.....	66
7.2	Scheduling .....	69
<b>8.</b>	<b>SOURCE CODE</b>	
8.1	Source Code.....	72
8.2	Code efficiency.....	100
8.3	Optimization Of Code.....	100
<b>9.</b>	<b>TESTING</b>	
9.1	Testing phase.....	102
9.2	Levels of testing.....	103
9.3	Verification And Validation (V&V).....	108
<b>10.</b>	<b>SYSTEM IMPLEMENTATION MAINTENANCE AND REVIEW</b>	
10.1	Hardware Evaluation Factors.....	110
10.2	Software Evaluation Factors.....	111
10.3	Conversion And Training.....	112
10.4	Training Needs.....	114
<b>11.</b>	<b>SECURITY MEASURES.....</b>	<b>115</b>
<b>12.</b>	<b>SCHEDULING.....</b>	<b>116</b>
<b>13.</b>	<b>GANT AND PERT CHART.....</b>	<b>118</b>
<b>14.</b>	<b>FURTHER SCOPE OF THE PROJECT.....</b>	<b>121</b>
<b>15.</b>	<b>CONCLUSION.....</b>	<b>121</b>
<b>16.</b>	<b>BIBLIOGRAPHY.....</b>	<b>122</b>

## **1. INTRODUCTION AND OBJECTIVE OF THE PROJECT**

### **1.1 INTRODUCTION:**

This is a Project work is undertaken in context of partial fulfillment of the MBA.

Usenet is used by many people, for many different purposes. The rise of the home network has made the job of keeping up with news groups incredibly difficult as articles read on one computer are not automatically marked as read on other machines. This project attempts to address these issues by producing a system which is both easy to use and which incorporates similar features to be found in the more popular news readers. Also, the software can be configured to provide a remotely accessible, centralized system where a user may read news articles.

I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error. This project is used by two types of users

- i. Online Users.
- ii. Administrator (management of the news agency).

Online users can see the required articles or news Administrator can maintain daily updates in the news articles. Administrator is must be an authorized user. He can further change the password. There is the facility for password recovery, logout etc.

### **OBJECTIVES OF THE PROJECT**

The objectives of the project are to:

- Research user needs & requirements in a Web based application.

- Develop a functional & usable E-NEWSPAPER.COM.
  - Delivers the latest breaking **News** and information on the latest top stories, weather, business, entertainment, politics, and more.
  - Official site of an english daily providing **News**, views along with complete entertainment with chat, forum shopping, e-cards and **News** columbs, bollywood.
- iii. Indian **News** for Indians Worldwide. Get Live **News** From India. Latest **News** from The Leading **News** Publications In India.
  - iv. To get Cricket details-features,scheduling,teams,domestic cricket.
  - v. To get Cinema details – features,mirch masala,films realeased.
  - vi. Matrimonial, Classified, Astrology.
  - vii. General lifestyles-Fashion & Designs, Art & Culture, Food, Books, E-Shopping etc.
  - viii. Users interested in job placement will must use this site.
  - ix. Users
    - Collecting Feedbacks from the users.
    - Report Generation.

## **1.2 DEFINITION OF THE PROBLEM**

This is a Project work undertaken in context of partial fulfillment of the MBA.

Since E-NEWSPAPER.COM is associated with the NEWS of DAILY ROUTINES so I decided to work on this project.

The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-to-day activities like Searching life partner, searching jobs, Registration of the new candidate for matrimonial & placement , bussiness,real-estate and placed the jobs to job seekers and match the life partner for the matrimonial candidate. And finally compute the bill etc.

I have tried my best to make the complicated process of E-NEWSPAPER.COM as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is perform each E-newspaper activity in computerized way rather than manually which is time consuming.

I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

### **1.3 DRAWBACKS OF CURRENT MANUAL- SYSTEM**

1. The current manual system has a lot of paper work and it does not deal with exact details.
2. To maintain the records of sale and service manually, is a Time-consuming job.
3. With the increase in database, it will become a massive job to maintain the database.
4. Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previously news.
5. The retrieval of records of previously sold vehicle will be a tedious job.
6. Lack of security for the records, anyone disarrange the records of your system.

Projecthelp.in

## **1.4 ESTABLISH THE NEED OF NEW SYSTEM**

- 1) **Problem of Reliability:** Current system is not reliable. It seems to vary in quality from one month to the, next. Some times it gives good output, but some times the output is worst.
- 2) **Problem of Accuracy:** There are too many mistakes in reports.
- 3) **Problem Of Timeliness:** In the current system the reports and output produced is mostly late and in most of the cases it is useless because it is not on time.
- 4) **Problem of Validity:** The output and reports mostly contains misleading information. The customer's information is sometimes not valid.
- 5) **Problem of Economy:** The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired results.
- 6) **Problem of Capacity:** The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work.

## **1.5 PROPOSED SYSTEM**

**Employee Details:** The new proposed system stores and maintains all the employees details.

**Calculations:** The new proposed system calculates salary and income tax automatically and it is very fast and accurate.

**Registers:** There is no need of keeping and maintaining salary and employee register manually. It remembers each and every record and we can get any report related to employee and salary at any time.

**Speed:** The new proposed system is very fast with 100% accuracy and saves time.

**Manpower:** The new proposed system needs less manpower. Less people can do the large work.

**Efficiency:** The new proposed systems complete the work of many executive in less time. Past details: the new proposed system contains the details of every past vehicle for future assistance.

**Reduces redundancy:** The most important benefit of this system is that it reduces the redundancy of data within the data.

**Work load:** Reduces the work load of the data store by helping in easy updates of the products and providing them with the necessary details together with financial transactions management.

## **1.6 SOFTWARE & HARDWARE REQUIREMENTS**

### **HARDWARE:**

Processor	:	Pentium 2.4 GHz or above
Memory	:	256 MB RAM or above
Cache Memory	:	128 KB or above
Hard Disk	:	3 GB or above [at least 3 MB free space required]
Floppy Disk Drive	:	3.5” [At least one drive labeled a: required]
Printer	:	Dot Matrix / DeskJet connected to LPT port

### **SOFTWARE:**

Operating System	:	Windows XP (Professional).
Font-End Tool	:	JSP, Servlets ,My Eclipse
Back-End	:	My SQL

The need of today's software development is competence in a GUI based front-end tool, which can connect to Relational Database engines. This gives the programmer the opportunity to develop client server based commercial applications.

## **FRONT END**

### **INTRODUCTION TO JSP**

Adding dynamic content via expressions

As we saw in the previous section, any HTML file can be turned into a JSP file by changing its extension to .jsp. Of course, what makes JSP useful is the ability to embed Java. Put the following text in a file with .jsp extension (let us call it **hello.jsp**), place it in your JSP directory, and view it in a browser.

```
<HTML>
<BODY>
Hello! The time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```

Notice that each time you reload the page in the browser, it comes up with the current time.

The character sequences `<%=` and `%>` enclose Java expressions, which are evaluated at run time.

This is what makes it possible to use JSP to generate dynamic HTML pages that change in response to user actions or vary from user to user.

*Exercise:* Write a JSP to output the values returned by `System.getProperty` for various system properties such as `java.version`, `java.home`, `os.name`, `user.name`, `user.home`, `user.dir` etc.

### Scriptlets

We have already seen how to embed Java expressions in JSP pages by putting them between the `<%=` and `%>` character sequences.

But it is difficult to do much programming just by putting Java expressions inside HTML.

JSP also allows you to write blocks of Java code inside the JSP. You do this by placing your Java code between `<%` and `%>` characters (just like expressions, but without the `=` sign at the start of the sequence.)

This block of code is known as a "scriptlet". By itself, a scriptlet doesn't contribute any HTML (though it can, as we will see down below.) A scriptlet contains Java code that is executed every time the JSP is invoked.

Here is a modified version of our JSP from previous section, adding in a scriptlet.

```
<HTML>
<BODY>
<%
    // This is a scriptlet. Notice that the "date"
    // variable we declare here is available in the
    // embedded expression later on.
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello! The time is now <%= date %>
</BODY>
```

```
</HTML>
```

If you run the above example, you will notice the output from the "System.out.println" on the server log. This is a convenient way to do simple debugging (some servers also have techniques of debugging the JSP in the IDE. See your server's documentation to see if it offers such a technique.)

By itself a scriptlet does not generate HTML. If a scriptlet wants to generate HTML, it can use a variable called "out". This variable does not need to be declared. It is already predefined for scriptlets, along with some other variables. The following example shows how the scriptlet can generate HTML output.

```
<HTML>
<BODY>
<%
    // This scriptlet declares and initializes "date"
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello! The time is now
<%
    // This scriptlet generates HTML output
    out.println( String.valueOf( date ) );
%>
</BODY>
</HTML>
```

Here, instead of using an expression, we are generating the HTML directly by printing to the "out" variable. The "out" variable is of type [javax.servlet.jsp.JspWriter](#).

Another very useful pre-defined variable is "request". It is of type [javax.servlet.http.HttpServletRequest](#)

A "request" in server-side processing refers to the transaction between a browser and the server. When someone clicks or enters a URL, the browser sends a "request" to the server for that URL, and shows the data returned. As a part of this "request", various data is available, including the file the browser wants from the server, and if the request is coming from pressing a SUBMIT button, the information the user has entered in the form fields.

The JSP "request" variable is used to obtain information from the request as sent by the browser. For instance, you can find out the name of the client's host (if available, otherwise the IP address will be returned.) Let us modify the code as shown:

```
<HTML>
<BODY>
<%
```

```

// This scriptlet declares and initializes "date"
System.out.println( "Evaluating date now" );
java.util.Date date = new java.util.Date();
%>
Hello!  The time is now
<%
    out.println( date );
    out.println( "<BR>Your machine's address is " );
    out.println( request.getRemoteHost() );
%>
</BODY>
</HTML>

```

A similar variable is "response". This can be used to affect the response being sent to the browser. For instance, you can call `response.sendRedirect( anotherUrl );` to send a response to the browser that it should load a different URL. This response will actually go all the way to the browser. The browser will then send a different request, to "anotherUrl". This is a little different from some other JSP mechanisms we will come across, for including another page or forwarding the browser to another page.

*Exercise:* Write a JSP to output the entire line, "Hello! The time is now ..." but use a scriptlet for the complete string, including the HTML tags.

### Mixing Scriptlets and HTML

We have already seen how to use the "out" variable to generate HTML output from within a scriptlet. For more complicated HTML, using the out variable all the time loses some of the advantages of JSP programming. It is simpler to mix scriptlets and HTML.

Suppose you have to generate a table in HTML. This is a common operation, and you may want to generate a table from a SQL table, or from the lines of a file. But to keep our example simple, we will generate a table containing the numbers from 1 to N. Not very useful, but it will show you the technique.

Here is the JSP fragment to do it:

```

<TABLE BORDER=2>
<%
    for ( int i = 0; i < n; i++ ) {
        %>
        <TR>
            <TD>Number</TD>
            <TD><%= i+1 %></TD>
        </TR>
    }
%>

```

```
</TABLE>
```

You would have to supply an `int` variable "n" before it will work, and then it will output a simple table with "n" rows.

The important things to notice are how the `%>` and `<%` characters appear in the middle of the "for" loop, to let you drop back into HTML and then to come back to the scriptlet.

The concepts are simple here -- as you can see, you can drop out of the scriptlets, write normal HTML, and get back into the scriptlet. Any control expressions such as a "while" or a "for" loop or an "if" expression will control the HTML also. If the HTML is inside a loop, it will be emitted once for each iteration of the loop.

Another example of mixing scriptlets and HTML is shown below -- here it is assumed that there is a boolean variable named "hello" available. If you set it to true, you will see one output, if you set it to false, you will see another output.

```
<%
    if ( hello ) {
        %>
        <P>Hello, world
        <%
    } else {
        %>
        <P>Goodbye, world
        <%
    }
%>
```

It is a little difficult to keep track of all open braces and scriptlet start and ends, but with a little practice and some good formatting discipline, you will acquire competence in doing it.

*Exercise:* Make the above examples work. Write a JSP to output all the values returned by `System.getProperties` with "`<BR>`" embedded after each property name and value. Do not output the "`<BR>`" using the "out" variable.

### JSP Directives

We have been fully qualifying the `java.util.Date` in the examples in the previous sections. Perhaps you wondered why we don't just import `java.util.*`;

It is possible to use "import" statements in JSPs, but the syntax is a little different from normal Java. Try the following example:

```
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%
```

```

    System.out.println( "Evaluating date now" );
    Date date = new Date();
%>
Hello! The time is now <%= date %>
</BODY>
</HTML>

```

The first line in the above example is called a "directive". A JSP "directive" starts with `<%@` characters.

This one is a "page directive". The page directive can contain the list of all imported packages. To import more than one item, separate the package names by commas, e.g.

```
<%@ page import="java.util.*,java.text.*" %>
```

There are a number of JSP directives, besides the page directive. Besides the page directives, the other most useful directives are include and taglib. We will be covering taglib separately.

The include directive is used to physically include the contents of another file. The included file can be HTML or JSP or anything else -- the result is as if the original JSP file actually contained the included text. To see this directive in action, create a new JSP

```

<HTML>
<BODY>
Going to include hello.jsp...<BR>
<%@ include file="hello.jsp" %>
</BODY>
</HTML>

```

View this JSP in your browser, and you will see your original `hello.jsp` get included in the new JSP.

*Exercise:* Modify all your earlier exercises to import the `java.util` packages.

### JSP Declarations:

The JSP you write turns into a class definition. All the scriptlets you write are placed inside a single method of this class.

You can also add variable and method declarations to this class. You can then use these variables and methods from your scriptlets and expressions.

To add a declaration, you must use the `<%!` and `%>` sequences to enclose your declarations, as shown below.

```

<%@ page import="java.util.*" %>
<HTML>
<BODY>

```

```

<%!
    Date theDate = new Date();
    Date getDate()
    {
        System.out.println( "In getDate() method" );
        return theDate;
    }
%>
Hello! The time is now <%= getDate() %>
</BODY>
</HTML>

```

The example has been created a little contrived, to show variable and method declarations.

Here we are declaring a Date variable `theDate`, and the method `getDate`. Both of these are available now in our scriptlets and expressions.

But this example no longer works! The date will be the same, no matter how often you reload the page. This is because these are declarations, and will only be evaluated once when the page is loaded! (Just as if you were creating a class and had variable initialization declared in it.)

*Exercise:* Modify the above example to add another function `computeDate` which re-initializes `theDate`. Add a scriptlet that calls `computeDate` each time.

**Note:** Now that you know how to do this -- it is in general not a good idea to use variables as shown here. The JSP usually will run as multiple *threads* of one single instance. Different threads would interfere with variable access, because it will be the same variable for all of them. If you do have to use variables in JSP, you should use *synchronized* access, but that hurts the performance. In general, any data you need should go either in the *session* object or the *request* object (these are introduced a little later) if passing data between different JSP pages. Variables you declare inside *scriptlets* are fine, e.g. `<% int i = 45; %>` because these are declared inside the local scope and are not shared.

### JSP Tags:

Another important syntax element of JSP are tags. JSP tags do not use `<%`, but just the `<` character. A JSP tag is somewhat like an HTML tag. JSP tags can have a "start tag", a "tag body" and an "end tag". The start and end tag both use the tag name, enclosed in `<` and `>` characters. The end starts with a `/` character after the `<` character. The tag names have an embedded colon character `:` in them, the part before the colon describes the type of the tag. For instance:

```

<some:tag>
body
</some:tag>

```

If the tag does not require a body, the start and end can be conveniently merged together, as

```
<some:tag/>
```

Here by closing the start tag with a `/>` instead of `>` character, we are ending the tag immediately, and without a body. (This syntax convention is the the same as XML.)

Tags can be of two types: loaded from an external tag library, or predefined tags. Predefined tags start with **jsp:** characters. For instance, `jsp:include` is a predefined tag that is used to include other pages.

We have already seen the `include` directive. `jsp:include` is similar. But instead of loading the text of the included file in the original file, it actually calls the included target at run-time (the way a browser would call the included target. In practice, this is actually a simulated request rather than a full round-trip between the browser and the server). Following is an example of `jsp:include` usage

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<jsp:include page="hello.jsp"/>
</BODY>
</HTML>
```

Try it and see what you get. Now change the `"jsp:include"` to `"jsp:forward"` and see what is the difference. These two predefined tags are frequently very useful.

