Web Enabling a Bibliographic Database of Indian Biomedical Journals: IndMED

A Dissertation submitted for the partial fulfillment of M.S. *(Software Systems)* Degree.

By
Mr. Sukhdev Singh

http://www.geocities.com/esukhdev
esukhdev@hotmail.com

BITS ZG629T: DISSERTATION

Web Enabling a Bibliographic Database of Indian Biomedical Journals: IndMED

Work Carried at

National Informatics Centre, New Delhi.

by

Name : **Mr. Sukhdev Singh**
ID_No. : **2000HZ12435**

Dissertation submitted for the partial fulfillment of M.S. *(Software Systems)* Degree under the Supervision of ***Mr. Vinay Thakur***, *Principal Systems Analyst.*

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)
November 2001

CERTIFICATE

This is to certify that the Dissertation entitled, ***Web Enabling a Bibliographic Database of Indian Biomedical Journals: IndMED*** And submitted by **Mr. Sukhdev Singh** ID-No. **2000HZ12435** for the partial fulfillment of the requirements of M.S. (Software Systems) degree of BITS, embodies the work done by him/her under my supervision.

_____

Signature of the Supervisor

Date: _____

Mr. Vinay Thakur
Principal Systems Analyst
National Informatics Centre
CGO Complex, Lodhi Road,
New Delhi-110003

_____

Name, Designation & Organization

Contents

SUMMARY

This project aims to provide access through web to a bibliographic database called IndMED. The database is a product of Indian Medlars Centre (IMC, a division of National Informatics Centre) and has data relating to references to articles published in India's learned biomedical journals. The scope of this project is limited to providing a web-based user interface for database searching and developing the CGI for Web-Database Connectivity.

The project was planned to ensure its scheduled completion. A survey of relevant research and technologies pertaining to Web Interface, Web-Database Connectivity and DBMS have been made and briefly described in the dissertation. Interview, Questionnaire and Observation of similar systems were utilized to determine the requirements of the proposed system. Specification of the proposed system was then based on the requirements.

After the requirements and the specification of the proposed system was finalized an Analysis Model consisting of Data Flow Diagrams (DFDs) and Entity-Relationship Diagram (ERD) was developed. Design of the system was then developed for the sub-systems – Web-Interface, Web-database Connectivity and DBMS - so the requirements and specifications are met. The Web-Interface mainly consists of html pages i.e. Simple Search Form, Minimal Search Form and Advanced Search Form and the dynamically generated by CGI the Search Result Screen. The Web-Database Connectivity sub-system consists of PERL Scripts dealing with Processing of data submitted from html forms, Variables assignment, Error Trap, Formulation of query string which can passed on to the DBMS sub-system and Html Patching of the output of the database in response to the query.

The implementation of the system includes: the screens along with html code written for various search forms and generated by the CGI for the Web-Interface, Perl Scripts (IMSS.PL; IMAS.PL; IMRS.PL) for the CGI and the Database Structure, Fields included in Inverted File Index of the Database and Display Formats for the DBMS sub-system. The system thus developed was tested to meet the tests prescribed by the system specification. The final product has been integrated into the Home-page of IMC at the URL http://indmed.nic.in . A user help document for searching is also included. The system developed has scope for future research and refinement in regard to relevancy ranking, federated searching, links to full text articles and other material relevant to the user query available on the Internet along with the results from IndMED.

PREFACE

This project was an attempt to create a **Web-based User Interface** to a bibliographic database that will allow a user to conduct searches to retrieve relevant references to articles published in Indian biomedical journals. The bibliographic database – named IndMED - consists of references along with abstracts of articles published in Indian biomedical journals. A typical record of the database includes data items like Author(s), Address of the Author(s), Title of the Article, Name of the Journal, Volume of the Journal, Issue of the Journal, Pagination of the article, Abstract of the article and Keywords representing the theme of the article.

Common Gateway Interface (**CGI**) has been developed to link Web Interface to the bibliographic database. This allows user to enter search words and choose search options through search Web forms. Submit buttons of these forms activates various scripts to extract data from the database and present it in HTML. PERL 5 has been used to develop server-side scripts for this purpose.

The Web Interface has been tested on three different Internet Browser namely on Internet Explorer, Netscape Communicator and Opera so as to ensure cross browser compatibility of the interface. The interface has been presented to a group of users for testing and their feed back The PERL scripts also have tested and all the known errors has been rectified. The system developed has been integrated to the official web site of Bibliographic Informatics Division (Indian Medlars Centre), National Informatics Centre, New Delhi at the URL http://indmed.nic.in .Web based user guide has also been developed to help the users in using the interface.

For the purpose of citing references, Harvard Style (author-date system) has been followed in the dissertation. In this style, references in the text generally requires only the name of the author(s) and the year of publication (and specific page(s) if necessary). The end of the report contains bibliography identifying each item in sufficient detail listed alphabetically.

Chapter One:

INTRODUCTION

1. Overview

This report will discuss a number of issues pertinent to the creation of a web based interface. The report is broken up into a number of chapters, which discuss the following topics:

1. Chapter One provides an introduction to the report and project. Included in this chapter is a discussion of the reasons for attempting this project, the operational environment of the proposed system and the benefits to be gained.
2. Chapter Two gives a brief explanation of the spectrum of relevant research and technologies used for the project and development of the system. These are presented under three sections: Interface, CGI and DBMS.
3. Chapter Three details the plan for ensuring the timely completion of the project and the dissertation.
4. Chapter Four details the methodology used for collecting the requirements of the system. Then a set of requirements and specifications had been laid down for the project and the system to be developed.
5. Chapter Five details the analysis model of the system in terms of DFDs and ERD.
6. Chapter Six highlights the design of the system in terms of its sub-systems.
7. Chapter Seven is the lengthiest of all and includes the implementation.
8. Chapter Eight deals with Testing and Integration of the final product into the home page of IMC.
9. Chapter Nine reproduces the Help Document for the Users to search in IndMED from the Web-based interface.
10. Chapter Ten is the concluding chapter giving some future directions to research and improvement to the system developed.
11. The Dissertation ends with a bibliography given according to the Harvard Style of Referencing.

2. Operational Environment

2.1 National Informatics Centre

National Informatics Centre (NIC) (National Informatics Centre, 2001, http://home.nic.in), is a premier Information Technology organization in India which is committed to providing state-of-the-art, solutions for the IT needs of the Government of India at all levels. NIC carries the distinction of being the largest IT

Organization in the Country and has set up a satellite based nationwide computer communication network, called NICNET, with over 1400 nodes connecting the National Capital, the State Capitals and the District Headquarters to one another. The IT services of NIC range from Consultancy, Software Design & Development, Office Automation and Networking Services to Training, Video Conferencing, CAD, EDI, Multimedia and Internet Services including Web Site Development and Hosting. NIC has a nationwide presence, with its offices spread all across the Country, from Leh to Andaman & Nicobar Islands.

The organizational set up of NIC encompasses its Headquarters at New Delhi, State Units in all the 29 State capitals and 6 Union Territory Headquarters and District centres in more than 540 Districts of India. The Organization employs a large pool of efficient technical manpower. At the NIC Headquarters, a large number of Application Divisions exist which provide total Informatics Support to the Ministries and Departments of the Central Government. NIC computer cells are located in almost all the Ministry Bhawans of the Central Government and Apex Offices including the Prime Minister's Office, the Rashtrapati Bhawan and the Parliament House. Apart from this, NIC has various Resource Divisions at the Headquarters which specialize into different areas of IT and facilitate the Application Divisions as well as other NIC Centres in providing state-of-the-art services to the Govt. At the State level, NIC State Centres provide informatics support to their respective State Government and at the District level lie the NIC District Informatics Offices which strive hard to provide effective informatics support to the Development, Revenue and Judiciary administration of the District. In order to promote Information Technology in the country, focussing on INTERNET Technology based Value-added Services in the area of Industry, Business and Commerce, NIC has also established 22 National Information Technology Promotional Units (NITPUs) at major industrial/ Commercial cities viz.- Ahmedabad, Bangalore , Bhopal, Bhubaneswar, Calcutta, Chandigarh Chennai, Guwahati, Hyderabad, Jaipur, Kanpur, Kochi, Lucknow, Ludhiana, Mumbai, Nagpur, Panipat, Patna, Pune, Shimla, Tripur and Vishakapatnam. These centres provide Video-conferencing facilities, EDI services and other INTERNET Technology based Services over NICNET.

2.2 Indian Medlars Centre

National Informatics Centre (NIC) and Indian Council of Medical Research (ICMR) (http://www.icmr.nic.in) jointly set up the *ICMR-NIC Centre for Biomedical Information* in 1986. The primary goal of the Centre was to meet the information needs of the medical professionals in India from the MEDLARS (Medical Literature Analysis and Retrieval Systems) databases of United States National Library of Medicine (NLM), USA (http://www.nlm.nih.gov). The Centre provides information from the MEDLARS databases as well as various biomedical resources, accessible over Internet, to the users in the country. In 1990, the Centre came to be recognized as the 17th International MEDLARS Centre (IMC) and now is known as Indian Medlars Centre (http://indmed.nic.in).

3. IndMED: What, Why and How?

3.1 What

IndMED is a bibliographic database. *"**Bibliographic databases** are text databases that contain references to journal articles and sometimes chapters in books and conference proceedings. References are usually supported by abstracts or summaries of the contents of the article to help you gain a clear idea of the information that each article contains."* (Bournemouth University, http://www.bournemouth.ac.uk/library_catalogue/html/biblio.html). A bibliographic record usually have information like Author(s), Address of the Author(s), Title of the Article, Name of the Journal, Volume of the Journal, Issue of the Journal, Pagination of the article, Abstract of the article and Keywords representing the theme of the article. A bibliographic database is characterized by having variable number of fields and variable record length.

3.2 Why

One of the most common tasks in any research project is to perform complete search of the literature to determine what, if anything has been written on the topic before. One of the reasons is to make certain that one is not repeating someone else's work. Moreover literature review is necessary not only to a researcher but also for the practitioner who wishes to develop their practice based on research evidence. It systematically gathers together, critically appraise and summarize research findings relating to a particular topic. Each individual piece of research literature included in the review is studied in order to assess whether its conclusions are supported by the evidence presented, and are therefore trustworthy. Finally the review findings are related to current practice.

For the researcher, the literature review process is critical in determining the research question since examining other's work in the area will show what has and has not been researched, what questions have and have not been answered. The research methods to be used may also be influenced, for example, when the literature review indicates a standard approach taken by experts in the field.

The practitioner (nurse, health visitor, midwife, paramedic, etc.) may undertake a literature review for very different reasons, not initially related with research. For example, an area of practice may be of concern, causing the practitioner to turn to the research literature in search of evidence of what is best practice. This situation may arise as the result of an audit, and ultimately end up as a piece of research when the literature review reveals a gap in knowledge.

Regardless of the motivation for carrying out a literature review the process should be the same, the key being that a systematic approach is taken at every step of the way. The most common method of disseminating research findings is through publishing research articles (papers) in journals. As a result, the majority of references included in a literature review will be journal articles.

The easiest way to locate literature in the topic area is by searching the many electronic bibliographic databases. These databases contain listings of the references (and possibly abstracts) from selected research journals, accumulated over the years. They are normally available on CD-ROMs, on-line and on the Internet. The mostly

commonly used databases by a biomedical researcher/practitioner are: Medline (Index Medicus online) and EMBASE (Exerpta Medica online) These two databases combined provide coverage of the literature in many health care areas, but they do not record all publications from all medical journals.

Now let us look from the angle of the authors of the research publications (journal articles etc.). Impact of the ideas conveyed by the authors is directly related to amount of exposure of their publications to the target audience. The inclusion of the publications in bibliographic databases provides tremendous and long lasting exposure. This is because; bibliographic databases are very important tool for locating the literature in a particular area. The greater the exposure the greater are the chance of their works being cited and recognized.

Indian contribution in the areas of biomedical research and health care has been significant and conforming to international standards. However, only a small fraction of it is available for reference through international bibliographic databases. One of the factors could be that third world countries, including India, are facing the predicament of neglect of biomedical research, irrespective of its value to mankind. The issue has been distinctly addressed in the 1st November 1997 issue of "New Scientist" under the editorial page (It's not what you know - If there are biases in scientific publication editors must take the blame, 1997). and one of the articles entitled "The price of prejudice" (Day, M. 1997). The ICMR-NIC Centre for Biomedical Information, better known as Indian MEDLARS Centre (IMC), decided to put efforts to provide greater exposure to Indian biomedical research literature. To achieve this objective, a bibliographic database has been designed and developed on the lines of MEDLINE database (the largest and the comprehensive of the MEDLARS databases family). It is a bibliographic database and covers articles published in from Indian biomedical journals. To start with 75 prominent Indian journals have been selected to build up the database. More journals would be added to the list as their quality improves in coming years. This database has been named IndMED to reflect its scope of covering Indian biomedical literature. It is proposed to cover the journals from 1985 onwards in the database.

3.3 How

IMC has been making efforts to build bibliographic databases from biomedical journals published in India. An initial effort to this was the database on Neurology, developed by IMC in 1989. In 1993 a database on Population was compiled by IMC and in 1995 Tuberculosis Research Centre (TRC), Chennai was assigned a project to index articles from journals pertaining to "Tuberculosis & Allied Chest Diseases". Another project was given to Kidwai Memorial Institute of Oncology (KMIO), Bangalore to index articles from journals related to "Oncology". A common database structure based on the “Common Communication Format, 2nd Edition” (Hopkinson, A. & Simmons, P. 1988) was used for these databases. CDS/ISIS Version 3.07 (UNESCO, 2001) was used as the DBMS for these databases. Later it was decided to include more journals from other subject areas as well and club all these databases into one national database called IndMED. A committee of experts examined the biomedical journals published in India and selected journals to be included in IndMED. Journals already indexed in MEDLINE & other international databases were not included in this database to avoid duplication.

At presently 75 Indian biomedical journals are indexed in IndMED. The indexing work involves the articles be read by human experts; analyzed for key concepts; standard keywords assigned according to MeSH (United States National Library of Medicine, 2001); Bibliographic details rendered according to International Standards and keywords recorded in the standard prescribed data sheets.

4. Scope and Goals

The scope of this project is limited to make IndMED accessible through on Internet through World Wide Web (WWW) and develop suitable user interface permitting searching and display of records of this database. The following are specific goals that are to be achieved.

4.1 Web-Database (IndMED) Connectivity

To make a database available through Web suitable Web-database connectivity is required. Here this would be achieved by development of suitable server-side CGI scripts in PERL to link up the user interface and the IndMED database. These scripts will take the queries from the user interface and query the database. Suitable **HTML elements** would also be generated along with data. These would enable the user's browser to display the output properly.

4.2 Web-based User Interface

A **Web-based User Interface** to IndMED database needs to be developed. This would allow a user to conduct searches to retrieve relevant references to articles published in Indian biomedical journals. It would allow a user to perform **simple full text searching** as well as **qualified searching** for precise retrieval. Moreover this interface would also allow an advanced user to conduct **advanced searching** using **Boolean operators**.

The User Interface would provide various options to the users to select various **display formats** having different levels of details of the references retrieved. The user should also be able to specify the maximum **number** to records to be displayed. The executed query would also be displayed at the end. This would enable the user to modify and resubmit it if the user so desires.

Chapter Two:

RELEVANT RESEARCH AND TECHNOLOGIES USED

1. Interface

1.1 WWW

The *World Wide Web* is a widespread information system that is easy and inexpensive to use. Most people find using a web browser straightforward and unintimidating. Web browsers are easy to install and are beginning to be packaged as a standard component of some operating systems (Ford, A., 1995). There are many graphical web browsers currently in use. Some of the more popular browsers include Netscape, Internet Explorer and Opera.

The World Wide Web allows users to access information via hypertext documents that contain links to more documents. The web's hypermedia system also allows users to send and receive data requests for text, image, sound or digital video (Wilson, S., 1995).

Web browsers, in addition to providing the networking functions to retrieve documents, are also HTML formatters. When one load an HTML document into a browser it reads, or parses, the HTML information and formats the text and images on the screen. However different browsers may present the same document in different ways. This inconsistency must be taken into consideration when designing web pages particularly pages using the more advanced features such as tables, frames, image maps and forms.

The web is available virtually on all current computer platforms. It is also hardware-independent. As long as the information being sent is formatted according to the current standards it does not matter what type of computer is doing the sending or receiving (Wilson, S., 1995). To achieve this compatibility each communicating computer must have the capability to understand the TCP/IP format.

There are a number of advantages to using the web as the access point for an interface.

1. Text and graphics can be laid out on "pages" of the screen and titles, photographs, captions can be inserted.
2. Hypertext links can be used to retrieve online information. The web can provide access to any computer on the Internet. Web search programmes are also available for locating information.
3. The web browser provides a means for looking at the information retrieved.
4. Forms can be used to request or provide information.
5. Other applications or media may be easily incorporated in the web page e.g. audio clips, video clips, ftp, telnet and gopher facilities.
6. Information can be quickly and easily kept current.
7. The web supports the forms of information that were already in existence on the Internet - FTP, Gopher, WAIS, telnet and so on.
8. The web is interactive.
9. HTML documents are device-independent.
10. HTML is small and simple-to-learn.

11. Many freeware and shareware programmes are available for editing HTML files.

The web's ease of use, updating capabilities, widespread use by the academic and general population and availability means that it is a perfect forum for a graphical interface to bibliographic databases. The formatting capability of the web can be used to advantage to produce attractive pages.

The web is still in a state of rapid evolution as people are constantly discovering new ways to use it.

1.2 HTML

A web page is a document designed for viewing on a computer screen whilst using a browser such as Netscape or Internet Explorer. The document is written using *Hyper Text Markup Language*. HTML is not a programming language but a set of tags written into the document text to provide formatting instructions to the browser. It is based on SGML - the Standard Generalized Markup Language.

The hypertext part of HTML comes from the special tags that set up links within and between documents. Click on a hypertexted link and the browser will provide whatever information is at the end of the link. This information may be in the same document, on the same server or at a different geographical location altogether. If the link is to a CGI programme, the server does some data processing and computing before it returns the results (Muelver, J., 1996).

HTML started out as a simple markup language but then became more complicated with each new version. Unfortunately not all browsers support some of the more dazzling enhancements and not all browsers are graphical. Some enhancements are also browser specific particularly the Netscape enhancements. For this reason care must to be taken when designing web documents. There should not be a heavy reliance on some of the non-standard features. The use of these enhancements may look spectacular when viewed through the appropriate browser but a complete disaster when viewed through a different browser.

This project will take advantage of HTML's capability to format forms. A form on a web page is a collection of input fields that communicates its contents back to the server. All browsers support forms and display them to varying degrees (Muelver, J., 1996). The basic idea behind a form is simple. The visitor fills in the form, the form information is passed to the server and the CGI then process that information.

The data from the form is sent, along with the URL, in a special format called *URL encoding*. The encoding is necessary to make all non-printing characters in the data visible to the server. This coding also prevents the server from confusing the meaning of certain characters. The URL-encoded data is then translated back into readable text at the other end of the transmission (Muelver, J., 1996).

The browser sends form data to the server by using one of two methods (Muelver, J., 1996):

1. **get** - the form data is tacked on to the end of an URL request. The server passes the encoded string to the CGI for processing. The encoded string goes through the environmental variable, QUERY_STRING. The GET method is workable but limited as servers have length limits for URLs and environment variables. If the form contains long field names and values then the length limit may be reached before the end of the data. Because of this data length limit most forms use the post method.

2. **post** - can handle much longer coded strings of data. It encodes the form data the same way but sends the data directly to the CGI through STDIN. The server's CGI recognises that this add-on is a post and uses a different handling technique. The use of environmental variable, CONTENT_LENGTH, informs the server of how many bytes to read from STDIN. Thus the form data can be as long as required.

Form data reports on each field in the form. Each field has a NAME attribute and a VALUE attribute which are sent in pairs by the browser. The name/value pair is the basic data construct for forms and forms-processing programmes.

2. WEB-DATABASE CONNECTIVITY

2.1 CGI

The *Common Gateway Interface* emerged as the first way to present dynamically generated information on the web. It allows the computer to generate web pages instantly at the user's request (Gundavaram, S., 1996). Virtual or dynamic document creation is at the heart of CGI. It is still the most stable and well-understood method for creating such pages.

The Common Gateway Interface is a doorway between a web page and the server on which the page is stored. Gateway scripts enable the reader to interact with the web page. The URL points to the gateway script and the browser requests the URL from the server. The server receives the request and executes the script. The script performs some action and then formats its result in a manner that the web server can understand. The web server passes the result back to the browser, which formats and displays it for the reader (Lemay, L., 1995). The normal response provided by a CGI programme takes the form of another HTML page. This is frequently constructed in an ad-hoc manner to meet the needs of the requester (Muelver, J.,1996).

CGI programming is relatively simple and can be written in any language, although the most common language is Perl. Perl contains many powerfull features and is easy for the novice programmer to learn. Some other popular languages for CGI programming include AppleScript, C/C++, C Shell, Tcl and Visual Basic (Gundavaram,S., 1996).

One of the most common uses of the CGI is for processing forms. Forms allow the user to supply or request information. The interface of the proposed system would include forms that would allow the user to enter their search term(s). The user will enter the term or terms they wish to search on, then activate the submit button. This information will be passed to the CGI script, as a name/value pair, where it will be parsed and processed. The results from this processing will be sent back to the server and displayed by the browser. The CGI scripts would be written in Perl.

2.2 Perl

Perl is an acronym - *Practical Extraction and Report Language*. Perl was designed by Larry Wall, then given away as freeware (under some conditions). Larry Wall has continued to support, develop and enhance his creation (Muelver, J.,1996)
Perl is an interpreted language . It has the power and flexibility of a high-level programming language but does not require a special compiler and linker to execute

the programme (Till, D., 1995). Programming in Perl involves writing an instruction or a set of instructions (script) which consists of a sequence of statements and declarations. The script is written and made executable. The script can then be executed by a call from the command line. One of the key advantages of Perl is that it is platform-independent. It has been ported to most computer operating systems. Thus Perl scripts will run on any computer that has a Perl interpreter (Muelver, J.,1996).

Perl was originally designed for scanning text files, extracting information from these files and printing reports based on that information. HTML form data is passed as an URL-encoded string. Perl is capable of decoding this data into a useable format, processing the decoded information and formatting all the new data into a report with the necessary HTML tags. The page is then passed to the server which passes the page back to the requesting browser for display (Muelver, J.,1996).
Perl software is available free of charge as part of the Free Software Foundation's GNU project.

2.3 ActivePerl

ActivePerl is a port of core Perl to Windows. In this project it is this this flavor of the PERL that has been used. It is available from the ActiveState home page. (ActiveState Corporation, 2001, http://aspn.activestate.com/ASPN/Perl/).

2.4 CGI.pm: - a Perl5 CGI Library

"It is a PERL 5 Library. This perl 5 library uses objects to create Web fill-out forms on the fly and to parse their contents. It provides a simple interface for parsing and interpreting query strings passed to CGI scripts. However, it also offers a rich set of functions for creating fill-out forms. Instead of remembering the syntax for HTML form elements, you just make a series of perl function calls. An important fringe benefit of this is that the value of the previous query is used to initialize the form, so that the state of the form is preserved from invocation to invocation." (Stein L. 2001, http://stein.cshl.org/WWW/software/CGI/)

2.5 Windows NT

Windows NT is a Microsoft Windows personal computer operating system designed for users and businesses needing advanced capability. NT's technology is the base for the Microsoft successor operating system, Windows 2000. Windows NT (which may originally have stood for "New Technology," although Microsoft doesn't say) is actually two products: Microsoft NT Workstation and Microsoft NT Server. **Windows NT Server** is probably the second most installed network server operating system after Novell's NetWare operating system. Microsoft claims that its NT servers are beginning to replace both NetWare and the various UNIX-based systems such as those of Sun Microsystems and Hewlett-Packard.

This project uses Windows NT Server with IIS.

2.6 IIS

IIS (Internet Information Server) is a group of Internet servers (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with

additional capabilities for Microsoft's Windows NT and Windows 2000 Server operating systems. IIS is Microsoft's entry to compete in the Internet server market that is also addressed by Apache, Sun Microsystems, O'Reilly, and others. With IIS, Microsoft includes a set of programs for building and administering Web sites, a search engine, and support for writing Web-based applications that access databases. Microsoft points out that IIS is tightly integrated with the Windows NT and 2000 Servers in a number of ways, resulting in faster Web page serving.

