**DEVELOPMENT OF AN OPTIMIZED DATABASE-DRIVEN WEB SEARCH ENGINE**

**BY**

**AHAMEFULA, VICTORY NGOZI**

**DE.2013/2844**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE, RIVERS STATE UNIVERSITY, PORT HARCOURT, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE DEGREE (B.Sc) IN COMPUTER SCIENCE**

**AUGUST, 2017**

**DECLARATION**

I hereby declare that this project work was conducted by me and has been presented for the award of a Bachelor of Science (B.Sc) in the Department of Computer Science, Rivers State University and has never been previously submitted anywhere for the purpose of the award of a degree.

--------------------------------------------- ---------------------------

**Ahamefula Victory Ngozi Date**

**DE.2013/2844**

**CERTIFICATION**

This is to certify that this is the original work of the candidate and that the project be accepted as fulfillment of the requirements for the degree of Bachelor of Science (B.Sc) in the Department of Computer Science.

**Mr. D.J.S. Sako -------------------- -----------------**(Project Supervisor) Signature Date

**Dr. V. I.E. Anireh -------------------- -----------------**

(Head of Department) Signature Date

**Prof. V. B. Omubo-Pepple --------------------- -----------------**

(Dean, Faculty of Science) Signature Date

**External Examiner ---------------------- -----------------**

Signature Date

**DEDICATION**

This project work is dedicated to the Almighty God, My Heavenly Father who saw me through this project work from start to finish. And also, to my parents Rev. & Pst. Mrs. Chris Ahamefula, for their love, support and encouragement.

**ACKNOWLEDGEMENTS**

I am eternally grateful to God, first of all for this precious gift of life, secondly for His love, mercies and grace that has seen me through my time in Rivers State University.

My sincere appreciation and gratitude also goes to my project supervisor, Mr. D. J. Sako for his encouragement and constant direction and contribution that has made this project work a huge success.

I also wish to appreciate and acknowledge the relentless effort and constant encouragement of my father, Rev. Chris Ahamefula and Pst. Mrs. Nata Ahamefula, my sisters, Chinenye Ahamefula, Emmanuella Ahamefula, my brother, Ahamefula Chris (Jnr.) and my “BIG DADDY”, Dr. S. Ngah, thank you so much for your love.

I want to also appreciate my Pastors, Pst. Uchenna Juan Augustine, Pst. Mrs. Christabel Juan Augustine, Pst. David Bode Oluji, Pst. Washington Answer and all the pastors of The Sanctuary Missions Int’l for their love, constant support and fervent prayers.

My profound gratitude also goes out to the Head of Department of Computer Science, Dr. Ike Anireh and all the lecturers of the department, for their encouragement and support. God bless you all.

**ABSTRACT**

In the world today, the internet serves as one of the primary sources of information and the importance of web search engines to internet users cannot be over-emphasized. Database-driven web search engines are essentially web-based document retrieval systems. Specific examples of database-driven web search engines are academic search engines. Academic web search engines ensure and provide specificity of results for research activities carried out by various users (students, teachers, lecturers and scholars). In this study we will develop a web-based academic search engine for storing and retrieving project/research materials with ease.

**TABLE OF CONTENTS**

**Pages**

Title page i

Declaration ii

Certification iii

Dedication iv

Acknowledgement v

Abstract vi

Table of Contents vii

List of Figures x

List of Tables xii

**CHAPTER ONE: INTRODUCTION**

* 1. Background of the study 1
  2. Statement of the Problem 2
  3. Aim and Objectives of study 3
  4. Methodology of the study 4
  5. Scope of the study 4
  6. Definition of terms 5
  7. Project layout 7

**CHAPTER TWO: LITERATURE REVIEW**

2.1 Introduction 8

2.2 A brief history of search engines 9

2.3 Database-Driven Search Engines 11

2.4 Information Retrieval 13

2.5 Academic Research Web Search Engine 15

2.6 Review of the Existing System 15

2.6.1 Google Scholar 16

2.6.2 Google Books 18

2.6.3 Microsoft Academic 19

2.6.4 WorldWideScience 20

2.6.5 Refseek 22

2.6.6 ERIC 22

2.2.7 iSeek 23

2.6.8 BASE 24

2.6.9 UMR Repository 24

2.6 Features of Academic Web Search Engine 25

**CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN**

3.1 Introduction 28

3.2 Analysis of the system 28

3.2.1 System requirement specification 28

3.3 System design 29

3.3.1 Architectural design 30

3.3.2 Physical design 31

3.3.3 Process design 32

3.3.4 Input design 32

3.3.5 Output design 34

3.3.6 Database design 35

3.4 Use case diagram 37

3.5 System flow chart 39

**CHAPTER FOUR: SYSTEM IMPLEMENTATION**

4.1 Introduction 40

4.2 Sample output interfaces 40

4.3 System requirements 45

4.3.1Software requirements 45

4.3.2 Hardware requirements 46

4.4 Choice of programming language 46

4.4.1 Reasons for choice of programming language 46

**CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION**

5.1 Summary 49

5.2 Conclusion 49

5.3 Recommendation 49

References 51

Appendix 53

**LIST OF FIGURES**

Figure 2.1 The GUI for Google Scholar search engine 17

Figure 2.2 The GUI for Google Books search engine 19

Figure 2.3 The GUI for Microsoft Academic search engine 20

Figure 2.4 The GUI for WorldWideScience.org search engine 21

Figure 2.5 The GUI for RefSeek search engine 22

Figure 2.6 The GUI for ERIC search engine 23

Figure 2.7 The GUI for iSeek search engine 23

Figure 2.8 The GUI for BASE search engine 24

Figure 2.9 The GUI for UMR Repository search engine 25

Figure 3.1 The Three Tier Architecture 30

Figure 3.2 General Search Engine Architecture 31

Figure 3.3 Search Window 34

Figure 3.4 Search result 35

Figure 4.1 The Index Page 40

Figure 4.2 Sign Up Page 41

Figure 4.3 Sign In Page 41

Figure 4.4 The About Page 42

Figure 4.5 Document List Page 43

Figure 4.6 The Search Query Page 43

Figure 4.7 Search Result Page 44

Figure 4.8 The Add Document Page 44

Figure 4.9 Contacts Page 45

**LIST OF TABLES**

Table 3.1 Database Table 36

Table 3.2: Login (Admin Login Table) 36

Table 3.3: Research Table 37

**CHAPTER ONE**

**INTRODUCTION**

* 1. **BACKGROUND OF THE STUDY**

Numerous people all around the world access the World Wide Web daily for different purposes, it could either be to connect with friends, family, colleagues, business partners, research studies etc. or to discover and access a wide variety and quantity of information the internet offers. In this light, Web search engines were developed to enable users precisely describe and extract useful information in specific fields. Search Engines matter because they increasingly determine the information about brands, products, research studies and services that are relevant to users. Hence the need for the case study, Rivers State University to adopt this technology to ensure an organized and structured storage unit for Projects in order to achieve the purpose of proper documentation and constant availability of necessary research materials.

A [search engine](https://en.wikipedia.org/wiki/Search_engine) is an information retrieval program that discovers, crawls, transforms and stores information for retrieval and presentation in response to user queries. A search engine normally consists of four components e.g. search interface, crawler (also known as a spider or bot), indexer, and database (Vaughan et al, 2004).

A web search engine is a software system that is designed to search for information on the World Wide Web (Jansen & Rieh, 2010). Some search engines also [mine data](https://en.wikipedia.org/wiki/Data_mining) available in [databases](https://en.wikipedia.org/wiki/Database). A database-driven web search engine is a [web-based search engine](https://en.wikipedia.org/wiki/Search_engine) that operates on material stored in a digital [database](https://en.wikipedia.org/wiki/Database). Database search technology is used by large public and private entities including government database services, e-commerce companies, online advertising platforms, telecommunications service providers and other consumers with a need to access information in large repositories (Yu &Meng, 1999). An optimized database-driven web search engine is a web-based search engine that is designed to retrieve data stored in a database or multiple databases.

In this study, we will be developing an optimized database driven web search engine for research, that would search and retrieve research materials/students’ projects from a database.

**1.2 STATEMENT OF PROBLEM**

Problems associated with research system manually handled in academic institutions are as follows;

* Research materials/projects are sorted and stored manually on its appropriate shelf
* Retrieving of research materials/projects take a lot of time depending on the number of topics involved.
* Loss or damage of any of the research material may lead to loss of resources/ information.
* Too cumbersome and time consuming to maintain.

**1.3 AIM AND OBJECTIVES OF STUDY**

The aim of this study is to develop an optimized, computerized research system that is user-friendly and GUI-oriented to ensure proper storage and retrieval of research materials.

The objectives include;

1. To develop a search engine that is optimized.
2. To create a Database Management System (DBMS) which stores research materials and provide security for the stored data.
3. To provide quick and efficient means of retrieving student research/projects material through a search system.
4. To develop a system that would be open and easily accessible to various users.

At the completion of this work, this system will improve the management of a research library which can be used as a model in building digital research library.

**1.4 METHODOLOGY OF THE STUDY**

The following procedures will be carried out to achieve the desired goal and anticipated outcome of this project;

* Outline the requirements by carrying out a proper analysis on the existing manual system which involves a Literature Review conducted by searching books, articles and projects on the Internet, related to the needs of the end user based on the existing manual system.
* Create a front end which is a Graphic user interface (GUI) which is user friendly, using Hypertext Markup Language (HTML) and Cascading Style Sheet (CSS).
* Create a back end to the system which is a database, using MYSQL.
* Create a connection between the front end and the back end using a server-side programming language PHP Hypertext Preprocessor (PHP) and a client-side scripting language JavaScript.

**1.5 SCOPE OF THE STUDY**

The scope of this study is to develop a web based academic search engine for advanced research purposes. The case study of this research work centers on students project work and research materials in Rivers State University. The system can be seen as an online repository for students’ projects in Rivers State University.

**1.6 DEFINITION OF TERMS**

**Optimization:** Optimization is the act, process, or methodology of making something such as; a design, system, or decision as functional, or effective as possible (Merriam Webster dictionary).

**Search Engine:** This is computer software used to searchand retrieve data (such as text or a database) for specified information.

**Search engine results page (SERP):** This is the page displayed by a search engine in response to a query by a searcher. The main component of the SERP is the listing of results that are returned by the [search engine](https://en.wikipedia.org/wiki/Search_engine_%28computing%29) in response to a [keyword](https://en.wikipedia.org/wiki/Keyword_%28Internet_search%29) [query](https://en.wikipedia.org/wiki/Web_search_query).

**A web page (webpage or Web page):** This is a document that is suitable for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) and [web browsers](https://en.wikipedia.org/wiki/Web_browser). A web browser displays a web page on a [monitor](https://en.wikipedia.org/wiki/Computer_display) or [mobile device](https://en.wikipedia.org/wiki/Mobile_device). The web page is what displays, but the term also refers to a [computer file](https://en.wikipedia.org/wiki/Computer_file), usually written in [HTML](https://en.wikipedia.org/wiki/HTML) or comparable [markup language](https://en.wikipedia.org/wiki/Markup_language).

**Data mining:** This is a computational process of extracting information from a data set and transforming it into an understandable structure for further use.

**A database:** This is an organized collection of [data](https://en.wikipedia.org/wiki/Data_%28computing%29). It is the collection of [schemas](https://en.wikipedia.org/wiki/Database_schema), [tables](https://en.wikipedia.org/wiki/Table_%28database%29), [queries](https://en.wikipedia.org/wiki/Query_language), reports, [views](https://en.wikipedia.org/wiki/View_%28SQL%29), and other objects.