*Exercise:* Write a JSP to do either a `forward` or an `include`, depending upon a boolean variable (hint: The concepts of mixing HTML and scriptlets work with JSP tags also!)

### JSP Sessions:

On a typical web site, a visitor might visit several pages and perform several interactions.

If you are programming the site, it is very helpful to be able to associate some data with each visitor. For this purpose, `"session"`s can be used in JSP.

A `session` is an object associated with a visitor. Data can be put in the session and retrieved from it, much like a `Hashtable`. A different set of data is kept for each visitor to the site.

Here is a set of pages that put a user's name in the session, and display it elsewhere. Try out installing and using these.

First we have a form, let us call it `GetName.html`

```
<HTML>
```

```

<BODY>
<FORM METHOD=POST ACTION="SaveName.jsp">
What's your name? <INPUT TYPE=TEXT NAME=username SIZE=20>
<P><INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>

```

The target of the form is "SaveName.jsp", which saves the user's name in the session. Note the variable "session". This is another variable that is normally made available in JSPs, just like `out` and `request` variables. (In the `@page` directive, you can indicate that you do not need sessions, in which case the "session" variable will not be made available.)

```

<%
    String name = request.getParameter( "username" );
    session.setAttribute( "theName", name );
%>
<HTML>
<BODY>
<A HREF="NextPage.jsp">Continue</A>
</BODY>
</HTML>

```

The `SaveName.jsp` saves the user's name in the session, and puts a link to another page, `NextPage.jsp`.

`NextPage.jsp` shows how to retrieve the saved name.

```

<HTML>
<BODY>
Hello, <%= session.getAttribute( "theName" ) %>
</BODY>
</HTML>

```

If you bring up two different browsers (not different windows of the same browser), or run two browsers from two different machines, you can put one name in one browser and another name in another browser, and both names will be kept track of.

The session is kept around until a timeout period. Then it is assumed the user is no longer visiting the site, and the session is discarded.

## **MY SQL**

My SQL is an application used to create computer databases for the Microsoft Windows family of server operating systems. It provides an environment used to generate databases that can be accessed from workstations, the web, or other media such as a personal digital assistant (PDA). MY SQL is probably the most accessible and the most documented enterprise database environment right now. This also means that you can learn it a little quicker than most other database environments on the market.

Projecthelpline.in

# SYSTEM ANALYSIS

Projecthelp.com.in

## **2.1 PRINCIPLES OF SYSTEM ANALYSIS**

### **PRINCIPLES:**

- Understand the problem before you begin to create the analysis model.
- Develop prototypes that enable a user to understand how human machine interaction will occur.
- Record the origin of and the reason for every requirement.
- Use multiple views of requirements like building data, function and behavioral models.
- Work to eliminate ambiguity

System Analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- What type of information is required?
- What are the constraints on the investigation?
- What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system; the total system was designed and can be described as under:

**The three major parts of the system are:**

➤ **Providing Information:**

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide access to records of various Job seekers & users of matrimonial such as resume & profile of boys and girls those who want to search a life partner with quick update to latest modifications in the records. This thing is not at all possible in printed material, which are updated only once a few weeks. It also gives information about the general usage of the system for first time visitors. The system itself works as a information provider for company & life partner seekers.

**Preliminary Investigation**

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- Preliminary investigation

- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organisation initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

- 1) Request clarification
- 2) Feasibility study
- 3) Request approval

**Request clarification:** Many requests from employees and users in the organisations are not clearly defined, Therefore it becomes necessary that project request must be examined and clarified properly before considering systems investigation.

## **2.2 FEASIBILITY STUDY**

The feasibility study proposes one or more conceptual solution to the problem set of the project. In fact, it is an evaluation of whether it is worthwhile to proceed with project or not.

1. Evaluation of feasibility of such solutions. Such evaluation often indicates shortcomings in the initial goals. This step is repeated as the goals are adjusted and the alternative solutions are evaluated.

Feasibility analysis usually considers a number of project alternatives, one that is chosen as the most satisfactory solution. These alternatives also need to be evaluated in a broad

way without committing too many resources. Various steps involved in feasibility analysis are:

2. To propose a set of solution that can realize the project goal. These solutions are usually descriptions of what the new system should look like.

Four primary areas of interest in feasibility study are:

**Economic Feasibility**: An evaluation of development cost weighed against the ultimate income of benefit derived from the development system of product. In economic feasibility, cost benefit analysis is done in which expected cost and benefits are evaluated.

### **2.3 COST AND BENEFIT ANALYSIS**

Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected costs.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

1. **Development Costs** – Development costs is the costs that are incurred during the development of the system. It is one time investment.

2. **Operating Costs** – Operating Costs are the expenses required for the day to day running of the system. Examples of Operating Costs are Wages, Supplies and Overheads.
3. **Hardware/Software Costs** – It includes the cost of purchasing or leasing of computers and it's peripherals. Software costs involves required S/W costs.
4. **Personnel Costs** – It is the money spent on the people involved in the development of the system.
5. **Facility Costs** – Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lightning, and air-conditioning.
6. **Supply Costs** – These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and the like.

□ **BENEFITS**

We can define benefits as

**Profit or Benefit = Income – Costs**

Benefits can be accrued by:

- Increasing income, or
- Decreasing costs, or
- Both

## **2.4 TECHNICAL FEASIBILITY:**

Technical Feasibility includes existing and new H/W and S/W requirements that are required to operate the project using JSP. The basic S/W requirement is J2EE in which the front end of the online hospital management project has been done. The basic entry forms are developed in JSP and the data is stored in the MY SQL.

## **2.5 OPERATIONAL FEASIBILITY:**

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits? The essential questions that help in testing the technical feasibility of a system are following:

- Does management support the project?
- Are the users not happy with current business practices? Will it reduce the time considerably? If yes, then they will welcome the change and the new system.
- Have the users involved in the planning and development of the project? Early involvement reduced the probability of resistance towards the new system.
- Will the proposed system really benefit the organization? Does the overall response increase? Will accessibility of information be lost? Will the system affect the customers in considerable way?

### **Legal Feasibility:**

A determination of any infringement, violation, or liability that could result from development of the system. Legal feasibility tells that the software used in the project should be original purchased from the legal authorities and they have the license to use it or the software are pirated.

### **Alternatives:**

An evaluation of alternative approaches to the development of system or product.

## **2.6 SYSTEM OVERVIEW**

The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in E-NEWSPAPER.COM Company, but utmost care has been taken to make the system efficient and user friendly.

For the optimum use of practical time it is necessary that every session is planned. Planning of this project will include the following things:

- Topic Understanding.
- Modular Break – Up of the System.
- Processor Logic for Each Module.
- Database Requirements.

### **Topic Understanding:**

It is vital that the field of application as introduced in the project may be totally a new field. So as soon as the project was allocated to me, I carefully went through the project to identify the requirements of the project.

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, The subject System Analysis and Design (SAD), mainly deals with the software development activities.

### **DEFINING A SYSTEM**

A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objectives of the system demand that some output is produced as a result of processing the suitable inputs.

## **2.7 SYSTEM LIFE CYCLE**

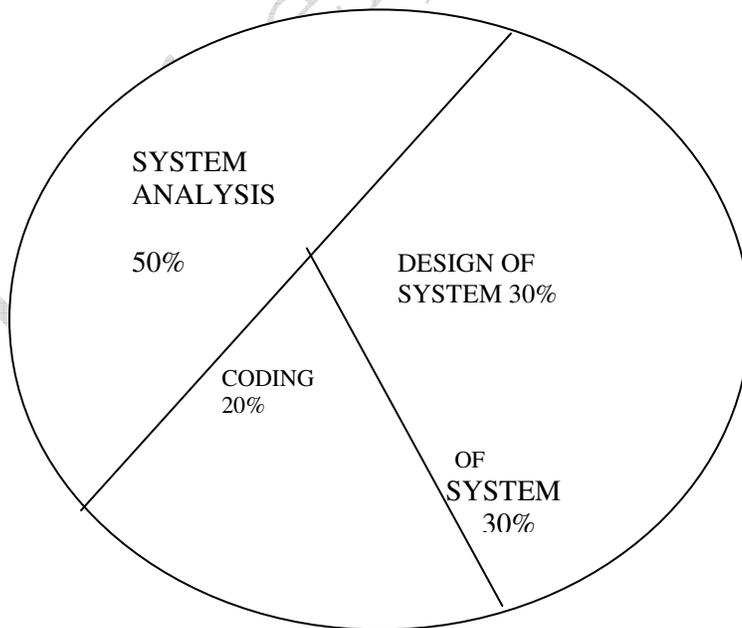
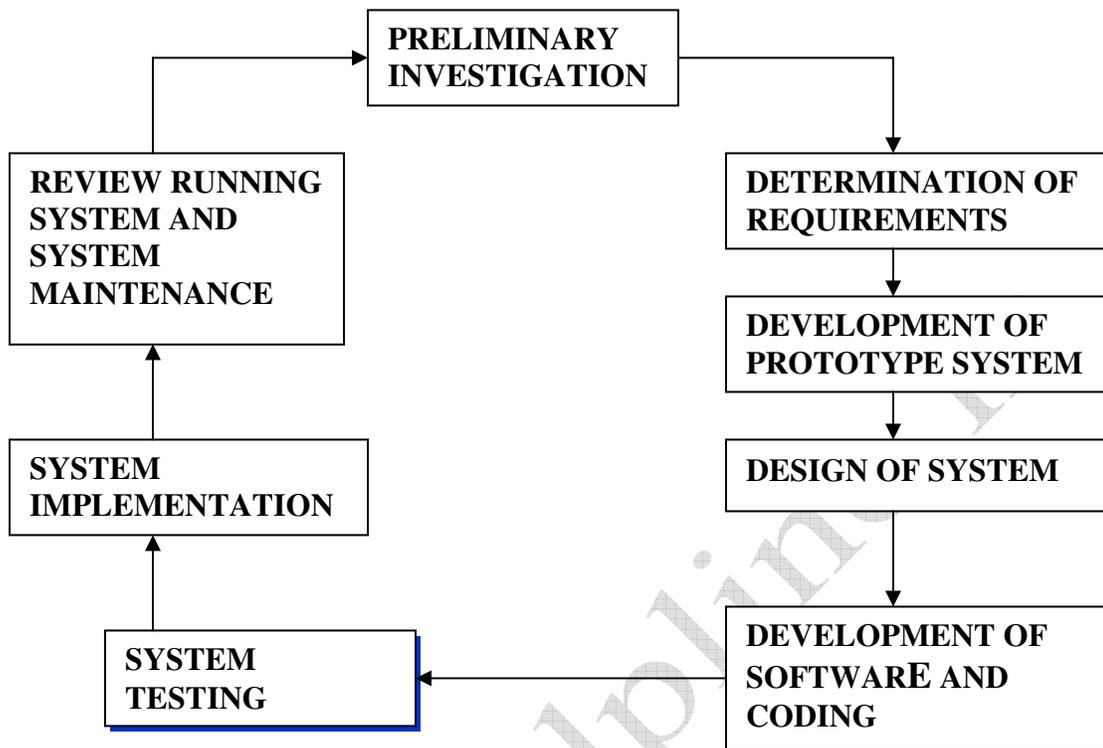
System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance

The different phases of software development life cycle is shown below.



**FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED**

## **2.8 IDENTIFICATION OF NEED**

I have designed the given proposed system in the JSP to automate the process of news sites. Many different people use Usenet, for many different reasons, ranging from discussions of various topics, requests for help or to trade media, which is often seen as the 'dark side' to Usenet — the abuse of public news groups to trade copyrighted or offensive material. Thankfully however, those that use Usenet responsibly far out weight the few who use it as a tool for piracy etc.

Many of the people using and contributing to Usenet are people who work with computer technology. These people often use Usenet to seek help with a piece of software or suggest improvements, indeed one of the early functions of Usenet was to help users identify and report bugs found in software.

Unfortunately, the massive growth seen within Usenet has also made it a difficult medium to keep track of, the great variety and number of articles can take considerable time to filter through in order to find those of interest. There have been a variety of software applications created to help deal with this problem and many are freely available for personal use.

The following steps that give the detailed information of the need of proposed system are:

**Performance:** During past several decades, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Company system, the computerized system is to be undertaken. The computerized project is fully computerized and user friendly even that any of the members can see the report and status of the company.

**Efficiency:** The basic need of this website is efficiency. The website should be efficient so that whenever a new user submit his/her details the website is updated automatically. This record will be useful for other users instantly.

**Control:** The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.

**Security:** Security is the main criteria for the proposed system. Since illegal access may corrupt the database. So security has to be given in this project.

## **2.9 FACT FINDING TECHNIQUES**

The functioning of the system is to be understood by the system analyst to design the proposed system. Various methods are used for this and these are known as fact-finding techniques. The analyst needs to fully understand the current system.

The analyst needs data about the requirements and demands of the project undertaken and the techniques employed to gather this data are known as fact-finding techniques. Various kinds of techniques and the most popular among them are interviews, questionnaires, record views, case tools and also the personal observations made by the analyst himself.

### □ **Interviews**

Interview is a very important data gathering technique as in this the analyst directly contacts system and the potential user of the proposed system.

One very essential aspect of conducting the interview is that the interviewer should first establish a rapport with the interviewee. It should also be taken into account that the interviewee may or may not be a technician and the analyst should prefer to use day to day language instead of jargon and technical terms.

The advantage of the interview is that the analyst has a free hand and he can extract almost all the information from the concerned people but then as it is a very time consuming method, he should also employ other means such as questionnaires, record reviews, etc. This may also help the analyst to verify and validate the information gained. Interviewing should be approached, as logically and from a general point of view the following guides can be very beneficial for a successful interview:

1. Set the stage for the interview.
2. Establish rapport; put the interview at ease.
3. Phrase questions clearly and succinctly.
4. Be a good listener; avoid arguments.
5. Evaluate the outcome of the interview.

The interviews are of the two types namely **structured** and **unstructured**.

### **I. Structured Interview**

Structured interviews are those where the interviewee is asked a standard set of questions in a particular order. All interviews are asked the same set of questions. The questions are further divided into two kinds of formats for conducting this type of interview.

### **II. Unstructured Interview**

The unstructured interviews are undertaken in a question-and-answer format. This is of a much more flexible nature than the structured and can be very rightly used to gather general information about the system.

#### **□ Questionnaires:**

Questionnaires are another way of information gathering where the potential users of the system are given questionnaires to be filled up and returned to the analyst.

Questionnaires are useful when the analyst need to gather information from a large number of people. It is not possible to interview each individual. Also if the time is very short, in that case also questionnaires are useful. If the analyst guarantees the anonymity of the respondent then the respondent answers the questionnaires very honestly and critically.

The analyst should sensibly design and frame questionnaires with clarity of it's objective so as to do just to the cost incurred on their development and distribution.

#### □ **Record Reviews**

Records and reports are the collection of information and data accumulated over the time by the users about the system and it's operations. This can also put light on the requirements of the system and the modifications it has undergone. Records and reports may have a limitation if they are not up-to-date or if some essential links are missing. All the changes, which the system suffers, may not be recorded. The analyst may scrutinize the records either at the beginning of his study which may give him a fair introduction about the system and will make him familiar with it or in the end which will provide the analyst with a comparison between what exactly is/was desired from the system and it's current working.