3. DBMS

DBMS stands for DataBase Management System. It simple terms it is a software that is used to create, manage and uses a database. This project uses **CDS/ISIS** as DBMS. Version meant for PCs is called Micro ISIS. *"Micro CDS/ISIS is an advanced non-numerical information storage and retrieval software developed by UNESCO in 1985 basing the mainframe version of CDS/ISIS started in the late '60s. This aims to satisfy the need of many institutions, especially in developing countries, to streamline their information processing activities by using modern and relatively inexpensive technologies. One of the major advantages of using CDS/ISIS is its capabilities to manipulate an unlimited number of databases each of which may consist of completely different data elements.*

The major features of this software include: handling of variable length records, fields and sub fields; handling of repeatable fields; information retrieval using powerful search language; free text searching; a data interchange using ISO 2709; an integrated application for programming language in CDS/ISIS Pascal and ISIS_DLL; functions allowing users to build relational database and powerful hypertext functions to design complex user interfaces." (Varun V.K., 2001).

"…CDS/ISIS is not based over a relational model" (UNESCO, 2001, http://www.unesco.org/webworld/isis/isis.htm). It can best be described as Text Retrieval Systems or "Free Form" Databases. In some ways they are at the opposite end of the spectrum to a relational database. *"A "Free Form" database is the better term since there is little structure imposed on the designer of these systems. …[Designers] can define their own record structures. The search "engine" then indexes every field in an inverted index structure so that any string of data may be used to retrieve a record. The search sophistication possible with such indexes is enormous - Boolean combinations between terms both within and across fields are simply done and the typical text retrieval functions of adjacency, truncation, set manipulation etc are all built in. …The main advantage is the power of search and retrieve functions. Where there are large numbers of technical reports to be searched for information in an unpredictable way then a text retrieval engine is required since its efficiency and flexibility of indexing mean that you can index the "full text" of the report. A text retrieval engine verges on textual analysis."* (Biblio Tech Review, 2001, http://www.biblio-tech.com/html/databases.html). Further according to Deco C and others (Deco, C., Bender C. & Crespo F., 1995) a relational model is not adequate for the necessary type of searches on a textual database. Although some relational systems have included functions such as word indexes, these do not solve specific problems about queries put to a text retrieval system, such as the case of adjacent words.

Method used for providing access to Database.

The current version of CDS/ISIS for MS DOS is 3.08 and for Windows is 1.4. Versions are also available for UNIX/LINUX. (UNESCO, 2001,

http://www.unesco.org/webworld/isis/isis.htm). As such it do not have build-in support for publishing databases over Internet, however Varun (Varun VK, 2001) has surveyed various attempts for the same using third party technologies. This project would put up IndMED database under this DBMS by using a different (and new) approach using a combination PERL and an utility called “MX”(from a set of CISIS Utilities) from BEREME (BEREME, 2001, http://www.bireme.br/).

Chapter Three:

PLANNING PROCESS

This project had several stages. A list of these is given as under:

- Requirements and Specification
- Planning
- Research
- Analysis of the system
- Design of the system
- Implementation
- Testing and Debugging
- Preparation of User Support Document
- Integration into IMC's Webpages.

1. Planning

The dissertation had to be submitted by 10th November 2001. Completing the Project in time and writing dissertation without a proper plan could had been difficult. Planning was very essential to have well specified targets and goals and it helped in keeping track of status of the project.

1.1 Method

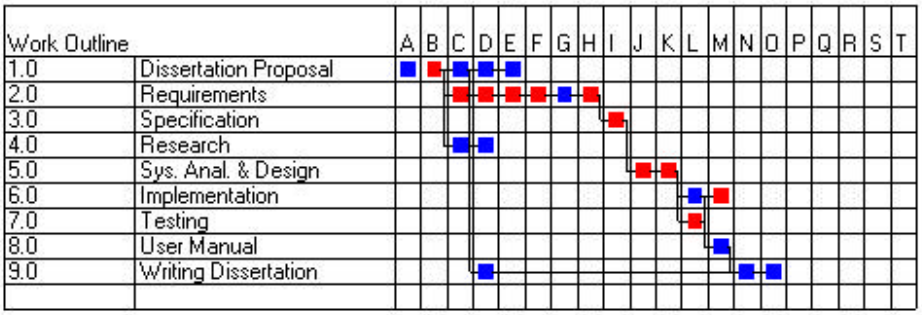
Project Management Software called "MinuteMan" version 6.4 was used for planing and keeping track of the project. The project was broken into tasks. The requirements and deliverables from the specification were used as tasks. Each task (whether milestone or deliverable) was given duration (the length of time to be spent on that task). It also contained a start and a finish date that coincides with the duration of the task. The software created three charts – Outline View, Pert Chart and Time Line Chart. The Outline View and Time Line Charts are given in next sections.

1.2 Critical Path

The critical path of the plan was drawn automatically by the software. The critical path is a linear path through the project. This path indicates the tasks that fall on it have to be completed by the date specified on the task. If the deadlines for the tasks on the critical path are not met then the project will be late as the project has now fallen behind the schedule. The tasks that do not fall on the critical path now have what is called slack time. Slack time means that these tasks do not have to be completed as early and have less priority than the tasks on the critical path.

The charts shown here shows that the **Project Outline, Requirements, Specification, Systems Analysis & Design, Integration into Website** and **Testing** are on critical path. Hence extra care was taken to finish these on time. However there were some delays in few tasks, however had been managed by making suitable adjustments.

Project = Dissertation for MS(SS) from BITS, Pilani Page 1



Tasks		Start	End
1.0	Dissertation Proposal		
A	Dissertation Proposal	8/1/01	8/3/01
B	Project Outline	8/6/01	8/10/01
C	Planning	8/13/01	8/13/01
D	Seminar Prepration	8/14/01	9/10/01
E	Seminar	9/11/01	9/12/01
2.0	Requirements		
C	Requirements Outline	8/13/01	8/16/01
D	Interviews	8/17/01	8/23/01
E	Questionnaire Design	8/24/01	8/24/01
F	Filled Questionnaire	8/27/01	9/7/01
G	Study of Similar Systems	8/17/01	8/21/01
H	Set of Requirements	9/10/01	9/10/01
3.0	Specification		
I	Specification	9/11/01	9/12/01
4.0	Research		
C	Literature Review	8/13/01	9/21/01
D	Study of Related Technologies	8/13/01	9/21/01
5.0	Sys. Anal. & Design		
J	System Analysis	9/13/01	9/26/01
K	System Design	9/27/01	10/10/01
6.0	Implementation		
L	Implementation	10/11/01	10/31/01
M	Integration into WebSite	11/8/01	11/9/01
7.0	Testing		
L	Testing	10/11/01	11/7/01
8.0	User Manual		
M	User Manual	11/1/01	11/7/01
9.0	Writing Dissertation		
D	Writing Dissertation	8/14/01	11/5/01
N	Dissertation Binding	11/8/01	11/8/01
O	Dissertation Submission	11/9/01	11/9/01

Figure: **Outline View Chart of the Project**. ***Date Shown in MM/DD/YY***

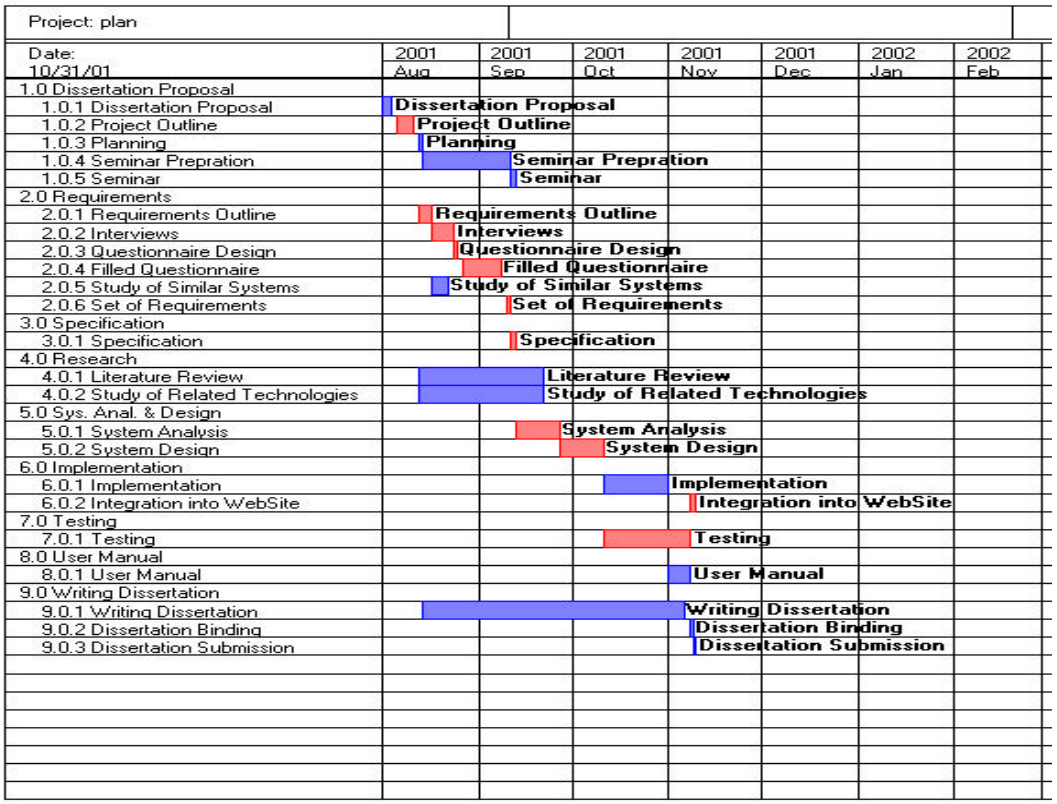


Figure: **Time Line Chart of the Project**.

Chapter Four:

REQUIREMENTS AND SPECIFICATION

1. Requirements

1.1 Methods

This project was undertaken to fulfill the work-related requirements at employing organization NIC (National Informatics Centre, New Delhi) and simultaneously fulfill the requirements of MS (Software Systems) degree pursued from the BITS (Birla Institute of Technology and Science), Pilani. This project was not given with a prescribed set of requirements. When the project was undertaken, there was no specific set of requirements in job work environment except the broad general requirement that the IndMED database has to be put on Internet so that it is available globally and can be searched without much of efforts from the end users. Thus initially the requirements from drawn from the work related environment, experience and the project proposal submitted at the time of registering for the semester. Later, precise requirements were established by employing methods of Informal Interviews, Questionnaires and Observation of similar systems.

The potential users of the proposed system fall into two groups of people. One group consists of the officers and the staff of the Indian Medlars Centre (a Division of NIC) who are responsible for providing service to doctors, biomedical researchers and others seeking information. This group can be labeled as "Service Providers". The other group consists of the doctors, biomedical researchers and others seeking research oriented information through web. This group may be called as "Service Consumers".

The Service Providers were easily available for long interviews and discussions. They were interviewed to access their needs and search behavior. Moreover since they had considerable experience in serving the Service Consumers, the interviews were also directed at their perception of the needs of the Service Consumers. In contrast the Service Consumers were not readily available for long interviews. However few of them were interviewed on their visit Indian Medlars Centre (IMC) for getting information related to their research works. A group of biomedical researchers were also interviewed who participated in a training-cum-workshop on "Internet and Biomedical Research" at NIC, New Delhi. However Questionnaire method was more effective as it consumed less of their time.

1.1.1 Interviewing

The idea behind this approach to collecting a set of requirements is that the users of the potential system are interviewed. For this project, the interviews were kept informal and unstructured. This helped to gather the requirements on a broad spectrum of the needs and the search behavior of the users. The theme of the interviews were the following:

1.1.1.1 User Search Behavior

- Word or Phrase Searching
- Author Searching
- Subject Searching
- Title Searching
- Journal Name Searching
- Free-Text Searching
- Control Vocabulary Searching
- Use of Boolean Operators
- Field Restricted Searching

1.1.1.2 Web-Interface

- Type of Search Boxes
- Number of References required to be displayed
- Facility to edit previously executed query
- Selecting references for further processing
- Kind and level of help required

1.1.1.3 Display of the Search Results

- Default display of the format.
- Different formats for the references.

1.1.2 Questionnaire

The Questionnaires were used to find out the requirements precisely in a structured manner. The interviews provided the groundwork for framing the Questionnaire. It was kept as short as possible so that the answers would be of better quality. The Questionnaire is shown as follows:

Questionnaire for Users' Requirements for Web-Based IndMED

We are in a process of developing a Web-based database covering references along with abstracts to articles published in Indian Biomedical Journals. Could you please spend some time in answering the following questions? It will help us to satisfy the needs of the biomedical community for a database of Indian biomedical literature.

I. **Searching**

a) How many words would you be normally using to search your desired information from the proposed web-based database?
One Two Three More

b) How the words are used in searching?
Individual Words.
As a Phase.
Combined with Boolean operators (AND, OR, NOT)?

c) The words entered should be picked up from?
Anywhere
Keywords
Restricted to fields specified

d) If the words are to be restricted to fields, which of following would be useful?
Article Title
Journal Title
Authors
Address of the Authors
Year of Publication
Keywords
Abstract
_____
_____
_____
_____

II. Web-based Interface:

a) Would you like to have the following?
Simple Search Box: Suitable for simple query of few words.
Advanced Search Box: Suitable for complex query, using Boolean Operators (AND, OR, NOT) and words restricted to specific fields.
Minimal Search Box: Just like simple search but having most of the default values and occupying very small space.

b) How many references would you like to be displayed to your query at a time?

10 20
50 100
200 500
1000 5000

III. **Display Formats:**

a) How the references in the results should be displayed?

Each field of the reference be labeled
Display should be according to academically accepted format.
No labels with the individual fields of the references, but distinguishable typographically and use of colours.

b) Please click against each item that you would like to be present in the following three formats of different level of details?

Reference Details (You are free to add your own)	**Minimal Details Format**	**Medium Details Format**	**Full Details Format**
Author(s)			
Author Address			
Article Title			
Article Abstract			
Journal Name			
ISSN			
Publication Year			
Journal Volume			
Journal Issue No.			
Article Pagination			
No. of Reference			
MeSH Keywords			
Other Keywords			
Record ID.			

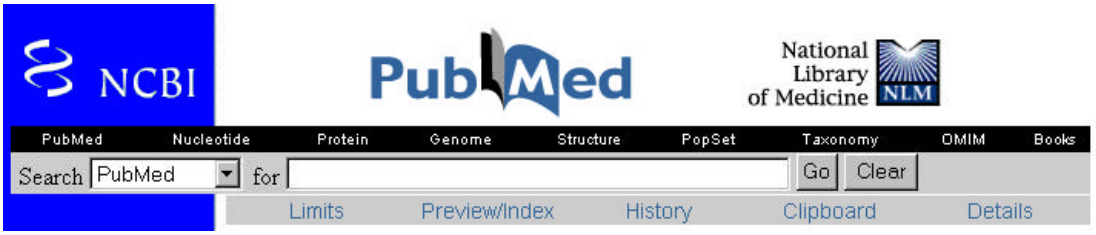
Thanks for your Cooperation.

1.1.3 Observation of similar systems

There are number of web-based database interfaces to bibliographic databases available on Internet. A study of few of them would give insight into the requirements of the proposed system. Following three interfaces were studied. They were selected as they provide the MEDLINE database (which is very similar to IndMED) database but through their different interfaces. The study provided insight into different ways for providing access to bibliographic databases on the web and requirement for such a system. The following sites were studied:

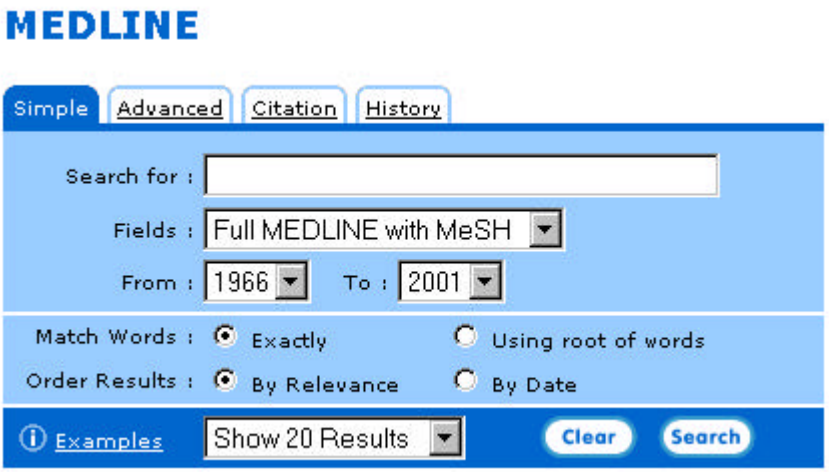
1.1.3.1 Pubmed

"PubMed, a service of the (United States) National Library of Medicine, provides access to over 11 million MEDLINE citations back to the mid-1960's and additional life science journals. PubMed includes links to many sites providing full text articles and other related resources" (United States National Library of Medicine, 2001c, http://www.ncbi.nlm.nih.gov/entrez/query.fcgi).



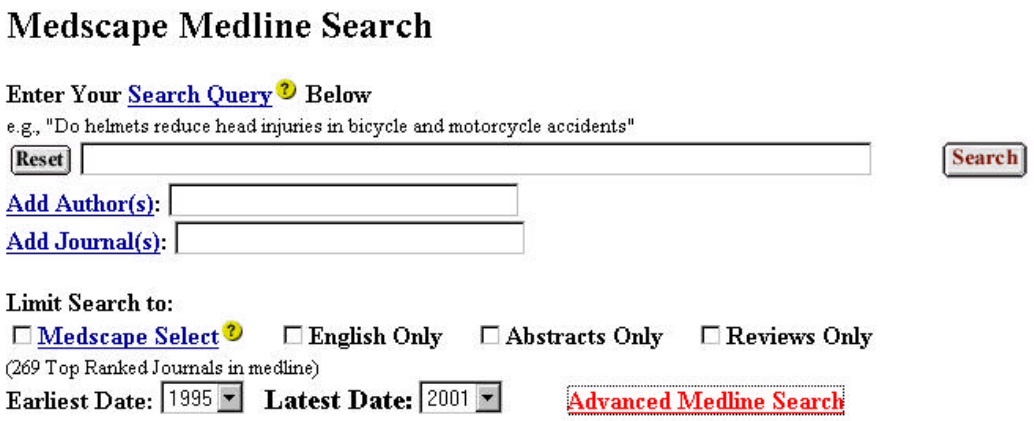
1.1.3.2 BioMedNet

It is a biomedical portal providing wide range of information services in the areas of biomedical sciences. (BioMedNet, 2001, http://research.bmn.com/medline)



1.1.3.3 Medscape

It is also portal directed towards medical practitioners and the patients. It has a interface to search Medline database along with many others. (Medscape, 2001, http://www.medscape.com/server-java/MedlineSearchForm)



1.2 Set of Requirements

From the methods employed as discussed above, the following set of requirements were formulated:

- Web-base interface to search the database.
- Three types of Search Forms to satisfy different needs:
  - Simple Search: Suitable for simple query of few words.
  - Advanced Search: Suitable for complex query, using Boolean Operators (AND, OR , NOT) and words restricted to specific fields.
  - Minimal Box: Just like simple search but having most of the defaults values and occupying very small space so it may be available at number of web-pages of the site.
- Allow searching of words without restricting to a particular field i.e. anywhere in the record.
- Allow searching of words restricted to the following fields:
  - -Article Title
  - -Authors of the Article
  - -Title of Journal Publishing the Article.
  - -ISSN of Journal Publishing the Article.
  - -Volume, Issue or Pagination of Journal Publishing the Article.
  - -Year of Publication.
  - -Keywords assigned to the Article.
- Allow the user to select number to references to be displayed:
  - -10
  - -20
  - -50
  - -100
  - -200

- Display the total number of hits to the search query.
- Display of the references to articles should be displayed without labels of individual fields but with proper typographic distinctions.
- By default initially the references should be displayed in "Citation Format" i.e. showing the following fields:
  -Author(s)
  -Title
  -Journal Name along with Publication Date, Volume, Issue and Page Numbers.
- Option to select all or individual references shown as the result of the search query.
- Option to show the selected references in the following formats:
  "**Citation Format**"
    -Author(s)
    -Title
    -Journal Name along with Publication Date, Volume, Issue and Page Numbers.

  "**Citation + Abstract Format**"
    -Author(s)
    -Address of first the Author
    -Title
    -Journal Name along with Publication Date, Volume, Issue and Page Numbers
    -Abstract of Article

  "Full Record Format"
    -Author(s)
    -Address of first the Author
    -Title
    -Journal Name along with Publication Date, Volume, Issue and Page Numbers
    -Abstract of Article
    -Keywords assigned using MeSH.
    -Other Keywords
    -Number of References cited in the Article
    -Record Number
- Display the executed query in a search form along with the search results. The user may modify the query and resubmit it.

2. Specification

Now that the requirements for the system had been compiled the next step was to introduce the specification of the project. The specification of the project was intended to lay the project out in a formal manner. This provided the whole picture and thus helped in planning and fixing approximate time to be spent on each aspect.

2.1 Aim

The aim of the project is to enable a legacy database to be searched over the Internet through web-based interface. This database had been build using a DBMS (CDS/ISIS Version 3.07) under DOS that do not has in-built support accessing it through web. The database entitled IndMED is of bibliographic nature and consists of references to articles published in learned biomedical journals. It contains information like Authors and their Addresses, Article Title, Journal Title, Year of Publication, Volume, Issue and Pagination. Beside this basic reference information, it also contains Abstract of the articles and other information like Keywords assigned by human subject experts according to according to a controlled vocabulary of medical terms called Medical Subject Heading (**MeSH®**).

2.2 System

It would be a web-based system where the users can search and retrieve references related to their subject of interest by submitting a query to the system.

The proposed system would have has three **sub-systems** as described below:

2.2.1 Web-Based Interface

The web-based user interface would consist of web pages having html forms and dynamically generated html pages by the CGI scripts. Suitable html pages and output would be designed and produced to satisfy the searching behavior of the users. Care would be taken to produce suitable output from the database along with html tags for displays on the users' browsers.

2.2.2 Web-Database Connectivity

The users would be interacting with web-based user interface only. Connectivity mechanism would be required between the user interface and the database. This would have to be achieved through by **Common Gateway Interface** (CGI). The CGI would consist of few PERL scripts. The scripts would take the search query from the web-based user interface (html forms) and pass on to the database system. The results would be returned back and would be displayed in the users' browsers.

2.2.3 Database subsystem

This subsystem would consist of the existing database (IndMED - having bibliographic information relating to references published in Indian Biomedical Journals) in the CDS/ISIS version 3.07 under DOS. An interface to pass on the query and retrieve the records in suitable format by using the "inverted file" index of the DBMS.

2.3 Scope

The scope of the project is limited to the searching of the database through web. Other aspects of the database like data entry, editing and management of the database is out of scope for this project. These are assumed being carried out under the DBMS.

The project involves designing a system that will allow users to submit their searches over Internet using their web browsers to the IndMED database and retrieve relevant references from Indian biomedical literature. The use of systems analysis techniques to break the system down for analysis would be the starting point, leading to the design of the system. The subsystems would be implemented and tested. A user help document for searching would also be produced.

2.4 Deliverables

The following would be the deliverable of the project:

- Analysis of the system
- Design for the system
- Web-based interface
- CGI Scripts
- User Help Document for searching
- Integrated web-pages for IMC website
- Written report on the project

2.5 Test Specifications

The system would be tested for:

- Compatibility with Internet Browsers: Internet Explorer, Netscape Communicator and Opera
- Handling user errors: submitting empty request and Unbalanced parentheses
- Retrieval of correct set of references
- Proper display formats
- Number of references as per selection by the users
- Logical errors in the CGI scripts

2.6 Required Skills

Following are skills the would be required:

- System Requirement collection and Analysis
- Systems Analysis and Design
- Project Planning and Management
- Knowledge of PERL
- Knowledge of HTML
- Familiarity with the IndMED database
- Windows NT and IIS
- Project report skills

2.7 Duration of the Project

It should be completed in approximately 70 working Days.

2.8 Resources Available

- NIC Library
- World Wide Web
- Httpd Server over Windows NT Server
- Windows NT Workstation

Chapter: Five

ANALYSIS

Next logical step of the project was the analysis modeling. There are two dominant methods for analysis modeling. The first one, which is the traditional modeling method, is the *"Structured Analysis"*. The other one, which is the newer, is known as *"Object Oriented Analysis"*. The heart of the distinction between them lies in their primary focus. The traditional method focuses on the functions of the system – What is it doing? – Object Oriented method centers on the Object, which combines *data and functionality.* Structured Analysis method was adopted for the project. The reasons were the complexity and unfamiliarity with the other method and because of the primary emphasis of the project is on functionality and interface.