**A database management system (DBMS):** This is a [computer software](https://en.wikipedia.org/wiki/Computer_software) application that interacts with the user, other applications, and the database itself to capture and analyze data. Well-known DBMSs include [MySQL](https://en.wikipedia.org/wiki/MySQL), Microsoft SQL Server, Oracle, Sybase etc.

**Web indexing (or Internet indexing):** This refers to various methods for indexing the contents of a [website](https://en.wikipedia.org/wiki/Website), database or the [Internet](https://en.wikipedia.org/wiki/Internet) as a whole.

**Algorithm:** This is a set of detailed instructions for accomplishing a task by proceeding through a series of well-defined successive stages. Search Engines use algorithms to assign Quality Score and rankings to websites. Algorithms can be extremely complex (Google looks at over 200 variables) and are closely guarded by the Search Engines.

**1.7 Project Layout**

This research work is organized into five chapters. A brief description of each chapter is given below;

* Chapter one is concerned with the introduction of the research study and it presents the preliminaries of the system to be built.
* Chapter two focuses on the literature review, the contributions of other scholars on the existing and similar systems.
* Chapter three is concerned with the system analysis and design. It presents the research methodology used in the development of the system, it analyses the present system to identify the problems and provides information on the advantages and disadvantages of the proposed system.
* Chapter four presents the system implementation and documentation, the choice of programming language, analysis of modules, choice of programming language and system requirements for implementation.
* Chapter five focuses on the summary, constraints of the study, conclusion and recommendations are provided in this chapter based on the study carried out.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 INTRODUCTION**

A literature review is a critical analysis of published sources, or literature, on a particular topic. It is an assessment of the literature and provides a summary, classification, comparison and evaluation. An understanding of the research study is shown by analyzing and then synthesizing the information to:

* Determine what has already been written or developed on the topic.
* Provide an overview of key concepts.

This chapter takes a concise look at database-driven search engines, concentrating specifically on academic search engines. Academic search engines are web-based search engines that provide a simple way to broadly search for scholarly literature. From one place, you can search across many disciplines and sources: articles, theses, books, abstracts and court opinions, from academic publishers, professional societies, online repositories, universities and other web sites.

Academic search engines are often database-driven. This simply means that, the articles provided by the publisher of the search engine are not placed on the Surface Web (i.e., there are no URLs for them) but they are accessible by Web users through querying the publisher’s database via the search engine. This chapter goes further to survey recent academic search engines.

**2.2 A BRIEF HISTORY OF SEARCH ENGINES**

The concept of hypertext and a memory extension really came to life in July of 1945, when after enjoying the scientific camaraderie that was a side effect of world war II, Vannevar Bushs urged scientists to work together to help build a body of knowledge for all mankind. He not only was a firm believer in storing data, but he also believed that if the data source was to be useful to the human mind we should have it represent how the mind works to the best of our abilities. He then proposed the idea of a virtually limitless, fast, reliable, extensible, associative memory storage and retrieval system. He named this device a MEMEX (searchenginehistory.com, 2015).

## Prior to this time there was no World Wide Web (www), the main way people shared data back then was via File Transfer Protocol (FTP).If you had a file you wanted to share you would set up an FTP server. If someone was interested in retrieving the data they could using an FTP client. This process worked effectively in small groups, but the data became as much fragmented as it was collected. The first Web site was built by Tim Berners-Lee at “http://info.cern.ch/” and was first put online on August 6, 1991. It provided an explanation about what the World Wide Web was, how one could own a browser and how to set up a Web server. It was also the world's first Web directory, since Berners-Lee maintained a list of other Web sites apart from his own (Coffman et al, 1997).

In 1994, Berners-Lee founded the World Wide Web Consortium (W3C) at the Massachusetts Institute of Technology (MIT).

The World Wide Web has emerged as the largest information source in recent years. People all over the world use the Web to find needed information on a regular basis. Students use the Web as a library to find references and customers use the Web to purchase various products. It is safe to say that the Web has already become an important part in many people's daily lives.

The precise size of the Web is a moving target as the Web is expanding very quickly. The Web can be divided into the Surface Web and the Deep Web (or Hidden Web) (Bergman, 2000). The former refers to the collection of Web pages that are publicly indexable. Each such page has a logical address called Uniform Resource Locator or URL. It was estimated that the Surface Web contained about 2 billion Web pages in 2000 (Bergman, 2000). The Hidden Web contains Web pages that are not publicly indexable. As an example, a publisher may have accumulated many articles in digital format. If these articles are not placed on the Surface Web but they are accessible by Web users through the publisher’s search engine, then these articles belong to the Deep Web. Web pages that are dynamically generated using data stored in database systems belong to the **Hidden Web**. A recent study estimated the size of the Hidden Web to be about 500 billion pages (Bergman, 2000).

In the last several years, many search engines have been created to help users find desired information on the Web. Search engines are easy-to-use tools for searching the Web. Based on what type of data is searched, there are **document-driven search** **engines** and **database-driven search engines**. The former searches documents (Web pages) while the latter searches data items from a database system through a Web interface (Yu &Meng, 1999). Database-driven search engines are mostly employed for e-commerce applications such as buying cars or books and research purposes. This study anchors on database-driven search engines.

**2.3 DATABASE DRIVEN SEARCH ENGINES**

Yu & Meng (1999) described a database-driven search engine as a [search engine](https://en.wikipedia.org/wiki/Search_engine) that operates on material stored in a digital [database](https://en.wikipedia.org/wiki/Database). Database-driven search technology is used by large public and private entities including government database services, e-commerce companies, online advertising platforms, telecommunications service providers, academic research platforms and other consumers with a need to access information in large repositories.

A database-driven search engine generally consists of the following components:

• User Interface

• Query Engine

• Indexer

• Database

**USER INTERFACE**

A Graphical User Interface (GUI) is generally provided by every search engine. The users looking for information provide the search string based on which the search engine searches its local database and displays result in an orderly fashion. In this case, the users simply write their queries in the text box provided as a word or phrase which could be a department, faculty, author or topic.

**QUERY ENGINE**

The goal behind using a search engine is to provide high quality search result to users in response to search queries provided by them. On behalf of the search engine it is the query engine which performs this activity using some query processing algorithms.

**INDEXER**

Indexing of documents is very important as without an index the search engine will have to scan every document in its database whereby consuming considerable time and computing power. This requires processing of web pages, deciding which of them to index and building various data structures representing the downloaded pages. This helps in fast and accurate retrieval of information from the search engine repository. The document index keeps information about each document such as current document status, a pointer into the repository, a document checksum and various other statistics.

**DATABASE**

The database is used to store and manage large collection of documents. [Databases](https://en.wikipedia.org/wiki/Databases) allow logical queries such as the use of multi-field [Boolean logic](https://en.wikipedia.org/wiki/Boolean_logic), while full-text searches do not. "Crawling" (a human by-eye search) is not necessary to find information stored in a database because the data is already structured. [Indexing](https://en.wikipedia.org/w/index.php?title=Index_%28datatbase%29&action=edit&redlink=1) the data allows for faster searches. Database-driven search engines are usually included with major database software products.

**2.4 INFORMATION RETRIEVAL**

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of [relevancy](https://en.wikipedia.org/wiki/Relevance_%28information_retrieval%29).

An object is an entity that is represented by information in a content collection or [database](https://en.wikipedia.org/wiki/Database). User queries are matched against the database information. However, as opposed to classical SQL queries of a database, in information retrieval the results returned may or may not match the query, so results are typically ranked. This ranking of results is a key difference of information retrieval searching compared to database searching (Jansen & Rieh, 2010).

Information Retrieval systems like the “ProSearch” compute a numeric score on how well each object in the database matches the query, and rank the objects according to this value. The top ranking objects are then shown to the user. The process may then be iterated if the user wishes to refine the query (William, 1992).

The retrieval effectiveness of a text retrieval system is often measured by a pair of quantities known as **recall**and **precision**. Suppose, for a given user query, the set of relevant documents in the document collection is known. Recall and precision are defined as follows:

1. Recall = {relevant documents} {retrieved documents}

{retrieved documents}

Recall is the fraction of the documents that are relevant to the query that are successfully retrieved.

1. Precision = {relevant documents} {retrieved documents}

{retrieved documents}

Precision is the fraction of the documents retrieved that are [relevant](https://en.wikipedia.org/wiki/Relevance_%28information_retrieval%29) to the user's information need.precision = | { relevant documents } ∩ { retrieved documents } | | { retrieved documents } | {\displaystyle {\mbox{precision}}={\frac {|\{{\mbox{relevant documents}}\}\cap \{{\mbox{retrieved documents}}\}|}{|\{{\mbox{retrieved documents}}\}|}}}

To evaluate the effectiveness of a document retrieval system, a set of test queries is often used. For each query, the set of relevant documents is identified in advance. For each test query, a precision value for each distinct recall value is obtained. Usually only eleven recall values, 0.0, 0.1, ..., 1.0, are considered. When the precision values at each recall value are averaged over all test queries, an average recall-precision curve is obtained.

An ideal document retrieval system should retrieve exactly the set of relevant documents for each query. In other words, a perfect system has recall = 1 and precision = 1 at the same time. In practice, perfect performance is not achievable due to many reasons. For example, a user's needs may be incorrectly or imprecisely specified by the query used and the representation of documents and queries as vectors does not capture their contents completely (Yu &Meng, 1999).

**2.5 ACADEMIC-RESEARCH WEB SEARCH ENGINES**

Niche-specific content is usually not readily available through regular generic search engines. One example is the academic and scholarly content. While running a search query about an academic topic through a generic search engine such as Google would probably render fairly decent results, it, however, usually takes digging into so much fluff before finally landing on relevant results. This is where having access to topic-specific search engines comes in handy. Such search engines do not only provide specific content tailored to the topic under study but their content is more likely to be reliable and authoritative **(**educatorstechnology.com, 2016).

To this end, academic search engines are designed for teachers, student researchers and academicians and scholars to enable them quickly locate and access scholarly works and publications.

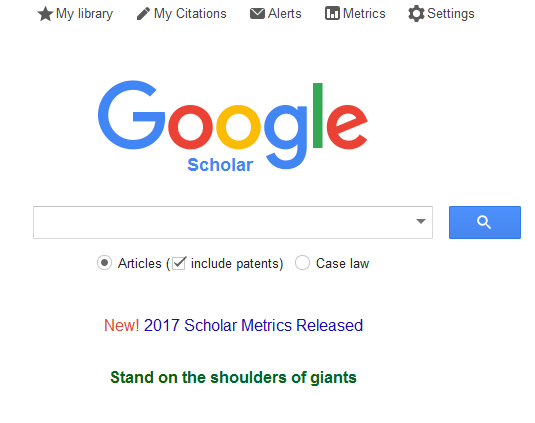
**2.6 REVIEW OF THE EXISTING SYSTEM**

Some existing academic search engines that exhibit similar or partially similar characteristics with that of the proposed system are highlighted below;

### 2.6.1. Google Scholar

Google Scholar is a freely accessible [web search engine](https://en.wikipedia.org/wiki/Web_search_engine) (developed by *Google* and launched in November 20, 2004; 12 years ago) that indexes the full text or metadata of [scholarly literature](https://en.wikipedia.org/wiki/Scholarly_literature) across an array of publishing formats and disciplines. Released in [beta](https://en.wikipedia.org/wiki/Beta_release) in November 2004, the Google Scholar index includes most [peer-reviewed](https://en.wikipedia.org/wiki/Peer_review) online [academic journals](https://en.wikipedia.org/wiki/Academic_journal) and books, conference papers, [theses](https://en.wikipedia.org/wiki/Thesis) and [dissertations](https://en.wikipedia.org/wiki/Dissertation), [preprints](https://en.wikipedia.org/wiki/Preprint), [abstracts](https://en.wikipedia.org/wiki/Abstract_%28summary%29), [technical reports](https://en.wikipedia.org/wiki/Technical_report), and other scholarly literature, including [court opinions](https://en.wikipedia.org/wiki/Legal_opinion) and [patents](https://en.wikipedia.org/wiki/Patent). While [Google](https://en.wikipedia.org/wiki/Google) does not publish the size of Google Scholar's database, third-party researchers estimated it to contain roughly 160 million documents as of May 2014 and an earlier statistical estimate published in [PLOS ONE](https://en.wikipedia.org/wiki/PLOS_ONE) using a [Mark and recapture](https://en.wikipedia.org/wiki/Mark_and_recapture) method estimated approximately 80-90% coverage of all articles published in English with an estimate of 100 million. This estimate also determined how many documents were freely available on the web.