#### □ **On-Site Observation**

On-site observations are one of the most effectively tools with the analyst where the analyst personally goes to the site and discovers the functioning of the system. As a observer, the analyst can gain first hand knowledge of the activities, operations, processes of the system on-site, hence here the role of an analyst is of an information seeker. This information is very meaningful as it is unbiased and has been directly taken by the analyst. This exposure also sheds some light on the actual happenings of the system as compared to what has already been documented, thus the analyst gets closer to system. This technique is also time-consuming and the analyst should not jump to conclusions or

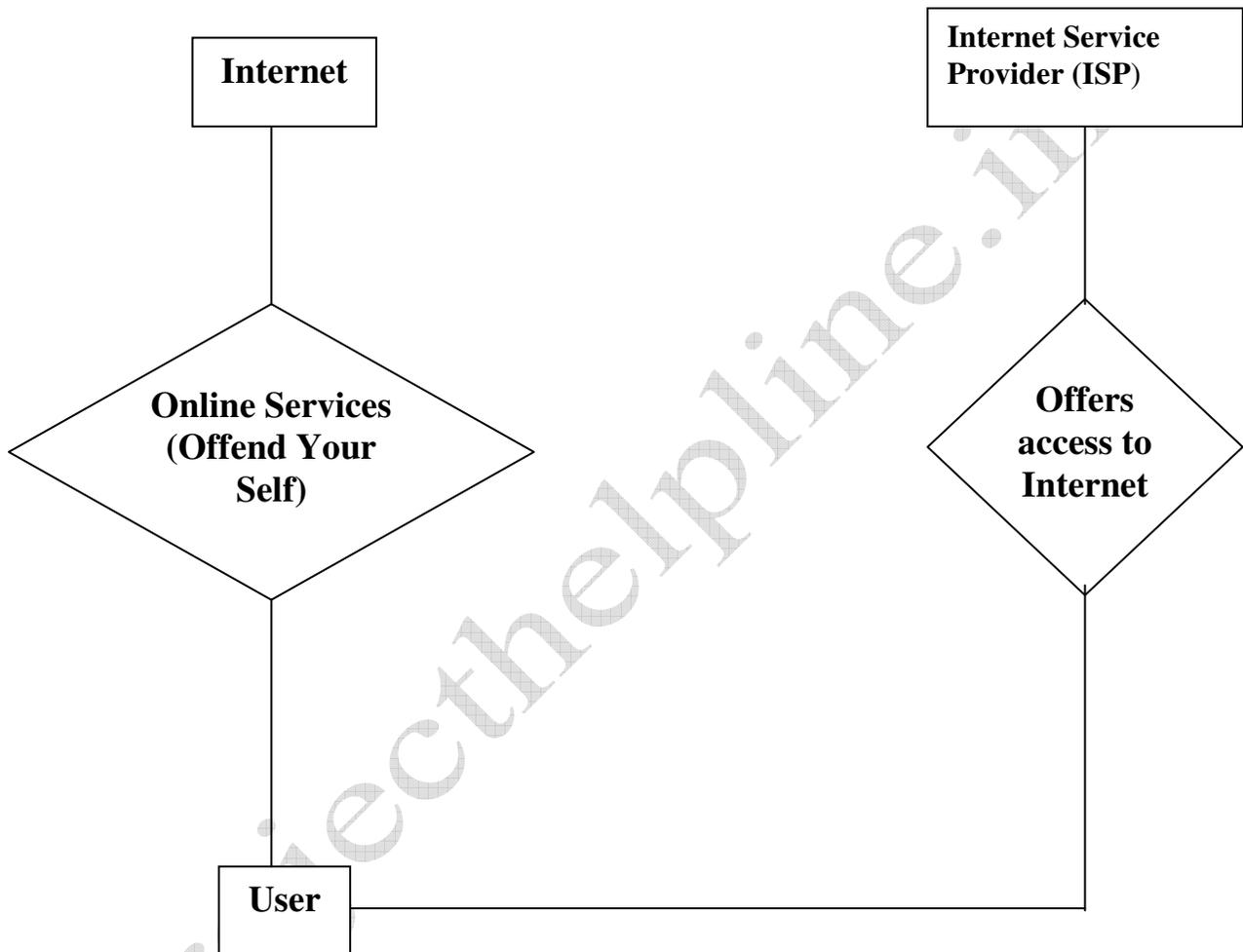
draw inferences from small samples of observation rather the analyst should be more patient in gathering the information. This method is however less effective for learning about people's perceptions, feelings and motivations.

Projecthelpline.in

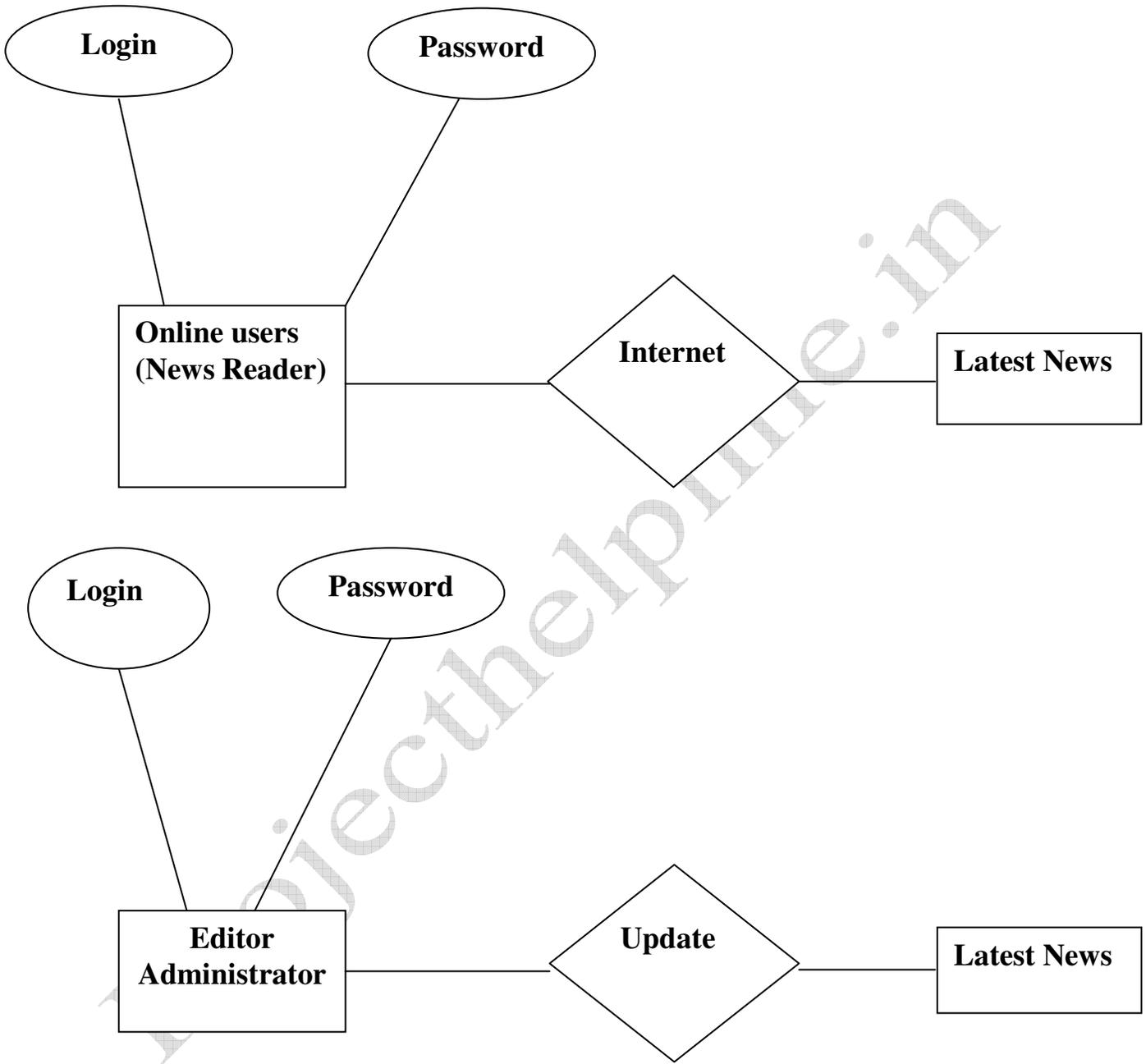
# E-R DIAGRAM

Projecthelpzone.in

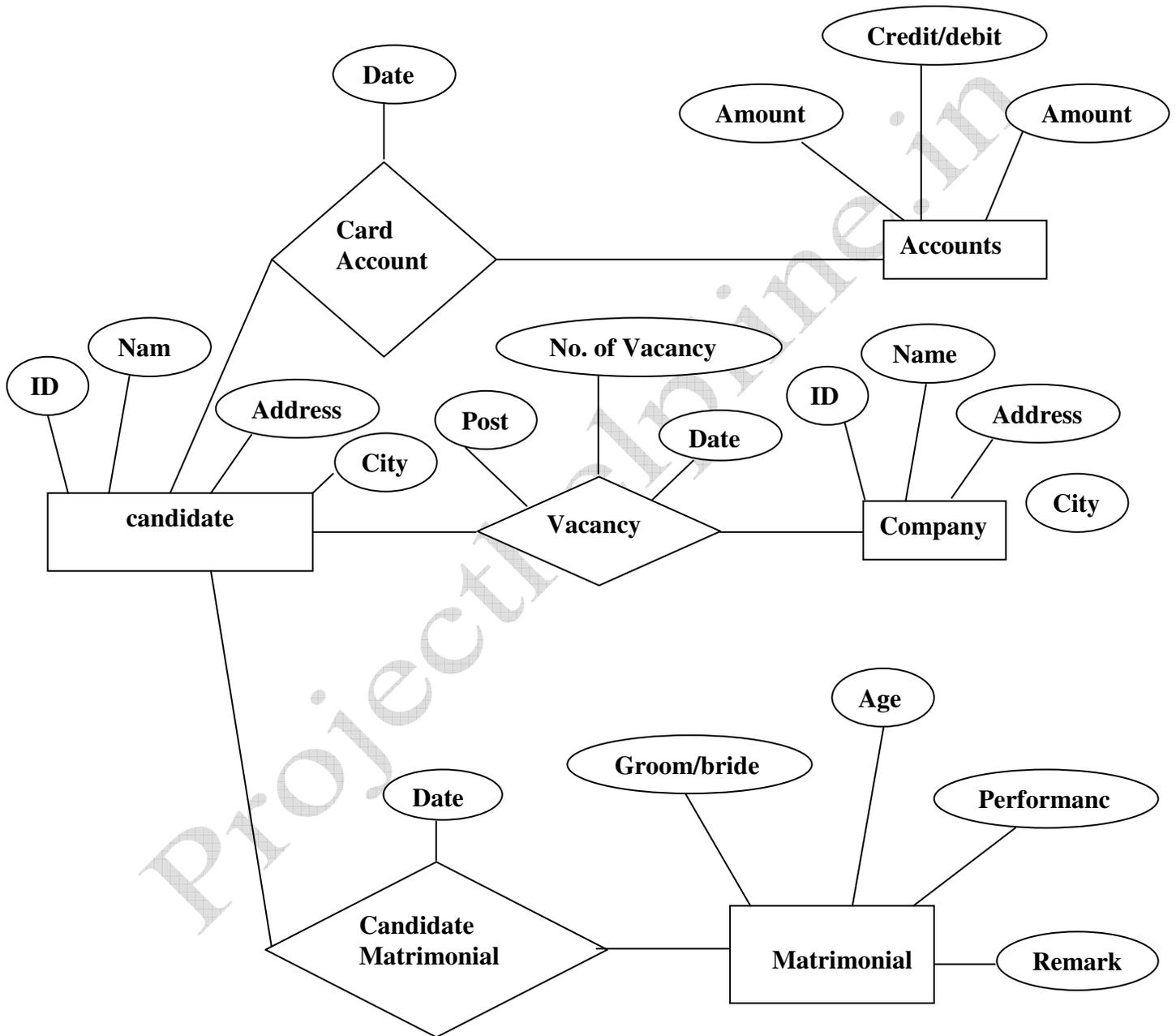
### 3.1 E-R DIAGRAM



**Online Services connect you to the internet but encourage you to explore their offerings, whereas ISPs just connect you to the internet and list you send for yourself.**