"Structured Analysis is not a single method applied consistently by all who use it. Rather, it is an amalgam that was evolved over almost 20 years. There is probably no other software engineering method that has generated as much interest, been tried (and often rejected and the tried again) by as many people, provoked as much criticism, and sparked as much controversy, the method has prospered and gained a substantial following in the software engineering community" (Pressman R.S., 1997, pp. 298).

Structured analysis began as an information flow modeling technique. As information moves through software, it is modified by a series of transformations. A **data flow diagram (DFD)** is a graphical technique that depicts information flow and transformations that are applied as data move from input to out. The data flow diagrams are developed in stages, the first diagram being the context diagram. This contains the external entities and data stores, that are using or being used by the system. The next stage is to expand the system into a set of processes. This is called the Level 1 DFD. These can be decomposed further into lower level processes to show in more detail how the process works.

1. Analysis Model

The system consists of three sub-systems. The Web-Based User Interface, The Common Gateway Interface (GCI) and the DBMS Subsystem. While the first two subsystems were need to created from the scratch. The Database already exited in the DBMS Subsystem.

1.1 Data Flow Diagrams (DFDs)

The DFDs of the model are given in the following sections.

1.1.1 Context Diagram

The Context Diagram or the Level 0 DFD as given here shows:

- User
- IndMED System
- Display

The User is the external entity, which sends a query to the IndMED System. The IndMED System responds by outputting list of references to the Display.

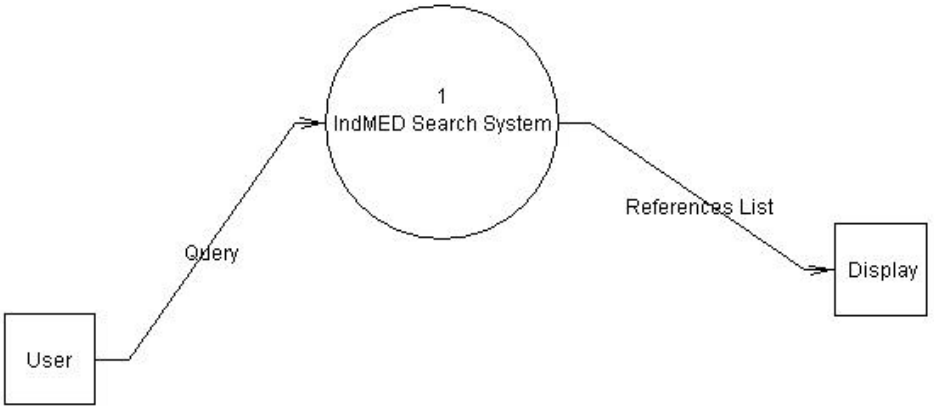


Figure**: CONTEXT DIAGRAM**

1.1.2 Level 1 DFD

Exploding the Context Diagram into Level 1 DFD shows three processes (subsystems):

- Web-Interface (1.1)
- CGI (1.2)
- DBMS (1.3)

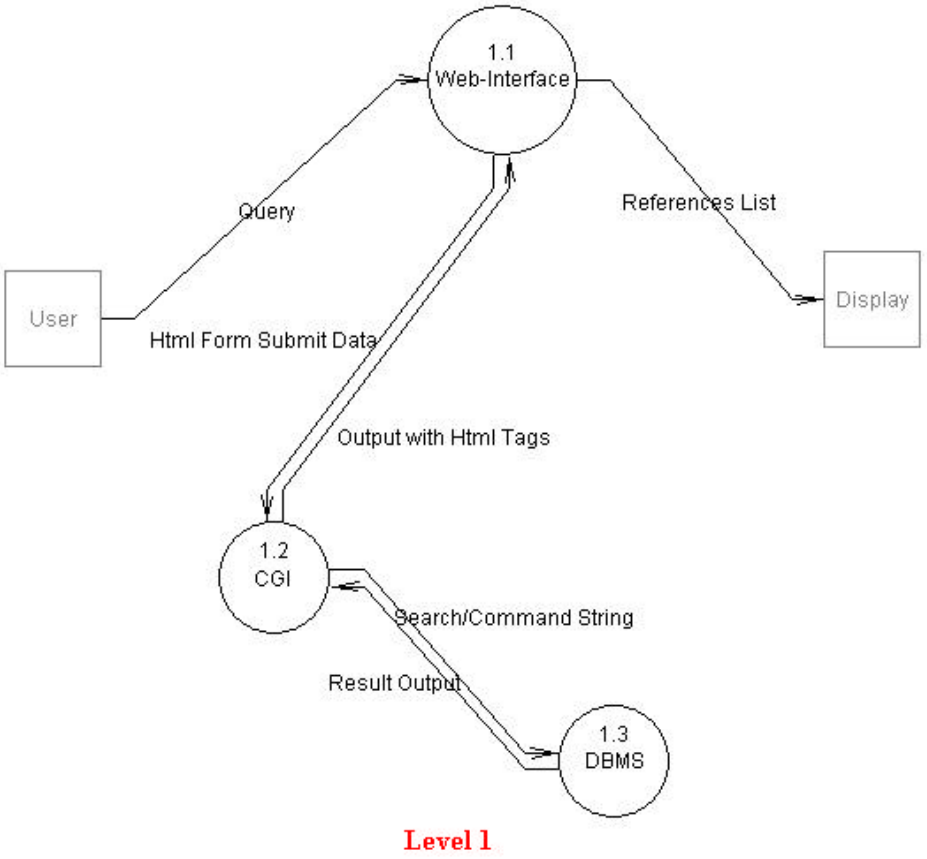


Figure**: Level 1 DFD**

1.1.2.1 Level 2 DFD for Web-Interface (1.1)

The Process Web-Interface has been further exploded as follows:

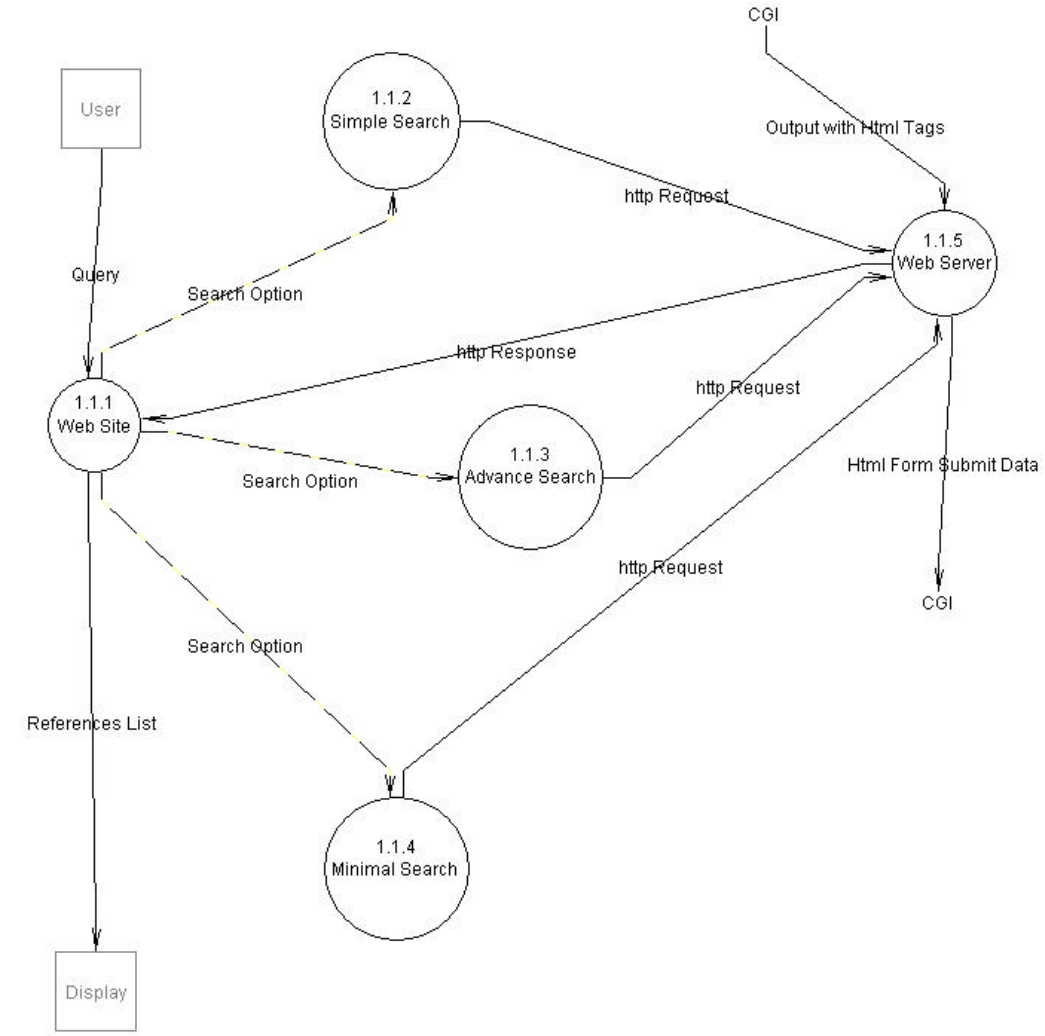


Figure**: Level 2 DFD for Web-Interface (1.1)**

1.1.2.2 Level 2 DFD for CGI (1.2)

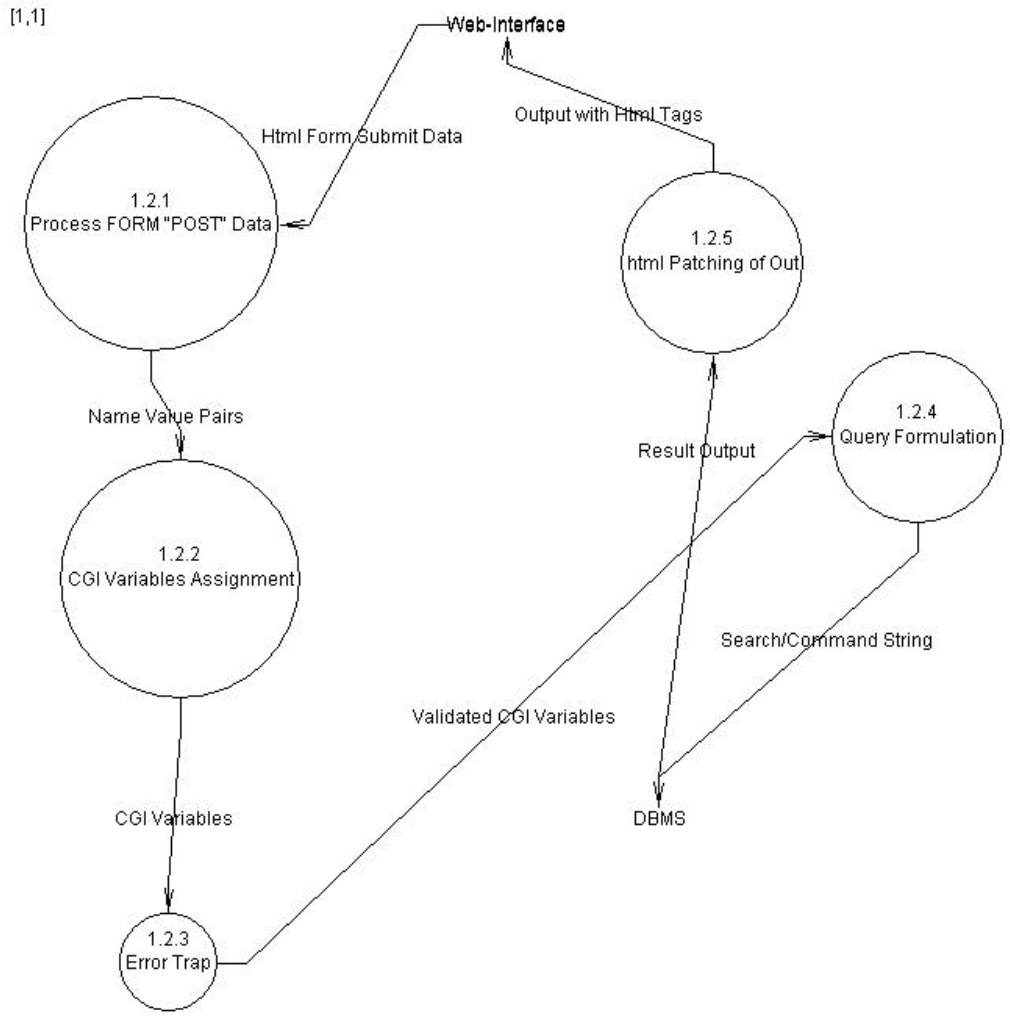


Figure**: Level 2 DFD for CGI (1.2)**

1.1.2.3 Level 2 DFD for DBMS (1.3)

[2,1

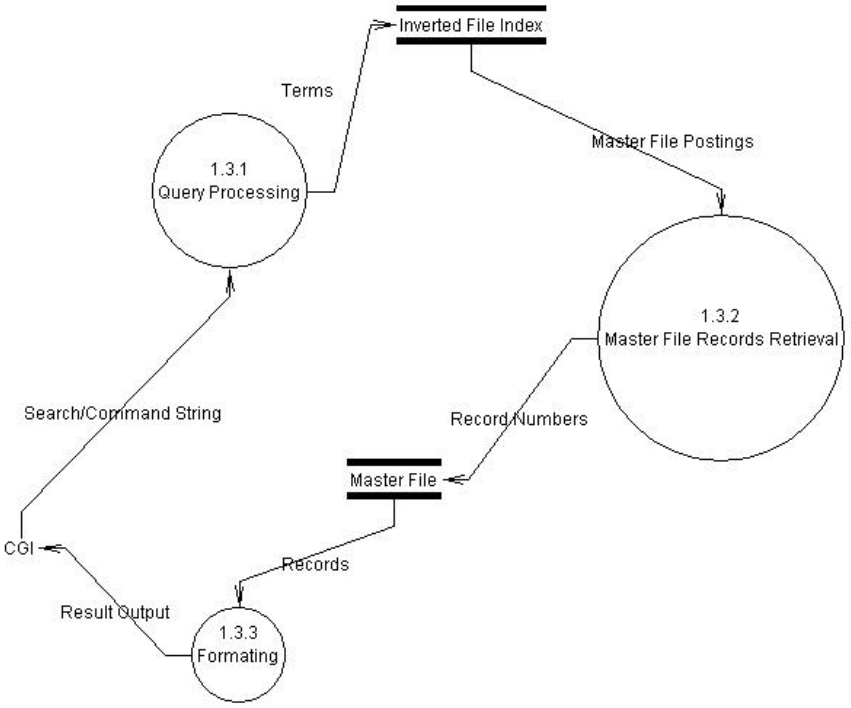
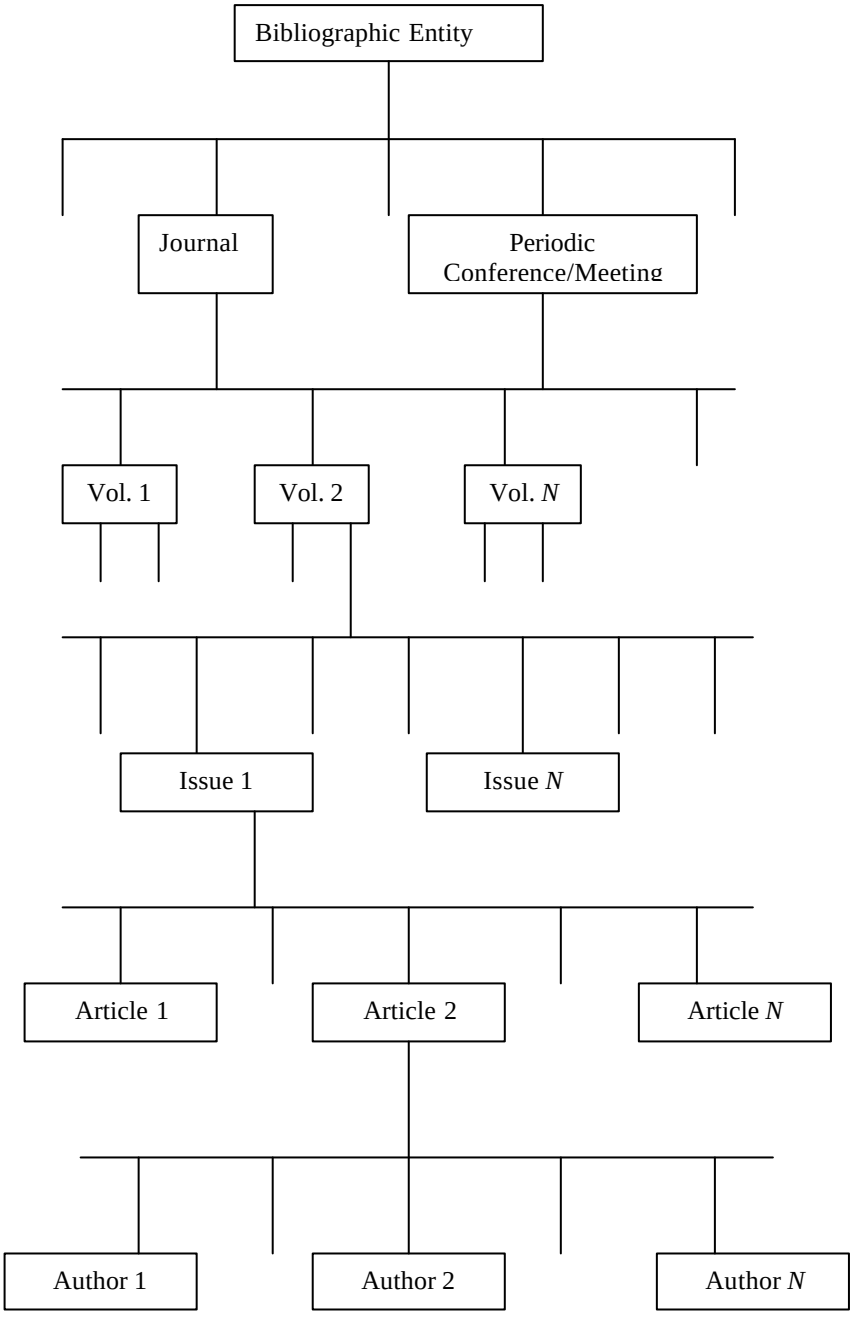


Figure**: Level 2 DFD for DBMS (1.3)**

2.1 Data Model

The system deals with data about research articles. Which are mostly published in learned journals. However these can also be papers presented in periodic conferences and academic meetings. Data Model of the system is complex one. The relationships of the entities can best be described in a hierarchical fashion as shown ahead.



2.1.1 Entity-Relationship Diagram (ERD)

The ERD diagram of the data model is shown as follows:

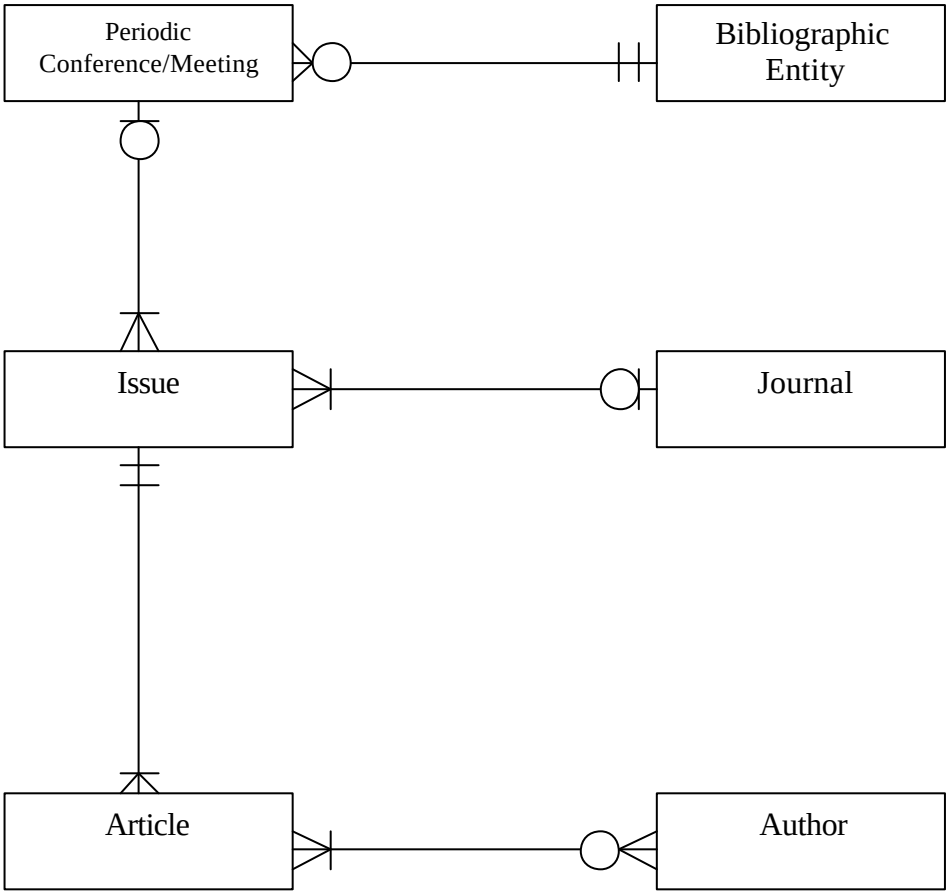


Figure: ERD to show the Data Model

Attributes of the Entities shown in the ERD:

The Entities shown in the ERD have the following attribute:

Bibliographic Entity:

- Type of Bibliographic Entity
- Bibliographic Level
- Literary Indicator
- Note on Bibliographic Relationship

Journal:

- ISSN
- Title
- Person(s) associated with Journal [editor(s) etc.]
- Publisher
- Place of Publication

Periodic Conference/Meeting:

- Title of the Conference/Meeting
- Person(s) associated with conference/Meeting [editor(s) etc.]
- Date of the Conference/Meeting
- Publisher
- Place of Publication

Issue:

- Date of Publication
- Volume No.
- Issue No.
- Library holding the Issue if any.

Article:

- Title
- Language
- Abstract
- Pagination
- Abstract
- Keywords representing the theme of the article
- Number of References mentioned in the article
- Note
- Degree in case the article is based on Dissertation/Theses
- Indexer Identifier

Author:

- Name of the Personal Author(s)
- Name of the Corporate Author(s)
- Affiliation/Address

Chapter: Six

DESIGN

The design of the system had been explored under three subsystems:

1. **Web-Interface** (corresponding to process 1.1 of Level 1 DFD of Analysis Model).
2. **Web-Database Connectivity** (corresponding to CGI process 1.2 of Level 1 DFD of Analysis Model).
3. **The Database** (corresponding to DBMS process 1.3 of Level 1 DFD of Analysis Model).

1. Web-Interface

1.1 Design Considerations

According to Schneiderman (Schneiderman, B.,1992) novices are best served by a simple set of actions but as the users' experience increases so does their desire for more functionality and more rapid performance. Thus in this project to cater for the simplicity the novice users would be provided **Simple Search Form**. To provide more functionality and rapid performance to the more experienced user an **Advanced Search Form** would be provided. A small box would have to be provided so as to place it every important web page of **IMC Home Page** to promote IndMED. This would be referred as **Minimal Search Box** in this project. It would be a small box with most of the options selected by default as hidden *form* tags.

Ambler (Ambler S.W. 2000) presented several tips and techniques for designing effective user interfaces. The key tips are summarized below:

1.1.1 General Guidelines

- Be consistent in a user interface, it's critical.
- Set user interface standards and stick to them.
- Choose industry standards so as to increase the chance that your applications will look and feel like other applications developed externally to your organization.
- Explain the rules of how your application works to your users. If it's consistent, then the rules should be simple and few in number.
- Support both novices and experts.
- Word text consistently, positively, and in full English.
- Look at other applications with a grain of salt because not everyone understands good user interface design.
- Display the objects that your users need to do their jobs on the desktop.

- Think in terms of clusters of business objects and their corresponding interface objects, not in terms ofapplications.
- Interface objects should look, feel, and behave exactly like the real-world objects that they represent.

1.1.2 Screen Design

- Navigation between screens and on screens are both important.
- Understand your widgets so that you know how to apply them properly.
- Use color sparingly and always have a secondary indicator.
- Follow the contrast rule - put dark text on light backgrounds and light text on dark backgrounds.
- Use fonts sparingly and consistently.
- When items are unavailable gray them out, don't remove them if you want your users to form accurate mental models.
- Use non-destructive default buttons.
- Left justify edit fields and right justify their labels.
- Right justify integers, decimal-align floating point numbers, and left justify strings.
- Don't create busy/crowded screens.
- Use group boxes and whitespace to group logically related items on the screen.
- Open windows in the center of the action.
- Pop-up menus shouldn't be the only source of functionality.