[Google Scholar](https://scholar.google.com/) was created as a tool to congregate scholarly literature on the web. From one place, students have the ability to hunt for peer-reviewed papers, theses, books, abstracts and articles from academic publishers, professional societies, preprint repositories, universities and other scholarly organizations (googlescholar.com/about, 2016).



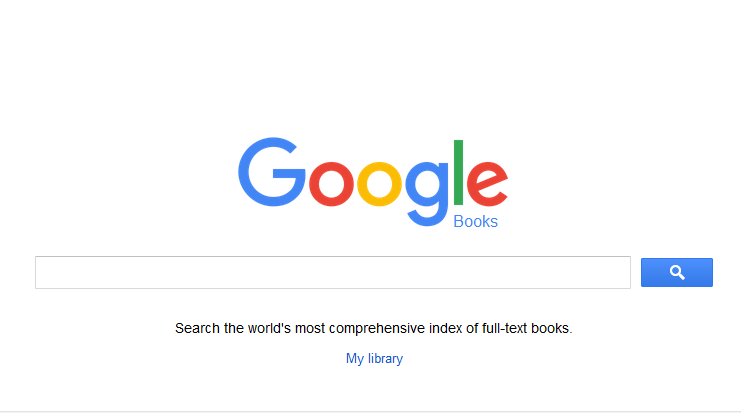
**Fig 2.1 The GUI for Google Scholar search engine**

### 2.6.2. Google Books

Google Books (previously known as Google Book Search and Google Print and by its codename Project Ocean) is a service from [Google Inc.](https://en.wikipedia.org/wiki/Google) that searches the full text of books and magazines that Google has scanned, converted to text using [optical character recognition](https://en.wikipedia.org/wiki/Optical_character_recognition) (OCR), and stored in its digital database. Books are provided either by publishers and authors, through the Google Books Partner Program, or by Google's library partners, through the Library Project. Additionally, Google has partnered with a number of magazine publishers to digitize their archives.

The Publisher Program was first known as 'Google Print' when it was introduced at the [Frankfurt Book Fair](https://en.wikipedia.org/wiki/Frankfurt_Book_Fair) in October 2004. The Google Books Library Project, which scans works in the collections of library partners and adds them to the digital inventory, was announced in December 2004. The Google Books initiative has been hailed for its potential to offer unprecedented access to what may become the largest online body of human knowledge and promoting the [democratization of knowledge](https://en.wikipedia.org/wiki/Democratization_of_knowledge). But it has also been criticized for potential copyright violations, and lack of editing to correct the many errors introduced into the scanned texts by the OCR process. As of October 2015, the number of scanned book titles was over 25 million, but the [scanning process](https://en.wikipedia.org/wiki/Book_scanning) has slowed down in American academic libraries (Stephen 2015). Google estimated in 2010 that there were about 130 million distinct titles in the world, and stated that it intended to scan all of them (googlebooks.com/about, 2010)*.*

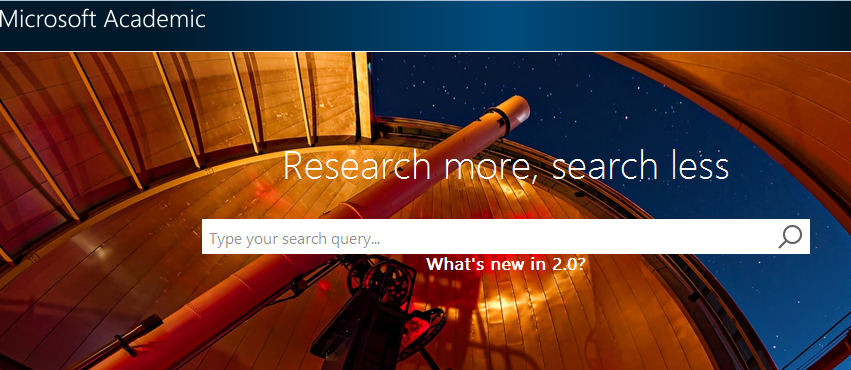
[Google Books](https://books.google.com/?hl=en) allows web users to find the book you are looking for; you can look through pages, find online reviews and learn where you can get a hard copy.



**Fig 2.2 The GUI for Google Books search engine**

### 2.6.3. Microsoft Academic

Microsoft Academic is a free public [search engine](https://en.wikipedia.org/wiki/Search_engine) for [academic publications](https://en.wikipedia.org/wiki/Academic_publication) and literature, developed by [Microsoft Research](https://en.wikipedia.org/wiki/Microsoft_Research). Re-launched in 2016 (February 22, 2016; 15 months ago to be precise), the tool features an entirely new data structure and search engine using [semantic search](https://en.wikipedia.org/wiki/Semantic_search) technologies and it currently indexes over 150 million entities (Arnab et al, 2015). The Academic Knowledge API offers information retrieval from the underlying database using REST endpoints for advanced research purposes. [Microsoft Academic](http://academic.research.microsoft.com/) is a reliable, comprehensive research tool. You can search directly by topic, or you can search by an extensive list of fields of study. For example, if you’re interested in computer science, you can filter through topics such as artificial intelligence, computer security, data science, programming languages and more (microsoftacademic.com/about, 2015).



**Fig 2.3 The GUI for Microsoft Academic search engine**

### 2.6.4. WorldWideScience

WorldWideScience.org is a global science search engine ([Academic databases and search engines](https://en.wikipedia.org/wiki/Academic_databases_and_search_engines)) designed to accelerate scientific discovery and progress by accelerating the sharing of scientific knowledge. Through a multilateral partnership, WorldWideScience.org enables anyone with internet access to launch a single-query search of national scientific databases and portals in more than 70 countries, covering all of the world’s inhabited continents and over three-quarters of the world’s population. From a user’s perspective, WorldWideScience.org makes the databases act as if they were a unified whole.

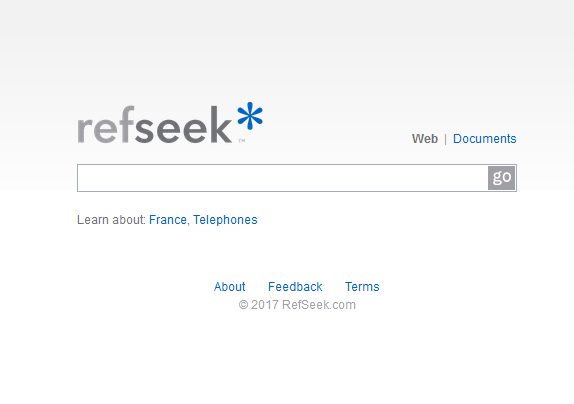
WorldWideScience.org implements [federated searching](https://en.wikipedia.org/wiki/Federated_search) to provide its coverage of global science and research results. Federated searching technology allows the information patron to search multiple data sources with a single query in real time. It provides simultaneous access to "[deep web](https://en.wikipedia.org/wiki/Deep_Web_%28search_indexing%29)" scientific databases, which are typically not searchable by commercial search engines. In June 2010, WorldWideScience.org implemented multilingual translations capabilities. Using Microsoft's [Bing Translator](https://en.wikipedia.org/wiki/Bing_Translator), Multilingual WorldWideScience.org offers the user the ability to search across databases in ten languages and then have the results translated into their preferred language. "One to many" and "many to one" machine translations can be performed for Arabic, Chinese, English, French, German, Japanese, Korean, Portuguese, Russian, and Spanish ([WorldWideScience.org, 2010](http://blogs.msdn.com/b/msr_er/archive/2010/06/16/microsoft-research-and-worldwidescience-org-collaborate-to-break-down-language-barriers.aspx)).



**Fig 2.4 The GUI for WorldWideScience.org search engine**

### 2.6.5. Refseek

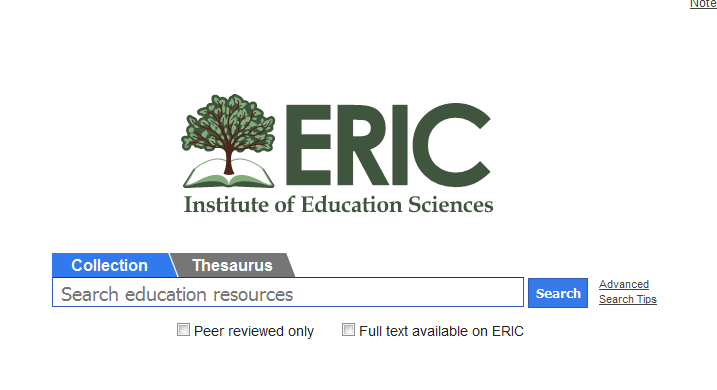
[Refseek](http://www.refseek.com/) pulls from over one billion web pages, encyclopedias, journals and books. It is similar to Google in its functionality, except that it focuses more on scientific and academic results—meaning more results will come from .edu or .org sites, as well as online encyclopedias. It also has an option to search documents directly—providing easy access to PDFs of academic papers ([Heinrich,](http://www.rasmussen.edu/student-life/blogs/author-archives/Anna%20Heinrich) 2009).



**Fig 2.5 The GUI for RefSeek search engine**

### 2.6.6. Education Resources Information Center

Populated by the U.S. Department of Education, the [Education Resources Information Center](https://eric.ed.gov/) (ERIC) is a great tool for academic research with more than 1.3 million bibliographic records of articles and online materials. ERIC provides access to an extensive body of education-related literature including journal articles, books, research syntheses, conference papers, technical reports, policy papers and more. With more than eight million searches each month, it’s no wonder why this search engine is a great web source for education ([Heinrich,](http://www.rasmussen.edu/student-life/blogs/author-archives/Anna%20Heinrich) 2009).

****

**Fig 2.6 The GUI for ERIC search engine**

### 2.6.7. iSeek

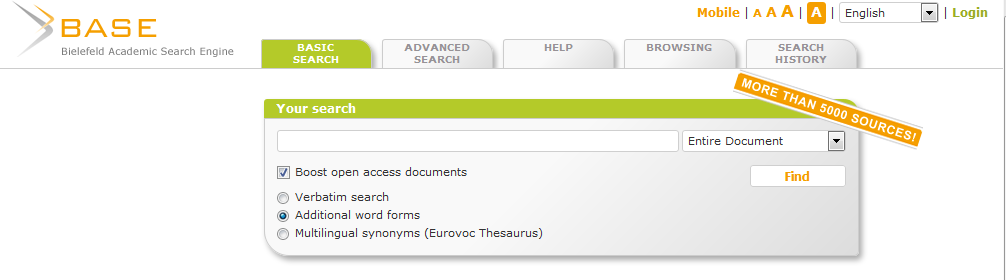
[iSeek](http://education.iseek.com/iseek/home.page) is a great search engine for students, teachers and administrators alike. Simply ask a question or enter search topics or tools, and iSeek will pull from scholastic sources to find exactly what you are looking for. The search engine is safe, intelligent and timesaving—and it draws from trusted resources from universities, government and established non-commercial sites ([Heinrich,](http://www.rasmussen.edu/student-life/blogs/author-archives/Anna%20Heinrich) 2009).