## ER DIAGRAM

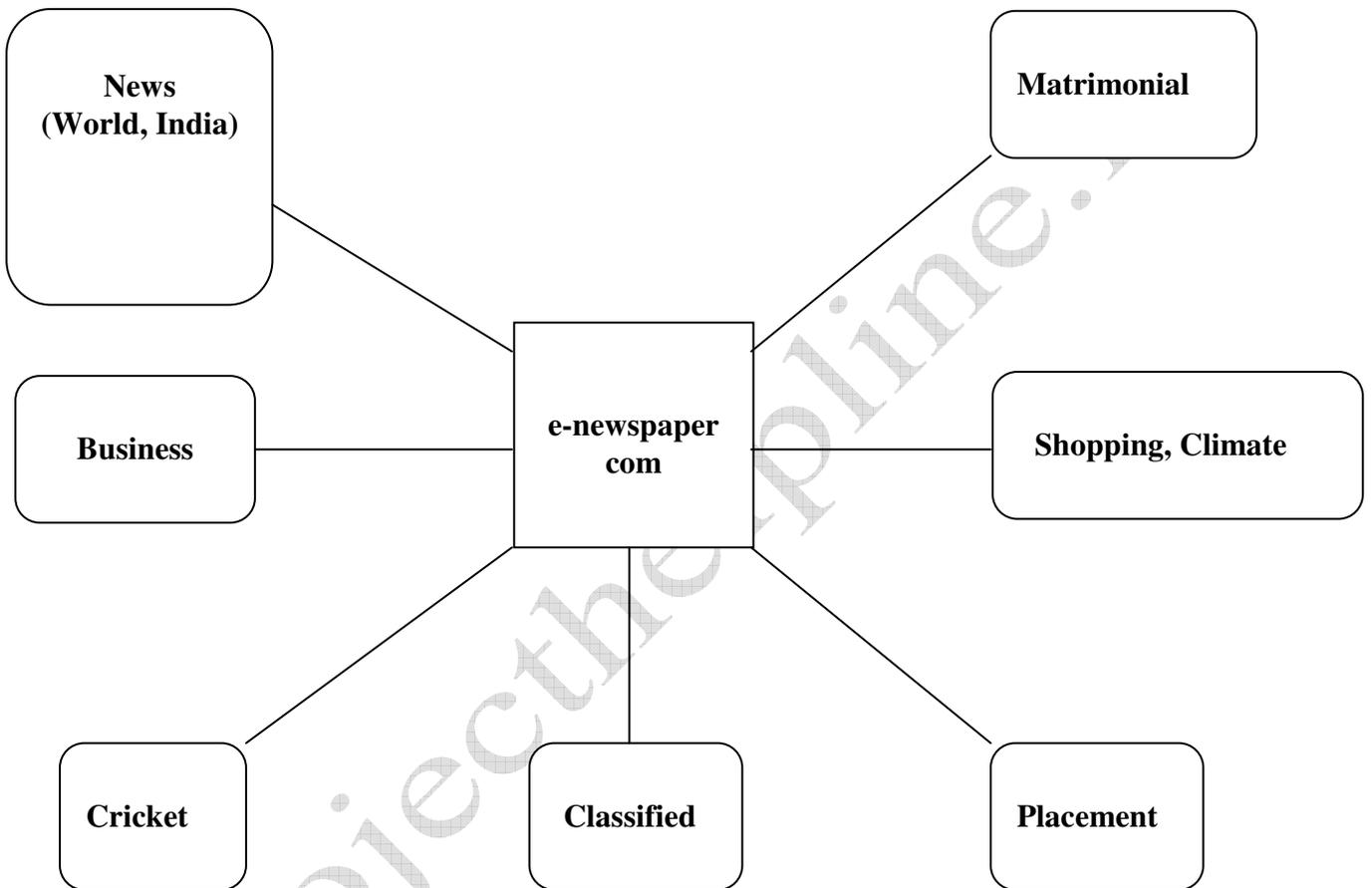


# **E-NEWSPAPER.COM DATA FLOW DIAGRAM**

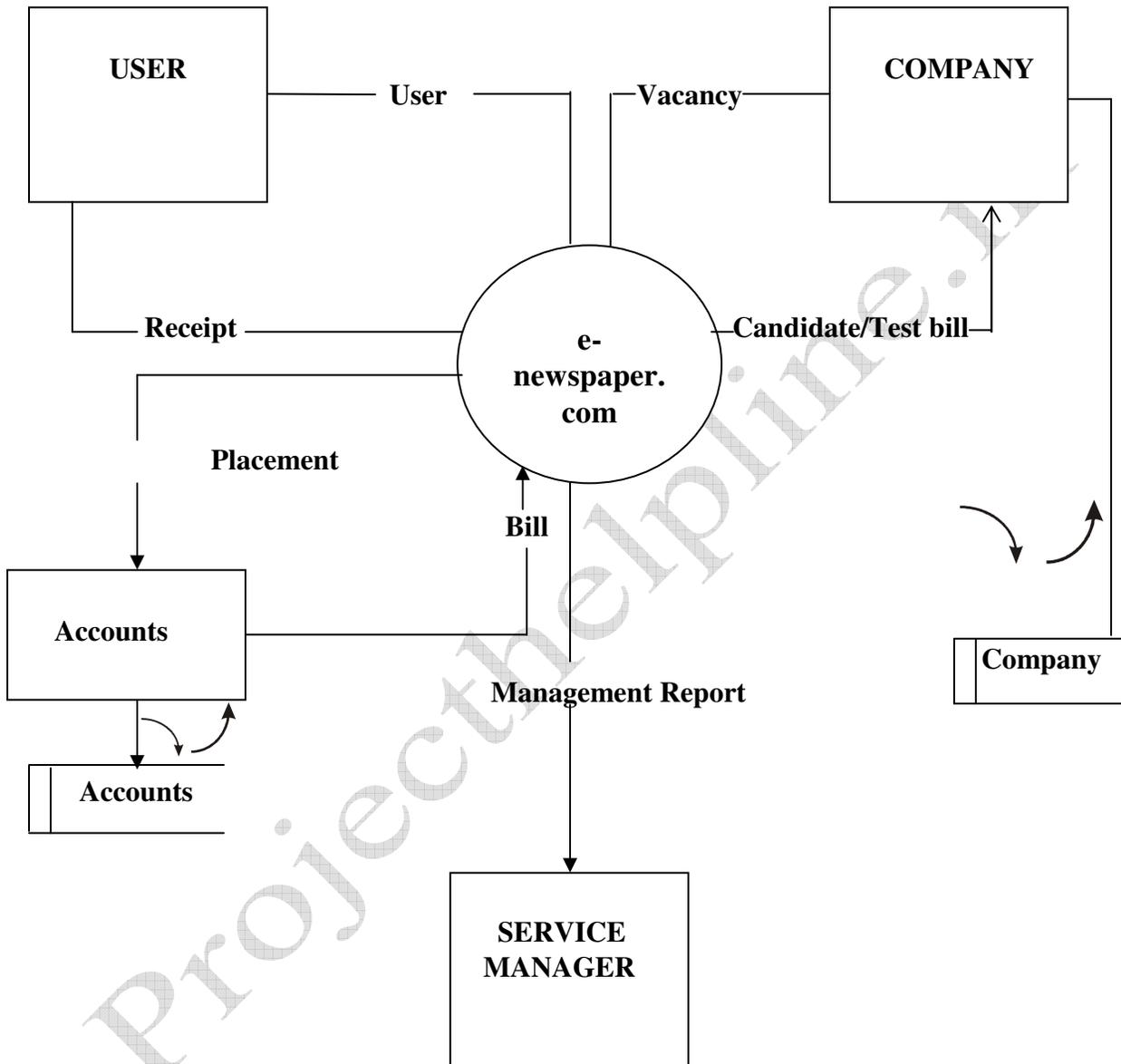
ProjectHomeLibrary.in

**4.1 E-NEWSPAPER.COM**

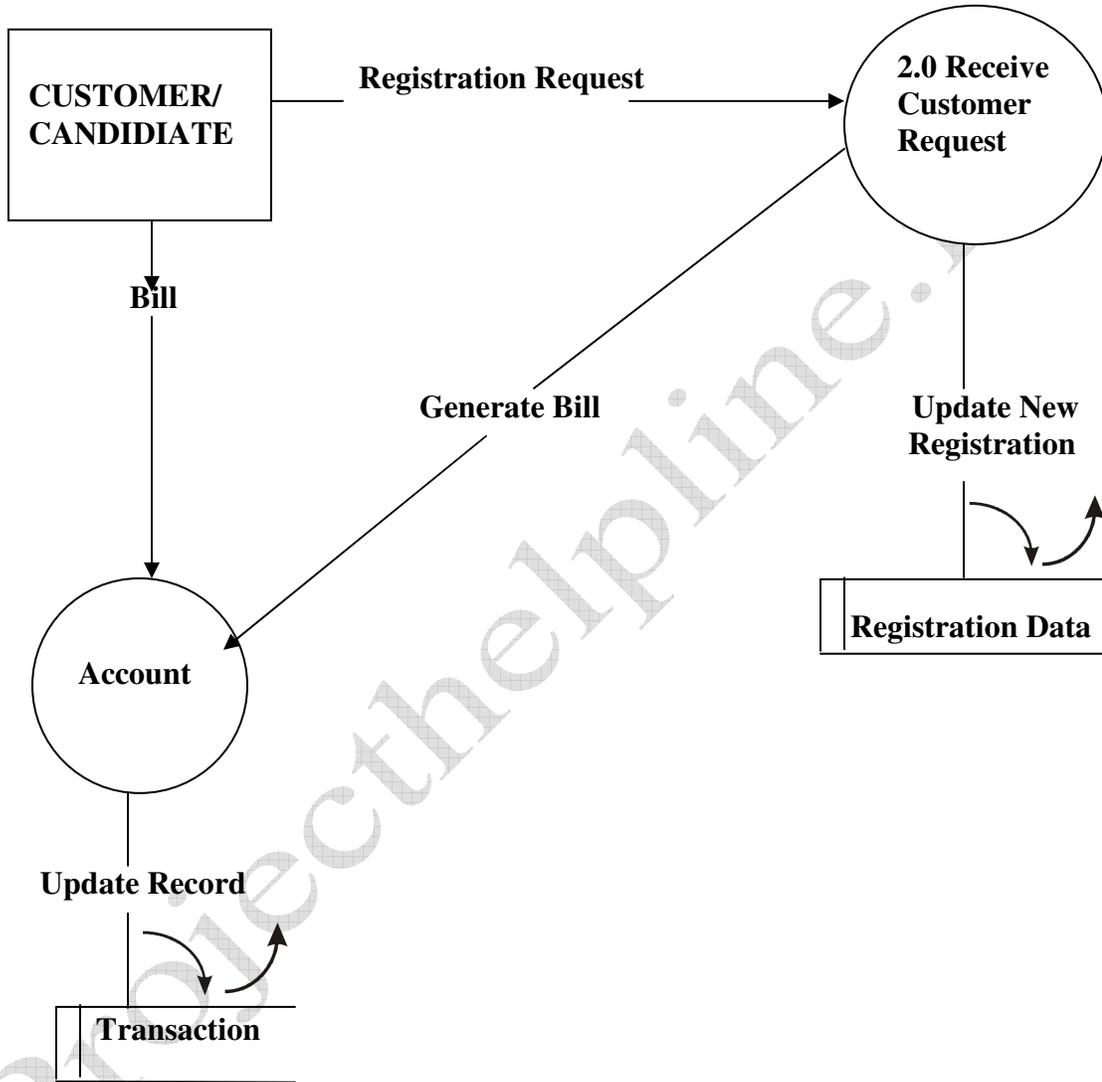
**CONTEXT LEVEL DFD**



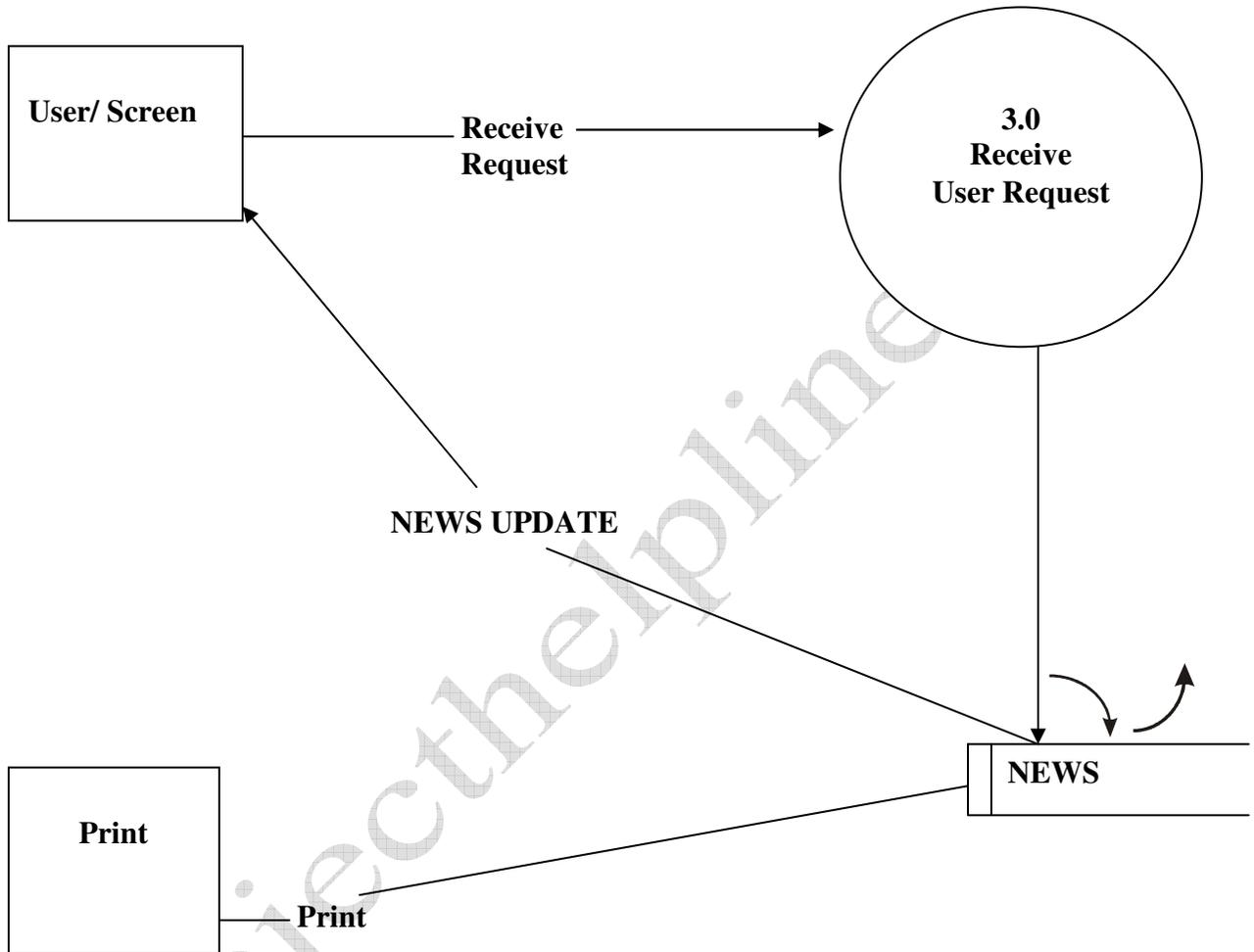
### PLACEMENT MODULE



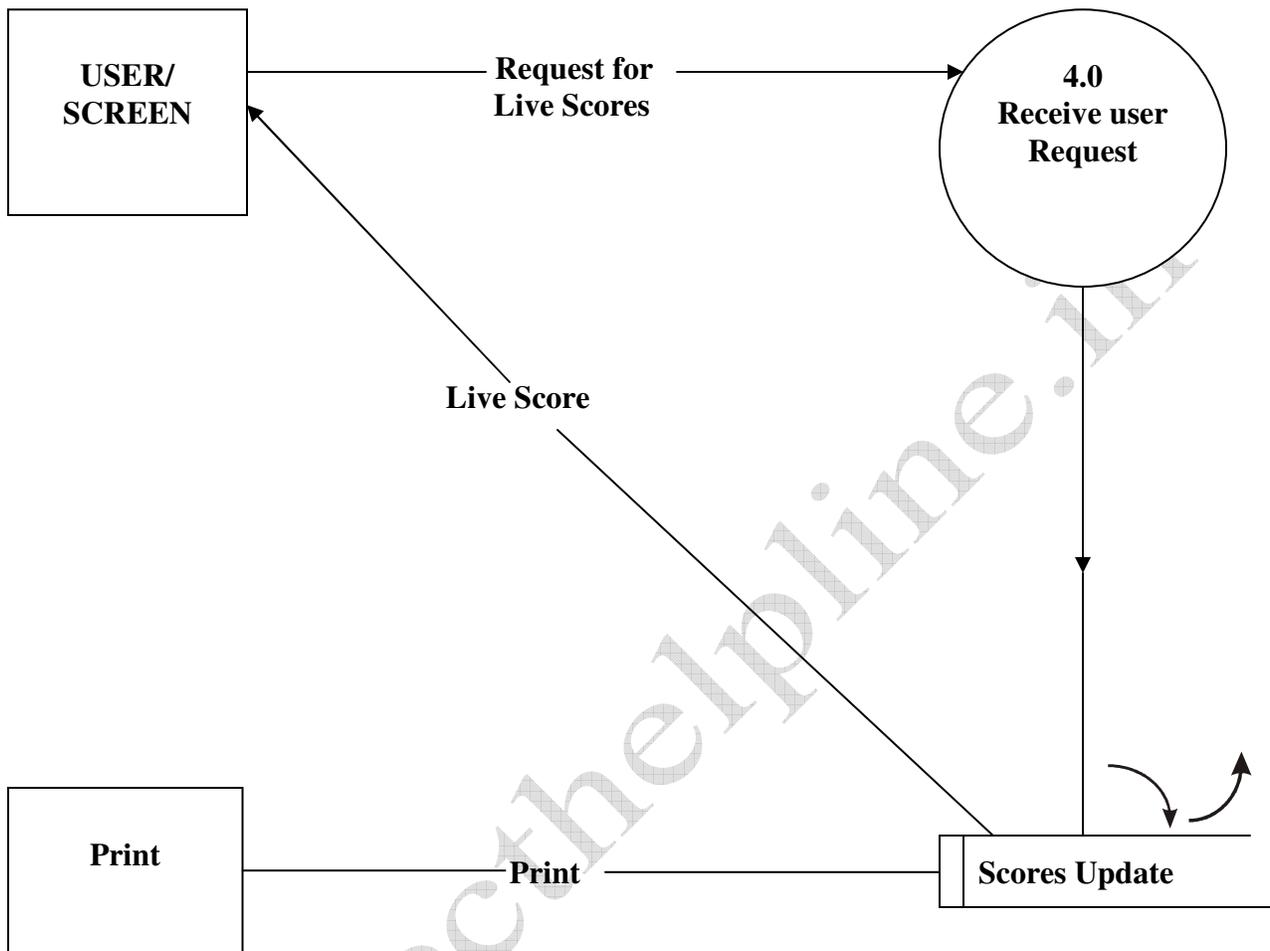
## DFD FOR MATRIMONIAL DATA



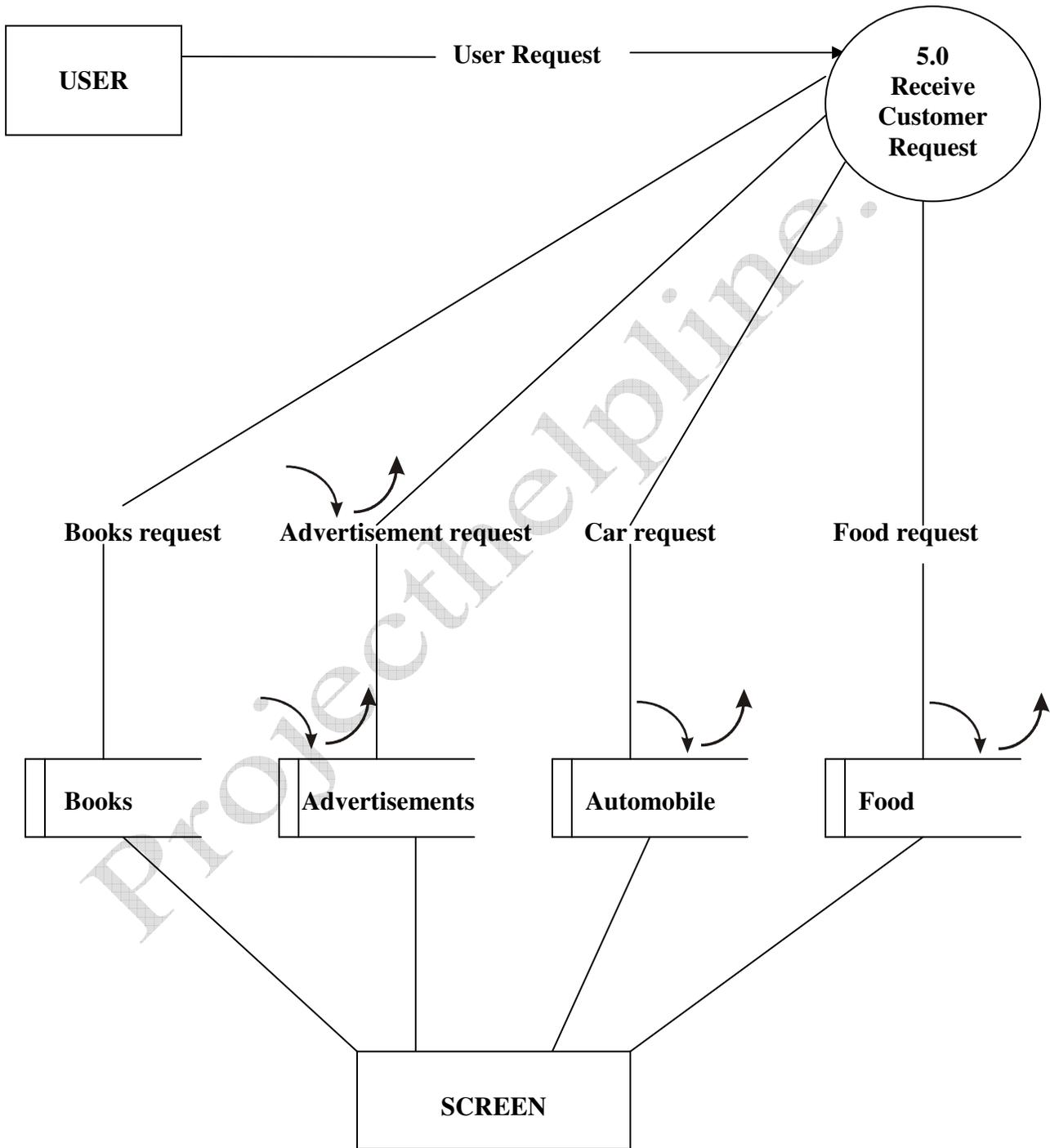
**DFD FOR NEWS**



**DFD FOR CRICKET**



**DFD FOR LIFE STYLE**



# SYSTEM DESIGN

Projecthelp.com.in

## **5.1 SYSTEM DESIGN**

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: -

Define the boundaries along which to break;  
Determine into how many pieces to break; and  
Identify the proper level of detail when design should stop and

implementation should start.

Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain.

The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design activity begins when the requirements document for the software to be developed is available. This may be the SRS for the complete system, as is the case if the waterfall model is being followed or the requirements for the next "iteration" if the iterative enhancement is being followed or the requirements for the prototype if the prototyping is being followed. While the requirements specification activity is entirely in the problem domain, design is the first step in moving from the problem domain toward the solution domain. Design is essentially the bridge between requirements specification and the final solution for satisfying the requirements.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module.

The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these

modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

The input to the design phase is the specifications for the system to be designed. Hence, a reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc.

The output of the top-level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. A reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

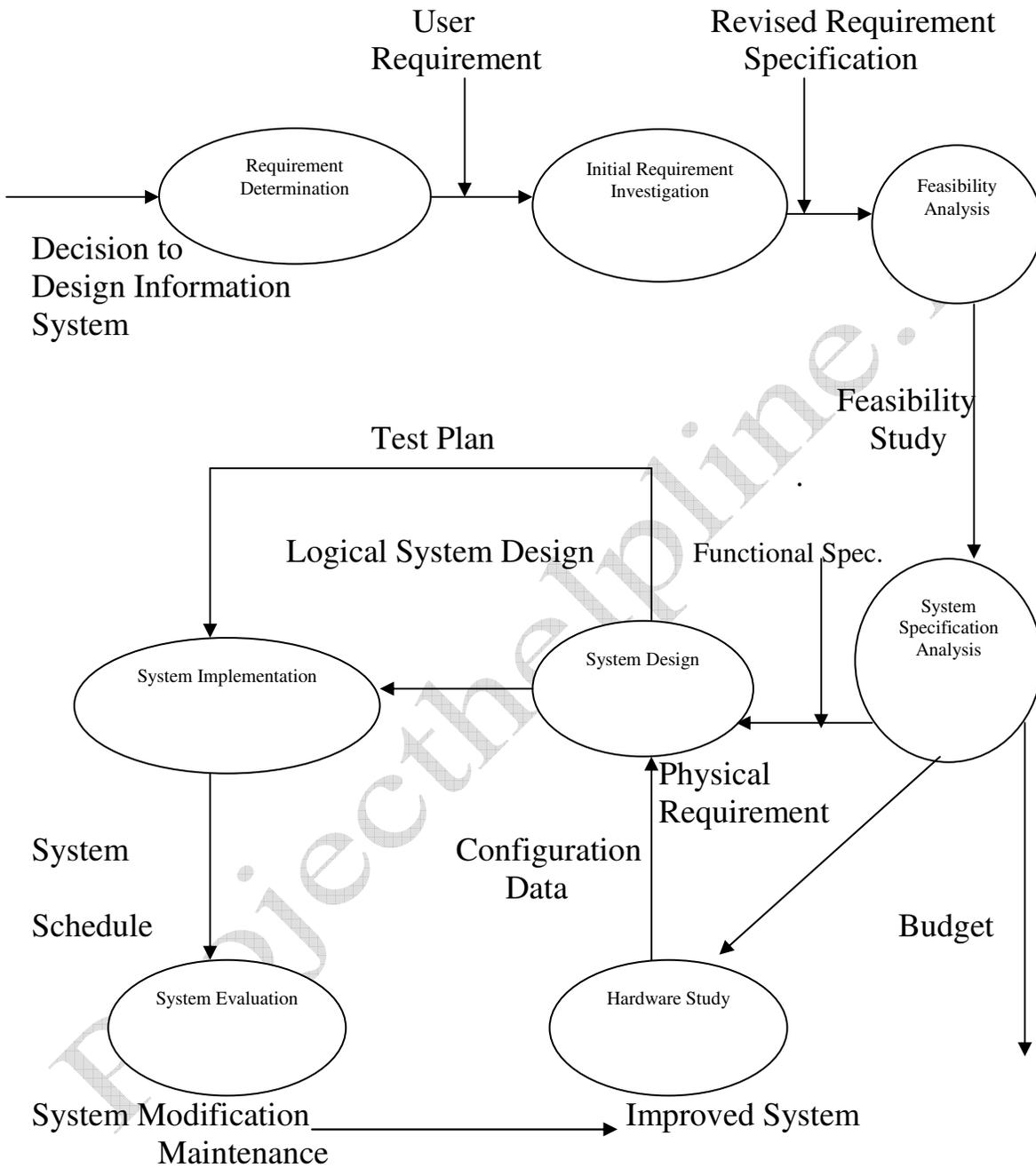
A design can be object-oriented or function-oriented. In function-oriented design, the design consists of module definitions, with each module supporting a functional abstraction. In object-oriented design, the modules in the design represent data abstraction (these abstractions are discussed in more detail later). In the function-oriented methods for design and describe one particular methodology the structured design methodology in some detail. In a function-oriented design approach, a system is viewed as a transformation function, transforming the inputs to the desired outputs. The purpose of the design phase is to specify the components for this transformation function, so that each component is also a transformation function. Hence, the basic output of the system

design phase, when a function oriented design approach is being followed, is the definition of all the major data structures in the system, all the major modules of the system, and how the modules interact with each other.

Once the designer is satisfied with the design he has produced, the design is to be precisely specified in the form of a document. To specify the design, specification languages are used. Producing the design specification is the ultimate objective of the design phase. The purpose of this design document is quite different from that of the design notation. Whereas a design represented using the design notation is largely to be used by the designer, a design specification has to be so precise and complete that it can be used as a basis of further development by other programmers. Generally,

Projecthelpline.in

## 5.2 SYSTEM DEVELOPMENT LIFE CYCLE



### **5.3 MODULARISATION DETAILS**

The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in E-NEWSPAPER.COM Company, but utmost care has been taken to make the system efficient and user friendly.

For the optimum use of practical time it is necessary that every session is planned. Planning of this project will include the following things:

- Topic Understanding.
- Modular Break – Up of the System.
- Processor Logic for Each Module.
- Database Requirements.

#### **Topic Understanding:**

It is vital that the field of application as introduced in the project may be totally a new field. So as soon as the project was allocated to me, I carefully went through the project to identify the requirements of the project.

#### **Modular Break –Up of the System:**

- Identify The Various Modules In The System.
- List Them In The Right Hierarchy.
- Identify Their Priority Of Development
- Description Of The Modules:

## **(i).Modules**

### **Module 1: Password Module**

In this module, this website is for multiple users. If a User enters a password and the software checks its validity. This module is used in astrology, matrimony, lifestyle etc.. If the password is valid then option is given to change the password, otherwise “Invalid User/Password” message is displayed. There is an option for password recovery, log out, login, new users sign in. The Administrator can also update changes in the site after login.

**Module 2: Creating new Entities ( Users for Matrimonial, Lifestyle, Astrology, Cricket, Users, Administrator)** In this module, whenever a new entity is required to be added the corresponding forms are opened and the database is manipulated to check whether the data is already existing or not. If it already exists, then it prompts that “Entry already existing” and if not than the data is entered with the various validation checks.

### **Module 3: Modifying / Updating Existing Entities**

In this module, whenever an existing entity is required to be modified the corresponding forms are opened and the database is manipulated and the data is fetched. Now the administrator can made the required changes and then accordingly, he updates the data. Again, the checks are followed in case there is any invalid entry.

### **Module 4: Searching**

In this module, whenever an existing entity is required to be searched the corresponding forms are opened and the database is manipulated and the data is fetched. Again, the checks are followed in case there is any invalid entry.

### **Module 5: Validation of Data Entered by the User & Error Handling**

In this module, the validity of data entered by the user during the various business processes is checked through various validation checks. For example, there should not be

any characters entered in the numeric fields, likewise if there is any error occurs than it should handle that particular error and give the required messages.

### **Module 6: Feedback**

This module keeps track of all the feedbacks given by the users for future records.

### **Module 7: Transactions**

This module keeps track of all the transactions done by the company for future records and income tax purposes.

**Module 8: Breaking News** Delivers the latest Breaking news and information on the latest top stories, weather, business, entertainment, politics, cricket, lifestyle, cinema, matrimonial, classified,.

## **5.4 PROCESS LOGIC FOR EACH MODULE:**

In the **first** module, validity of password is checked against a particular user.

In the **second** module, whenever a new entity is entered it should be checked for the duplicate data.

In the **third** and **fourth** module, just like the first module it should have the proper checks for every entity being modified or updated.

In the **fifth** and **sixth** module, again the validation checks are made and the different reports are generated to ease the business processes and decision making.

## **(ii).DataStructure Requirements according to the modules:**

- Identify The Various Tables Required.
- Fields for These Tables.
- The Various Key Fields (for example Primary key and foreign key).
- Identify The Various Constraints like Not Null, Unique etc.

## **5.5 NUMBER OF FUNCTIONS USED IN THIS PROJECT**

### **1. Function Password()**

This is the function used for password authentication for users and administrator.

### **2. Function New Articles ()**

This is the function used to add new articles in the database. In that screen, the automatic current date is created. The record is appended in the database according to the date.

### **3. Function Display Records ()**

This function is used to display all the transaction regarding any Indian or regional news, matrimonial, placement, astrology, cricket, user etc.. This is a global report to display all the transaction records in the screen.

### **4. Functions Business()**

This function is used to display all the details regarding Stock Market, Corporate, Economy, Private sectors etc.

### **5. Function Cricket()**

This function is used to display all the details regarding Cricket, Cricket Teams, Scheduling of matches etc.

**6. Functions Matrimonial()**

This function is used by the users who are interested in Matrimony.

**7. Function Classified()**

This function is related to Recruitment, Automobiles, Real Estate, Fashion & Beauty, Books, Food, Art & Culture etc.

**8. Functions Astrology()**

This function is related to Astrology, offers Free Horoscope, Career Guidance, Stock Market information etc.

**9. Functions Search()**

This function is used to search records regarding any Indian or regional news, matrimonial, placement, astrology, cricket, user etc.

**10. Functions Reports()**

This function is used to generate reports by the administrator for

- i. Users
- ii. Users interested in Matrimonial
- iii. Users interested in Placement
- iv. Users interested in Astrology
- v. News Articles.
- vii. Cricket Scheduling
- x. Cinema-latest gossips
- xi. Cricket Teams.
- xii. Any general information

# DATABASE DESIGN

Projecthelp.in

## 6.1 TABLE STRUCTURE

### Tables:

<b>Table: Login</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
User Name	Char	Primary
Password	Char	Not Null

<b>Table 2: New Users</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Name	Char	Not Null
Address	Char	Not Null
Phone	Number	Not Null
Email	Char	Not Null
Id	Number	Primary

<b>Table3: Matrimonial</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Name	Char	Not Null
Email	Char	Not Null
Address	Char	Not Null
Age	Number	Not Null
Qualification	Char	Not Null
Cast	Char	Not Null
Family Background	Char	Not Null

<b>Table 4: Placement</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Name	Char	Not Null
Address	Char	Not Null
E-mail	Char	Not Null
Age	Number	Not Null
Qualification	Char	Not Null
Experience	Number	Not Null

<b>Table 5: Administrator</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Username	Char	Not Null
Password	Char	Not Null

<b>Table6: Lifestyle</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Books	Char	Not Null
Art and Culture	Char	Not Null
Fashion and Beauty	Char	Not Null

<b>Table7: News</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
News_India	Char	
News_World	Char	

<b>Table 7: Cricket</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Team	Number	Not Null
Country	Char	Not Null
Scheduling	Number	Not Null

<b>Table 7: business</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Economy	char	Not Null
Market	char	Not Null

Projecthelp.in

# DATA DICTIONARY

Projecthelpinc.com

## **7.1 DATA DICTIONARY**

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: -

Define the boundaries along which to break;

Determine into how many pieces to break; and

Identify the proper level of detail when design should stop and implementation should start.

Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain.

The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design activity begins when the requirements document for the software to be developed is available. This may be the SRS for the complete system, as is the case if the waterfall model is being followed or the requirements for the next "iteration" if the iterative enhancement is being followed or the requirements for the prototype if the prototyping is being followed. While the requirements specification activity is entirely in the problem domain, design is the first step in moving from the problem domain toward the solution domain. Design is essentially the bridge between requirements specification and the final solution for satisfying the requirements.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module.

The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact

on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

The input to the design phase is the specifications for the system to be designed. Hence, a reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc. The output of the top-level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. A reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

A design can be object-oriented or function-oriented. In function-oriented design, the design consists of module definitions, with each module supporting a functional abstraction. In object-oriented design, the modules in the design represent data abstraction (these abstractions are discussed in more detail later). In the function-oriented methods for design and describe one particular methodology the structured design methodology in some detail. In a function-oriented design approach, a system is viewed as a transformation function, transforming the inputs to the desired outputs. The purpose of the design phase is to specify the components for this transformation function, so that each component is also a transformation function. Hence, the basic output of the system design phase, when a function oriented design approach is being followed, is the definition of all the major data structures in the system, all the major modules of the system, and how the modules interact with each other.

Once the designer is satisfied with the design he has produced, the design is to be precisely specified in the form of a document. To specify the design, specification languages are used. Producing the design specification is the ultimate objective of the design phase. The purpose of this design document is quite different from that of the design notation. Whereas a design represented using the design notation is largely to be used by the designer, a design specification has to be so precise and complete that it can be used as a basis of further development by other programmers. Generally, design specification uses textual structures, with design notation helping in understanding.

## **7.2 SCHEDULING**

Scheduling of a software project does not differ greatly from scheduling of any multi-task engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification to software projects.

Program evaluation and review technique (PERT) and critical path method (CPM) are two project scheduling methods that can be applied to software development. Both techniques are driven by information already developed in earlier project planning activities.

### **Estimates of Effort**

- A decomposition of the product function.
- The selection of the appropriate process model and task set.
- Decomposition of tasks.

Interdependencies among tasks may be defined using a task network. Tasks, sometimes called the project Work Breakdown Structure (WBS) are defined for the product as a whole or for individual functions.

Both PERT and CPM provide quantitative tools that allow the software planner to (1) determine the critical path-the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical models; and (3) calculate "boundary times" that define a time window" for a particular task.