1.2 Interface-Flow Diagram

To users the user interface is the system. An Interface-Flow Diagram (Page-Jones, 1995; Ambler1998a; Ambler, 1998b; Ambler, 2001b), was utilized as the mechanism to help design the user interface. The following Interface-Flow Diagram was used to model the high level relationships and interactions between User Inteface Screens. The boxes represent the user interface objects (Screens, Reports or Forms) and the arrows represent the possible flow between the screens. This Diagram presents a high-level overview of the web-based user interface.

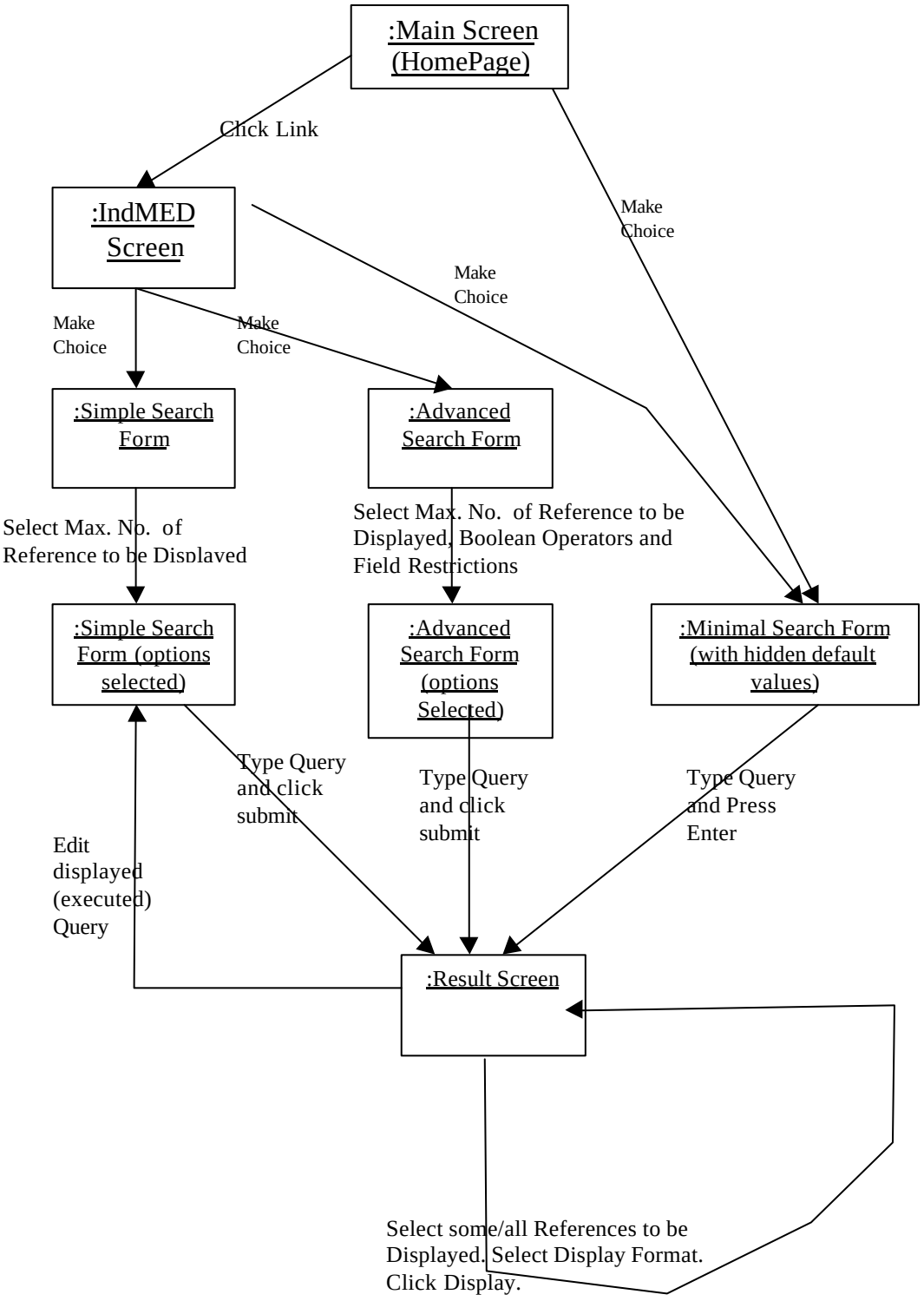


Figure: **Interface-Flow Diagram for Web-User Interface of IndMED**

1.3 Design of Web-Interface

Using the Interface-Flow Diagram as guide then following components of the User Interface had been designed:

1.3.1 Main Screen (IMC's HOME PAGE)

The IMC's Home Page would provide link to **IndMED webpage** as well as have the **Minimal Search Form.**

1.3.2 IndMED webpage

It would be an ordinary web page having **Simple Search Form, Advanced Search Form** and **Minimal Search Form.**

1.3.3 Simple Search Form

It would be an html form.
It would provide:

-A single line window for entering query.
-Options to select maximum number references that can be displayed from a set of 10, 20, 50, 100 and 200 references.
-A Clear Button
-A Submit Button

It would display the references in the default format i.e. "citation format".
Form Action: Server Side Perl script "**imss.pl**" Method: **Post**
Hidden type tags for:

-Display Format = "citation format" (plmx.pft)

1.3.4 Minimal Search Form

It would provide the same functionality as that of "simple search form".
It would require minimum web space.
It would provide small single line box for entering query.
It would have **no "**clear button".
It would have **no** "submit button".
User would have to press "ENTER" key after entering query. This would initiate the "submit" action of the form.
Its action would return maximum number of 50 references only.
Form Action: Server Side Perl script "**imss.pl**" Method: **Post**
Hidden type tags for:

-Display Format = "citation format" (plmx.pft)
-Maximum Number of references = 50

1.3.5 Advanced Search Form

It would serve the needs of the advanced users.
It would have four window query lines for typing of query by the user.
Each query line would be followed by "*field restriction*" drop down menu.
The search for the words entered in the query line would be restricted to the "*field restriction*".

The following would be included in the *"field restriction":*

-Article Title
-Authors of the Article
-Title of Journal Publishing the Article.
-ISSN of Journal Publishing the Article.
-Volume, Issue or Pagination of Journal Publishing the Article.
-Year of Publication.
-Keywords assigned to the Article.

In-between the two adjustant query lines there would be a drop down menu for selection of Boolean operator between the query entered in them.
There would be three such *Boolean operator* drop-down menus.
The default Boolean operator would be "AND".
It would provide:

-Options to select maximum number references that can be displayed from a set of 10, 20, 50, 100 and 200 references.
-A Clear Button
-A Submit Button

It would display the references in the default format i.e. "citation format".
Form Action: Server Side Perl script "**imas.pl**" Method: **Post**
Hidden type tags for:

-Display Format = "citation format" (plmx.pft)

1.3.6 Search Result Screen

The Search Result Screen would be *dynamically generated* by the CGI every-time a *"submit form"* action is initiated or "**display button**" (present on itself) is clicked. It would have following three parts:

- **Top Menu**
- **List of References**
- **Bottom query Box**

1.3.6.1 Top Menu

The Top Menu would serve two purposes. One to display information regarding the search results. The other is to act as menu for providing mechanism for the users to do the following:

- Option to choose to display ALL or SELECTED references.
- Option to choose the Display Format from the following:

  Citation Format (plmx.pft)
  Citation + Abstract Format (plmxb.pft)
  Full Record Format (plmxf.pft)

The Top Menu would be a ***dynamically generated*** **html FORM** by the CGI.
It would have a "**display button**".
The "**display button**" would initiate *"submit form"* action when clicked.
It would have radio button set to choose "ALL" or "SELECTED" references.
The default would be ALL.
It would have a drop-down menu for selection of Display Format.
The default selected Display Format would be *"Citation + Abstract Format"*.
Form Action: Server Side Perl script "**imrs.pl**" Method: **Post**

Hidden type tags for:

-Carry forward of the executed search query.

-Carry forward the previously selected maximum no. of references that can be displayed.

The **List of References** (to be discussed below) is also included this FORM.

1.3.6.2 List of References

These are the response of the query users. The list of references will be displayed in the default Display Format (Citation Format) or other Display Format depending upon the situation. There would be three Display Formats. These formats along with the element of information displayed for each returned reference are given below:

Citation Format (plmx.pft):

-A check box which the user can click to select the reference
-Author(s)
-Title
-Journal Name along with Publication Date, Volume, Issue and Page Numbers.

Citation + Abstract Format (plmxb.pft):

-A check box which the user can click to select the reference
-Author(s)
-Address of first the Author
-Title
-Journal Name along with Publication Date, Volume, Issue and Page Numbers
-Abstract of Article

Full Record Format (plmxf.pft):

-A check box which the user can click to select the reference
-Author(s)
-Address of first the Author
-Title
-Journal Name along with Publication Date, Volume, Issue and Page Numbers
-Abstract of Article
-Keywords assigned using MeSH.
-Other Keywords
-Number of References cited in the Article
-Record Number

1.3.6.3 Bottom Query Box

This part of the *Search Result Screen* would also be a ***dynamically generated*** **html FORM** by the CGI. It would be very similar to the **Simple Search Form.** The only difference would be that the executed query is already displayed in the form window. It would serve two purposes. First to remind user the query executed for the set of references displayed. Secondly to allow his to refine the query and execute it again by clicking submit button.

It would provide:

-A single line window for displaying the executed query.

-Options to select maximum number references that can be displayed from a set of 10, 20, 50, 100 and 200 references.

-A Clear Button

-A Submit Button

It would display the references in the default format i.e. “citation format”.

Form Action: Server Side Perl script “**imss.pl**” Method: **Post**

Hidden type tags for:

-Display Format = “citation format” (plmx.pft)

-Carry forward of the executed search query.

2. Web-Database Connectivity

The connectivity between the web-based user interface and the database would be achieved by “Common Gateway Interface (CGI)”. It would consist of CGI Server Side scripts written in PERL 5 (ActivePERL). The following the scripts would be written:

- **imss.pl**: For connecting **Simple Search Form**, **Bottom query Box (**of **Search Result Screen)** and **Minimal Search Form** to the database.
- **imas.pl**: For connecting **Advanced Search Form** to the database to the database.
- **imrs.pl**: For connecting **Top Menu (**of **Search Result Screen)** to the database.

For designing the CGI, Level 2 DFD for CGI (1.2) would serve as the basic model. Accordingly the following modules would be incorporated in all the scripts:

2.1 Processing of FORM data (1.2.1)

This module would be designed for *parsing* and *interpreting* query strings passed to CGI scripts from the web-based user interface. I would separate the **field – value** pairs of the html forms.

2.2.CGI Variables Assignment (1.2.2)

This module would assign the values passed from the html forms to CGI variables. These CGI variable would be used for further processing data received.

2.3 Error Trap (1.2.3)

This module would take care of the following errors by the users:

- Unbalanced Parentheses.

- Submitting of empty form.

- Choosing “Selected References” to display but no referenced had been selected from the list of references displayed in Search Result Screen.

An error messages would be generated and the scripts would be terminated in case of error.

2.4 Query Formulation (1.2.4)

This module would formulate a query string suitable for passing on to the interface to the DBMS. This query string would consist of the following:

- Name of the database to be searched.
- Maximum No. of references to be output.
- Display Format to be used for retrieving the references.
- The Boolean Query.
- Field Restrictions if any to the search terms.

2.5 Html Patching of DBMS Output (1.2.5)

This module would take the output of the DBMS after the query had been executed. After the output is stored in CGI variable it would *patch* the out for the following:

- Generate HTML Document Header.
- Generate HTML Document Title.
- Generate HTML Document Body Tag.
- Generate HTML Document Table Tag.
- Generate HTML Document Form Tag.

-Generate suitable html code for *Search Result Screen*'s Top Menu.

-Pick up all search terms from the query.

-Patch up search terms occurring in the output for highlighting.

-Extract the total number of hits (i.e. references present in the database against the query) from the output and display at appropriate place.

-Generate a check-box before each reference retrieved. This would enable the to select the references ticked.

-Generate suitable tags for displaying the *list of references.*

-Generate form closing tag to mark the end of *Top Menu Form.*

-Generate *start form tag* for the *bottom search box form.*

-Generate tags for the *bottom search box form* on the *Search Result Screen* having the *executed query* displayed.

-Generate the form closing tag.

-Generate the closing tags for *table, body* and *html document*.

3. The Database

The database IndMED had been created in CDS/ISIS 3.07. Unlike other common *Database Management Systems* (DBMS) it is not based on *Relational Model*. It can best be described as Text Retrieval Systems or "Free Form" DBMS. The IndMED database's structure and the fields indexed are discussed in the following sections.

3.1 Database Structure

The make the database compatible with International Standards, the database structure had been based on an international standard called "Common Communication Format, 2nd Edition" (Hopkinson, A. & Simmons, P. 1988). The structure of the database is given in the table given ahead.

The table has the following columns:

Field Name: It is the descriptive name given to each field. A field may be optional (i.e. it may be absent in one or more records). The fields are of variable length. This would make the records of the database having variable fields and variable length.

Sub-Field: The field may co*n*tain single data element, or two or more variable length data elements. In case the field contains two or more data elements it is said to have subfields. Which are identified by subfield delimiters.

Tag: Each is assigned a tag. It is numerical indicative of the contents of the field. A field is known by its tag by the DBMS.

Length: It is the maximum length in character that each occurrence of the field can have.

Type: It indicates the type of data the field would contain. Here "C" - stands for character type (both alphabetic and numeric) of data; "A" – stands for alphabetic type; N – stands for numeric type.

Repeatable: It indicates whether a record can have more the one instance or occurrence of the field.

Field Name	Sub-field	TAG	Length	Type	Repeatable?
RECORD IDENTIFIER		1	9	C	
BIBLIOGRAPHIC ENTITY		14	1	A	
BIBLIOGRAPHIC LEVEL		15	5	C	
LITERARY INDICATOR		16	10	C	
SOURCE OF RECORD		20	8	C	
ENTRY DATE		22	6	N	
ENTRY MONTH		23	4	N	
LANGUAGE OF DOCUMENT		40	5	C	
ISSN		100	9	C	
DOCUMENT IDENTIFICATION NUMBER		120	10	C	
TITLE OF ARTICLE		200	250	C	
TITLE OF SERIAL		201	150	C	
TITLE OF HIGHER RECORD		202	100	C	
PERSON ASSOC. HIGHER RECORD		260	100	C	
AUTHOR	a,b	300	250	C	YES
VARIANT AUTHOR		301	150	C	
CORPORATE AUTHOR		310	250	C	YES
NAME OF MEETING		320	250	C	
LOCATION OF MEETING		321	150	C	
DATE FORMAL		322	25	C	
DATE IN ISO FORMAT		323	20	C	
NUMBER OF MEETING		324	6	C	
AFFILIATION		330	150	C	
PUBLISHER NAME		400	150	C	YES
PLACE OF PUBLICATION		401	50	C	
DATE OF PUBLICATION		440	40	C	
VOLUME		490	5	C	
ISSUE		491	6	N	
PAGE		492	10	C	
YEAR		493	2	N	
NOTE		500	100	C	
NOTES ON BIBLIO RELATIONSHIP		510	150	C	
ABSTRACT		600	1000	C	
AUTHOR ABSTRACT		601	1	A	
MeSH DESCRIPTORS		620	300	C	YES
NON-MeSH DESCRIPTORS		621	250	C	YES
PART STMT OF HIGHER RECORD		700	10	C	
LIBRARY NAME		701	50	C	
TYPE OF DEGREE		702	50	C	
NUMBER OF REFERENCES		703	5	N	
INDEXER IDENTIFIER		800	3	A	

Table: **Database Structure of IndMED**

3.2 Fields indexed

There would be a single logical inverted file index. The Inverted file would contain all terms, which may be used as access points during retrieval. For each term, there would a list of references to the Master file record(s) from which the term was extracted. Each posting would contain the record number and additional information precisely identifying the field from which the data was extracted, as well as the relative word position within the field.

The selection of the fields for indexing of the database is given in the table given ahead. The table has the following columns:

Fields Selected for Indexing: These are the descriptive names of the fields that would be selected for indexing. The terms exacted from these fields would go into the inverted file of the system.

Index Tag: The terms extracted from the field would also have an index tag along with these. These index tags would be used by the system to restrict the query to particular field/group of fields. Thus the user would be able to specify the fields restriction and thus serve the needs of the advanced users.

Indexing Type: This indicates how the terms would be extracted from the fields.

Fields Selected for Indexing	*Index Tag*	*Indexing Type*
Record Number	910	Whole Field as Phrase
ISSN	100	Whole Field as Phrase
Date of Publication	440	Each Sub-Field/Occurrence as Phrase
Volume	490	Whole Field as Phrase
Issue	491	Whole Field as Phrase
Page	492	Whole Field as Phrase
Year	493	Whole Field as Phrase
Record Identifier	1	Whole Field as Phrase
Author(s)	300	Each Sub-Field/Occurrence as Phrase
Variant Author(s)	301	Each Word of Field
Name of the Meeting	320	Each Word of Field
Affiliation	330	Each Word of Field
Title of the Article	200	Each Word of Field
Author Abstract	600	Each Word of Field
MeSH DESCRIPTORS	620	Each Sub-Field/Occurrence as Phrase
NON-MeSH DESCRIPTORS	621	Each Word of Field
Article Source (Journal Title, Year, Volume, Issue, Pagination)	903	Each Word of Field
Journal Title	912	Each Sub-Field/Occurrence as Phrase

Table: **Fields to be Selected for IndMED's Inverted File Index.**

Chapter: Seven

IMPLEMENTATION

The Implementation of the system would be discussed under the following three subsystems:

1. **Web-Interface**
2. **Web-Database Connectivity**
3. **The Database**

1. Web-Interface

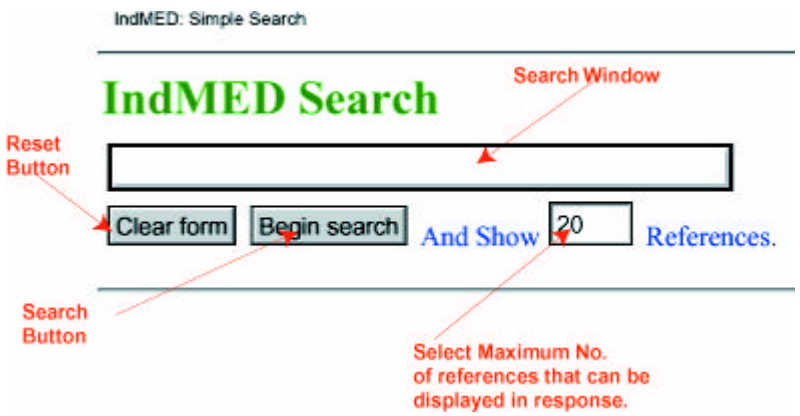
The Web-Interface consists of html forms. These forms were either hard coded with the Html Editors (FrontPage Express and 1st Page 2001) or were implemented in CGI and DBMS to generate dynamically as response to user action.

The User Web Interface consists of the following:

1.1 Simple Search Form

1.1.1 Screen

The Simple Search Form is shown ahead:



It provides line window for entering query where the user can enter the query. It also has a drop down menu to select maximum number references that can be displayed from a set of 10, 20, 50, 100 and 200 references. The Clear Button can be used to scratch the previous entered query and start again. Once the query is entered then user can click on Submit Button. On clicking the Submit Button, the perl script "imss.pl" would execute on the server. The results would be shown on the Search Result Screen in the default format i.e. Citation Format. The name of the display format would be send to the CGI as a hidden html tag.

1.1.2 HTML Code

The html code along with comments for the Simple Search Form is given below:

Simplesearch.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>

<Head>
<title>IndMED: Simple Search</title>
</head>

<body>

<hr>
<table border="0" width="100%">

<tr>
<td colspan="3" width="80%">
<font color="#008000" size="5">
<b>IndMED Search</b>
</font>

<form action="http://164.100.9.16/bin/im/imss.pl" method="post">

        <table border="0" width="90%">
        <tr>
        <td width="55%">
        <input type="text" size="45" maxlength="256" name="search">
        </td>
        <td width="30%" nowrap> </td>
        </tr>
        <tr>

        <td align="middle" colspan="4">
        <p align="left">
        <input type="reset" name="clearbutton" value="Clear form">
        <input type="submit" name="submitbutton" value="Begin search">
        <font color="#0000FF"> And Show
        <select name="maximum" size="1">

                <option value="10">10</option>
                <option selected value="20">20</option>
                <option value="50">50</option>
                <option value="100">100</option>
                <option value="200">200</option>

        </select>References</font>.
        </p>
        </td>
        </tr>
        <tr>
```

Form Action

Max. No. of Ref.s

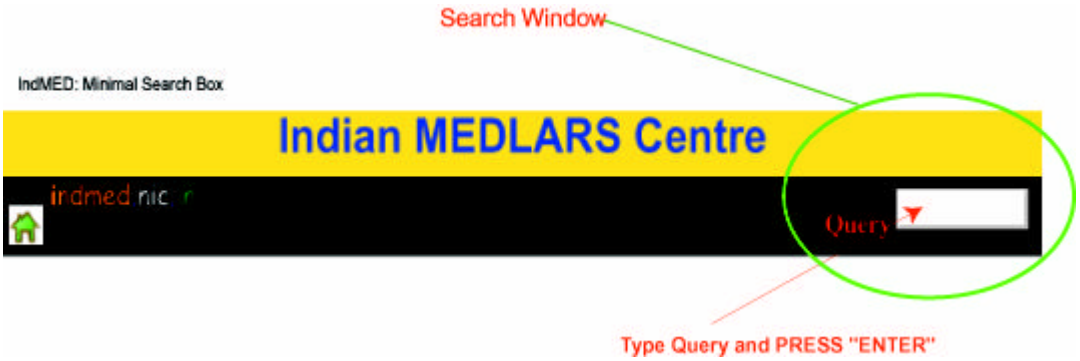
Hidden Format Value

```
        <td>
        <input name="format" type="hidden" value="@plmx.pft">
</form>
</td>
</tr>
</table>
</table>

<hr>

</body>
</html>
```

1.2 Minimal Search Form

1.2.1 Screen

The Minimal Search Form is shown here:



It provides the same functionality as that of “simple search form” at the same time it requires minimum of the web space. It provides small single line box for entering query. It can be seen that there is **no “**clear button” and “submit button”. Therefor the user would have to press “ENTER” key after entering query. This would initiate the “submit” action of the form. Which would result in execution of the perl “**imss.pl**” action. Upto 50 references would be shown on the Search Result Screen in the default format i.e. Citation Format. The maximum number of references and the name of the display format would be send to the CGI as a hidden html tag.

1.2.2 HTML Code

The html code along with comments for the Minimal Search Form is given below:

Minimalsearch.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
<title>IndMED: Minimal Search Box</title>
</head>

<body>
```

```
<div align="center">
<center>
<table border="0" cellpadding="3" cellspacing="0" width="100%">
<tr>
<td align="center" colspan="3" bgcolor="#FFFF00">
<font color="#2200BF" size="5" face="Arial">
<strong>Indian MEDLARS Centre</strong>
</font>
</td>
</tr>
<tr>

<td valign="top" width="50%" bgcolor="#000000">
      <a href="http://indmed.nic.in" target="_top">
      <img src="homeg.gif"
            alt="HOME: Indian Medlars Centre"
            align="top" border="0" width="20" height="20">
      </a>

<font color="#FF8000" size="3" face="Comic Sans MS">
<strong>i</strong>ndmed</font>
<font color="#0000FF" size="3" face="Comic Sans MS">.</font>
<font color="#FFFFFF" size="3" face="Comic Sans MS">nic</font>
<font color="#0000FF" size="3" face="Comic Sans MS">.</font>
<font color="#008040" size="3" face="Comic Sans MS">in </font>
</td>

<td align="right" valign="top" bgcolor="#000000"> </td>
<td align="right" valign="top" bgcolor="#000000">

      <form action="http://164.100.9.16/bin/im/imss.pl" method="POST">

            <input type="hidden" name="maximum" value="50">
            <input type="hidden" name="format" value="@plmx.pft">
            <p>
            <font color="#FF0000">
            <strong>Query</strong>
            </font>
            <input type="text" size="10" maxlength="100" name="search">
            </p>

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

</center>
</div>

</body>
</html>
```

Form Action

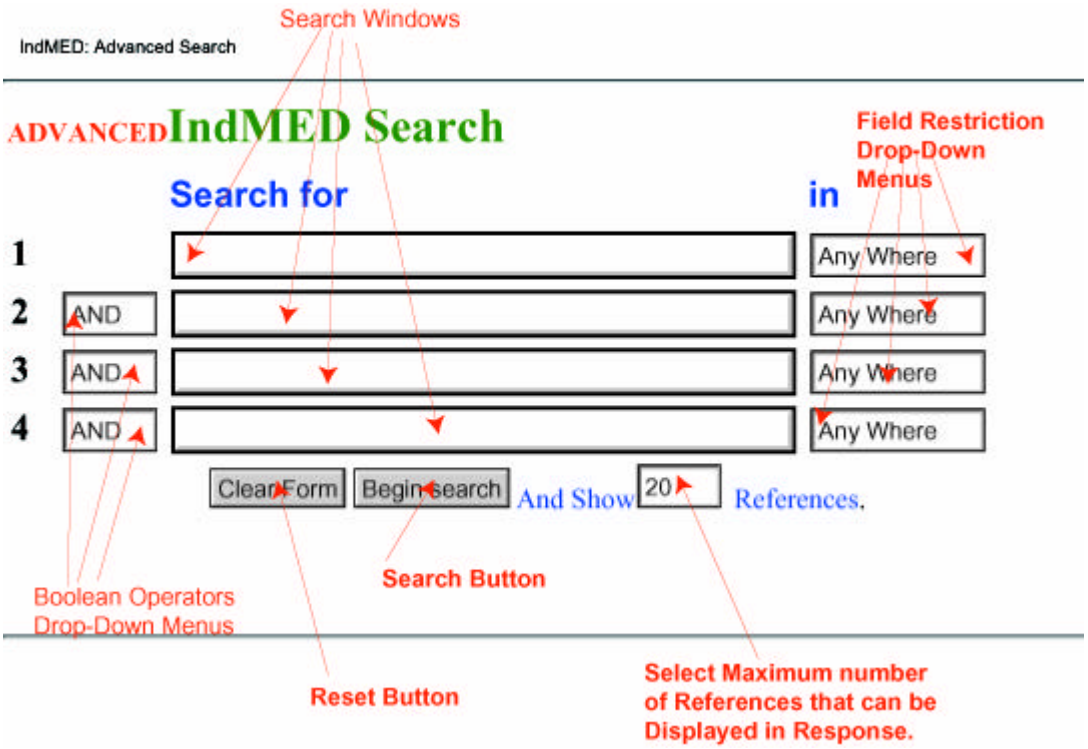
Form does not have any Buttons

Hidden Value for Format & Maximum No. of References.