**Fig 2.7 The GUI for iSeek search engine**

### 2.6.8. BASE

The [Bielefeld Academic Search Engine](https://www.base-search.net/) (BASE) prides itself as being “one of the world’s most voluminous search engines especially for academic web resources.” Utilizing 4,000 sources, the site contains results from over 100 million documents. The advanced search option allows users to narrow their research—so whether you’re looking for a book, review, lecture, video or thesis, BASE can provide the specific format you need ([Heinrich,](http://www.rasmussen.edu/student-life/blogs/author-archives/Anna%20Heinrich) 2009).



**Fig 2.8 The GUI for BASE search engine**

**2.6.9. UM Research Repository**

UM Research Repository is the University of Malaya’s perpetual, accessible and growing collection of research work which includes peer-reviewed articles, conferences and working papers representing their rich intellectual community. It has been established to provide a deposit service for the academic staff and researchers. It is an initiative of the UM Library.

Where possible the details of each item described in the archive will include a link to a freely available electronic copy of the full text or other electronic documentation of the research output (repository.um.edu.my, 2017).



**Fig 2.9 The GUI for UMR Repository search engine**

* 1. **FEATURES OF ACADEMIC SEARCH ENGINE**

Based on the reviewed existing academic search engines, information obtained from conducted research showed that the system offers specificity as regards the users’ (students’) required information. The system is best for college level research and when a user needs to find credible information quickly.

Some features of the academic search engine are outlined as follows;

1. **Type of Information Retrieved**: The information retrieved as a search result includes;

* Scholarly journal articles
* Popular magazine articles
* Newspaper articles
* Reference book articles (e.g., directories, encyclopedias)
* Books
* Project works and references.

1. **Creditability:** All material in the [database](http://library.vccs.edu/license-bin/linker.plx?ebscoa9h) is evaluated for accuracy and credibility by subject experts and the publishers. The articles and books are written by journalists or experts in a professional field and these documents are reviewed and updated regularly.
2. **Usability:** The organization and various search capabilities of  [databases](http://library.vccs.edu/license-bin/linker.plx?ebscoa9h)-driven academic search engine users ensure the search and retrieval of focused and relevant results.
3. **Constancy / Permanence / Stability:** Publishedcontent from journals, magazines, newspapers and books does not change. Most material remains in [database](http://library.vccs.edu/license-bin/linker.plx?ebscoa9h) for a significant length of time and can be easily retrieved again. Integrity of data is also assured due to the fact that documents stored in the database cannot be compromised.

**ADVANTAGES OF THE ACADEMIC SEARCH ENGINE**

* **Wider dissemination:** E-print repositories enable free worldwide web-based access to research outputs.
* **Increased impact:** Studies show that research deposited in open access repositories have considerably increased impact as citation rates rise.
* **A showcase for research:** The repository provides a means to bring together your research output. The repository can be linked from your research centre or faculty web pages.

**CHAPTER 3**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 INTRODUCTION**

This chapter describes the system design methodology. It starts by outlining the systems requirement specification, then goes on to further elaborate on the system design by illustrating with various UML diagrams such as the Use case diagram which models users of the system and use cases alike; it also specifies activities of each of these users using the activity diagram and then shows the relationship between various classes using the UML class diagrams. Interface designs are also shown via the menu, input and output designs.

**3.2 ANALYSIS OF THE SYSTEM**

This is a method of studying a system by examining its component parts and their interactions to achieve the desired goal.

**3.2.1 SYSTEM REQUIREMENT SPECIFICATION**

This section would focus only on the functional systems requirements of the proposed system. It is centered on the functionality that the proposed system is to provide. These are outlined below;

* The system should be able to validate search filled form in its database.
* The system should be able to handle exception.
* The system should be able to return search entry.
* It should be able to retrieve successful search records.
* The system should be able to return appropriate results given the student name or the lecturer name or the research topic.
* The system should be able to return search result for multiple research with the same name (parse information).

**3.3 SYSTEM DESIGN**

The systems design, defines the architecture, components, modules, interface and data for a system that satisfy specified requirements. In this section, there are different and separate steps to be followed which are ensured in the logical design process. The system’s design would also incorporate interface designs and UML diagrams used to explain the architecture of the system. The system’s design allows the user gain a detailed understanding of how the system functions. This usage profile can be leveraged to develop future architecture changes. A baseline performance level is established against which benefits can be compared and changes to the system predicted or foreseen.

### 3.3.1 ARCHITECTURAL DESIGN

The architectural design of the system emphasizes the design of the [system architecture](https://en.wikipedia.org/wiki/System_architecture) that describes the [structure](https://en.wikipedia.org/wiki/Structure), [behavior](https://en.wikipedia.org/wiki/Behavior) and more [views](https://en.wikipedia.org/wiki/View_model) of that system and analysis.

The system is a web-based system and it follows a three-tier architecture design in its development. The three-tier architecture is a client–server [software architecture pattern](https://en.wikipedia.org/wiki/Software_Architecture_styles_and_patterns) in which the [user interface](https://en.wikipedia.org/wiki/User_interface) (presentation), [functional process logic](https://en.wikipedia.org/wiki/Business_logic_layer) ("business rules"), [computer data storage](https://en.wikipedia.org/wiki/Computer_data_storage) and [data access](https://en.wikipedia.org/wiki/Data_access) are developed and maintained as independent [modules](https://en.wikipedia.org/wiki/Modular_programming), most often on separate [platforms](https://en.wikipedia.org/wiki/Platform_%28computing%29). It was developed by [John J. Donovan](https://en.wikipedia.org/wiki/John_J._Donovan) in Open Environment Corporation (OEC), a tools company he founded in [Cambridge, Massachusetts](https://en.wikipedia.org/wiki/Cambridge,_MA).

CLIENT

APPLICATION SERVER

DATABASE SERVER

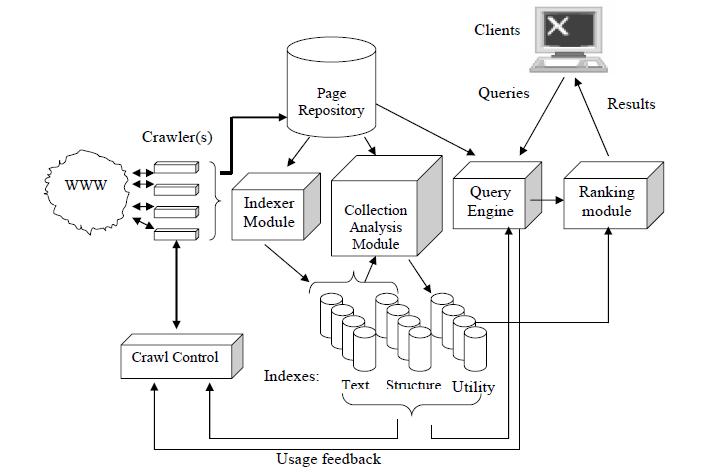
Request

Reply

Command

Result

**Fig 3.1 the Three Tier Architecture**



**Fig 3.2 General Search Engine Architecture**

**3.3.2 PHYSICAL DESIGN**

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is processed and how it is displayed. In physical design, the following requirements about the system are decided.

1. **Input requirement:** the input requirement for this system is a query in text form. The text is imputed into the provided search box in the interface.
2. **Output requirements:** the output requirements are pdf documents.
3. **Storage requirements:** the storage requirement for this system is the database.
4. **Processing requirements:** the process requirements involve good internet access on the device.

**3.3.3 PROCESS DESIGN**

The search engine typically does the following for every search query entered;

* Accept the user’s inputted query, checking to match any advanced syntax and checking to see if the query is misspelled to recommend more popular or correct spelling variations.
* Check to see if the query is relevant to other vertical search databases and place relevant links to a few items from that type of search query near the regular search results.
* Gather a list of relevant documents for the organic search results. These results are ranked based on content similarity.

**3.3.4 INPUT DESIGN**

The input design facilitates the entry of data into the computer system. This entails the selection of the best strategy for getting data into the computer system at the right time and as accurately as possible. The use of well-defined interfaces encourages users to input data accurately without omission. Input design must capture all the data that the system needs without introducing any errors.

* **User Interface Design**

User interface design (UI) or user interface engineering is the [design](https://en.wikipedia.org/wiki/Design) of [user interfaces](https://en.wikipedia.org/wiki/User_interface) for [machines](https://en.wikipedia.org/wiki/Machine) and [software](https://en.wikipedia.org/wiki/Software), such as [computers](https://en.wikipedia.org/wiki/Computer), [home appliances](https://en.wikipedia.org/wiki/Home_appliance), [mobile devices](https://en.wikipedia.org/wiki/Mobile_device), and other [electronic devices](https://en.wikipedia.org/wiki/Electronics), with the focus on maximizing [usability](https://en.wikipedia.org/wiki/Usability) and [user experience](https://en.wikipedia.org/wiki/User_experience). The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals ([user-centered design](https://en.wikipedia.org/wiki/User-centered_design)).

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. [Graphic design](https://en.wikipedia.org/wiki/Graphic_design) and [typography](https://en.wikipedia.org/wiki/Typography) are utilized to support its [usability](https://en.wikipedia.org/wiki/Usability), influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface.

**Graphical User Interface**

The graphical user interface (GUI), is a type of [user interface](https://en.wikipedia.org/wiki/User_interface) that allows [users](https://en.wikipedia.org/wiki/User_%28computing%29) to [interact with electronic devices](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction) through graphical [icons](https://en.wikipedia.org/wiki/Computer_icon) and visual indicators such as secondary notation, instead of [text-based user interfaces](https://en.wikipedia.org/wiki/Text-based_user_interface), typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep [learning curve](https://en.wikipedia.org/wiki/Learning_curve) of [command-line interfaces](https://en.wikipedia.org/wiki/Command-line_interface) (CLIs), which require commands to be typed on a [computer keyboard](https://en.wikipedia.org/wiki/Computer_keyboard).

The actions in a GUI are usually performed through [direct manipulation](https://en.wikipedia.org/wiki/Direct_manipulation) of the graphical elements.

**THE SEARCH WINDOW**

The Search window prompts the user to search for research materials using various search methods. The search window is on the index page of the Search engine and it prompts the user to search using the project date, project topic, faculty, department, and project description for a more refined search.

Enter Project Date

Enter Project Title

Enter Project Description

**SEARCH**

Select Faculty

Choose Department

**Fig 3.3 Search Window**

**3.3.5 OUTPUT DESIGN**

The output design allows the system to indicate the effect of the users’ manipulation. This system has only one output design which is the Search Result Window.

Search Result field

|  |
| --- |
| Search Result 1 |
| Search Result 2 |
| Search Result 3 |
| Search Result … |

**Fig 3.2.3 Search Result Window**

**Fig 3.4 Search result**

As shown in Fig 3.4, the Search Result window of the system presents the user the results of the search listed in table form ranked by similarity to the query.

**3.3.6 DATABASE DESIGN**

MySQL (My Structured Query Language) Relational database management system (RDBMS) will be used to store information about the system. Table 3.1 below shows the various tables and a description of what each table stores.