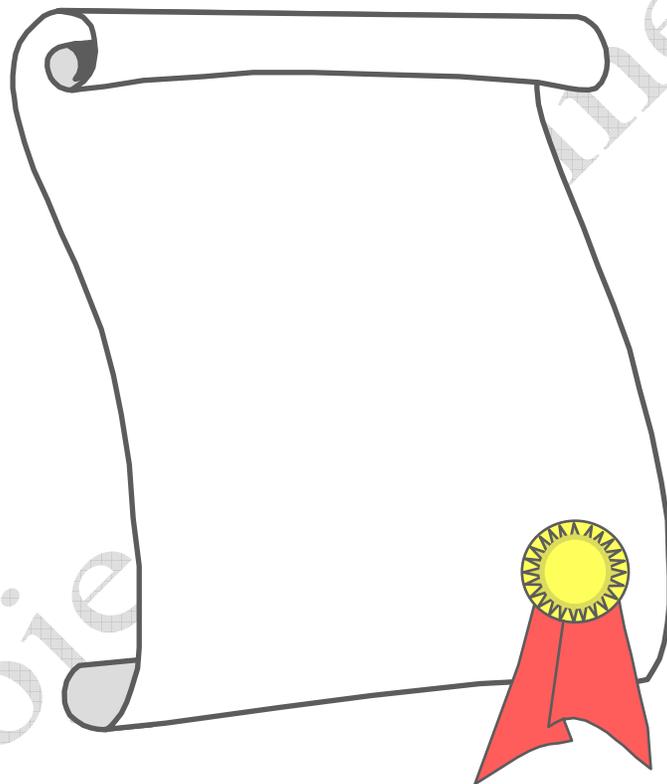
Boundary time calculations can be very useful in software project scheduling. Slippage in the design of one function, for example, can retard further development of other functions. It describes important boundary times that may be discerned from a PERT or CPM network: (1) the earliest time that a task can begin when preceding tasks are completed in the shortest possible time, (2) the latest time for task initiation before the minimum project completion time is delayed, (3) the earliest finish-the sum of the

earliest start and the task duration, (4) the latest finish- the latest start time added to task duration, and (5) the total float-the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path maintained on schedule. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as tasks are completed.

Both PERT and CPM have been implemented in a wide variety of automated tools that are available for the personal computer. Such tools are easy to use and take the scheduling methods described previously available to every software project manager.

Projecthelpline.in

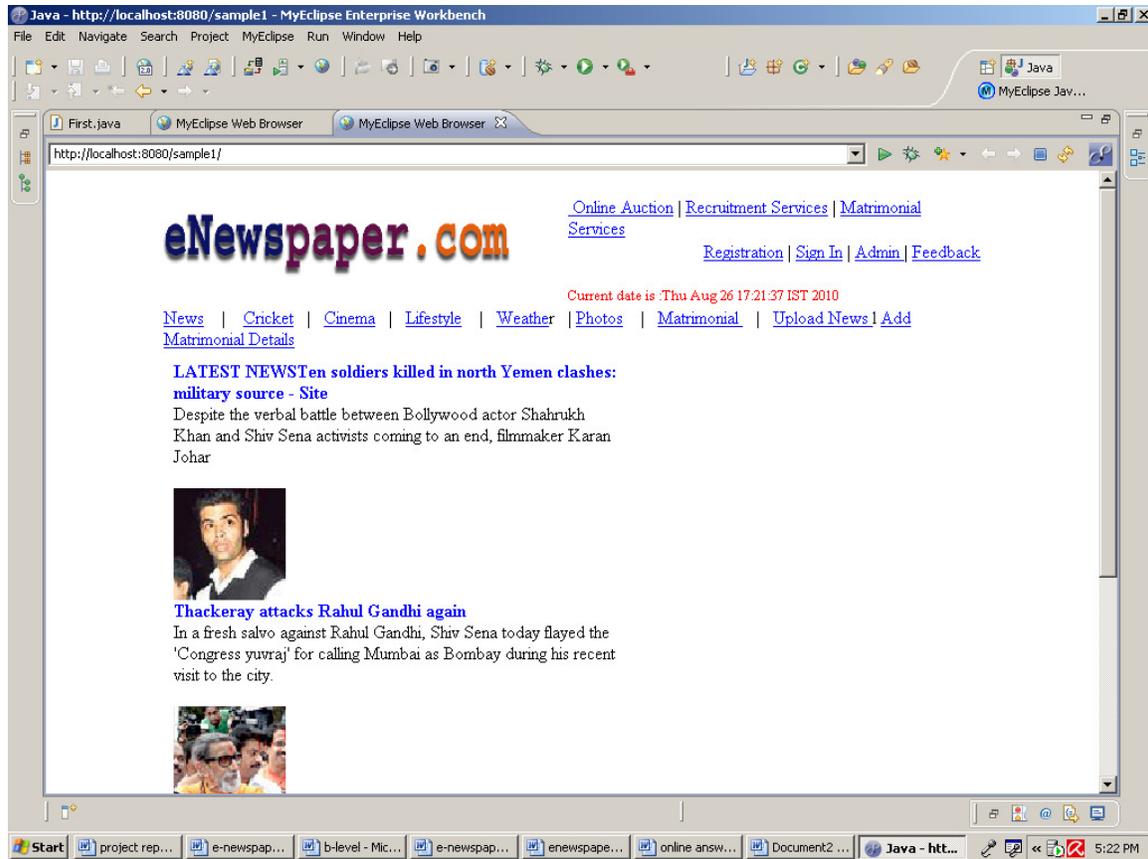
# SOURCE CODE



Proje  
me.in

## 8.1 SOURCE CODE

### Index.jsp



```

<% @ page language="java" import="java.sql.*" %>
<% @ page session="true" %>
<html>
<head>
<title>eNewspaper.com</title>
<style type="text/css">
<!--
.style2 {
    font-size: 16px;
    font-weight: bold;
    color: #990000;
}
.style3 {color: #0000FF}
-->
</style>
</head>
<body>
<form method="post" action="Log.jsp">

```

```

<table width="728" border="0" align="center" cellspacing="0">
<tr>
  <td colspan="2" height="154"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="2" align="left" valign="top">&nbsp;</td>
<td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
  <tr>
    <td height="230" align="left" valign="top"><p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p></td>
    <td width="398" rowspan="2" align="left" valign="top"><a href="products.jsp?cat=Jewellery
and Watches&type=Rings"></a><a href="products.jsp?cat=Mobile and
Accessories&type=Mobile"></a>
    <% Class.forName("com.mysql.jdbc.Driver");

    Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/sample","root","1234");
    Statement statement = connection.createStatement();

    String query="select * from headnews where category='General';
    ResultSet rs = statement.executeQuery(query);
    while(rs.next())
    {
    out.println("<b> <span class='style3'>" + rs.getString(2)+ "</span></b>");
    out.println("<br>" + rs.getString(3)+ "<br>");
    out.println("<br>");
    %>
    <br>
    <%
    }
    %>
  </td>
  <td align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td width="4" align="left" valign="top">&nbsp;</td>
  <td width="215" align="left" valign="top"><a href="products.jsp?cat=Jewellery and
Watches&type=Ladies Watch"></a></td>
</tr>
<tr>
<td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top"> <p>

```

```
<table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
<tr></tr><tr>

<td width="522"> </td>

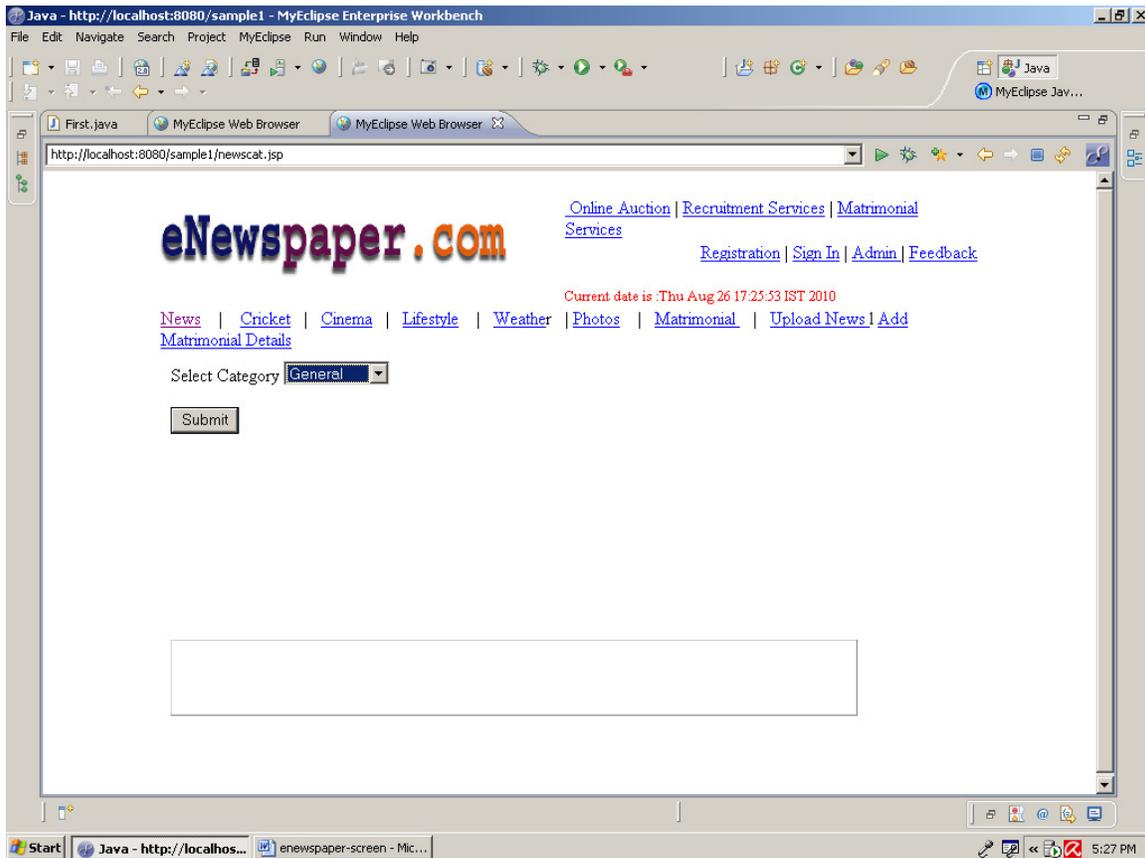
</table></td>
</tr>

<tr>
<td align="left" valign="top">&nbsp;</td>
<td align="left" valign="top">&nbsp;</td>
<td align="center" valign="middle">&nbsp;</td>
</tr>
</table>
<p>&nbsp;</p></td>
</tr>
</table>

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

Projecthelpline.in

## Newscat.jsp



```
<%@ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>My JSP 'newscat.jsp' starting page</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
</head>
```

```
<body>
```

```
<form method="post" action="allnews.jsp">
```

```

<table width="728" border="0" align="center" cellspacing="0">
  <tr>
    <td colspan="2" height="154"><%@ include file="header1.jsp" %></td>
  </tr>
  <tr>
    <td width="2" align="left" valign="top">&nbsp;</td>
    <td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
  <tr>
    <td height="230" align="left" valign="top"><p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p></td>
    <td width="398" rowspan="2" align="left" valign="top"><p><a
href="products.jsp?cat=Jewellery and Watches&type=Rings"></a><a
href="products.jsp?cat=Mobile and Accessories&type=Mobile"></a>

    Select Category
    <select name="select">
      <option>General</option>
      <option>Cricket</option>
      <option>Cinema</option>
      <option>Weather</option>
      <option>Fashion</option>
      <option>Matrimonial</option>
      <option>Classified</option>
    </select>
  </p>
  <p>
    <input type="submit" name="Submit" value="Submit">
</p></td>
  <td align="left" valign="top">&nbsp;</td>
</tr>
</tr>
<tr>
  <td width="4" align="left" valign="top">&nbsp;</td>
  <td width="215" align="left" valign="top"><a href="products.jsp?cat=Jewellery and
Watches&type=Ladies Watch"></a></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top"><p>
    <table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
      <tr></tr>
      <tr>
        <td width="522"></td>
      </tr>
    </table></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">&nbsp;</td>
  <td align="center" valign="middle">&nbsp;</td>
</tr>

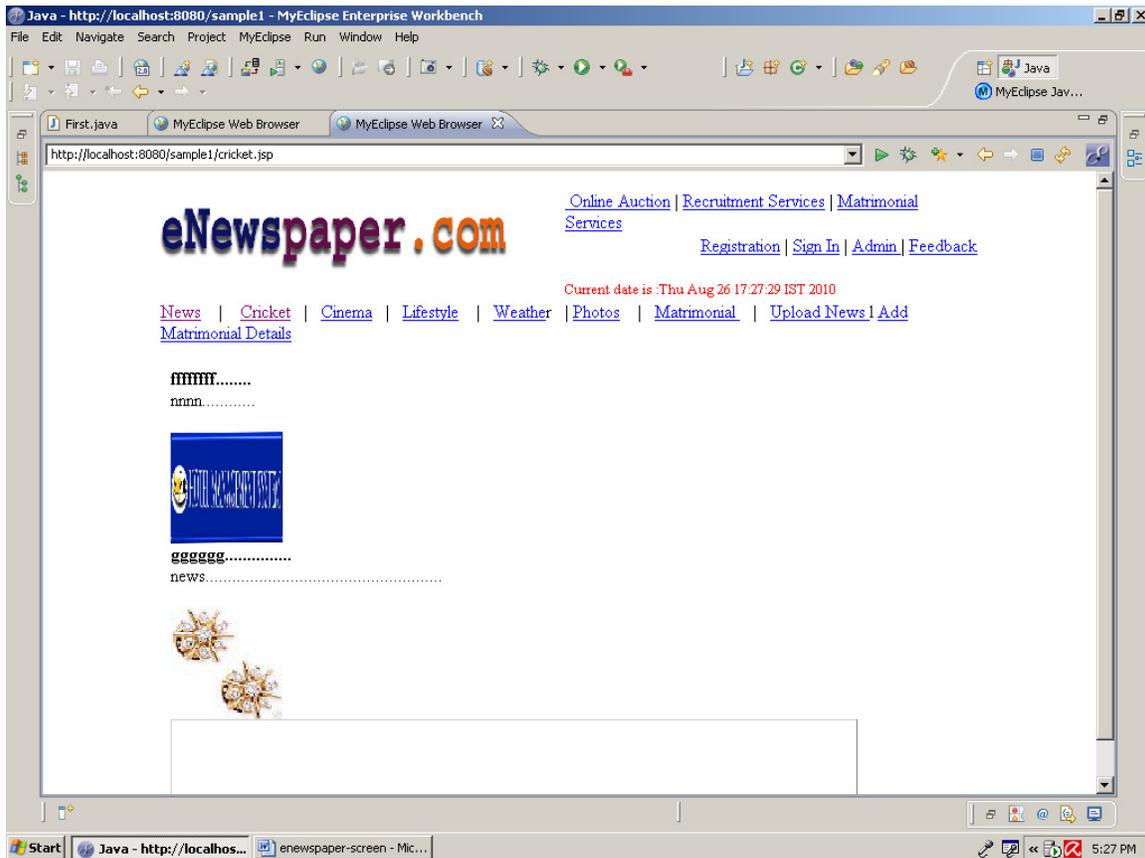
```

```
</table>
  <p>&nbsp;</p></td>
</tr>
</table>
</form>
<br>

</body>
</html>
```

Projecthelpline.in

## Cricket.jsp



```
<% @ page language="java" import="java.sql.*,java.util.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>enewspaper.com</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
<style type="text/css">
```

```
<!--
```

```
.style2 { font-size: 16px;
```

```
font-weight: bold;
```

```
color: #990000;
```

```

}
-->
</style>
</head>

<body>
<form method="post" action="Log.jsp">
<table width="728" border="0" align="center" cellspacing="0">
<tr>
<td colspan="2" height="154"><p>
<%@ include file="header1.jsp" %>
</p>
<p>
</p></td>
</tr>
<tr>
<td width="2" align="left" valign="top">&nbsp;</td>
<td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
<tr>
<td height="230" align="left" valign="top"><p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p></td>
<td width="398" rowspan="2" align="left" valign="top"><a
href="products.jsp?cat=Jewellery and Watches&type=Rings"></a><a
href="products.jsp?cat=Mobile and Accessories&type=Mobile"></a>
<% Class.forName("com.mysql.jdbc.Driver");

Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/sample","root","1234");
Statement statement = connection.createStatement();

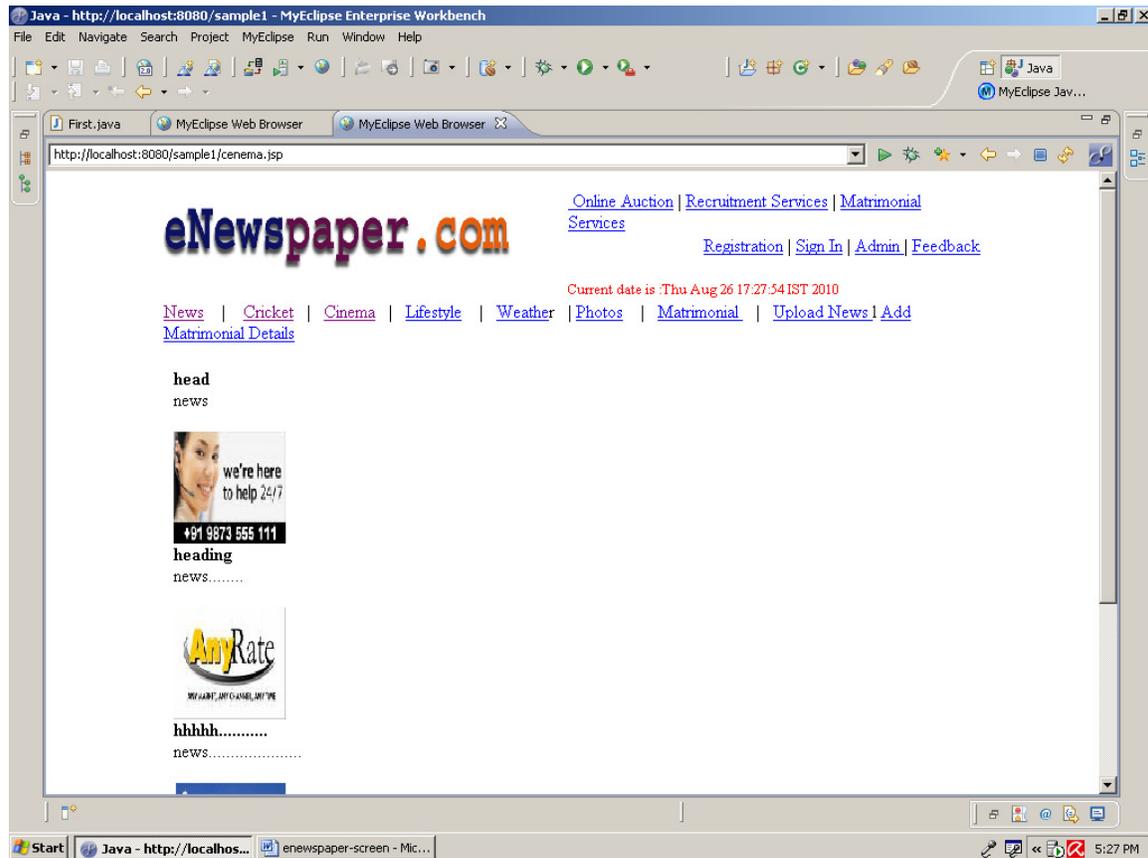
String query="select * from headnews where category='Cricket'";
ResultSet rs = statement.executeQuery(query);
while(rs.next())
{
out.println("<b> <span class='style3'>" + rs.getString(2)+ "</span></b>");
out.println("<br>" + rs.getString(3)+ "<br>");
out.println("<br>");
%>
<br>
<%
}
%>
</td>
<td align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td width="4" align="left" valign="top">&nbsp;</td>
<td width="215" align="left" valign="top"></td>

```

```
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top"><p>
    <table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
      <tr></tr>
      <tr>
        <td width="522"></td>
      </table></td>
    </tr>
  <tr>
    <td align="left" valign="top">&nbsp;</td>
    <td align="left" valign="top">&nbsp;</td>
    <td align="center" valign="middle">&nbsp;</td>
  </tr>
</table>
<p>&nbsp;</p></td>
</tr>
</table>
</form>
</body>
</html>
```

Projecthelppline.in

## Cinema.jsp



```
<% @ page language="java" import="java.sql.*,java.util.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>enewspaper.com</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
<style type="text/css">
```

```
<!--
```

```
.style2 { font-size: 16px;
```

```
font-weight: bold;
```

```
color: #990000;
```

```

}
-->
</style>
</head>

<body>
<form method="post" action="Log.jsp">
<table width="728" border="0" align="center" cellspacing="0">
<tr>
<td colspan="2" height="154"><p>
<%@ include file="header1.jsp" %>
</p>
<p>
</p></td>
</tr>
<tr>
<td width="2" align="left" valign="top">&nbsp;</td>
<td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
<tr>
<td height="230" align="left" valign="top"><p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p></td>
<td width="398" rowspan="2" align="left" valign="top"><a
href="products.jsp?cat=Jewellery and Watches&type=Rings"></a><a
href="products.jsp?cat=Mobile and Accessories&type=Mobile"></a>
<% Class.forName("com.mysql.jdbc.Driver");

Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/sample","root","1234");
Statement statement = connection.createStatement();

String query="select * from headnews where category='Cricket'";
ResultSet rs = statement.executeQuery(query);
while(rs.next())
{
out.println("<b> <span class='style3'>" + rs.getString(2)+ "</span></b>");
out.println("<br>" + rs.getString(3)+ "<br>");
out.println("<br>");
%>
<br>
<%
}
%>
</td>
<td align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td width="4" align="left" valign="top">&nbsp;</td>
<td width="215" align="left" valign="top"></td>

```

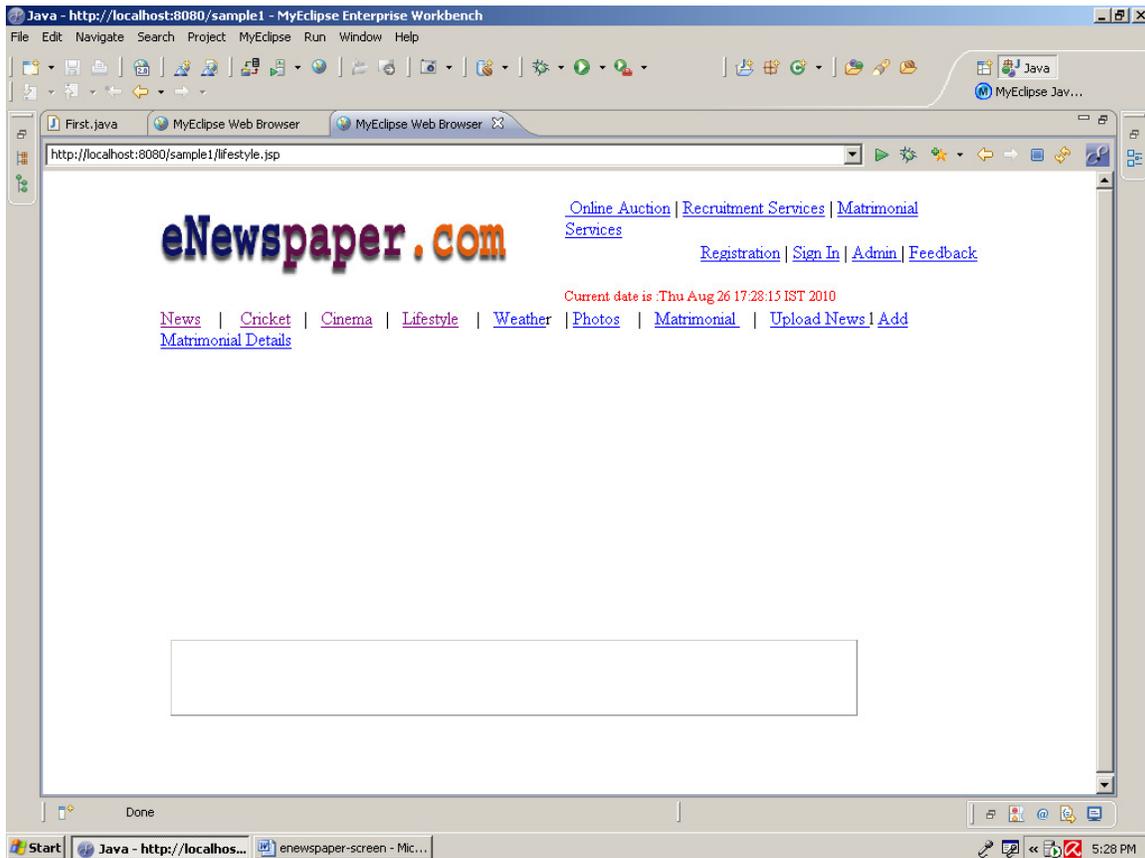
```

</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top"><p>
    <table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
      <tr></tr>
      <tr>
        <td width="522"></td>
      </table></td>
    </tr>
  <tr>
    <td align="left" valign="top">&nbsp;</td>
    <td align="left" valign="top">&nbsp;</td>
    <td align="center" valign="middle">&nbsp;</td>
  </tr>
</table>
<p>&nbsp;</p></td>
</tr>
</table>
</form>
</body>
</html>

```

Projecthelppline.in

## Lifestyle.jsp



```
<% @ page language="java" import="java.sql.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Life Style</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
</head>
```

```
<body>
```

```
<form method="post" action="Log.jsp">
```

```
<table width="728" border="0" align="center" cellspacing="0">
```

```

<tr>
  <td colspan="2" height="154"><%@ include file="header1.jsp" %></td>
</tr>
<tr>
  <td width="2" align="left" valign="top">&nbsp;</td>
  <td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
  <tr>
    <td height="230" align="left" valign="top"><p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p></td>
    <td width="398" rowspan="2" align="left" valign="top"><a
href="products.jsp?cat=Jewellery and Watches&type=Rings"></a><a
href="products.jsp?cat=Mobile and Accessories&type=Mobile"></a>
    <% Class.forName("com.mysql.jdbc.Driver");

    Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/sample","root","1234");
    Statement statement = connection.createStatement();

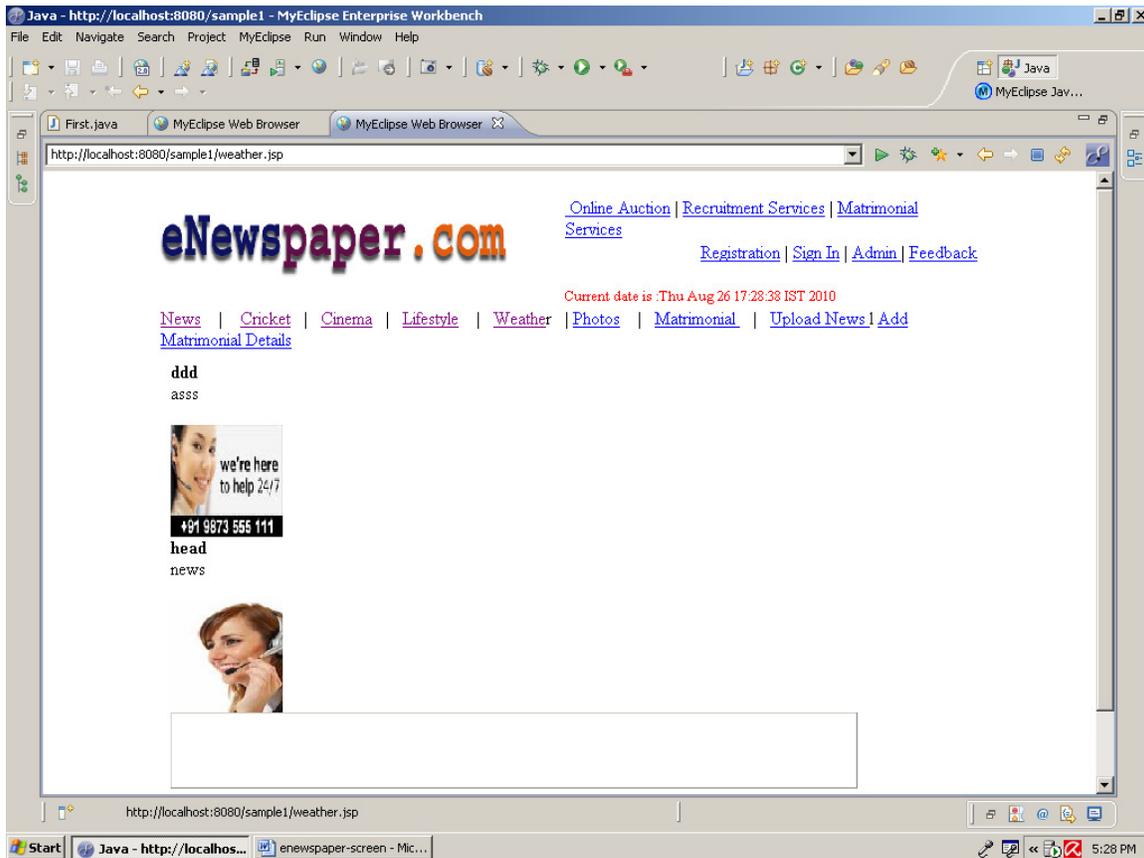
    String query="select * from headnews where category='Fashion'";
    ResultSet rs = statement.executeQuery(query);
    while(rs.next())
    {
    out.println("<b> <span class='style3'>" + rs.getString(2)+ "</span></b>");
    out.println("<br>" + rs.getString(3)+"<br>");
    out.println("<br>");
    %>
      <br>
    <%
    }
    %>
  </td>
  <td align="left" valign="top">&nbsp;</td>
</tr>
<tr>
  <td width="4" align="left" valign="top">&nbsp;</td>
  <td width="215" align="left" valign="top"><a href="products.jsp?cat=Jewellery and
Watches&type=Ladies Watch"></a></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top"><p>
    <table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
      <tr></tr>
      <tr>
        <td width="522"></td>
      </tr>
    </table></td>
</tr>
<tr>

```

```
<td align="left" valign="top">&nbsp;</td>  
<td align="left" valign="top">&nbsp;</td>  
<td align="center" valign="middle">&nbsp;</td>  
</tr>  
</table>  
<p>&nbsp;</p></td>  
</tr>  
</table>  
</form>  
<br>  
</body>  
</html>
```

Projecthelpline.in

## Weatherjsp



```
<%@ page language="java" import="java.sql.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Life Style</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
</head>
```

```
<body>
```

```
<form method="post" action="Log.jsp">
```

```
<table width="728" border="0" align="center" cellspacing="0">
```

```
<tr>
```

```

        <td colspan="2" height="154"><%@ include file="header1.jsp" %></td>
    </tr>
    <tr>
        <td width="2" align="left" valign="top">&nbsp;</td>
        <td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
    <tr>
        <td height="230" align="left" valign="top"><p>&nbsp;</p>
        <p>&nbsp;</p>
        <p>&nbsp;</p></td>
        <td width="398" rowspan="2" align="left" valign="top"><a
href="products.jsp?cat=Jewellery and Watches&type=Rings"></a><a
href="products.jsp?cat=Mobile and Accessories&type=Mobile"></a>
        <% Class.forName("com.mysql.jdbc.Driver");

    Connection connection =
    DriverManager.getConnection("jdbc:mysql://localhost/sample","root","1234");
    Statement statement = connection.createStatement();