1.3 Advanced Search Form

1.3.1 Screen

The Advanced Search Form is shown here:



The Advanced Search Form is meant for advanced user who would like to enter precise search queries to retrieve relevant references. It has four lines for typing out the query. The words entered into each of these lines can be qualified by "field restriction drop down menu". These words would be search only if they occur in the fields qualified. The default values for these "field restriction" menus are "Any Where". The user can choose the following from the menu:

The following would be included in the *"field restriction":*

-Title
-Authors
-Keywords
-ISSN
-Source (Journal [words], Volume, Issue and Pagination)
-Year
-Journal Title [Phrase]

In-between the two adjustant query lines there is drop down menu for selection of Boolean operator between the query entered in them. The default Boolean operator is be "AND". The user has the option of selecting "AND", OR, and "NOT". Like the Simple Search Form it also provides a Clear Button, Submit Button and option to select maximum number references that can be displayed from a set of 10, 20, 50,

100 and 200 references. On clicking the Submit Button, the perl script “imas.pl” would execute on the server. The results would be shown on the Search Result Screen in the default format i.e. Citation Format. The name of the display format would be send to the CGI as a hidden html tag

1.3.2 HTML Code

The html code along with comments for the Advanced Search Form is given below:

Advancedsearch.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
<title>IndMED: Advanced Search</title>
</head>

<body>
<hr>
<table border="0" width="100%">
<tr>
<td colspan="3" width="80%">
<font color="#FF0000" size="3"><b>ADVANCED</b></font>
<font color="#008000" size="5"><b>IndMED Search</b></font>

<form action="http://164.100.9.16/bin/im/imas.pl" method="POST">
       <table border="0" width="90%" bgcolor="#FFFFFF">
       <tr>
       <td width="7%"> </td>
       <td width="8%"> </td>
       <td width="55%">
       <h2>
       <font color="#0000FF" size="4">Search for </font>
       <font color="#0000FF" size="2"> </font> 
       </h2>
       </td>
       <td width="30%" nowrap>
       <h2>
       <font color="#0000FF" size="4">
       <b>in</b>
       </font> 
       </h2>
       </td>
       </tr>
       <tr>
       <td width="7%">
       <font size="4">
       <b>1 </b>
       </font>
       </td>
       <td width="8%"> </td>
       <td width="55%">
              <input type="text" size="45" maxlength="256" name="s21">
              </td>
              <td width="30%" nowrap>
              <select name="s31" size="1">
              <option value="braceopen200braceclose">Title</option>
              <option value="braceopen300braceclose">Authors</option>
              <option value="braceopen620braceclose">Keywords</option>
              <option value="braceopen100braceclose">ISSN</option>
              <option value="braceopen903braceclose">Source</option>
              <option value="braceopen912braceclose">Journal Title</option>
              <option value="braceopen440braceclose">Year</option>
```

Form Action

Search Window

Field Restriction Menu

```
<option selected value="">Any Where</option>
</select>
</td>
</tr>
<tr>
<td width="7%">
<font size="4">
<b>2 </b>
</font>
</td>
<td width="8%"><select name="s11" size="1">
<option selected value="*">AND</option>
<option value="+">OR</option>
<option value="^">NOT</option>
</select></td>
<td width="55%">
<input type="text" size="45" maxlength="256" name="s22">
</td>
<td width="30%" nowrap>
<select name="s32" size="1">
<option value="braceopen200braceclose">Title</option>
<option value="braceopen300braceclose">Authors</option>
<option value="braceopen620braceclose">Keywords</option>
<option value="braceopen100braceclose">ISSN</option>
<option value="braceopen903braceclose">Source</option>
<option value="braceopen912braceclose">Journal Title</option>
<option value="braceopen440braceclose">Year</option>
<option selected value="">Any Where</option>
</select>
</td>
</tr>
<tr>
<td width="7%">
<font size="4"><b>3 </b></font>
</td>

<td width="8%"><select name="s12" size="1">
<option selected value="*">AND</option>
<option value="+">OR</option>
<option value="^">NOT</option>
</select></td>

<td width="55%">
<input type="text" size="45" maxlength="256" name="s23">
</td>
<td width="30%" nowrap>
<select name="s33" size="1">
<option value="braceopen200braceclose">Title</option>
<option value="braceopen300braceclose">Authors</option>
<option value="braceopen620braceclose">Keywords</option>
<option value="braceopen100braceclose">ISSN</option>
<option value="braceopen903braceclose">Source</option>
<option value="braceopen912braceclose">Journal Title</option>
<option value="braceopen440braceclose">Year</option>
<option selected value="">Any Where</option>
</select>
</td>
</tr>
```

```
                <tr>
                <td width="7%">
                <font size="4"><b>4 </b></font>
                </td>
                 <td width="8%">
                <select name="s13" size="1">
                 <option selected value="*">AND</option>
                 <option value="+">OR</option>
                 <option value="^">NOT</option>
                 </select>
                </td>
                 <td width="55%">
                <input type="text" size="45" maxlength="256" name="s24">
                </td>
                <td width="30%" nowrap>
                <select name="s34" size="1">
                 <option value="braceopen200braceclose">Title</option>
                 <option value="braceopen300braceclose">Authors</option>
                 <option value="braceopen620braceclose">Keywords</option>
                 <option value="braceopen100braceclose">ISSN</option>
                 <option value="braceopen903braceclose">Source</option>
                 <option value="braceopen912braceclose">Journal Title</option>
                 <option value="braceopen440braceclose">Year</option>
                 <option selected value="">Any Where</option>
                 </select>
                </td>
                 </tr>

                <tr>
                 <td align="center" colspan="4">

        <input type="reset" name="clear" value="Clear Form">
        <input type="submit" name="begin" value="Begin search">
        <font color="#0000FF">
        And Show
        <select name="maximum" size="1">
        <option value="10">10</option>
        <option selected value="20">20</option>
        <option value="50">50</option>
        <option value="100">100</option>
        <option value="200">200</option>
        </select>
        References
        </font>.
        </p>
        </td>
        </tr>
        <tr>
        <td>
        <input name="format" type="hidden" value="@plmx.pft">
        </tr>
        </table>

</form>
</td>
</tr>
</table>
<br> 

<p>

<hr>

</body>
</html>
```

Boolean Operators Menu

Field Restriction Menu

Reset & Search Buttons

Max. No. of Ref.s

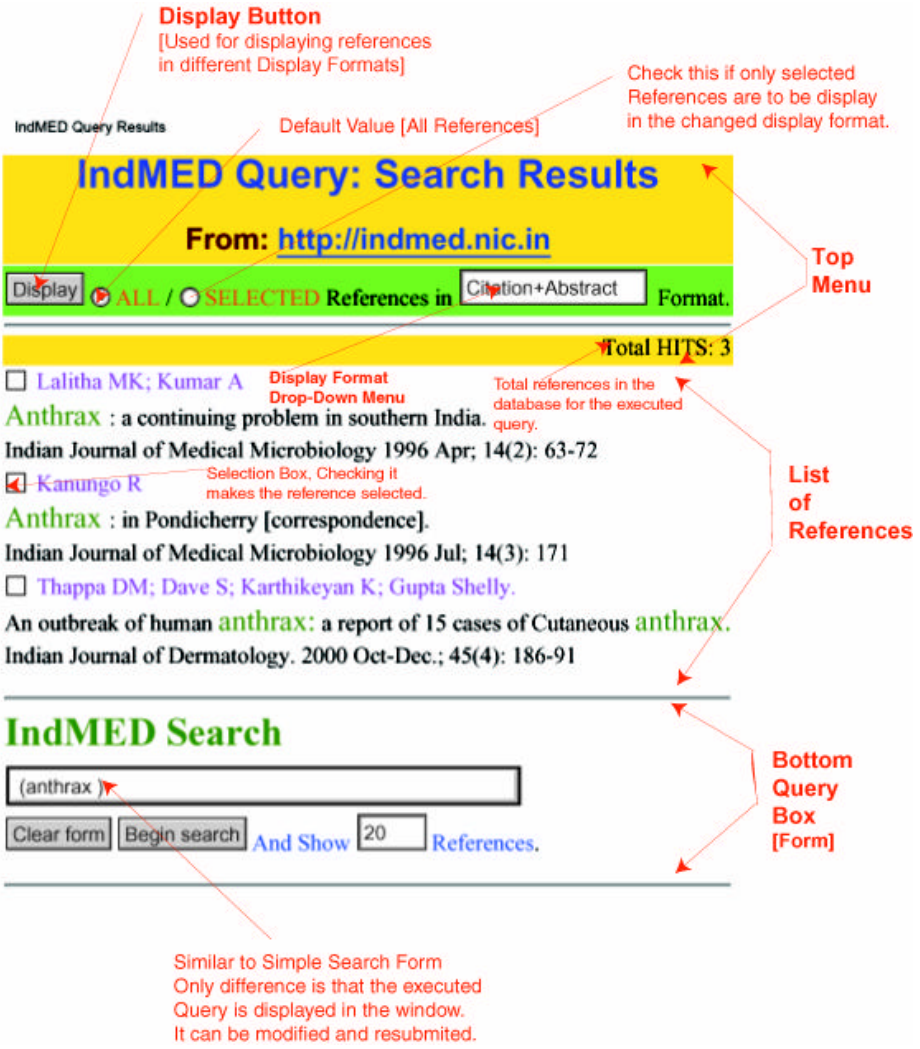
Hidden Format Value

1.4 Search Result Screen

The Search Result Screen would be *dynamically generated* by the CGI every-time a *"submit form"* (on any of the search form) action is initiated or "**display button**" (present on itself) is clicked.

1.4.1 Screen

The Search Result Screen is shown ahead:



It has the following three parts:

- **Top Menu**
- **List of References**
- **Bottom query Box**

1.4.1.1 Top Menu

Top Menu provides options to selection the display format from a drop down menu and displays either ALL or SELECTED (which ever option is selected from a radio button set) references when the "Display Button" is clicked.

The user has the following options to choose from:

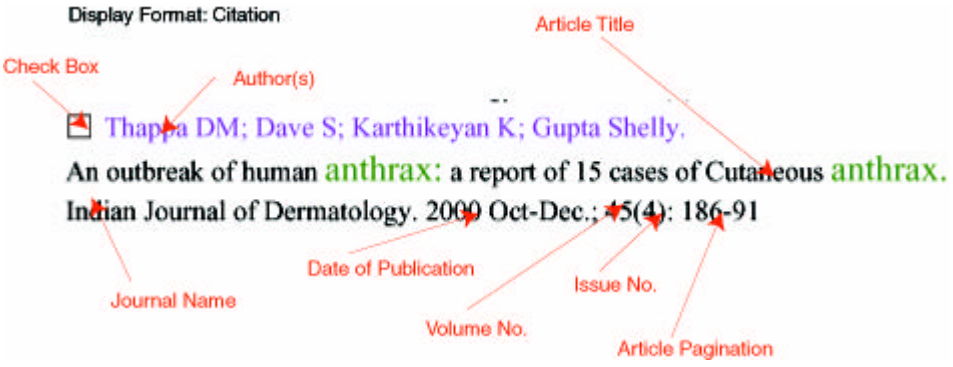
- **Citation Format**
- **Citation + Abstract Format**
- **Full Record Format**

1.4.1.2 List of References

The second and occupying the middle part of the screen is the List of References retrieved in response to the query. The references are displayed according to varying degree of details as per the format selected by the user. First time they are displayed in Citation Format.

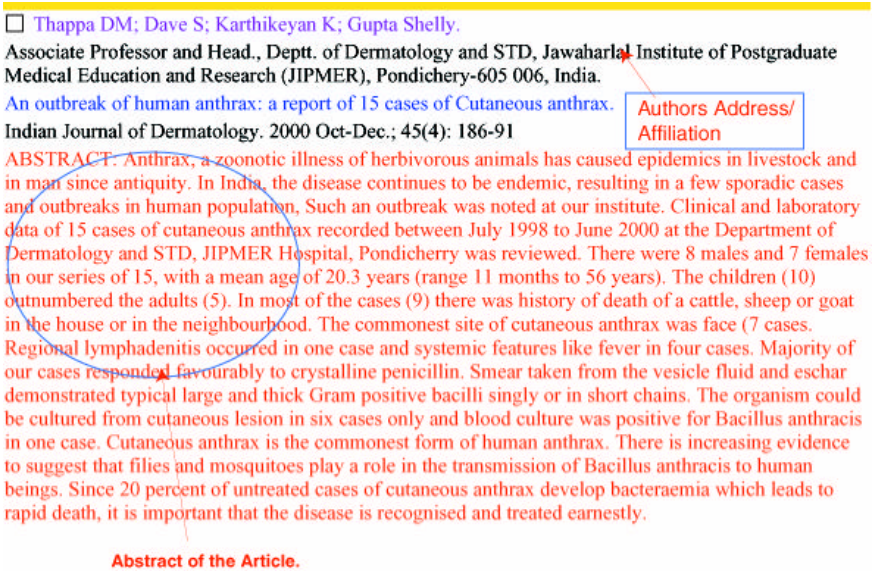
1.4.1.2.1 Citation Format

Citation Format display is shown ahead:



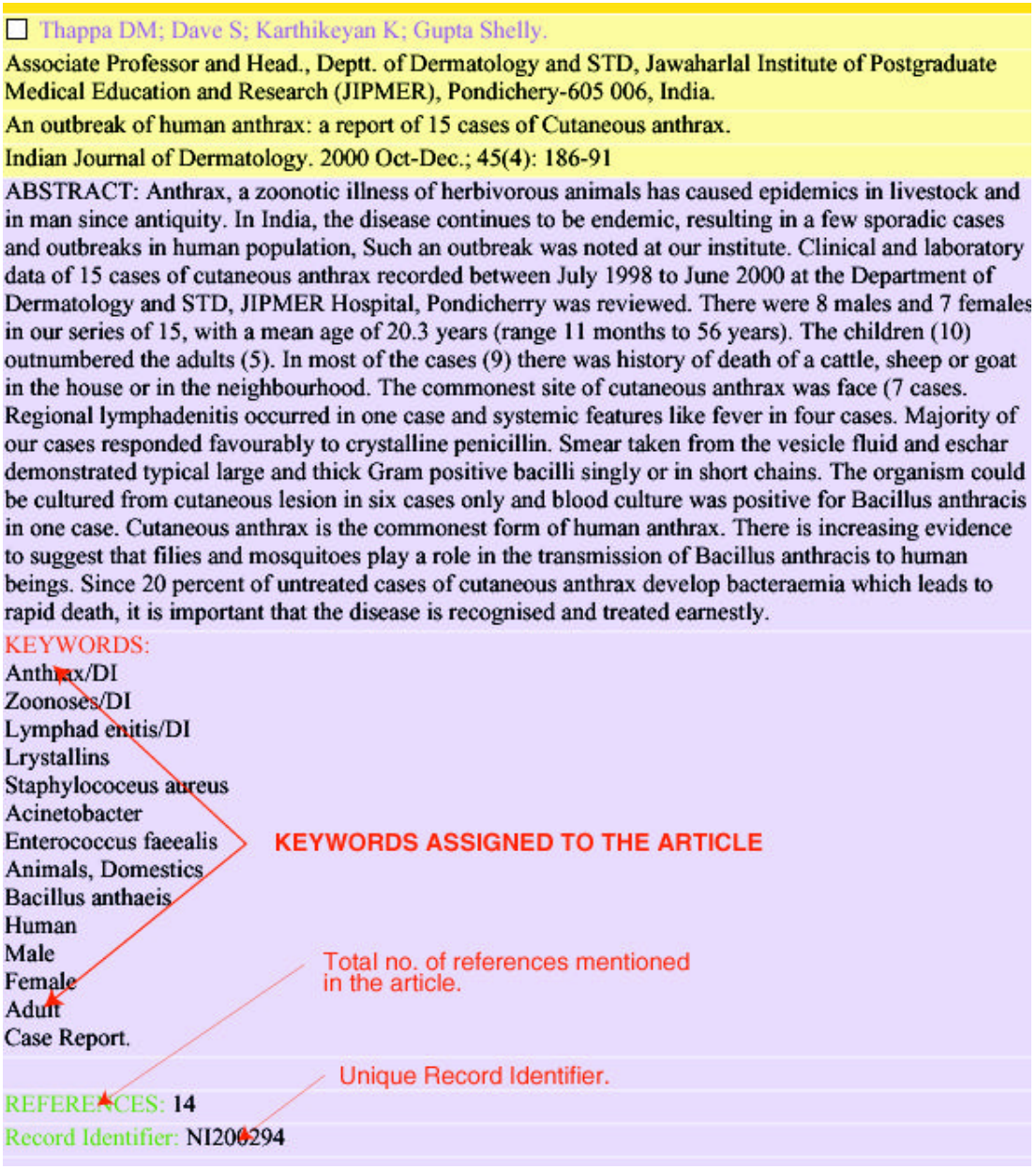
1.4.1.2.2 Citation+Abstract Format

Citation+Abstract Format display is shown ahead:



1.4.1.2.3 Full Format

Full Format display is shown ahead:



The *Top Menu* and the *List of References* together constitute a ***dynamically generated* html FORM** by the CGI. On clicking the Display Button, the perl script "imrs.pl" would execute on the server. The results would be shown again on the Search Result Screen but now in the format selected (default Citation+Abstract Format) by the user. The maximum number of reference and the previous executed query are carried forward as hidden html tag. Carrying forward of such data is necessary considering the state-less mode of the web.

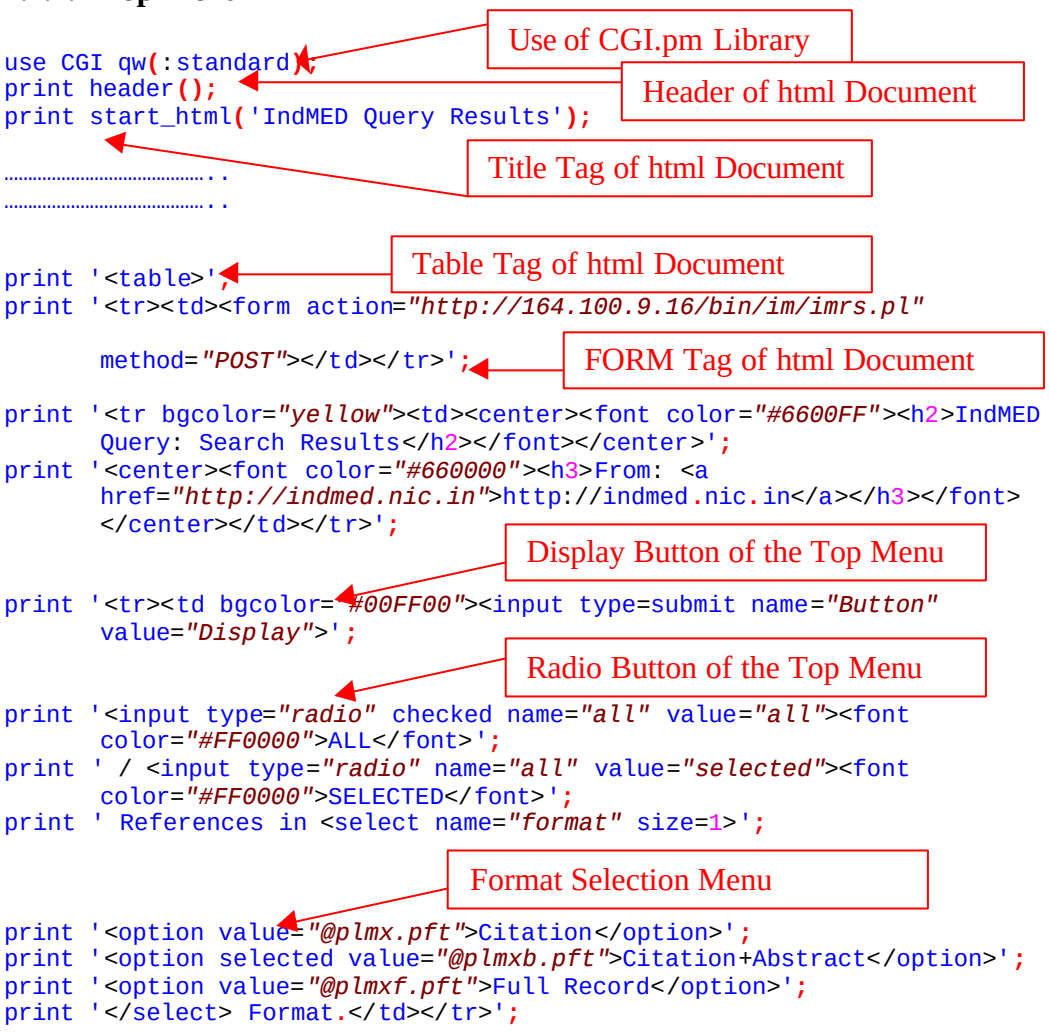
1.4.1.3 Bottom Query Box

This part of the *Search Result Screen* is also ***dynamically generated*** **html FORM** by the CGI. It is be very similar to the **Simple Search Form.** The only difference is that the executed query is already displayed in the form window. It provides the context to user about the query executed for the set of references being displayed. At the same time it allows to refine the query and execute it again by clicking submit button. On clicking the Submit Button, the perl script "imss.pl" would execute on the server. The results would be shown on the Search Result Screen in the default format i.e. Citation Format. The name of the display format would be send to the CGI as a hidden html tag.

1.4.2 Implementation of Search Result Screen

As already mentioned the html code for the *Search Result Screen* is not hard coded but it is generated dynamically by the CGI. The html code for Top Menu and the Bottom Query Box are generated from the perl scripts and the Display of the References are generated by the DBMS using various display formats coded in the "print formatting language" of the DBMS. The code thus generated is in turn passed on to the perl scripts. Relevant sections of the perl scripts and the "print format" of the database are given ahead with comments:

1.4.2.1 Top Menu




```
use CGI qw(:standard);
print header();
print start_html('IndMED Query Results');
.....
.....

print '<table>';
print '<tr><td><form action="http://164.100.9.16/bin/im/imrs.pl"
       method="POST"></td></tr>';
print '<tr bgcolor="yellow"><td><center><font color="#6600FF"><h2>IndMED
       Query: Search Results</h2></font></center>';
print '<center><font color="#660000"><h3>From: <a
       href="http://indmed.nic.in">http://indmed.nic.in</a></h3></font>
       </center></td></tr>';

print '<tr><td bgcolor="#00FF00"><input type=submit name="Button"
       value="Display">';

print '<input type="radio" checked name="all" value="all"><font
       color="#FF0000">ALL</font>';
print ' / <input type="radio" name="all" value="selected"><font
       color="#FF0000">SELECTED</font>';
print ' References in <select name="format" size=1>';

print '<option value="@plmx.pft">Citation</option>';
print '<option selected value="@plmxb.pft">Citation+Abstract</option>';
print '<option value="@plmxf.pft">Full Record</option>';
print '</select> Format.</td></tr>';
```

```
print '<tr><td><hr></td></tr>';
……
……..