**TABLE 3.1 DATABASE TABLE**

|  |  |  |
| --- | --- | --- |
| **S/N** | **DATABASE TABLE** | **DISCRIPTION** |
| 1 | admin | Stores login details of the database administrator(s). |
| 2 | agriculture | Stores all the documents for the departments in the faculty of AGRICULTURE. |
| 3 | engineering | Stores all the documents for the departments in the faculty of ENGINEERING. |
| 4 | management\_science | Stores all the documents for the departments in the faculty of MANAGEMENT SCIENCE. |
| 5 | science | Stores all the documents for the departments in the faculty of SCIENCE. |
| 6 | environmental\_science | Stores all the documents for the departments in the faculty of ENVIRONMENTAL SCIENCE. |
| 7 | law | Stores all the documents for the departments in the faculty of LAW. |
| 8 | education | Stores all the documents for the departments in the faculty of EDUCATION. |

**TABLE 3.2: LOGIN (ADMIN LOGIN TABLE)**

This table holds the information of the database administrator(s) authorized to manage records.

|  |  |  |
| --- | --- | --- |
| FIELD NAME | FIELD TYPE | CONSTRAINT |
| SN | INTEGER | NOT NULL |
| First Name | VARCHAR | NOT NULL |
| Last Name | VARCHAR | NOT NULL |
| Email Address | VARCHAR | NOT NULL |
| Password | VARCHAR | NOT NULL |

**TABLE 3.3: RESEARCH TABLE**

This table holds all the details of the projects as shown in the table below.

|  |  |  |
| --- | --- | --- |
| FIELD NAME | FIELD TYPE | CONSTRAINT |
| SN | INTEGER | NOT NULL |
| Full Name | VARCHAR | NOT NULL |
| Email Address | VARCHAR | NOT NULL |
| Department | TEXT | NOT NULL |
| Faculty | TEXT | NOT NULL |
| Project Title | VARCHAR | NOT NULL |
| Date uploaded | VARCHAR | NOT NULL |
| FilePath | TEXT | NOT NULL |
| Supervisor’s Name | VARCHAR | NOT NULL |
| Document Description | TEXT | NOT NULL |

**3.4 THE USE CASE DIAGRAM**

**ProSearch Engine System**

**Admin**

**User**

One can visualize high level system functions or requirements by drawing use case diagrams, which contain primary actors and use cases. Actors are entities that interact with the system, while use cases are system, that functions actors involve in. it should be noted that the use case diagram is used to tell readers what the system should do, but not how to do it. Fig 3.3 shows a typical user interaction with the system. First is the user with the system which is validating search entries and display search detail

**3.5 THE SYSTEM FLOW CHART**

No

Yes

No

Add Project

Edit Project

Delete Project

Validate Result

Yes

Database

Home Page

Search Project?

Verify Admin?

Validate Search

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

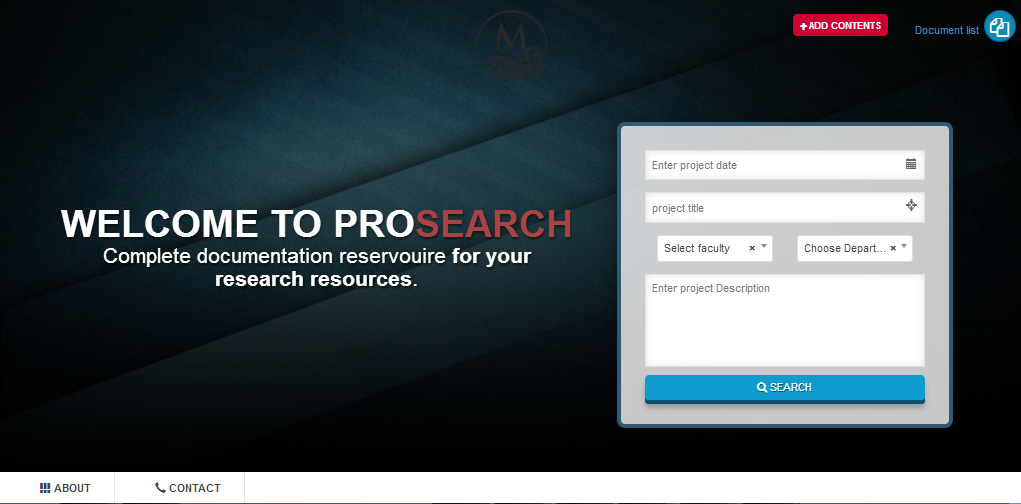
**4.1 INTRODUCTION**

This chapter describes the implementation and evaluation of the proposed system based on the analysis done in the previous chapters. System implementation enables the software developer to specify software function and evaluate performance. This includes software interfaces with other system elements.

**4.2 SAMPLE OUTPUTS (INTERFACES)**

The output interfaces are the graphical user interfaces of the designed system. Below are the various interfaces through which the user sends and retrieves data to and from the system.

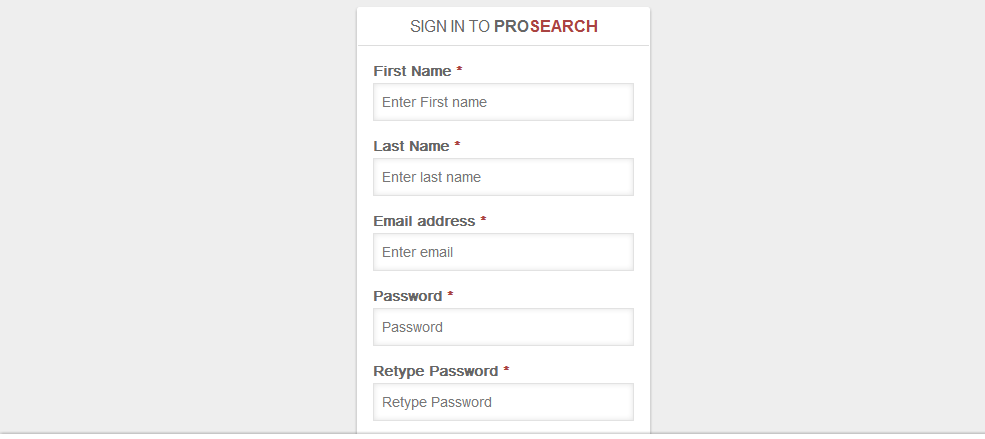
**THE INDEX PAGE**



**Fig 4.1: The Index Page**

The index page is the first page viewed by the user as the web site loads. This page displays a brief welcome note to the user and it also displays the search window wher the user enters the query for the search.

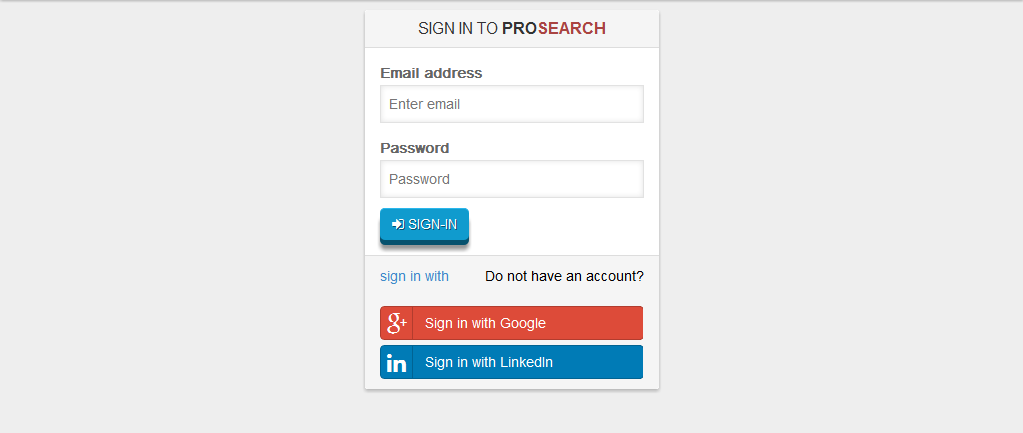
**THE SIGN UP PAGE**

****

**Fig 4.2: Sign Up Page**

This page gives the system administrator(s) the platform to be register his/her details in the system’s database.

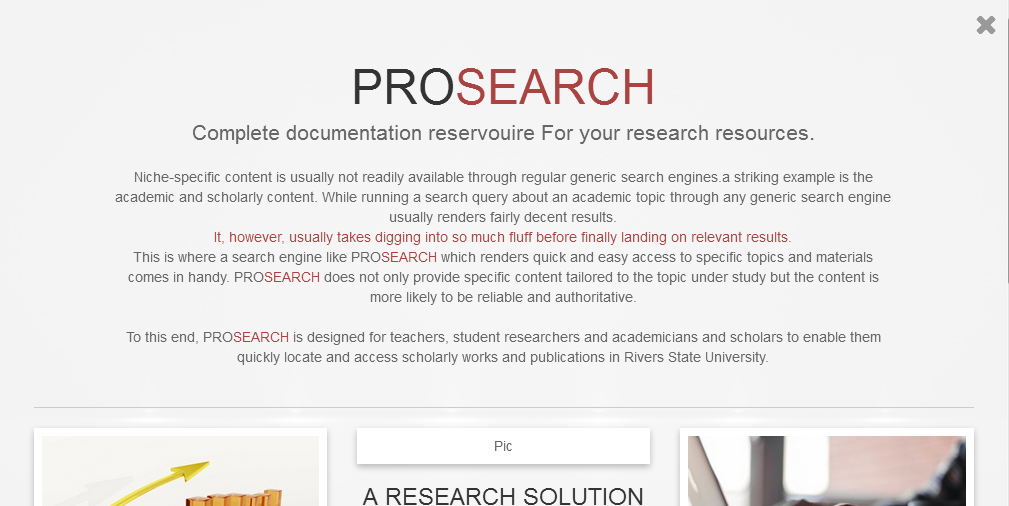
**THE SIGN IN PAGE**

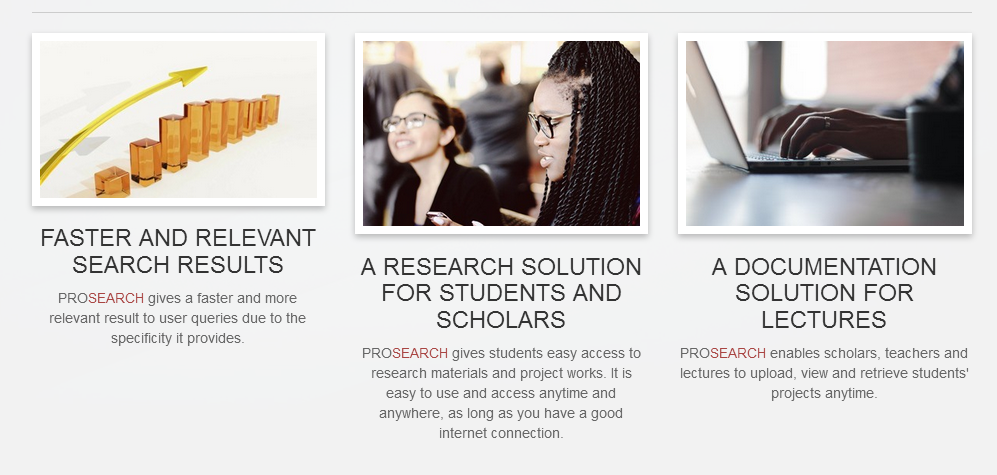
****

**Fig 4.3: Sign In Page**

This is an authentication page for verification of the identity of the said administrator. It contains the various text fields; email address and password.