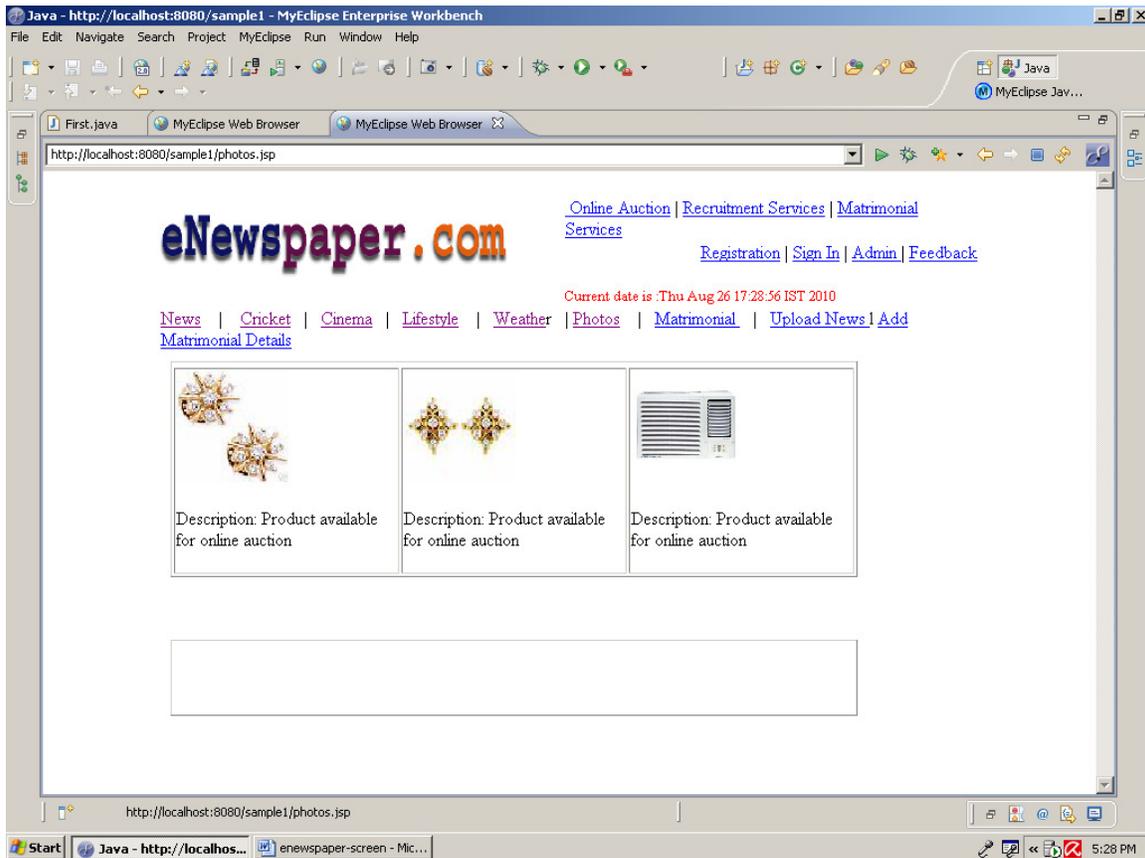
    String query="select * from headnews where category='Fashion'";
    ResultSet rs = statement.executeQuery(query);
    while(rs.next())
    {
        out.println("<b> <span class='style3'>" + rs.getString(2)+ "</span></b>");
        out.println("<br>" + rs.getString(3)+"<br>");
        out.println("<br>");
        %>
        <br>
        <%
    }
    %>
    </td>
    <td align="left" valign="top">&nbsp;</td>
</tr>
<tr>
    <td width="4" align="left" valign="top">&nbsp;</td>
    <td width="215" align="left" valign="top"><a href="products.jsp?cat=Jewellery and
Watches&type=Ladies Watch"></a></td>
</tr>
<tr>
    <td align="left" valign="top">&nbsp;</td>
    <td colspan="2" align="left" valign="top"><p>
        <table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
            <tr></tr>
            <tr>
                <td width="522"></td>
            </tr>
        </table></td>
</tr>
<tr>
    <td align="left" valign="top">&nbsp;</td>

```

```
<td align="left" valign="top">&nbsp;</td>  
<td align="center" valign="middle">&nbsp;</td>  
</tr>  
</table>  
<p>&nbsp;</p></td>  
</tr>  
</table>  
</form>  
<br>  
</body>  
</html>
```

Projecthelpline.in

## Photos.jsp



```
<% @ page language="java" import="java.sql.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>eNewspaper.com</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
</head>
```

```
<body>
```

```
<form method="post" action="Log.jsp">
```

```
<table width="728" border="0" align="center" cellspacing="0">
```

```

<tr>
  <td colspan="2" height="154"><%@ include file="header1.jsp" %></td>
</tr>
<tr>
  <td width="2" align="left" valign="top">&nbsp;</td>
  <td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
  <tr>
    <td height="230" align="left" valign="top"><p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p></td>
    <td width="398" rowspan="2" align="left" valign="top"><a
href="products.jsp?cat=Jewellery and Watches&type=Rings"></a><a
href="products.jsp?cat=Mobile and Accessories&type=Mobile"></a>
    <% Class.forName("com.mysql.jdbc.Driver");

    Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/sample","root","1234");
    Statement statement = connection.createStatement();

    String query="select * from photos";
    ResultSet rs = statement.executeQuery(query);%>
    <table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
<tr>
<% int i=0;
while(rs.next())
{
if(i%3==0)
{
%>
</tr><tr>
<% } %>

<td width="522">

  <p><img src='<%=rs.getString(3)%>' height="100" width="100" /> <br />
  <span class="style3"></span><br>Description: <span class="style3"><span
class="style4"><%=rs.getString(2)%></span><br />
  </span> <br />
  </td>

  <%

i=i+1;} %>
</table>

</td>

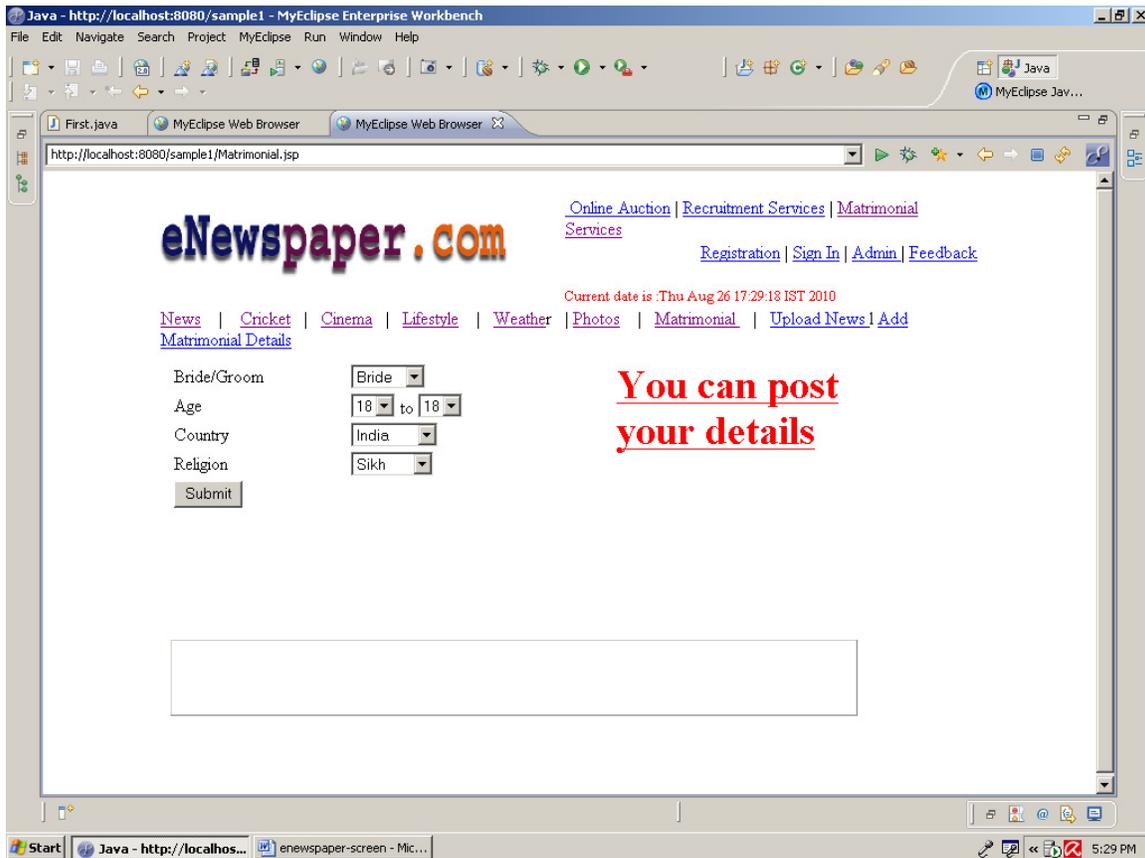
```

```

        <td align="left" valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td width="4" align="left" valign="top">&nbsp;</td>
        <td width="215" align="left" valign="top"><a href="products.jsp?cat=Jewellery and
Watches&type=Ladies Watch"></a></td>
    </tr>
    <tr>
        <td align="left" valign="top">&nbsp;</td>
        <td colspan="2" align="left" valign="top"><p>
            <table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
                <tr></tr>
                <tr>
                    <td width="522"></td>
                </tr>
            </table></td>
    </tr>
    <tr>
        <td align="left" valign="top">&nbsp;</td>
        <td align="left" valign="top">&nbsp;</td>
        <td align="center" valign="middle">&nbsp;</td>
    </tr>
</table>
    <p>&nbsp;</p></td>
</tr>
</table>
</form>
</body>
</html>

```

## Matrimonial.jsp



```
<% @ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Matrimonial</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
<style type="text/css">
```

```
<!--
```

```
.style1 {
```

```
color: #FF0000;
```

```
font-weight: bold;
```

```
}
```

```

-->
</style>
</head>

<body>

<form method="post" action="matrimonialsearch.jsp">
  <table width="728" border="0" align="center" cellspacing="0">
    <tr>
      <td colspan="2" height="154"><%@ include file="header1.jsp" %></td>
    </tr>
    <tr>
      <td width="2" align="left" valign="top">&nbsp;</td>
      <td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
        <tr>
          <td height="230" align="left" valign="top"><p>&nbsp;</p>
          <p>&nbsp;</p>
          <p>&nbsp;</p></td>
          <td width="398" rowspan="2" align="left" valign="top"><a
href="products.jsp?cat=Jewellery and Watches&type=Rings"></a><a
href="products.jsp?cat=Mobile and Accessories&type=Mobile"></a>

          <table width="396" border="0">
            <tr>
              <td width="154">Bride/Groom</td>
              <td width="178"><select name="select4">
                <option>Bride</option>
                <option>Groom</option>
              </select></td>
              <td width="50">&nbsp;</td>
            </tr>

            <tr>
              <td>Age</td>
              <td><select name="age" id="age">
                <option>18</option>
                <option>19</option>
                <option>20</option>
                <option>21</option>
                <option>22</option>
                <option>23</option>
                <option>24</option>
                <option>25</option>
                <option>26</option>
                <option>27</option>
                <option>28</option>
                <option>29</option>
                <option>30</option>
                <option>31</option>
                <option>32</option>
              </select></td>
            </tr>
          </table>
        </tr>
      </td>
    </tr>
  </table>

```

```

        <option>33</option>
        <option>34</option>
        <option>35</option>
        <option>36</option>
        <option>37</option>
        <option>38</option>
        <option>39</option>
        <option>40</option>
        <option>41</option>
    </select>
    to
    <select name="age1" id="age1">
        <option>18</option>
        <option>19</option>
        <option>20</option>
        <option>21</option>
        <option>22</option>
        <option>23</option>
        <option>24</option>
        <option>25</option>
        <option>26</option>
        <option>27</option>
        <option>28</option>
        <option>29</option>
        <option>30</option>
        <option>31</option>
        <option>32</option>
        <option>33</option>
        <option>34</option>
        <option>35</option>
        <option>36</option>
        <option>37</option>
        <option>38</option>
        <option>39</option>
        <option>40</option>
        <option>41</option>
    </select></td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>Country</td>
    <td><select name="country" id="country">
        <option>India</option>
        <option>Australia</option>
        <option>England</option>
        <option>London</option>
        <option>America</option>
        <option>Other</option>
    </select></td>
    <td>&nbsp;</td>

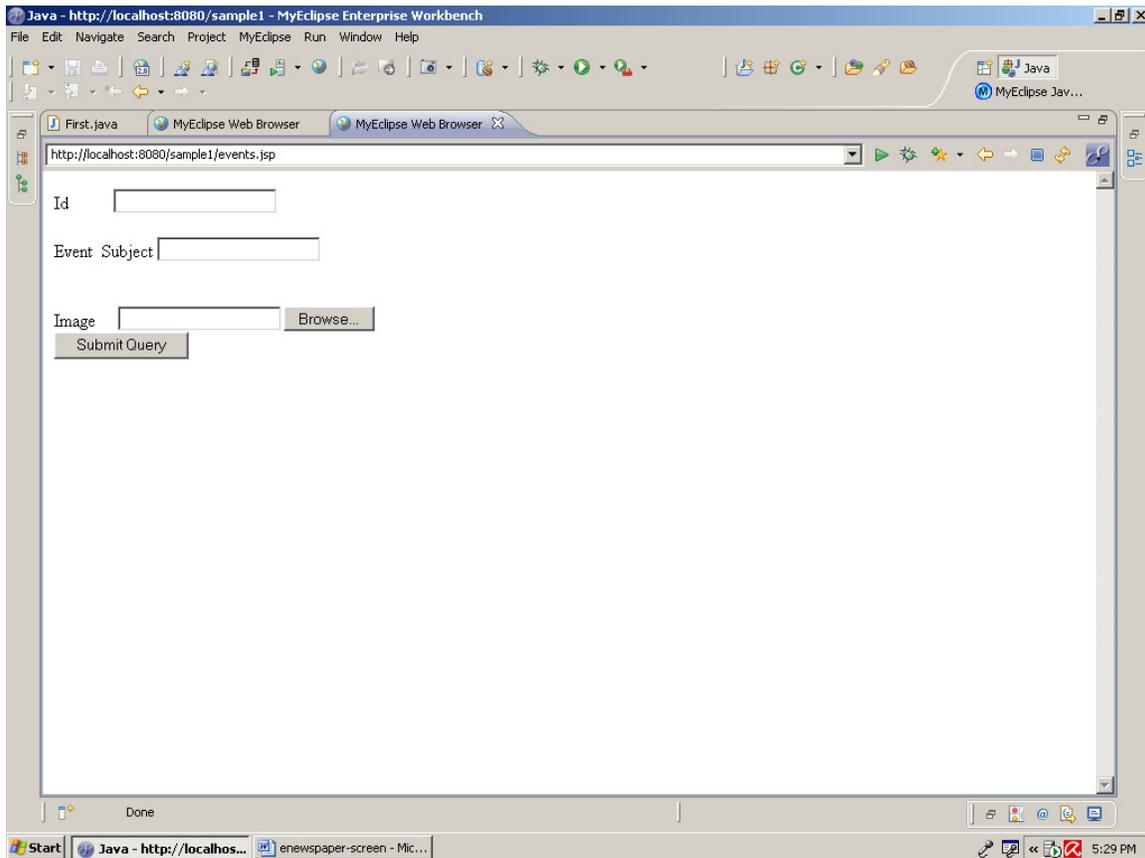
```

```

</tr>
<tr>
  <td>Religion</td>
  <td><select name="rel" id="rel">
    <option>Sikh</option>
    <option>Hindu</option>
    <option>Muslim</option>
    <option>Christen</option>
    <option>Punjabi</option>
    <option>Bengali</option>
    <option>Tamil</option>
  </select></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><input type="submit" name="Submit" value="Submit"></td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
</table></td>
<td align="left" valign="top"><a href="addmatrimonial.jsp" class="style1">You can
post your details</a> </td>
</tr>
<tr>
  <td width="4" align="left" valign="top">&nbsp;</td>
  <td width="215" align="left" valign="top"><a href="products.jsp?cat=Jewellery and
Watches&type=Ladies Watch"></a></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top"><p>
<table width="613" height="68" border="1" align="left" bgcolor="#FFFFFF">
  <tr></tr>
  <tr>
    <td width="522"></td>
  </tr>
</table></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">&nbsp;</td>
  <td align="center" valign="middle">&nbsp;</td>
</tr>
</table>
<p>&nbsp;</p></td>
</tr>
</table>
</form>
<br>
</body></html>

```

## Events.jsp



```
<%@ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>enewspaper.com</title>
```

```
<meta http-equiv="pragma" content="no-cache">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

```
<meta http-equiv="expires" content="0">
```

```
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
```

```
<meta http-equiv="description" content="This is my page">
```

```
<!--
```

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

```
-->
```

```
</head>
```

```
<body>
```

```
<form ENCTYPE="multipart/form-data" ACTION="uploadevents.jsp" METHOD="POST">
```