$out =~s/(.*Hits)/<tr><td bgcolor=yellow align=right>Total HITS: /s;
……
……..
```

Total Hits Display

```
$out =~ s/($querymatch)/<font color = "green" size="4">\1<\/font>/ig;
……
……..
```

Patch for “search term” highlighting in Display

```
print $out;
print '<tr><td><input type="hidden" name="srch"
       value=',$search,'></td></tr>';
print '<tr><td><input type="hidden" name="maxi"
       value=',$maximum,'></td></tr>';
print '<tr><td></form></td></tr>';
```

Display of list of references

Hidden tag for passing on Query & No. of References.

1.4.2.2 Bottom Query Box

```
print '<tr><td><hr></td></tr>';
print '<tr><td><form action="http://164.100.9.16/bin/im/imss.pl"
       method="POST"></td></tr>';
print '<tr><td><font color="#008000" size="5"><b>IndMED
       Search</b></font></td></tr>';
```

FORM tag for Bottom Query Box

Search Window with Query

```
print '<tr><td><input type="text" size="45" maxlength="256" name="search"
       value=',$search,'></td></tr>';
print '<tr><td><input type="reset" name="reset2" value="Clear form">';
print '<input type="submit" name="submit2" value="Begin search">';
print '<font color="#0000FF">';
print 'And Show <select name="maximum" size="1">';
```

Drop Down Menu for Selecting Max. No. References to be displayed

```
print '<option value="10">10</option>';
print '<option selected value="20">20</option>';
print '<option value="50">50</option>';
print '<option value="100">100</option>';
print '<option value="200">200</option>';
print '</select>References</font>.</td></tr>';
print '<tr><td><input name="format" type="hidden" value="@plmx.pft">';
print '<tr><td><input type="hidden" name="srch"
       value=',$search,'></td></tr>';
print '<tr><td></form></td></tr>';
print '<tr><td><hr></td></tr>';
print '</table>';

END {
print end_html;
}
```

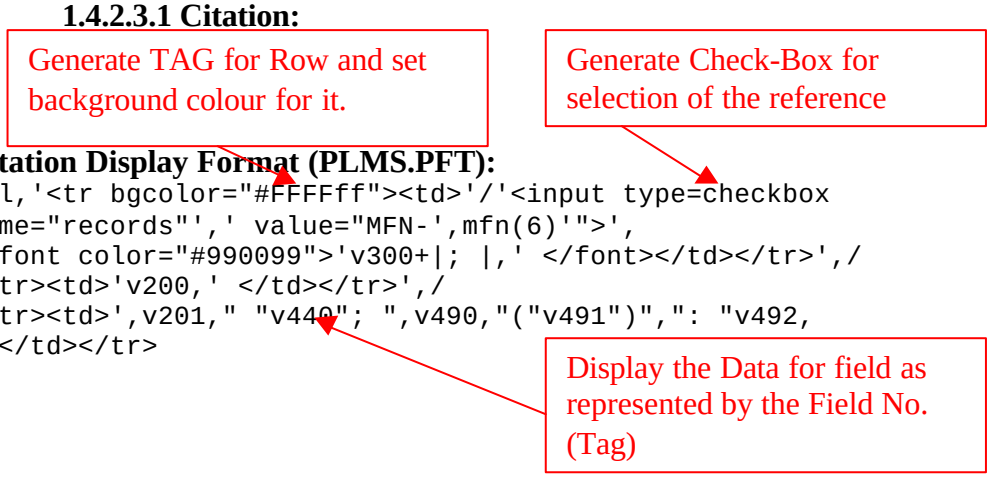
Hidden tag for passing on Query & Display Format

Generate End of Html Document Tag

1.4.2.3 List of References

The display of the references in the "List of References" in terms of the degree of details is controlled by the display formats defined in the DBMS. Such formats are text files containing commands in the "Formatting Language" of the DBMS. The language uses the numerical tag of the field to display its contents. Various types of strings can be appended to the contents of the fields. This feature makes it possible to attach the HTML Document Tags to the contents of the records and thus generate the required html code.

1.4.2.3.1 Citation:



Citation Display Format (PLMS.PFT):

```
mpl,'<tr bgcolor="#FFFFff"><td>'/'<input type=checkbox
name="records"',' value="MFN-',mfn(6)'">',
'<font color="#990099">'v300+|; |,' </font></td></tr>',/
'<tr><td>'v200,' </td></tr>',/
'<tr><td>',v201," "v440"; ",v490,"("v491")",": "v492,
' </td></tr>
```

1.4.2.3.2 Citation+Abstract:

Citation+Abstract `Display Format` (PLMXB.PFT):

```
mpl,'<tr bgcolor="#FFFFff"><td>'/'<input type=checkbox
name="records" value="MFN-',mfn(6)'">',
'<font color="#990099">'v300+|; |,' </font></td></tr>',/
'<tr><td>'v330,' </td></tr>',/
'<tr><td><font color="#0000AA"><STRONGE>'v200,'
</STRONGE></font></td></tr>',/
'<tr><td><STRONGE>',v201," "v440"; ",v490,"("v491")",": "v492,
' </STRONGE></td></tr>',/
'<tr bgcolor="FFFFFF"><td><font color="#ff0000">',
"ABSTRACT: "v600,' </font></td></tr>'/
'<tr><td><hr></td></tr>'
```

1.4.2.3.3 Full Record:

`Full Record Format(`*`PLMXF.PFT`*`):`

```
mpl,'<tr bgcolor="#FFFFBB"><td>'/'<input type=checkbox
name="records"',' value="MFN-',mfn(6)'">',
'<font color="#FF00FF">'v300+|; |,' </font></td></tr>',/
'<tr><td>'v330,' </td></tr>',/
'<tr><td>'v200,' </td></tr>',/
'<tr><td>',v201," "v440"; ",v490,"("v491")",": "v492,
' </td></tr>',/
'<tr bgcolor="FFDDFF"><td>'"ABSTRACT: "v600,' </td></tr>'/
'<tr><td>',if p(v620) then '<font color="#FF0000"> KEYWORDS:</font>
' fi,
"<br>"d620,(v620|<br>|),' </td></tr>'/,
```

```
'<tr><td>',if p(v621) then '<font color="#0000FF">OTHER
KEYWORDS:</font> ' fi,
"<br>"d621,(v621|<br>|),' </td></tr>'/,
'<tr><td>',if p(v703) then '<font color="#00AA00">
REFERENCES:</font> ' fi,v703,' </td></tr>'/,
'<tr><td>',"Indexer: "v800,' </td></tr>'/,
'<tr><td><hr></td></tr>'
```

1.4.3 Sample of the dynamically generated html code of the Search Result Screen

To give an idea what html code would be generated; a sample of the code is given below. It was generated by the CGI in response to “anthrax”.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html
       PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
       "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
<head>
<title>IndMED Query Results</title>
</head>
<body>
<table>
<tr>
<td>
<form action="http://164.100.9.16/bin/im/imrs.pl" method="POST">
</td>
</tr>
<tr bgcolor="yellow">
<td>
<center><font color="#6600FF">
<h2>IndMED Query: Search Results</h2>
</font></center>
<center>
<font color="#660000">
<h3>From: <a href="http://indmed.nic.in">http://indmed.nic.in</a></h3>
</font>
</center>
</td>
</tr>
<tr>
<td bgcolor="#00FF00">
<input type=submit name="Button" value="Display">
<input type="radio" checked name="all" value="all">
<font color="#FF0000">ALL</font> /
<input type="radio" name="all" value="selected">
<font color="#FF0000">SELECTED</font> References in
<select name="format" size=1>
<option value="@plmx.pft">Citation</option>
<option selected value="@plmxb.pft">Citation+Abstract</option>
<option value="@plmxf.pft">Full Record</option>
</select> Format.
</td>
</tr>
<tr>
<td>
<hr>
</td>
</tr>
<tr>
<td bgcolor=yellow align=right>Total HITS: 3
```

```
<tr bgcolor="#FFFFff">
<td>
<input type=checkbox name="records" value="MFN-000834">
<font color="#990099">
Lalitha MK; Kumar A </font></td></tr>
<tr>
<td>
<font color = "green" size="4">Anthrax </font>
: a continuing problem in southern India.
</td></tr>
<tr>
<td>
Indian Journal of Medical Microbiology 1996 Apr; 14(2): 63-72
 </td>
 </tr>
 <tr bgcolor="#FFFFff">
 <td>
<input type=checkbox name="records" value="MFN-008056">
<font color="#990099">
Kanungo R </font>
</td>
</tr>
<tr>
<td>
<font color = "green" size="4">Anthrax </font>
: in Pondicherry [correspondence].
</td></tr>
<tr>
<td>
Indian Journal of Medical Microbiology 1996 Jul; 14(3): 171
</td></tr>
<tr bgcolor="#FFFFff"><td>
<input type=checkbox name="records" value="MFN-010590">
<font color="#990099">
Thappa DM; Dave S; Karthikeyan K; Gupta Shelly. </font>
</td>
</tr>
<tr>
<td>An outbreak of human
<font color = "green" size="4">anthrax:</font>
 a report of 15 cases of Cutaneous
<font color = "green" size="4">anthrax.</font>
</td></tr>
<tr>
<td>
Indian Journal of Dermatology. 2000 Oct-Dec.; 45(4): 186-91
</td></tr>
<tr>
<td>
<input type="hidden" name="srch" value="(anthrax )">
</td>
</tr>
<tr>
<td>
<input type="hidden" name="maxi" value=50>
</td>
</tr>
<tr>
<td>
</form>
</td>
</tr>
<tr>
<td>
<hr></td>
</tr>
<tr>
<td>
```

```
<form action="http://164.100.9.16/bin/im/imss.pl" method="POST">
</td>
</tr>
<tr>
<td>
<font color="#008000" size="5"><b>IndMED Search</b></font>
</td>
</tr>
<tr>
<td>
<input type="text" size="45" maxlength="256" name="search" value="(anthrax
)">
</td>
</tr>
<tr>
<td>
<input type="reset" name="reset2" value="Clear form">
<input type="submit" name="submit2" value="Begin search">
<font color="#0000FF">
And Show
<select name="maximum" size="1">
<option value="10">10</option>
<option selected value="20">20</option>
<option value="50">50</option>
<option value="100">100</option>
<option value="200">200</option>
</select>
References
</font>.
</td>
</tr>
<tr>
<td>
<input name="format" type="hidden" value="@plmx.pft">
<tr>
<td><input type="hidden" name="srch" value="(anthrax )">
</td>
</tr>
<tr>
<td>
</form>
</td>
</tr>
<tr>
<td><hr></td>
</tr>
</table>
</body>
</html>
```

2. Web-Database Connectivity

The connectivity between the web-based user interface and the database have been implemented by "Common Gateway Interface (CGI)". It consists of three scripts written in PERL 5 (ActivePERL) for Windows NT Server running IIS. The scripts are given ahead. The comments appears both as in the source code (in green colour) after the commenting symbol hash mark ("#") as well as text boxes (red colour) added here for description.

2.1 IMSS.PL

This script is executed by the Html Forms in the Web-Based User Interface namely the **Simple Search Form, Bottom Query Box (**of **Search Result Screen)** and the Minimal **Search Form.** It extracts data from the database and creates a web document (Search Result Screen) dynamically.

```
#!/usr/local/bin/perl

use CGI qw(:standard);
print header();
print start_html('IndMED Query Results');

# Script for Query String Generation START:
$query = "";

# Check if atleast a single query is there.

if (param ('search'))
   { # START - Check if a query is present
   $string = param ('search');
   $restrict = " ";
    # END - Check if a query is present
   #Process the line of the search form
   $line = &createquery($string,$restrict);
   $query = $line;
   #print "<br><pre>THE QUERY LINE IS: $line ",'<br>',"QUERY IS
       :$query",'</pre>';
   }
   else
   {
   print '<p color="red">' , em ("PLEASE ENTER A QUERY IN THE SEARCH
FORM"),'</p>' ;
   die;
   }#END - If Check Brace

# Check if Second line is present.


#-----------------------------------------------------
#   SUBROUTINE Start
#-----------------------------------------------------
sub createquery { # Start of Subroutine "Create Query"
#Query String formulation subroutine.
#Check if the query starts with an operator
if ($string =~ /^ *not |^ *and |^ *or /)
```

Use of CGI.pm Library

Header of html Document

Title Tag of html Document

Error Trap: Empty Query

Error Message

Error Trap: Query starting with operator

Error Message

```
                    {
                    print "<br>SEARCH CAN'T START WITH AN OPERATOR";
                    die;
                    }
#END - Check if the query starts with an operator
my ($string,$restrict) = @_;
my (@open,@close);
my ($pattern,$patternrev);
#print '<br>',"$string"," : FIRST, String as Received.",'<br>';


$string =~ s/ not /$restrict ^ /gi;
$string =~ s/ and /$restrict * /gi;
$string =~ s/ or /$restrict + /gi;

#print $string, " : SECOND, And Or Not Replaced & Qualifier
attached.",'<br>';


if ($string =~ /\(|\)/)
   {
   $_ = $string;
   @open =  /\(/g;
   @close = /\)/g;
   if ($#open != $#close)
   {
   print $#open, ":", $#close,'<br>';
   print ("PLEASE ENTER BALANCED PARENTHESIS IN QUERY.");
   die;
   }
   $string =~ s/\)/ paren /g; # Replace Parentheses
   #print "$string"," : THIRD, Parenthesis are present and replaced with
PAREN",'<br>';
   $pattern = "paren +?" . $restrict;
   $patternrev = $restrict . " paren ";
   $string .= $restrict;
   #print $pattern," : PATTERN, Which is to be looked",'<br>';
   #print $string, " : STRING, in which the above PATTERN is to be
       matched",'<br>';
   if ($string =~ /$pattern/)
                    {
                    #print "MATCH THERE",'<br>';
                    }
                    else
                    {
                    #print "NO MATCH",'<br>';
                    }
                    while ($string =~ /$pattern/)
                           {
                           $string =~ s /$pattern/$patternrev/g;
                           }
                    #print $string, " : FORTH: String after patter
reversed.",'<br>';
       $string =~ s/paren / \)/g; # Parenthese re-introduced
       }
else
   {
   $string .= $restrict;
   }
#Now Convert as required by IM.

$string =~ s/braceopen/\/\(/g;
$string =~ s/braceclose/\)/g;
#print $string," :FINAL, This is Done";
return ('('.$string.')');
} #End of Subroutine "Create Query"
#---------------------------------------------------
#   SUBROUTINE END
```

Replace operator with equivalent symbols of the DBMS

Error Trap: Check Balance of Parenthesis

```
#-----------------------------------------------------
# Script for Query String Generation END:

$search = '"'.$query.'"'; #Query is to passed in quotes
$format = param ('format');
$maximum = param ('maximum');
print '<table>';
```

Start Generation of Top Menu of the Search Results Screen.

```
print '<tr><td><form action="http://164.100.9.16/bin/im/imrs.pl"
       method="POST"></td></tr>';
print '<tr bgcolor="yellow"><td><center><font color="#6600FF"><h2>IndMED
       Query: Search Results</h2></font></center>';
print '<center><font color="#660000"><h3>From: <a
       href="http://indmed.nic.in">http://indmed.nic.in</a></h3></font>
       </center></td></tr>';
print '<tr><td bgcolor="#00FF00"><input type=submit name="Button"
       value="Display">';
print '<input type="radio" checked name="all" value="all"><font
       color="#FF0000">ALL</font>';
print ' / <input type="radio" name="all" value="selected"><font
       color="#FF0000">SELECTED</font>';
print ' References in <select name="format" size=1>';
print '<option value="@plmx.pft">Citation</option>';
print '<option selected value="@plmxb.pft">Citation+Abstract</option>';
print '<option value="@plmxf.pft">Full Record</option>';
print '</select> Format.</td></tr>';
print '<tr><td><hr></td></tr>';
```

End: Generation of Top Menu of the Search Results Screen.

```
#START: THESE LINES ARE ADDED TO MAKE IT WORK ON INDMED SERVER
```

Query to the DBMS.

Source Code removed to safeguard the security of INDMED Server. Please email at esukhdev@hotmail.com to ask for it.

```
#END: THESE LINES ARE ADDED TO MAKE IT WORK ON INDMED SERVER
$out =~s/(.*Hits=)/<tr><td bgcolor=yellow align=right>Total HITS: /s; #To
print total hits.
#print "HERE IS THE MATCH: $1";
#print "HERE IS THE QUERY: $search";
# Start for showing the matched words
$querymatch = $search;
$querymatch =~ s/\W|\d|\s/ /ig;
$querymatch =~ s/^ +| +$//g;
if ($querymatch =~/MFN/i)
   {
   $mp='[ .,:\s]';
   }
   else
   {
   $mp='[ .,:-\s]';
   }
$querymatch =~ s/ +/$mp|/g;
if ($querymatch ne "")
   { # Atleast One Word should be there.
```

Read values for Format and Max. No. Ref.

Highlighting of Search Terms.

```
    $querymatch .= $mp;
    #print "<br>The Query words: $querymatch";
    $out =~ s/($querymatch)/<font color = "green" size="4">\1<\/font>/ig;
    }
# End for showing the matched words


print $out;
```

Output List of Reference to the Screen.

Start: Bottom Query Box on Display Screen.

```
print '<tr><td><input type="hidden" name="srch"
       value=',$search,'></td></tr>';
print '<tr><td><input type="hidden" name="maxi"
       value=',$maximum,'></td></tr>';
print '<tr><td></form></td></tr>';

print '<tr><td><hr></td></tr>';
print '<tr><td><form action="http://164.100.9.16/bin/im/imss.pl"
       method="POST"></td></tr>';
print '<tr><td><font color="#008000" size="5"><b>IndMED
       Search</b></font></td></tr>';
```

FORM tag for Bottom Query Box

Search Window with Query

```
print '<tr><td><input type="text" size="45" maxlength="256" name="search"
       value=',$search,'></td></tr>';
print '<tr><td><input type="reset" name="reset2" value="Clear form">';
print '<input type="submit" name="submit2" value="Begin search">';
print '<font color="#0000FF">';
print 'And Show <select name="maximum" size="1">';
```

Drop Down Menu for Selecting Max. No. References to be displayed

```
print '<option value="10">10</option>';
print '<option selected value="20">20</option>';
print '<option value="50">50</option>';
print '<option value="100">100</option>';
print '<option value="200">200</option>';
print '</select>References</font>.</td></tr>';
print '<tr><td><input name="format" type="hidden" value="@plmx.pft">';
print '<tr><td><input type="hidden" name="srch"
       value=',$search,'></td></tr>';
print '<tr><td></form></td></tr>';
print '<tr><td><hr></td></tr>';
print '</table>';

END {

print end_html;
}
```

Hidden tag for passing on Query & Display Format

Generate End of Html Document Tag

2.2 IMAS.PL

This script is executed by the Html Forms in the Web-Based User Interface namely the **Advanced Search Form.** It extracts data from the database and creates a web document (Search Result Screen) dynamically.

```
#!/usr/local/bin/perl

use CGI qw(:standard);
print header();
print start_html('IndMED Query Results');

       # Script for Query String Generation START:
$query = "";

# Check if atleast a single query is there.

if (param ('s21'))
   { #Outer If Check Brace
   $string = param ('s21');
        # Check if qualifier Specified
        if (param ('s31'))
              {#Inner If Check Brace
              $restrict = param ('s31');
              }
              else
              {
              $restrict = " ";
              } #Inner If Check Brace
        # End Checking
   #Process the First line of the search form
   $line = &createquery($string,$restrict);
   $query = $line;
   #print "<br><pre>THE QUERY LINE IS: $line ",'<br>',"QUERY IS
:$query",'</pre>';
   }
   else
   {
   print '<p color="red">' , qm ("PLEASE ENTER A QUERY IN THE SEARCH
       FORM"),'</p>' ;
   die;
   }#Outer If Check Brace
# End Checking

# Check if Second line is present.

if (param ('s22'))
   { #Outer If Check Brace
   $string = param ('s22');
        # Check if qualifier Specified
        if (param ('s32'))
              {#Inner If Check Brace
              $restrict = param ('s32');
              }
              else
              {
              $restrict = " ";
              } #Inner If Check Brace
        # End Checking
   #Process the Second line of the search form
```

Use of CGI.pm Library

Header of html Document

Title Tag of html Document

Error Trap: Empty Query

Assign Field Restriction else set it Null String.

Error Message

Check if Second Line is filled with query

```
   $line = &createquery($string,$restrict);
   #Get Operator
   $operator = param ('s11');
   $query .= " ".$operator." ".$line;
   #print "<br><pre>THE QUERY LINE IS: $line ",'<br>',"QUERY IS
       :$query",'</pre>';
}#Outer If Check Brace
# End Second Line

# Check if Third line is present.

if (param ('s23'))
   { #Outer If Check Brace
   $string = param ('s23');
        # Check if qualifier Specified
        if (param ('s33'))
              {#Inner If Check Brace
              $restrict = param ('s33');
              }
              else
              {
              $restrict = " ";
              } #Inner If Check Brace
        # End Checking
   #Process the Third line of the search form
   $line = &createquery($string,$restrict);
   #Get Operator
   $operator = param ('s12');
   $query .= $operator.$line;
  #print "<br><pre>THE QUERY LINE IS: $line ",'<br>',"QUERY IS
:$query",'</pre>';
}#Outer If Check Brace
# End Third Line
# Check if Fourth line is present.

if (param ('s24'))
   { #Outer If Check Brace
   $string = param ('s24');
        # Check if qualifier Specified
        if (param ('s34'))
              {#Inner If Check Brace
              $restrict = param ('s34');
              }
              else
              {
              $restrict = " ";
              } #Inner If Check Brace
        # End Checking
   #Process the Fourth line of the search form
   $line = &createquery($string,$restrict);
   #Get Operator
   $operator = param ('s13');
   $query .= $operator.$line;
   print "<br><pre>THE QUERY LINE IS: $line ",'<br>',"QUERY IS
       :$query",'</pre>';
}#Outer If Check Brace
# End Fourth Line
#-----------------------------------------------------
#   SUBROUTINE Start
#-----------------------------------------------------
sub createquery { # Start of Subroutine "Create Query"
#Query String formulation subroutine.
#Check if the query starts with an operator
if ($string =~ /^ *not |^ *and |^ *or /)
                  {
                  print "<br>SEARCH CAN'T START WITH AN OPERATOR";
                  die;
                  }
```

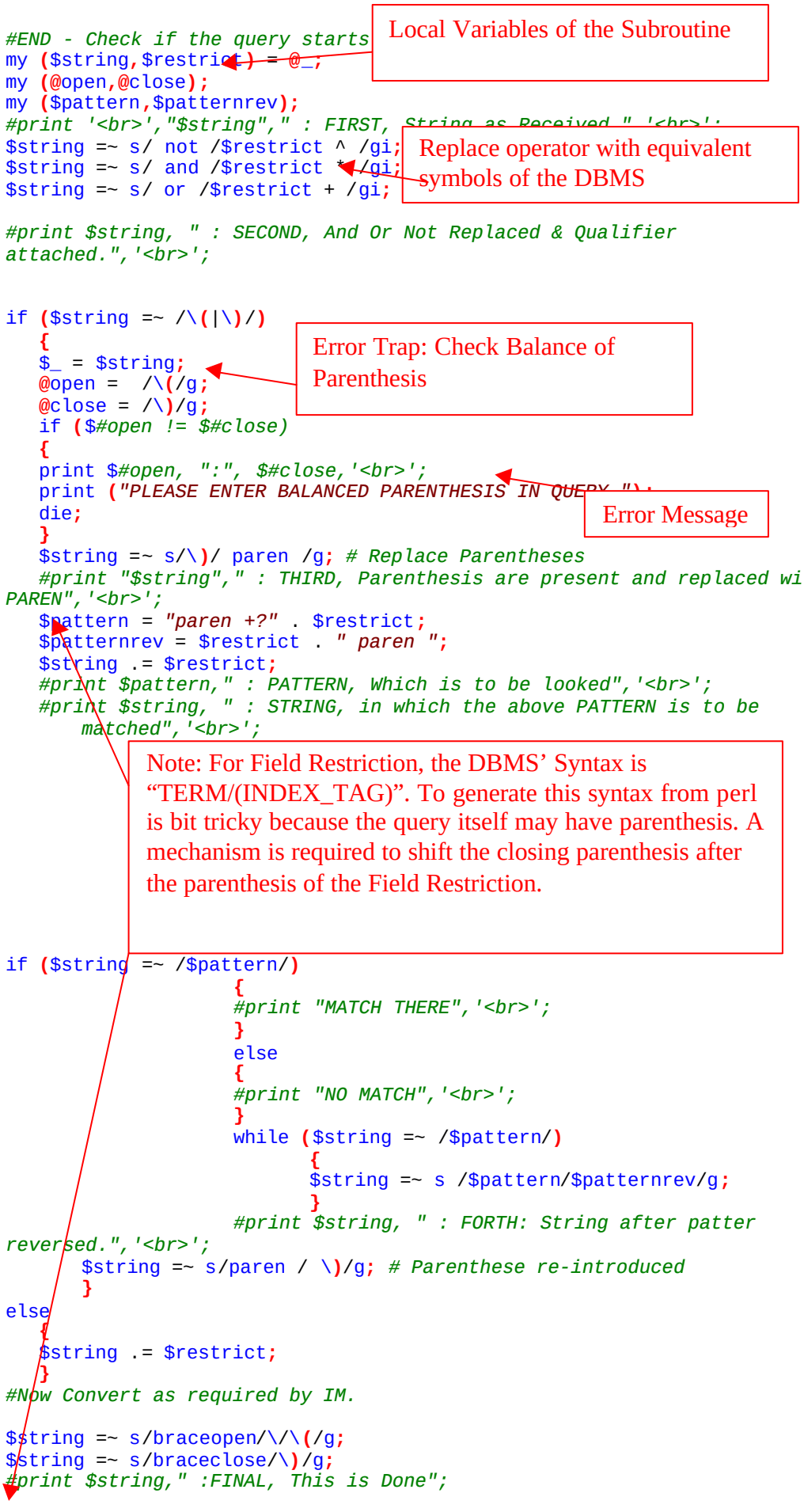
If Second Line is filled with query then get Boolean Operator

Check if Third Line is filled with query

If Third Line is filled with query then get Boolean Operator and So on.