**THE ABOUT PAGE**

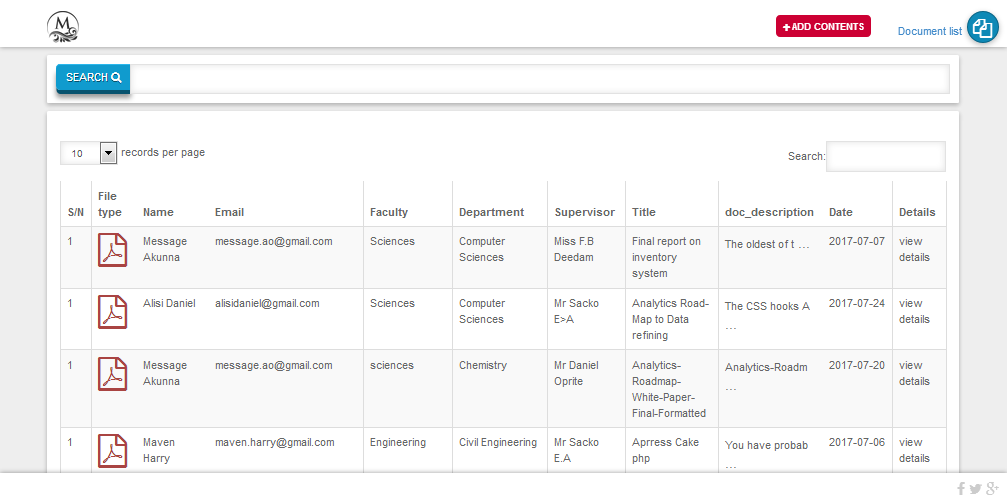
****

****

**Fig 4.4: The About Page**

The about page gives the users a brief description of the “ProSearch” search engine.

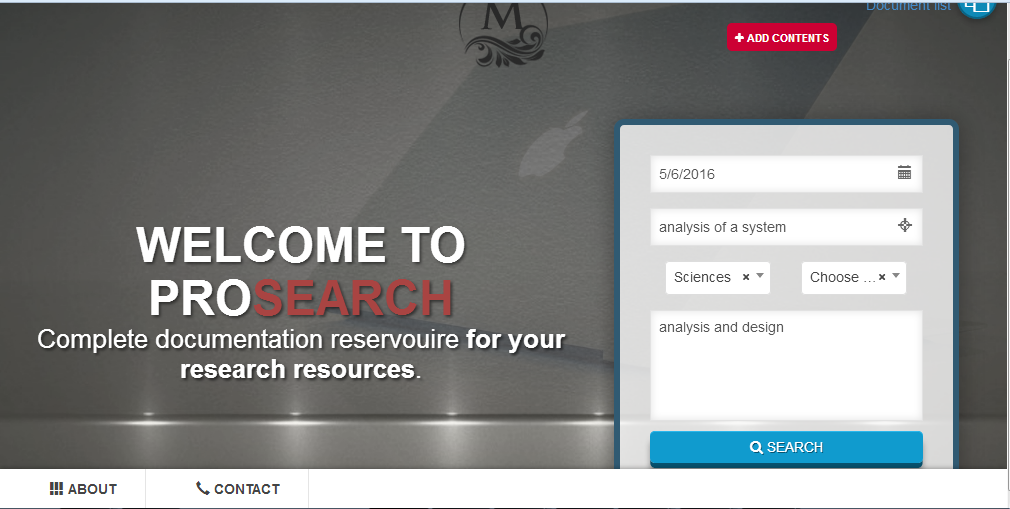
**THE DOCUMENTS LIST PAGE**

****

**Fig 4.5: Document List Page**

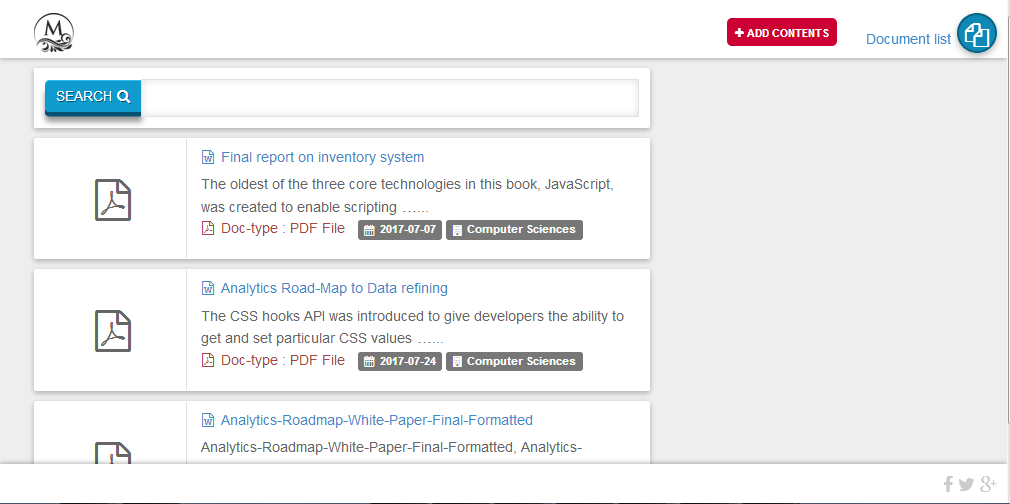
The document list page displays every document in the database. This page can only be viewed and modified by the admin. The admin is capable of viewing and deleting documents from the database from this page.

**THE SEARCH QUERY PAGE**

****

**Fig 4.6: The Search Query Page**

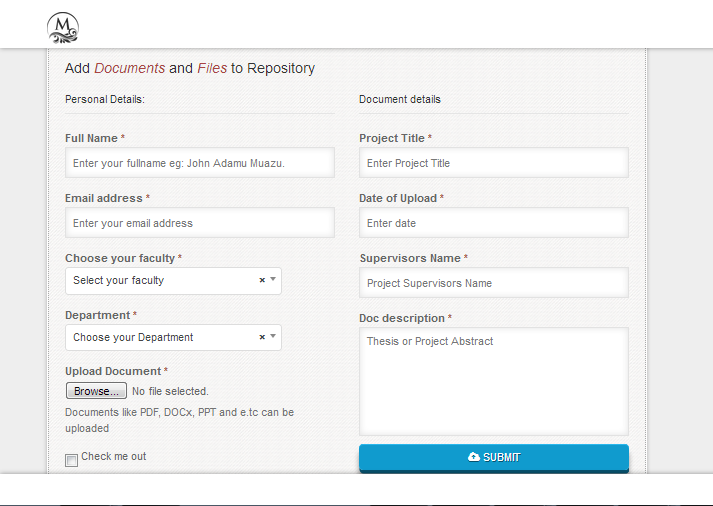
**THE SEARCH RESULT PAGE**

****

**Fig 4.7: Search Result Page**

The search result page displays all the results of the search query entered by the user ranking it by similarity and relevance.

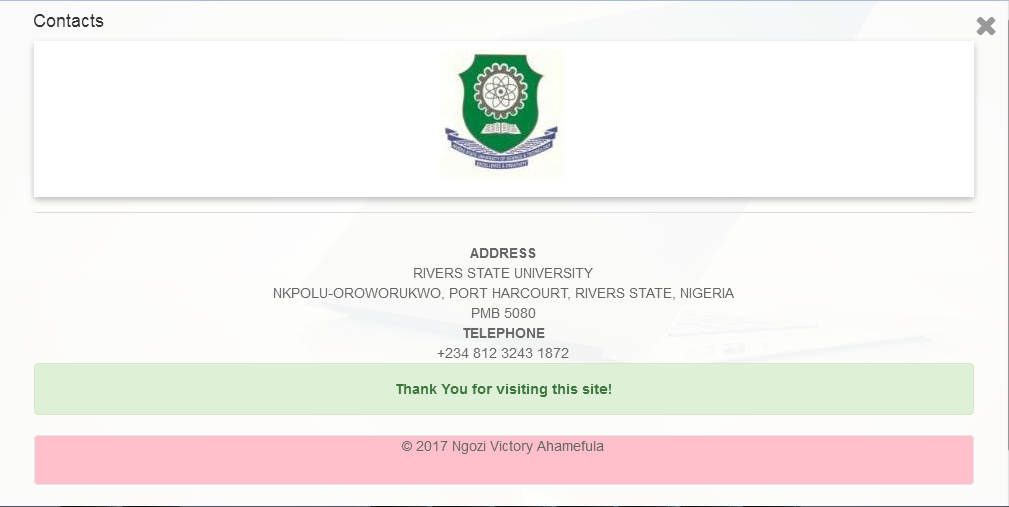
**ADD DOCUMENT PAGE**

****

**Fig 4.8: The Add Document Page**

This page is only accessible by the admin and it gives the admin the platform to add documents to the database.

**CONTACTS PAGE**

****

**Fig 4.9: Contacts Page**

* 1. **SYSTEM REQUIREMENTS**

The minimum requirements for the implementation of the system will be discussed below.

**4.3.1 SOFTWARE REQUIREMENTS**

The following are the software specifications required in order to access the search engine:

1. Operating System

* For systems (PCs): Windows, Linux, Unix.
* For mobile phones or tablets: Android OS, IPhone OS, and Windows OS etc.

1. Relational Database Management System: MySQL Database Server.
2. Internet Browser: Microsoft Internet Explorer, Mozilla Fire Fox, Google Chrome, Opera mini
3. PDF Reader (for viewing pdf documents): Adobe Reader etc
4. Document Viewer (for viewing word/text documents): Microsoft Word.

**4.3.2 HARDWARE REQUIREMENTS**

The following are the minimum hardware requirement to access the search engine:

1. Processor: Intel Centrino 1.6 Ghz Processor or higher or other equivalent processors.
2. 10GB memory capacity or higher
3. Memory: 512 RAM or Higher.
4. Others: Internet access
   1. **CHOICE OF PROGRAMMING LANGUAGE**

The programming languages used are;

* HTML
* PHP
* JAVA SCRIPT
* MySQL

**4.4.1 REASONS FOR THE CHOICE OF PROGRAMMING LANGUAGES**

* **HTML**

HTML, acronym for HyperText Markup Language, is a markup language used to describe the formatting of text in a document. It is useful in the sense that it allows text to be structured according to its purpose, namely as a heading, paragraph and so on. This is accomplished by writing the HTML in ‘tags’ that describes to the web browser how the text is to be displayed. A scripting language such as PHP and JavaScript can be easily embedded in HTML to enhance the functionality of HTML.

* **PHP**

To develop the proposed system, PHP (acronym for PHP Hypertext Preprocessor) was selected as the server-side scripting language. It is an open source language can be easily embedded into HTML codes to create interactive and dynamic web pages, which makes it especially appropriate for web development where the PHP code is executed in the web server which in turn creates the desired web page based on the codes. The platform compatibility of PHP is so vast that it can be used in most web servers, operating systems and can work in unison with majority databases

* **JAVASCRIPT**

Javascript is a programming / scripting language for browsers. It's most important use is to modify the HTML document tree, basically modifying the tags placed in the HTML source code. JavaScript itself is an interpreted programming language. It is plain text then downloaded by the web browser and executed by the JavaScript interpreter that’s built in into every web browser. In contrast to PHP which runs on the server side, JavaScript is a client side technology, which means that it runs inside the client (i.e. the browser). It allows the functionality of a website, manages user interaction, such as forms.

* **MySQL**

MySQL is a relational database management system (also known as an SQL Database Server) which is widely used around the globe due to it being open-sourced. Most SQL servers provide reliability but not ease of use unlike MySQL. It implements SQL functions using a highly optimized class library that should be as fast as possible. PHP and MySQL are tightly connected. Since PHP complements MySQL very well and given the fact that both technologies are widely used, it is can be certain that thorough testing has been done.

