Chapter 4

DESIGN OF HWCBSE

4.1 INTRODUCTION:

In the recent years, the World Wide Web has been enriched by the enormous amount of databases on the Internet. While the “Surface Web” has millions of static linked HTML pages, it is a fact that a far more significant amount of information is “hidden” in the online databases, behind the query search forms of searchable databases. This hidden information cannot be accessed directly through static URL links as it is only available as responses to dynamic queries submitted at the search query interface of a database. Since, this information is hidden behind search forms, this part of a web are often referred to as the “Hidden Web”.

Search query forms and dynamically generated pages are helpful to users because users can get the desired information they want. However, it is really tiresome task for users to visit all the hidden web sites for the same domain and fill out different forms provided by each site. Traditional crawlers wander throughout the Web and index the addresses of pages they discover. When a search form of Hidden website is encountered, such crawlers can only record the address, but can't deal with the information that page contains. Because current crawlers cannot effectively query these databases, such data are invisible to traditional search engines, and thus remain largely hidden from users.

Thus there is a need to develop an automated process that can fill the search query forms, extract the results, integrate them, store them at one location and search the results that match user query.

In this work, “A Hidden Web Crawler Based Search Engine (HWCBSE)” is designed that addresses the above issues. It fills out search forms automatically and extracts the relevant data behind those search forms. This system is implemented using the ASP.Net technology.
and SQL server is used as database management system. The architecture of proposed “HWCBSE” is explained in following section.

4.2 SOFTWARE ARCHITECTURE OF HWCBSE:

The basic idea behind designing of such an engine is to find several different web pages within the same domain like book domain, car domain etc., collect this information and after some compilation and normalization of data, present it as a free web search service to the user. The complete architecture of a Hidden Web Search engine is shown in figure 4.1.
The whole process consists of four main components. These are:

1. Hidden Web Crawler
2. Data Extractor
3. Indexer
4. Query Processor

A brief description of each component is given below:

1. **Hidden Web Crawler**

This component is designed to collect the web pages from different Hidden web sources. Since search interface represents a subset of queries that could be performed on the underlying Hidden web databases, *Hidden web crawler* starts its process by taking each URL one by one from predetermined list of URLs of Hidden websites and extracts the search interfaces from them by using some domain features.

Ontology is a description of concepts and their relationships in particular domain. It provides a structured way of describing knowledge about the web page [73]. The main function of ontology is to provide the knowledge base to classify the results. To build the domain ontology, some web pages (random web pages) are selected as sample web pages and attributes are extracted from them. Now, an ontology table is maintained using these attributes and the synonyms. The “wordnet” dictionary [115] which is an online dictionary for extracting meaning and synonyms of particular word is used for finding the synonyms.

Now, all the remaining websites from the websites list are searched for search interface forms by looking at the domain features stored in ontology table. As soon a search form is found, attributes and their values are extracted and sent to the attribute value table for storage. In the next step, crawler picks search form one by one for submission and if attribute of the form matches with attribute in the attribute-value table, the values corresponding to that attribute are retrieved. Crawler then submits the form with these values. After submission, result pages are retrieved and sent to next component of the search engine for extraction of the data that resides inside the pages.
Since the Hidden websites publish their data by embedding them in the tables or lists, the second component of proposed search engine analyzes every result page and then extracts individual data records from the table.

2. **Data Extractor**

A significant amount of valuable information on the web is generated from back-end databases and table/list is a commonly used presentation scheme for describing the relational information of these databases. Such tables further contain many rows and columns in which data is packed. *Data Extractor* has been designed in such a way that it extracts the data from rows, columns of table of each web site and collects this information at one place.

With the explosive growth of information sources available on the World Wide Web, it has become difficult to identify the relevant pieces of information. However, these web pages also contain some irrelevant data like advertisements, navigation-panels, copyright notices etc., surrounding the main content of the web page. Hence it is necessary to separate these unwanted components of the page to get relevant and accurate information.

Since HTML is the predominant markup language for web pages and it is the basic building-block of webpages, an HTML document can be represented by a tree-structure using HTML DOM [111]. As the nested structure of HTML tags automatically forms a tag tree, the main table area containing actual information is extracted and irrelevant area is discarded by analyzing every subtree of HTML DOM of the result page for domain attributes.

Next step is to find the data records (from rows and columns). Because all rows in a particular table share the same pattern and have same number of columns and as the nested structure of HTML tags automatically forms a tag tree, the data records are extracted by extracting subtrees of HTML DOM tree containing same number of childnodes, following the same pattern and containing domain attributes.

Now, one data record area is used to generate the dynamic rule for extracting result. Once the rule is made, all the records in that page are extracted in the same way. Since a particular site
follow the same pattern for all pages, all the pages returned from that site will follow the same rule.

After extracting rows and columns, separate tables are maintained for each web site that contains the result records. At the end, these tables are merged to form main repository which is used to build the index database.

3. **Indexer**

Indexer takes the hidden web data from the repository prepared by Hidden web crawler and stores the data records in an inverted index database. This index database is used to find the data records that match with user query. It contains all the attribute names corresponding to attribute values.

4. **Query Processor**

A traditional search engine normally queries a text database via a set of keywords provided by the user and returns the documents that are relevant to the search keywords. But retrieval of relevant information from a relational database requires writing and firing SQL queries. To make the Hidden web search engine works like traditional search engine, this search engine provides search interface to user which contains single search box where user can fill his/her query. When user submits the query, *Query Processor* translates user query into SQL query which is fired on the index database and results are returned to user. By building such a query processor, user does not need to know the database schema or SQL (Structured Query Language). Instead of that, he/she submits only a list of keywords.

The detailed description of all the components of HWCBSE is given in next chapter.