Error Trap: Query starting with operator

Error Message




```
#END - Check if the query starts
my ($string,$restrict) = @_;
my (@open,@close);
my ($pattern,$patternrev);
#print '<br>',"$string"," : FIRST, String as Received ",'<br>';
$string =~ s/ not /$restrict ^ /gi;
$string =~ s/ and /$restrict * /gi;
$string =~ s/ or /$restrict + /gi;

#print $string, " : SECOND, And Or Not Replaced & Qualifier
attached.",'<br>';


if ($string =~ /\(|\)/)
   {
   $_ = $string;
   @open =  /\(/g;
   @close = /\)/g;
   if ($#open != $#close)
   {
   print $#open, ":", $#close,'<br>';
   print ("PLEASE ENTER BALANCED PARENTHESIS IN QUERY ");
   die;
   }
   $string =~ s/\)/ paren /g; # Replace Parentheses
   #print "$string"," : THIRD, Parenthesis are present and replaced with
PAREN",'<br>';
   $pattern = "paren +?" . $restrict;
   $patternrev = $restrict . " paren ";
   $string .= $restrict;
   #print $pattern," : PATTERN, Which is to be looked",'<br>';
   #print $string, " : STRING, in which the above PATTERN is to be
       matched",'<br>';


if ($string =~ /$pattern/)
                  {
                  #print "MATCH THERE",'<br>';
                  }
                  else
                  {
                  #print "NO MATCH",'<br>';
                  }
                  while ($string =~ /$pattern/)
                         {
                         $string =~ s /$pattern/$patternrev/g;
                         }
                  #print $string, " : FORTH: String after patter
reversed.",'<br>';
       $string =~ s/paren / \)/g; # Parenthese re-introduced
       }
else
   {
   $string .= $restrict;
   }
#Now Convert as required by IM.

$string =~ s/braceopen/\/\(/g;
$string =~ s/braceclose/\)/g;
#print $string," :FINAL, This is Done";
```

```
return ('('.$string.')');
} #End of Subroutine "Create Query"
#-----------------------------------------------------
#   SUBROUTINE END
#-----------------------------------------------------
# Script for Query String Generation END:
```

The Code ahead is for generation of the three parts of the Search Result Screen i.e. Top Menu, List of References and the Bottom Query Box. The Code is Similar to the IMSS.PL Script as discussed earlier.

```
$search = '"'.$query.'"'; #Query is to passed in quotes
$format = param ('format');
$maximum = param ('maximum');
print '<table>';
print '<tr><td><form action="http://164.100.9.16/bin/im/imrs.pl"
       method="POST"></td></tr>';
print '<tr bgcolor="yellow"><td><center><font color="#6600FF"><h2>IndMED
       Query: Search Results</h2></font></center>';
print '<center><font color="#660000"><h3>From: <a
       href="http://indmed.nic.in">http://indmed.nic.in</a></h3></font>
       </center></td></tr>';
print '<tr><td bgcolor="#00FF00"><input type=submit name="Button"
       value="Display">';
print '<input type="radio" checked name="all" value="all"><font
       color="#FF0000">ALL</font>';
print ' / <input type="radio" name="all" value="selected"><font
       color="#FF0000">SELECTED</font>';
print ' References in <select name="format" size=1>';
print '<option value="@plmx.pft">Citation</option>';
print '<option selected value="@plmxb.pft">Citation+Abstract</option>';
print '<option value="@plmxf.pft">Full Record</option>';
print '</select> Format.</td></tr>';
print '<tr><td><hr></td></tr>';
#START: THESE LINES ARE ADDED TO MAKE IT WORK ON INDMED SERVER
```

Source Code removed to safeguard the security of INDMED Server. Please email at esukhdev@hotmail.com to ask for it.

```
#END: THESE LINES ARE ADDED TO MAKE IT WORK ON INDMED SERVER
$out =~s/(.*Hits=)/<tr><td bgcolor=yellow align=right>Total HITS: /s;
#To print total hits.
#print "HERE IS THE MATCH: $1";
#print "HERE IS THE QUERY: $search";
# Start for showing the matched words
$querymatch = $search;
$querymatch =~ s/\W|\d|\s/ /ig;
$querymatch =~ s/^ +| +$//g;
if ($querymatch =~/MFN/i)
   {
   $mp='[ .,:\s]';
   }
   else
   {
   $mp='[ .,:-\s]';
   }
$querymatch =~ s/ +/$mp|/g;
if ($querymatch ne "")
```

```
   { # Atleast One Word should be there.
   $querymatch .= $mp;
   #print "<br>The Query words: $querymatch";
   $out =~ s/($querymatch)/<font color = "green" size="4">\1<\/font>/ig;
   }
# End for showing the matched words
print $out;
print '<tr><td><input type="hidden" name="srch"
       value=',$search,'></td></tr>';
print '<tr><td><input type="hidden" name="maxi"
       value=',$maximum,'></td></tr>';
print '<tr><td></form></td></tr>';
print '<tr><td><hr></td></tr>';
print '<tr><td><form action="http://164.100.9.16/bin/im/imss.pl"
       method="POST"></td></tr>';
print '<tr><td><font color="#008000" size="5"><b>IndMED
       Search</b></font></td></tr>';
print '<tr><td><input type="text" size="45" maxlength="256" name="search"
       value=',$search,'></td></tr>';
print '<tr><td><input type="reset" name="reset2" value="Clear form">';
print '<input type="submit" name="submit2" value="Begin search">';
print '<font color="#0000FF">';
print 'And Show <select name="maximum" size="1">';
print '<option value="10">10</option>';
print '<option selected value="20">20</option>';
print '<option value="50">50</option>';
print '<option value="100">100</option>';
print '<option value="200">200</option>';
print '</select>References</font>.</td></tr>';
print '<tr><td><input name="format" type="hidden" value="@plmx.pft">';
print '<tr><td><input type="hidden" name="srch"
       value=',$search,'></td></tr>';
print '<tr><td></form></td></tr>';
print '<tr><td><hr></td></tr>';
print '</table>';

END {
#print "<br> It works!!!!!!!!";
print end_html;
}
```

2.3 IMRS.PL

The dynamically generated Html Form - **Top Menu** on the **Search Result Screen**, executes this script**.** It extracts data from the database in one of the two ways depending upon the value of the radio buttons "ALL" or "SELECTED" of the Top Menu". In case "ALL" is ticked, it re-executes the query now with new Display Format selected. When "SELECTED" radio button is ticked, it picks up the unique code numbers of the selected references and generates a Boolean "OR"ed query string. This query is then executed with the new Display Format selected by the user. The results again dynamically generate the web document (Search Result Screen).

```
#!/usr/local/bin/perl
use CGI qw(:standard);
print header;
print start_html('IndMED Query Results');
```

This module picks – up the unique codes prefixed to the selected references and generates a query string by putting a "+" sign ("OR" Boolean Operator) between them.

```
@records = param ('records');
$srch = '"'. param ('srch').'"';
$sel=$records[0];
for($i=1; $i <= $#records; $i++) {
$sel = "$sel" . '+' . "$records[$i]";
}
#print "$sel";

if (param ('all') ne "all") {
if ($sel) {
$search = $sel;
}

else {

print "NO REFERENCE HAS BEEN SELECTED, GO BACK AND SELECT!!!";
die;
}
#print  "\$search-sel is $search";

}
else {
$search = $srch;
}
```

If the radio button "ALL" is not clicked, then assign the selected references query to the variable holding the query.

However if the radio button "ALL" is not clicked, and no reference is selected as well then error would be generated.

However if the radio button "ALL" is clicked then the previous query is assigned to the query variable.

Rest of the query is similar to the PLAS.PL as discussed above.

```
$format = param ('format');
$maximum = param ('maxi');
print '<table>';
print '<tr><td><form action="http://164.100.9.16/bin/im/imrs.pl"
       method="POST"></td></tr>';
print '<tr bgcolor="yellow"><td><center><font color="#6600FF"><h2>IndMED
       Query: Search Results</h2></font></center>';
print '<center><font color="#660000"><h3>From: <a
       href="http://indmed.nic.in">http://indmed.nic.in</a></h3></font>
```

```
       </center> </td></tr>';

if ($search) {
print '<tr><td bgcolor="#00FF00"><input type=submit name="Button"
       value="Display">';
print '<input type="radio" checked name="all" value="all"><font
       color="#FF0000">ALL</font>';
print ' / <input type="radio" name="all" value="selected"><font
       color="#FF0000">SELECTED</font>';
print ' References in <select name="format" size=1>';
print '<option value="@plmx.pft">Citation</option>';
print '<option selected value="@plmxb.pft">Citation+Abstract</option>';
print '<option value="@plmxf.pft">Full Record</option>';
print '</select> Format.</td></tr>';
print '<tr><td><hr></td></tr>';
#START: THESE LINES ARE ADDED TO MAKE IT WORK ON INDMED SERVER
```

Source Code removed to safeguard the security of INDMED Server. Please email at esukhdev@hotmail.com to ask for it.

```
#END: THESE LINES ARE ADDED TO MAKE IT WORK ON INDMED SERVER
$out =~s/(.*Hits=)/<tr><td bgcolor=yellow align=right>Total HITS: /s; #To
print total hits.
#print "HERE IS THE MATCH: $1";
#print "HERE IS THE QUERY: $search";
# Start for showing the matched words
$querymatch = $search;
$querymatch =~ s/\W|\d|\s/ /ig;
$querymatch =~ s/^ +| +$//g;
if ($querymatch =~/MFN/i)
   {
   $mp='[ .,:\s]';
   }
   else
   {
   $mp='[ .,:-\s]';
   }
$querymatch =~ s/ +/$mp|/g;
if ($querymatch ne "")
   { # Atleast One Word should be there.
   $querymatch .= $mp;
   #print "<br>The Query words: $querymatch";
   $out =~ s/($querymatch)/<font color = "green" size="4">\1<\/font>/ig;
   }
# End for showing the matched words
print $out;
}
else
{
print '<tr><td bgcolor="red">' , em ("PLEASE ENTER A QUERY IN THE SEARCH
FORM"),'</tr></td>' ;
}
print '<tr><td><input type="hidden" name="srch"
       value=',$search,'></td></tr>';
print '<tr><td><input type="hidden" name="maxi"
       value=',$maximum,'></td></tr>';
print '<tr><td></form></td></tr>';
print '<tr><td><hr></td></tr>';
print '<tr><td><form action="http://164.100.9.16/bin/im/imss.pl"
       method="POST"></td></tr>';
print '<tr><td><font color="#008000" size="5"><b>IndMED
       Search</b></font></td></tr>';
print '<tr><td><input type="text" size="45" maxlength="256" name="search"
       value=',$search,'></td></tr>';
print '<tr><td><input type="reset" name="reset" value="Clear form">';
```

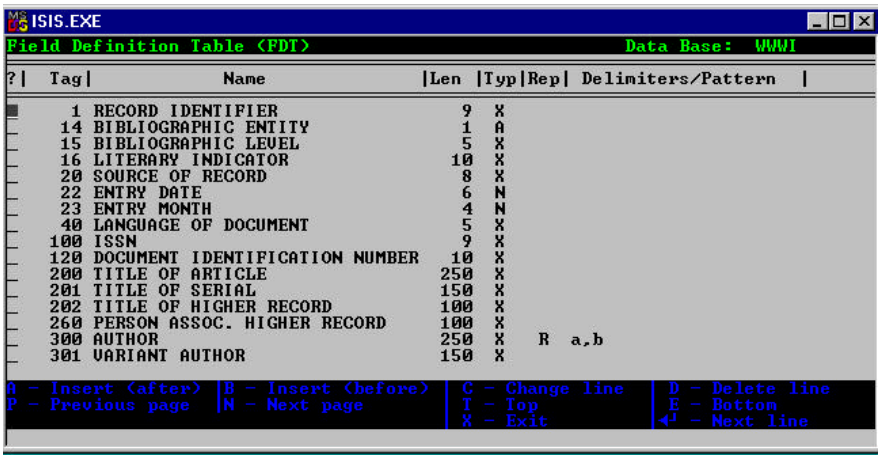
```
print '<input type="submit" name="submit" value="Begin search">';
print '<font color="#0000FF">';
print 'And Show <select name="maximum" size="1">';
print '<option value="10">10</option>';
print '<option selected value="20">20</option>';
print '<option value="50">50</option>';
print '<option value="100">100</option>';
print '<option value="200">200</option>';
print '</select>References</font>.</td></tr>';
print '<tr><td><input name="format" type="hidden" value="@plmx.pft">';
print '<tr><td></form></td></tr>';
print '<tr><td><hr></td></tr>';
print '</table>';
system ("del tmp???.???");
END {
#print "<br> It works!!!!!!!!";
print end_html;
}
```

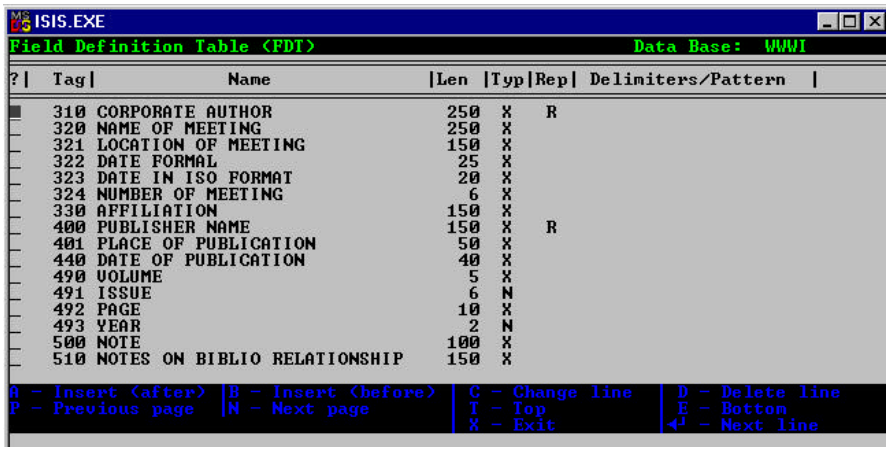
3. The Database

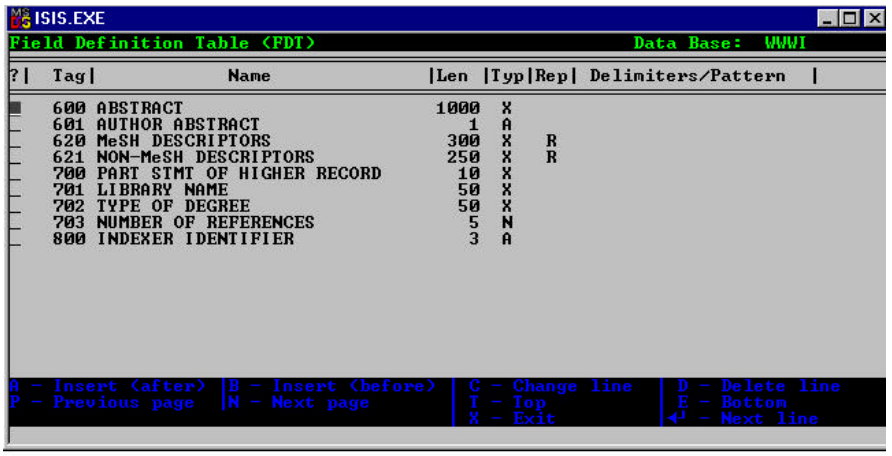
The IndMED database is under DBMS called CDS/ISIS Version 3.07. It does not provide as such for publishing the database over web. However a utility is available for retrieving data at system command level. This feature of the utility is used to pass query to the database by scripts issuing external system command in the perl. The utility is called "MX"(from a set of CISIS Utilities) by BEREME (BEREME, 2001, http://www.bireme.br/).

3.1 Database Structure

The structure of the database is shown here as three-captured screens:



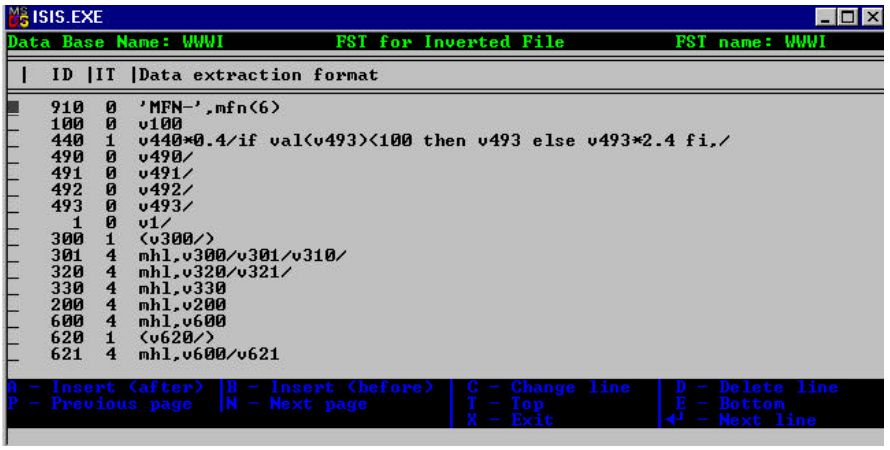


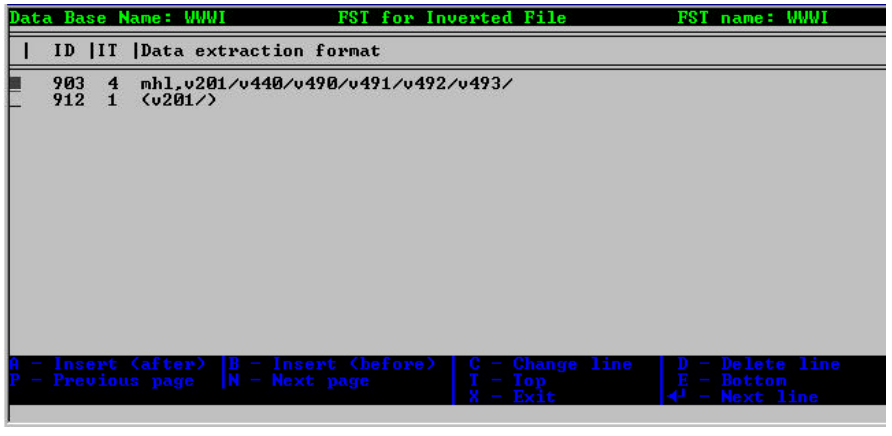


3.2 Fields indexed for the Inverted File

There is one inverted file index. The Inverted file contains terms which are extracted from the selected fields using a particular indexing technique. The fields here are selected are according to the design of the inverted file index as discussed in the previous chapter. For each term, there is a list of references to the Master file record(s) from which the term was extracted. Each posting contains the record number and additional information precisely identifying the field from which the data had been extracted.

The selection of the fields for indexing of the database is shown here as two captured screens:





3.3 Display Formats

The display formats have been written to display the records retrieved from the database. A listing of the formats is given in the following paragraphs:

Citation Display Format (PLMS.PFT):

```
mpl,'<tr bgcolor="#FFFFff"><td>'/'<input type=checkbox
name="records"',' value="MFN-',mfn(6)'">',
'<font color="#990099">'v300+|; |,' </font></td></tr>',/
'<tr><td>'v200,' </td></tr>',/
'<tr><td>',v201," "v440"; ",v490,"("v491")",": "v492,
' </td></tr>
```

Citation+Abstract `Display Format` (PLMXB.PFT):

```
mpl,'<tr bgcolor="#FFFFff"><td>'/'<input type=checkbox
name="records" value="MFN-',mfn(6)'">',
'<font color="#990099">'v300+|; |,' </font></td></tr>',/
'<tr><td>'v330,' </td></tr>',/
'<tr><td><font color="#0000AA"><STRONGE>'v200,'
</STRONGE></font></td></tr>',/
'<tr><td><STRONGE>',v201," "v440"; ",v490,"("v491")",": "v492,
' </STRONGE></td></tr>',/
'<tr bgcolor="FFFFFF"><td><font color="#ff0000">',
"ABSTRACT: "v600,' </font></td></tr>'/
'<tr><td><hr></td></tr>'
```

`Full Record Format(`*`PLMXF.PFT`*`):`

```
mpl,'<tr bgcolor="#FFFFBB"><td>'/'<input type=checkbox
name="records"',' value="MFN-',mfn(6)'">',
'<font color="#FF00FF">'v300+|; |,' </font></td></tr>',/
'<tr><td>'v330,' </td></tr>',/
'<tr><td>'v200,' </td></tr>',/
'<tr><td>',v201," "v440"; ",v490,"("v491")",": "v492,
' </td></tr>',/
'<tr bgcolor="FFDDFF"><td>'"ABSTRACT: "v600,' </td></tr>'/
'<tr><td>',if p(v620) then '<font color="#FF0000"> KEYWORDS:</font>
' fi,
"<br>"d620,(v620|<br>|),' </td></tr>'/,
'<tr><td>',if p(v621) then '<font color="#0000FF">OTHER
KEYWORDS:</font> ' fi,
"<br>"d621,(v621|<br>|),' </td></tr>'/,
'<tr><td>',if p(v703) then '<font color="#00AA00">
REFERENCES:</font> ' fi,v703,' </td></tr>'/,
'<tr><td>',"Indexer: "v800,' </td></tr>'/,
'<tr><td><hr></td></tr>'
```

Chapter: Eight

TESTING & INTEGRATION WITH IMC'S WEB-PAGES

1. Testing

There are two types of testing: - White Box and Black-box Methods. White box method relies on intimate knowledge of the code and a procedural design. It is used to determine that all-possible path within a module, to execute all loops and to test all logical expressions. Black-box Testing on the other hand focuses on the overall functionality of the software.

The specification of the proposed system had the following tests specified belonging to both types of methods:

- Compatibility with Internet Browsers: Internet Explorer, Netscape Communicator and Opera browsers.
- Handling user errors: submitting empty request and Unbalanced parentheses.
- Retrieval of correct set of references.
- Proper display formats.
- Number of references as per selection by the users.
- Logical errors in the CGI script.

The testing was done on continuing bases as well as at the end of the final product. Following tests were made:

1.1 Browsers Compatibility Tests

The User Web Interface opened with Internet Explorer (Version 5.5), Netscape Communicator (Version 6) and Opera (Version 5) and the compatibility was observed. It was also tested for various resolutions of 640 x 480, 800 x 600 and 1024 x 768 resolution on colored monitors. Besides testing on Windows based operating systems, cross-platform was attempted by using Netscape Communicator on the LINUX machine. The Interface performed satisfactorily.