**CHAPTER 5**

**CONCLUSION**

**5.1 SUMMARY**

This chapter entails the summary of the entire work crowning the efforts invested on the study. Also, projecting the reason for the work and achievements obtained via the study.

**5.2 CONCLUSION**

This research work was centered on the development of an optimized database-driven academic web search engine. The purpose has simply been to design an academic search engine that highlights the importance of optimization and electronic storing of projects/research books in the repository. The system uses database to store this research materials enhancing portability, reliability, efficiency with increased accessibility.

In conclusion, the system will increase the efficiency of research activities in Rivers State University if adopted. The system is easily accessible and does not limit access to students outside the university.

**5.3 RECOMMENDATION**

Considering the fact that the objectives of this project have been achieved, it is therefore necessary to make obvious recommendation that would be helpful to the users of this system. The institution should host this website i.e. buy a domain name to enable easy access of the system from any part of the world.

**REFERENCES**

[Heinrich, A.](http://www.rasmussen.edu/student-life/blogs/author-archives/Anna%20Heinrich)  (2009), 15 Educational Search Engines College Students Should Know About. 91(8), 167-170

Sinha, A., Shen, Z., Song, Y., Ma, Y., Eide, D., Bo-June, H. & Kuansan, W. (2015), An Overview of Microsoft Academic Service (MA) and Applications. In Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion). ACM, New York, NY, USA, 243-246.

Bergman M., (2000), *The Deep Web: Surfacing the hidden value, Bright Planet,*

Yu, C. & Meng, W. (1999), Web Search Technology, Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, Department of Computer Science, SUNY at Binghamton Binghamton, NY 13902

Hawking, D. and Craswell, N. (1998), Overview of TREC-7 very large collection track. In *Proc. of the Seventh Text Retrieval Conf.*, pages 91-104.

Coffman, E.G., Zhen, L., & Weber, R. (1997). Optimal robot scheduling for web search engines. Technical report, INRIA.

William, B. (1992). [*Information Retrieval Data Structures & Algorithms*](http://www.scribd.com/doc/13742235/Information-Retrieval-Data-Structures-Algorithms-William-B-Frakes). Prentice-Hall, Inc. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [0-13-463837-9](https://en.wikipedia.org/wiki/Special:BookSources/0-13-463837-9).

Gandal, N. (2001). "The dynamics of competition in the internet search engine market". International Journal of Industrial Organization. **19** (7): 1103–1117.

"Internet History - Search Engines" (from [Search Engine Watch](https://en.wikipedia.org/wiki/Search_Engine_Watch)), University Leiden, Netherlands, September 2001, web: [LeidenU-Archie](http://www.internethistory.leidenuniv.nl/index.php3?c=7).

Jansen, B. & Rieh, S. (2010) [The Seventeen Theoretical Constructs of Information Searching and Information Retrieval](https://faculty.ist.psu.edu/jjansen/academic/jansen_theoretical_constructs.pdf). Journal of the American Society for Information Sciences and Technology. 61(8), 1517-1534

Haveliwala, T. (1999) Efficient computation of pagerank, Database Group,

Computer Science Department, Stanford University, February 1999.

Vaughan, A., Liwen, G., Thelwall, M. (2004). "Search engine coverage bias: evidence and possible causes". Information Processing & Management. **40** (4): 693–707

<https://www.gogglebooks>.com/about, June 20, 2017

[https://www.gogglescholar](https://en.wikipedia.org/w/index.php?title=Google_Scholar).com/about, 2nd June, 2017.

<https://www.educatorstechnology.com>, June 15, 2017

<https://www.merriamwebster.com/dictionary/optimization>, 25th April 2017.

<https://www.microsoftacademic>.org/about,June 20, 2017.

http://www.searchenginehistory.com June 20, 2017.

http://repository.um.edu.my/information.html, June 21, 2017.

[https:// www.worldwidescience](https://en.wikipedia.org/w/index.php?title=WorldWideScience).org/about, June 20, 2017.

**APPENDIX**

**INDEX PAGE**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Pro-Search site with Search optimizer for your research documents.</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Favicons-->

<link rel="shortcut icon" href="img/favicon.ico" type="image/x-icon"/>

<link rel="apple-touch-icon" type="image/x-icon" href="img/apple-touch-icon-57x57-precomposed.png">

<link rel="apple-touch-icon" type="image/x-icon" sizes="72x72" href="img/apple-touch-icon-72x72-precomposed.png">

<link rel="apple-touch-icon" type="image/x-icon" sizes="114x114" href="img/apple-touch-icon-114x114-precomposed.png">

<link rel="apple-touch-icon" type="image/x-icon" sizes="144x144" href="img/apple-touch-icon-144x144-precomposed.png">

<!-- CSS -->

<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">

<link href="js/jquery-ui/jquery-ui.min.css" rel="stylesheet">

<link href="css/style.css" rel="stylesheet">

<link href="font-awesome/css/font-awesome.css" rel="stylesheet">

<link rel="stylesheet" type="text/css" href="select/select2.min.css">

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

<!-- Owl Carousel Assets -->

<link href="css/owl.carousel.css" rel="stylesheet">

<link href="css/owl.theme.css" rel="stylesheet">

<!--[if lt IE 9]>

<script src="http://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

<script src="http://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>

<![endif]-->

<script type="text/javascript">

var \_gaq = \_gaq || [];

\_gaq.push(['\_setAccount', 'UA-11097556-8']);

\_gaq.push(['\_trackPageview']);

(function() {

var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;

ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';

var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);

})();

</script>

</head>

<body>

<!-- Preloader -->

<div id="preloader">

<div id="status"><img src="img/logo.png" alt=""></div>

</div>

<!-- end Preloader -->

<div id="wrapper">

<div id="main">

<div class="container">

<div id="logo"><a href="index.php"><img src="img/logo.png" alt=""></a></div>

<div id="purchase"><a href="add\_doc.php"><i class="fa fa-plus"></i> add contents</a></div>

<?php

if (isset($\_SESSION['email'])) {

echo'<div id="lang">

<ul>

<li class="text text-primary">Document list</li>

**<!-- JQUERY -->**

<script src="js/jquery.min.js"></script>

<script src="js/jquery-ui/jquery-ui.min.js"></script>

<script src="js/calendar\_func.js"></script>

<script src="js/jquery.easing.1.3.min.js"></script>

<script src="js/jquery.superslides.min.js"></script>

<script type="text/javascript">

$('#slides').superslides({

play: 6000,

pagination:false,

animation\_speed: 800,

animation: 'fade'

});

</script>

**<!-- OTHER JS -->**

<script src="js/retina.min.js"></script>

<script src="js/jquery.placeholder.min.js"></script>

<script src="js/functions.js"></script>

<script src="js/validate.js"></script>

<script src="js/wow.min.js" type="text/javascript"></script>

</script>

</body>

</html>

**SIGN IN PAGE**

<div id="main">

<div class="container" style="margin-bottom: 40px; ">

<div id="search-list" style="margin-top: 70px;">

<div class="row" style="">

<div class="col-md-offset-4 col-md-4">

<div class="search-box panel panel-default" style=" border: none !important;">

<div class="panel-heading" style="border-bottom: 1px solid #dddddd">

<h3 class="panel-title">Sign in to <strong> PRO<span class="text text-danger">SEARCH</span></strong></h3>

</div>

<div class="panel-body">

<?php

if (isset($\_GET['err'])) {

echo '<div class="alert alert-danger alert-dismissable"><button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button><strong>Error!</strong> user name or password is not correct..</div>';

}

?>

<form role="form" class="" method="post" action="">

<div class="form-group">

<label for="">Email address</label>

<input type="email" class="form-control" id="" placeholder="Enter email" name="email">

</div>

<div class="form-group">

<label for="">Password</label>

<input type="password" class="form-control" id="" placeholder="Password" name="password">

</div>

<div class="search-bar">

<button type="submit" class="btn btn-primary" name="signUp"><i class="fa fa-sign-in"></i> Sign-in</button>

</div>

</form>

</div>

<div class="panel-footer">

<p class="text text-primary">sign in with <a href="sign\_up.php" class="pull-right">Do not have an account?</a></p>

<a class="btn btn-block btn-social btn-google">

<i class="fa fa-google-plus"></i> Sign in with Google

</a>

<a class="btn btn-block btn-social btn-linkedin">

<i class="fa fa-linkedin"></i> Sign in with LinkedIn

</a>

</div>

</div>

</div>

</div>

</div><!-- End book -->

<?php

if (isset($\_POST['signUp'])) {

$user = new user;

$user->login();

}

?>

</div><!-- End container -->

**THE SEARCH LIST**

<div class="row">

<div class="col-md-8">

<ul class="postList">

<?php

$pDate = $\_GET['pDate'];

$pTitle = $\_GET['pTitle'];

$pFaculty = $\_GET['pFaculty'];

$pDepartment = $\_GET['pDepartment'];

$pDescription = $\_GET['pDescription'];

if ($pFaculty == 'sciences') {

$searchQuery = mysql\_query("SELECT \* FROM sciences.project WHERE (p\_date >= '$pDate') OR (project\_title LIKE '%$pTitle%') OR (department='$pDepartment') OR (doc\_description='%$pDescription%')");

$searchNum = mysql\_num\_rows($searchQuery);

if ($searchNum > 0){

while ($row = mysql\_fetch\_assoc($searchQuery)) {

$doc\_description=strlen($row['doc\_description']);

if ($doc\_description>100) {

$doc\_description=substr($row['doc\_description'], 0,100)."<big> ...</big>";

}

else {

$doc\_description=$row['doc\_description'];

}

?>

<li class="search-box panel">

<div class="search-content">

<div class="row">

<div class="media-thumbnail col-md-3" style=" padding-top: 10px; padding-bottom: 10px; display: block;">

<i class="fa fa-file-pdf-o fa-3x" style="margin: 30px 30px 20px 50px;"></i>

<!--<img src="img/room\_1.jpg" width="100%">-->

</div>

<div class="media-content col-md-9" style="border-left: 1px ridge #999999;">

<a href=""><h5 class="text text-primary"><i class="fa fa-file-word-o"></i>&nbsp; <?php echo $row['project\_title'];?> </h5></a>

<p><?php echo $doc\_description; ?>...<br>

<a href="" class="text text-danger"><i class="fa fa-file-pdf-o"> </i>&nbsp; Doc-type : PDF File</a>&nbsp;&nbsp;

<span class="label label-default"><i class="fa fa-calendar"> </i>&nbsp; <?php echo $row['p\_date']; ?></span>

<span class="label label-default"><i class="fa fa-building"> </i>&nbsp; <?php echo $row['department']; ?> </span>

</p>

</div>

</div>

</div>

</li>

<?php

}

}

else{

echo '<li class="search-box panel">

<div class="search-content">

<h4 class="text text-info">No Content matches your search criteria...!!</h4>

</div>

</li>';

}

}

?>

<?php

if ($pFaculty == 'engineering') {

$searchQuery = mysql\_query("SELECT \* FROM engineering.project WHERE (p\_date >= '$pDate') OR (project\_title LIKE '%$pTitle%') OR (department='$pDepartment') OR (doc\_description='%$pDescription%')");

$searchNum = mysql\_num\_rows($searchQuery);

if ($searchNum > 0){

while ($row = mysql\_fetch\_assoc($searchQuery)) {

$doc\_description=strlen($row['doc\_description']);

if ($doc\_description>150) {

$doc\_description=substr($row['doc\_description'], 0,150)."<big> ...</big>";

}

else {

$doc\_description=$row['doc\_description'];

}

?>

<li class="search-box panel">

<div class="search-content">

<div class="row">

<div class="media-thumbnail col-md-3" style=" padding-top: 10px; padding-bottom: 10px; display: block;">

<!-- <i class="fa fa-file-pdf-o fa-3x" style="margin: 30px 30px 20px 50px;"></i> -->

<img src="img/room\_1.jpg" width="100%">

</div>

<div class="media-content col-md-9" style="border-left: 1px ridge #999999;">

<a href=""><h5 class="text text-primary"><i class="fa fa-file-word-o"></i>&nbsp; <?php echo $row['project\_title'];?> </h5></a>

<p><?php echo $doc\_description; ?>...<br>

<a href="" class="text text-danger"><i class="fa fa-file-pdf-o"> </i>&nbsp; Doc-type : PDF File</a>&nbsp;&nbsp;

<span class="label label-default"><i class="fa fa-calendar"> </i>&nbsp; <?php echo $row['p\_date']; ?></span>

<span class="label label-default"><i class="fa fa-building"> </i>&nbsp; Department : <?php echo $row['department']; ?> </span>

</p>

</div>

</div>

</div>

</li>

<?php

}

}

else{

echo '<li class="search-box panel">

<div class="search-content">

<h4 class="text text-info">No Content matches your search criteria...!!</h4>

</div>

</li>';

}

}

?>

<?php

if ($pFaculty == 'agriculture') {

$searchQuery = mysql\_query("SELECT \* FROM agriculture.project WHERE (p\_date >= '$pDate') OR (project\_title LIKE '%$pTitle%') OR (department='$pDepartment') OR (doc\_description='%$pDescription%')");

$searchNum = mysql\_num\_rows($searchQuery);

if ($searchNum > 0){

while ($row = mysql\_fetch\_assoc($searchQuery)) {

$doc\_description=strlen($row['doc\_description']);

if ($doc\_description>150) {

$doc\_description=substr($row['doc\_description'], 0,150)."<big> ...</big>";

}

else {

$doc\_description=$row['doc\_description'];

}

?>

<li class="search-box panel">

<div class="search-content">

<div class="row">

<div class="media-thumbnail col-md-3" style=" padding-top: 10px; padding-bottom: 10px; display: block;">

<!-- <i class="fa fa-file-pdf-o fa-3x" style="margin: 30px 30px 20px 50px;"></i> -->

<img src="img/room\_1.jpg" width="100%">

</div>

<div class="media-content col-md-9" style="border-left: 1px ridge #999999;">

<a href=""><h5 class="text text-primary"><i class="fa fa-file-word-o"></i>&nbsp; <?php echo $row['project\_title'];?> </h5></a>

<p><?php echo $doc\_description; ?>...<br>

<a href="" class="text text-danger"><i class="fa fa-file-pdf-o"> </i>&nbsp; Doc-type : PDF File</a>&nbsp;&nbsp;

<span class="label label-default"><i class="fa fa-calendar"> </i>&nbsp; <?php echo $row['p\_date']; ?></span>

<span class="label label-default"><i class="fa fa-building"> </i>&nbsp; Department : <?php echo $row['department']; ?> </span>

</p>

</div>

</div>

</div>

</li>

<?php

}

}

else{

echo '<li class="search-box panel">

<div class="search-content">

<h4 class="text text-info">No Content matches your search criteria...!!</h4>

</div>

</li>';

}

}

?>

<?php

if ($pFaculty == 'education') {

$searchQuery = mysql\_query("SELECT \* FROM education.project WHERE (p\_date >= '$pDate') OR (project\_title LIKE '%$pTitle%') OR (department='$pDepartment') OR (doc\_description='%$pDescription%')");

$searchNum = mysql\_num\_rows($searchQuery);

if ($searchNum > 0){

while ($row = mysql\_fetch\_assoc($searchQuery)) {

$doc\_description=strlen($row['doc\_description']);

if ($doc\_description>150) {

$doc\_description=substr($row['doc\_description'], 0,150)."<big> ...</big>";

}

else {

$doc\_description=$row['doc\_description'];

}

?>

<li class="search-box panel">

<div class="search-content">

<div class="row">

<div class="media-thumbnail col-md-3" style=" padding-top: 10px; padding-bottom: 10px; display: block;">

<!-- <i class="fa fa-file-pdf-o fa-3x" style="margin: 30px 30px 20px 50px;"></i> -->

<img src="img/room\_1.jpg" width="100%">

</div>

<div class="media-content col-md-9" style="border-left: 1px ridge #999999;">

<a href=""><h5 class="text text-primary"><i class="fa fa-file-word-o"></i>&nbsp; <?php echo $row['project\_title'];?> </h5></a>

<p><?php echo $doc\_description; ?>...<br>

<a href="" class="text text-danger"><i class="fa fa-file-pdf-o"> </i>&nbsp; Doc-type : PDF File</a>&nbsp;&nbsp;

<span class="label label-default"><i class="fa fa-calendar"> </i>&nbsp; <?php echo $row['p\_date']; ?></span>

<span class="label label-default"><i class="fa fa-building"> </i>&nbsp; Department : <?php echo $row['department']; ?> </span>

</p>

</div>

</div>

</div>

</li>

<?php

}

}

else{

echo '<li class="search-box panel">

<div class="search-content">

<h4 class="text text-info">No Content matches your search criteria...!!</h4>

</div>

</li>';

}

}

?>

<?php

if ($pFaculty == 'environmental\_science') {

$searchQuery = mysql\_query("SELECT \* FROM environmental\_science.project WHERE (p\_date >= '$pDate') OR (project\_title LIKE '%$pTitle%') OR (department='$pDepartment') OR (doc\_description='%$pDescription%')");

$searchNum = mysql\_num\_rows($searchQuery);

if ($searchNum > 0){

while ($row = mysql\_fetch\_assoc($searchQuery)) {

$doc\_description=strlen($row['doc\_description']);

if ($doc\_description>150) {

$doc\_description=substr($row['doc\_description'], 0,150)."<big> ...</big>";

}

else {

$doc\_description=$row['doc\_description'];

}

?>

<li class="search-box panel">

<div class="search-content">

<div class="row">

<div class="media-thumbnail col-md-3" style=" padding-top: 10px; padding-bottom: 10px; display: block;">

<!-- <i class="fa fa-file-pdf-o fa-3x" style="margin: 30px 30px 20px 50px;"></i> -->

<img src="img/room\_1.jpg" width="100%">

</div>

<div class="media-content col-md-9" style="border-left: 1px ridge #999999;">

<a href=""><h5 class="text text-primary"><i class="fa fa-file-word-o"></i>&nbsp; <?php echo $row['project\_title'];?> </h5></a>

<p><?php echo $doc\_description; ?>...<br>

<a href="" class="text text-danger"><i class="fa fa-file-pdf-o"> </i>&nbsp; Doc-type : PDF File</a>&nbsp;&nbsp;

<span class="label label-default"><i class="fa fa-calendar"> </i>&nbsp; <?php echo $row['p\_date']; ?></span>

<span class="label label-default"><i class="fa fa-building"> </i>&nbsp; Department : <?php echo $row['department']; ?> </span>

</p>

</div>

</div>

</div>

</li>

<?php

}

}

else{

echo '<li class="search-box panel">

<div class="search-content">

<h4 class="text text-info">No Content matches your search criteria...!!</h4>

</div>

</li>';

}

}

?>

<?php

if ($pFaculty == 'management\_science') {

$searchQuery = mysql\_query("SELECT \* FROM management\_science.project WHERE (p\_date >= '$pDate') OR (project\_title LIKE '%$pTitle%') OR (department='$pDepartment') OR (doc\_description='%$pDescription%')");

$searchNum = mysql\_num\_rows($searchQuery);

if ($searchNum > 0){

while ($row = mysql\_fetch\_assoc($searchQuery)) {

$doc\_description=strlen($row['doc\_description']);

if ($doc\_description>150) {

$doc\_description=substr($row['doc\_description'], 0,150)."<big> ...</big>";

}

else {

$doc\_description=$row['doc\_description'];

}

?>

<li class="search-box panel">

<div class="search-content">

<div class="row">

<div class="media-thumbnail col-md-3" style=" padding-top: 10px; padding-bottom: 10px; display: block;">

<!-- <i class="fa fa-file-pdf-o fa-3x" style="margin: 30px 30px 20px 50px;"></i> -->

<img src="img/room\_1.jpg" width="100%">

</div>

<div class="media-content col-md-9" style="border-left: 1px ridge #999999;">

<a href=""><h5 class="text text-primary"><i class="fa fa-file-word-o"></i>&nbsp; <?php echo $row['project\_title'];?> </h5></a>

<p><?php echo $doc\_description; ?>...<br>

<a href="" class="text text-danger"><i class="fa fa-file-pdf-o"> </i>&nbsp; Doc-type : PDF File</a>&nbsp;&nbsp;

<span class="label label-default"><i class="fa fa-calendar"> </i>&nbsp; <?php echo $row['p\_date']; ?></span>

<span class="label label-default"><i class="fa fa-building"> </i>&nbsp; Department : <?php echo $row['department']; ?> </span>

</p>

</div>

</div>

</div>

</li>

<?php

}

}

else{

echo '<li class="search-box panel">

<div class="search-content">

<h4 class="text text-info">No Content matches your search criteria...!!</h4>

</div>

</li>';

}

}

?>

<?php

if ($pFaculty == 'law') {

$searchQuery = mysql\_query("SELECT \* FROM law.project WHERE (p\_date >= '$pDate') OR (project\_title LIKE '%$pTitle%') OR (department='$pDepartment') OR (doc\_description='%$pDescription%')");

$searchNum = mysql\_num\_rows($searchQuery);

if ($searchNum > 0){

while ($row = mysql\_fetch\_assoc($searchQuery)) {

$doc\_description=strlen($row['doc\_description']);

if ($doc\_description>150) {

$doc\_description=substr($row['doc\_description'], 0,150)."<big> ...</big>";

}

else {

$doc\_description=$row['doc\_description'];

}

?>

<li class="search-box panel">

<div class="search-content">

<div class="row">

<div class="media-thumbnail col-md-3" style=" padding-top: 10px; padding-bottom: 10px; display: block;">

<!-- <i class="fa fa-file-pdf-o fa-3x" style="margin: 30px 30px 20px 50px;"></i> -->

<img src="img/room\_1.jpg" width="100%">

</div>

<div class="media-content col-md-9" style="border-left: 1px ridge #999999;">

<a href=""><h5 class="text text-primary"><i class="fa fa-file-word-o"></i>&nbsp; <?php echo $row['project\_title'];?> </h5></a>

<p><?php echo $doc\_description; ?>...<br>

<a href="" class="text text-danger"><i class="fa fa-file-pdf-o"> </i>&nbsp; Doc-type : PDF File</a>&nbsp;&nbsp;

<span class="label label-default"><i class="fa fa-calendar"> </i>&nbsp; <?php echo $row['p\_date']; ?></span>

<span class="label label-default"><i class="fa fa-building"> </i>&nbsp; Department : <?php echo $row['department']; ?> </span>

</p>

</div>

</div>

</div>

</li>

<?php

}

}

else{

echo '<li class="search-box panel">

<div class="search-content">

<h4 class="text text-info">No Content matches your search criteria...!!</h4>

</div>

</li>';

}

}

?>

</ul>

</div>

<div class="col-md-4"></div>

</div>

</div><!-- End book -->