1.2 Users Error Handing Tests

The following tests were undertaken:

-Clicking Submit Button on various forms of the interface without any query.

-Submitting a query with unequal open and closing brackets (parenthesis).

-Starting the query with a Boolean Operators (And; Or; Not).

-Clicking "Display Button" with "Selected" radio button on Top Menu of the Search Result Screen but without actually selecting any reference.

Some problems were encountered while displaying the error messages because when the scripts terminated in response to "die" command proper html document end tags could not be generated. As the result the scripts terminated but error message could not been seen due to in sufficient end tags of the generated html documents. Adding a standard procedure "END" to the perl scripts rectified this problem. The "END" procedure ensured that html tags required for ending an html documents were executed always. The tested were performed again and the results were satisfactory.

1.3 Retrieval Comparison Test

This test was undertaken to compare the correctness and effectiveness of the retrieval of the web interface system with of the database under DBMS. Three queries of varying complexity were chosen. Each query was executed in both the platforms. The retrieved reference sets were compared. There was exact match between these, which test the web interface satisfactorily.

1.4 Display Formats Testing

Changing them one by one for a set of retrieved references tested all the display formats. The results were satisfactorily.

1.5 Testing CGI Scripts for Logical Errors

This testing was done continuous during the development of the perl scripts. It was ensured to test all paths in the scripts. As the application is to operate in web environment a method was followed to tract all the variables. For this, the scripts were interlaced with PRINT statements to print the intermediary variable and results. Along with print statements suitable html tag were patched so the display can be seen in the browser. After the purposes of such PRINT statements were served, these were commented with hash sign (#). These commented statements could be seen on the source code of the scripts. These were left there to make the code more readable.

There were some problems initially with the scripts. The method employed above helped in tracing the variables and paths and removing the bugs.

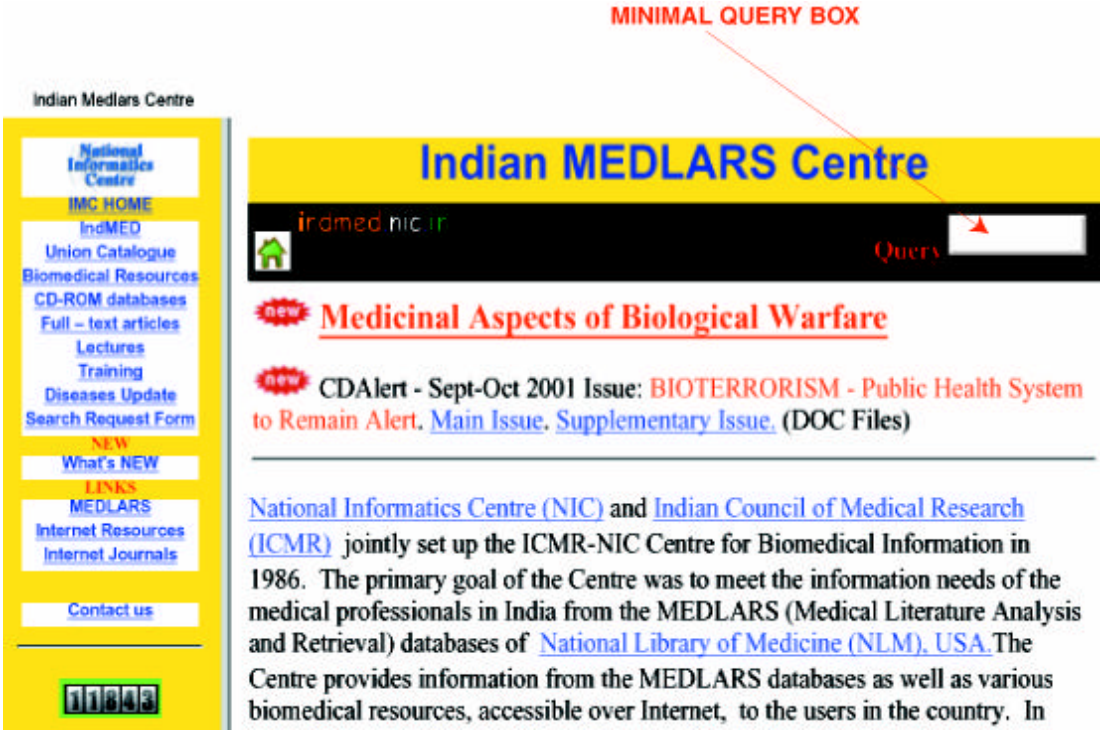
2. Integration with IMC's Web Pages

The system was meant to be available from the home page of the Indian Medlars Centre (IMC) at http://indmed.nic.in. Integration of the developed system with the IMC's Home Page was simple affair. It was most of the time Cut & Paste affair. The following tasks were specifically done.

- Placing of Minimal Search Box at exiting web pages.
- Development of IndMED Web Page.

2.1 Placing of Minimal Search Box.

The Minimal Search Box was placed on most of the important pages of the IMC's site. These pages were opened in html editors along with Minimal Search Box html file in different boxes. The relevant code of the search box was copied to that of the IMC's exiting pages. Moreover when a new page would be created, the code of the search box would be inserted. A sample is shown here:



2.2 Development of IndMED Web Page

A new web page was opened in html editor. First the code for Minimal Search Box was inserted at the top. Then description to the IndMED database was added. A link to HELP on Searching was placed. Then the codes for Simple Search Form and Advanced Search Form were inserted. This was then saved as IndMED Page. Appropriate link to this page was placed from the Home Page of IMC. It is shown as follows:

IndMED : A Bibliographic Database of Indian Biomedical Research

Indian MEDLARS Centre

indmed nic in

Query

NATIONAL INFORMATICS CENTRE NIC

Indian contribution in the areas of biomedical research and health care has been significant and conforming to international standards. However, only a small fraction of it is available for reference through international bibliographic databases. The ICMR-NIC Centre for Biomedical Information (Indian MEDLARS Centre) has designed and developed a bibliographic database from Indian biomedical literature. To startwith 75 prominent Indian journals, have been selected to build up the database entitled IndMED. More journals would be added to the list as their quality improves in coming years. It is proposed to cover the journals from 1985 onwards in this database.

Search your query here. HELP is also available.

IndMED Search

Clear form | Begin search | And Show 20 References.

ADVANCED IndMED Search

		Search for	in
1			Any Where
2	AND		Any Where
3	AND		Any Where
4	AND		Any Where

Clear Form | Begin search | And Show 20 References.

Chapter: Nine

USER HELP MANUAL FOR SEARCHING

A User Help Manual for Searching was prepared for the end user. It is available as a web document on the IMC's Site. A link from the IndMED page was provided to it.
A copy of the same is produced here:

IndMED HELP

Searching

- Forms
  - Simple Form
  - Minimal Search Box
  - Advance Form
- Truncation
- Operators
- Phrase Searching
- Specific Fields
  - Author Names
  - Journal Titles

Search Results

- Displaying
- Saving
- Printing

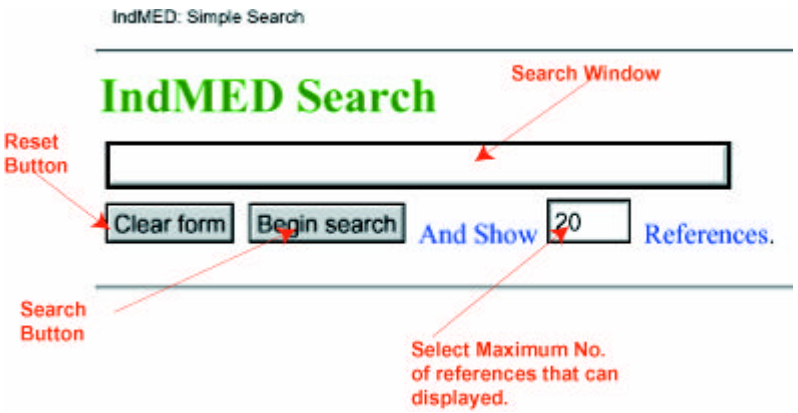
Display Formats

Subheadings

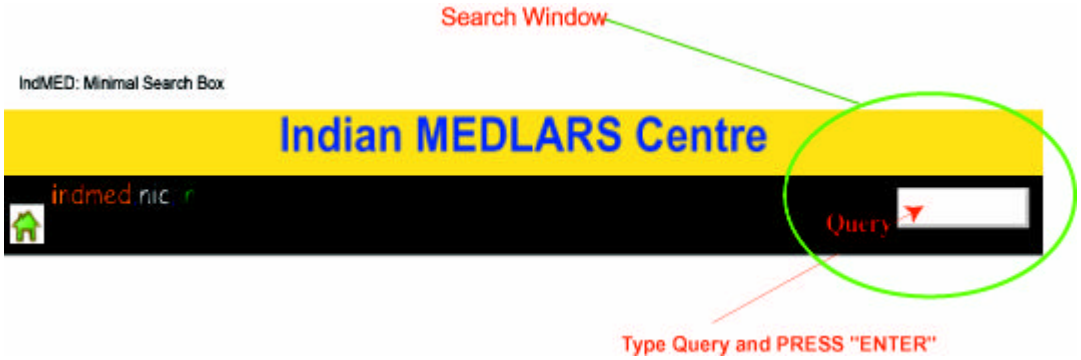
Searching

Searching is easy. Enter search terms in the query box. You may select Simple Form, Minimal Box **or** Advance Search Form, depending upon the complexity of your query. Minimal Box is available at the top of most of the pages (you do not need to return to the homepage/INDMed Page to enter a new search) as well as at the INDMed page. The Simple Form and Advance Form are available at the INDMed Page.

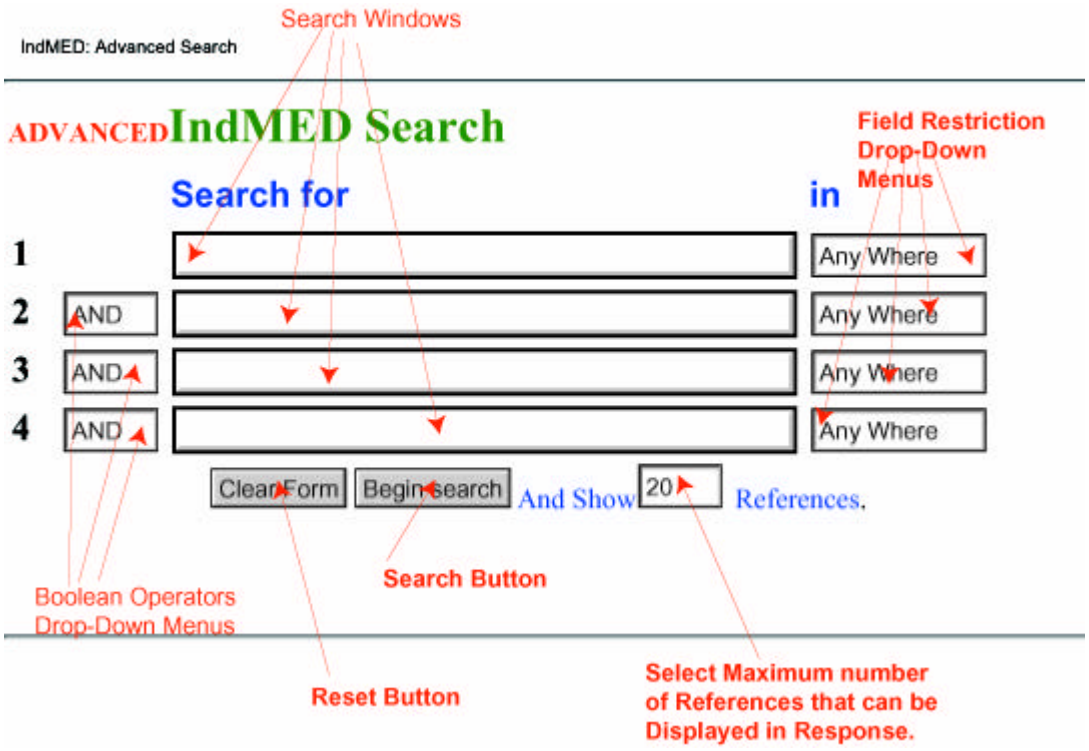
Simple Form



Minimal Search Box



Advance Form



Truncation

Truncation means finding all terms that begin with a given text string. Place an dollar ($) sign at the end of a term to search for all terms that begin with that word; for instance bacter$ will find all terms that begin with the letters bacter; e.g., bacteria, bacterium, bacteriophage, etc.

Operators

When there is more than one word or term in the same line, or from one line to the other, they must be separated by a boolean operator in order to be related one to the other. To type more than one word in the same line, it is necessary to use a boolean operator so as to relate one to the other.

Boolean Operators are AND, OR, and NOT are available.

operator	example	function
and	hepatitis b **and** hepatic cirrhosis	retrieves articles about topics **hepatitis b** and **hepatic cirrhosis**, obligatorily related.
or	hepatitis b **or** hepatitis c	retrieves articles about topics **hepatitis b** and/or the topic **hepatitis c**, not necessarily both topics in the same article.
not	arrhythmia/co **not** myocardial infarct	retrieves articles about **arrhythmia complications** eliminating the articles about **myocardial infarct.**

Phrase Searching

Phrase searching depends how the various fields are indexed, say for example authors are indexed as last name first then space then initials. MeSH keywords are also indexed are phares where applicable. Such fields can be searched as. e.g.

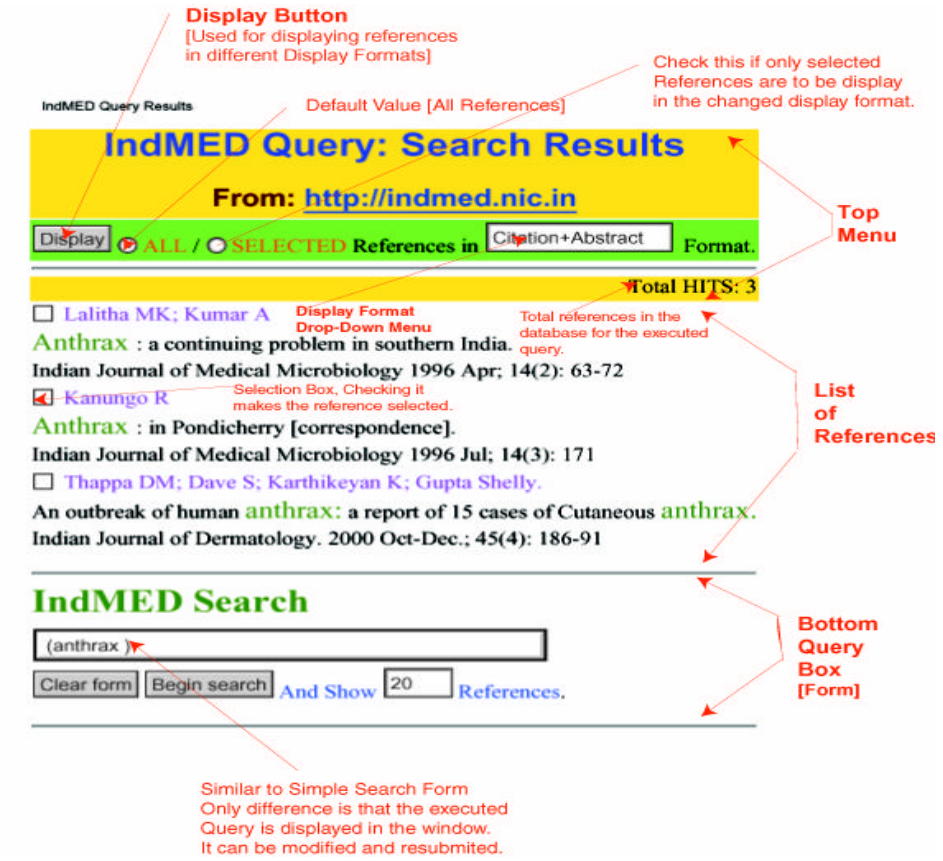
hepatitis b
Sharma PK

Specific Fields

- Author Names: Search by Last name followed by initials.
- Journal Titles: Enter the journal name in full.
- Title: Title words are searchable instead of full article title.
- Keywords: These are searchable according the format in MeSH.
- ISSN: Searchable in Nine Digit Format.

Search Results

- Displaying: Select the individual references or all references retrieved by the query. Select appropriate Display Format for the required fields to be displayed.



- Saving: Use same function of your browser. You may also use "Cut and Paste" technique to save selected portion of the results.
- Printing: Use print option from your browser.

Display Formats

- Citation: This displays Author(s), Address, Title and Source.
- Citation+Abstract: This display Abstract also along with the Citation.
- Full Record: This displays Keywords, Other Keywords and No. of References along with Citation+Abstract.

Subheadings

To search MeSH terms along with Subheadings use hypen "-". The following subheading have to used along with MeSH keywords:

AA analogs & derivatives AB abnormalities AD administration & dosage AE adverse effects AG agonists AH anatomy & histology AI antagonists & inhibitors AN analysis BI biosynthesis BL blood BS blood supply CF cerebrospinal fluid CH chemistry CI chemically induced CL classification CN congenital CO complications CS chemical synthesis CT contraindications CY cytology DE drug effects DF deficiency DH diet therapy DI diagnosis DT drug therapy DU diagnostic use EC economics ED education EH ethnology EM embryology EN enzymology EP epidemiology ET etiology GD growth & development GE genetics HI history IM immunology IN injuries IP isolation & purification IR innervation IS instrumentation	LJ legislation & jurisprudence MA manpower ME metabolism MI microbiology MO mortality MT methods NU nursing OG organization & administration PA pathology PC prevention & control PD pharmacology PH physiology PK pharmacokinetics PL PO poisoning PP phisiopathology PS parasitology PX psychology PY pathogenicity RA radiography RE radiation effects RH rehabilitation RI radionuclide imaging RT radiotherapy SC secundary SD supply & distribution SE secretion SN statistics & numerical data ST standards SU surgery TD trends TH therapy TM transmission TO toxicity TR transplantation TU therapeutic use UL ultrastructure UR urine US ultrasonography UT utilization VE veterinary VI virology

Chapter: Ten

FUTURE DIRECTIONS & CONCLUSION

1. Future directions

There is scope for further improvement and sophistication with this system.
Some features that could be added include:

- List of References sorted according to degree of relevance. The more relevant should be displayed at the top. A proper research into relevancy criteria and attaching differential weights to different fields could serve as the logic for facility.

- Links to the full text of the articles of some the journals available online can be considered.

- Search forms can have a facility for "federated searching" i.e. besides providing the references on the query from IndMED database, relevant material from Internet could also be presented along with.

- Links could be provided to the physical availability of the journals in medical libraries in India.

2. Conclusion

The aim of providing a Bibliographic Database through web has been successfully achieved. The crucial stages in this project were finding exact requirements of the users and writing Perl Scripts. In the initial stages there were some bugs that led to inadequate generation of html tag. This led to display of pages with junks along with visible html code. These bugs were fixed with tracking of the variables and displaying intermediate results. The interface was later on fully tested and accepted by end users. The system has been integrated with the web pages of Indian Medlars Centre at the URL: http://indmed.nic.in. A brief searching help manual is also available.

BIBLIOGRAPHY

[Bibliography According to Harvard Style (Modified for Internet Resources)]

ActiveState Corporation , 2001,
ASPN: PERL, [Online], Available:
http://aspn.activestate.com/ASPN/Perl/ [2001, Oct. 29].

Ambler, S.W. 1998a, Building Object Applications That Work: Your Step-By-Step Handbook for Developing Robust Systems with Object Technology, Cambridge University Press, New York.

Ambler, S.W. 1998b, Process Patterns: Building Large-Scale Systems Using Object Technology, Cambridge University Press, New York, [AVAILABLE],
http://www.ambysoft.com/processPatterns.html

Ambler, S.W. 2000, User Interface Design: Tips and techniques, [ONLINE], [AVAILABLE] http://www.ambysoft.com/userInterfaceDesign.pdf
[Material for this paper has been extracted from: The Object Primer: 2nd edition: Building Object Applications that Work and Process Patterns, all written by Scott W. Ambler and published by Cambridge University.]

Ambler, S.W. 2001a, Designing web-based user interfaces. [ONLINE],
[AVAILABLE] http://www.sdmagazine.com/articles/2001/0103/0103n/0103n.htm
[2001 Aug 7].

Ambler, S.W. 2001b, The Object Primer 2nd Edition: The Application Developer's Guide to Object Orientation, Cambridge University Press, New York, [AVAILABLE], http://www.ambysoft.com/theObjectPrimer.html

Aronson, L. 1995, HTML3 manual of style, Ziff-Davis Press,
Emeryville, Calif.

BEREME, 2001, *BVS – Home page*, [Online], Available:
http://www.bireme.br/ [2001, Oct. 9].

BioMedNet, 2001,
BioMedNet Research Tools: Medline, [Online], Available:
http://research.bmn.com/medline [2001, Oct. 29].

Biblio Tech Review, 2001,
Technology Review : Databases, [Online], Available:
http://www.biblio-tech.com/html/databases.html

Bournemouth University, *Bibliographic Databases*, [Online], Available:
http://www.bournemouth.ac.uk/library_catalogue/html/biblio.html [2001, Oct. 12]

Cherry, J.M., & Cox, J.1996, "World Wide Web displays of bibliographic records". *Canadian Association for Information Science. Proceedings of the 24th Annual Conference*, June 2-3, 1996, Toronto, Ontario, 100-114.

Day, M. 1997, 'The price of prejudice: the world may miss out because the fruits of research in developing countries seldom make it to the pages of top journals', *New Scientist*, Vol. 156, no. 2106, pp. 22-23.

Deco, C., Bender C. & Crespo F., 1995, "Why MicroIsis?: textual databases versus relational databases", *INFOISIS*, 1(3), pp. 42-55.

Deep, J & Holfelder, P. 1996, *Developing CGI Applications with Perl*. John Wiley and Sons: New York.

DESIRE, 2000, *DESIRE Information Gateway Handbook*, [Online], Available: http://www.desire.org/handbook/welcome.html [2001 Aug 7].

Ford, A. 1995, *Spinning the web: how to provide information on the internet*, International Thomson Computer Press, London.

Gundavaram, S. 1996, *CGI programming on the World Wide Web*, O'Reilly & Associates, Sebastopol, Ca.

Hopkinson, A. & Simmons, P. (eds) 1988, *CCF, the Common Communication Format*, 2nd Edn, UNESCO, Paris.

Indian Council of Medical Research, [Online], Available: http://www.icmr.nic.in/ [2001, Oct. 9].

"It's not what you know - If there are biases in scientific publication editors must take the blame", 1997, *New Scientist*, Vol. 156, no. 2106, pp.3.

Lemay, L. 1995, *Teach yourself web publishing with HTML in a week*, SAMS Publishing, Indianapolis.

Medscape, 2001, *Medscape Medline Search*, [Online], Available: http://www.medscape.com/server-java/MedlineSearchForm [2001, Oct. 29].

Muelver, J. 1996, *Creating cool web pages with perl*, IDG Books Worldwide Inc., Foster City, CA.

National Informatics Centre, 2001, [Online], Available: http://home.nic.in/ [2001, Oct. 9].

Page-Jones, M. 1995, What Every Programmer Should Know About Object-Oriented Design, Dorset-House Publishing, New York.

Pressman, R.S. 1997, Software engineering: a practitioner's approach, 4th ed., McGraw-Hill, New York.

Schneiderman, B. 1992,
Designing the user interface : strategies for effective human-computer interaction,
Addison-Wesley, Massachusetts.

Stein L. 2001, *CGI.pm - a Perl5 CGI Library*, [Online], Available:
http://stein.cshl.org/WWW/software/CGI/) [2001, Oct. 12].

Till, D. 1995, *Teach yourself Perl in 21 days*,
SAMS Publishing, Indianapolis.

UNESCO, 2001 *The UNESCO micro CDS/ISIS Software*, [Online],
Available: http://www.unesco.org/webworld/isis/ [2001, Oct. 15].

United States National Library of Medicine, 2001a, [Online], Available:
http://www.nlm.nih.gov/ [2001, Oct. 9].

United States National Library of Medicine, 2001b,
Medical Subject Headings, [Online], Available:
http://www.nlm.nih.gov/mesh/meshhome.html [2001, Oct. 16].

United States National Library of Medicine, 2001c,
PubMed, [Online], Available:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi [2001, Oct. 29].

Varun VK. 2001, "RU on Internet?: CDS/ISIS Resources", *Information Today and Tomorrow*, 20(1), pp 16-23, Available:
http://itt.nissat.tripod.com/itt0101/ruoi0101.htm [2001, Oct. 9].

Wilson, S. 1995, *World Wide Web design guide*,
Hayden Books, Indianapolis.