EAGLE SEARCH

Project submitted to University of Madras in partial Fulfilment of the requirement for the Award of the Degree of

(BACHELOR OF COMPUTER SCIENCE)

SUBMITTED BY

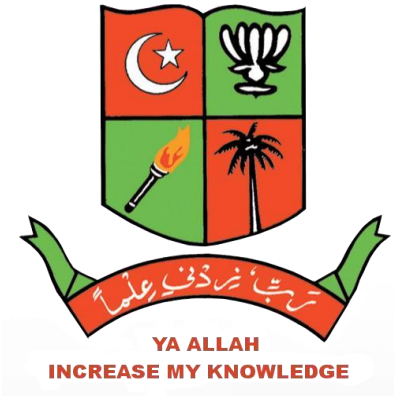
NAME : K BASLUR RAHMAN

REG.NO : 15UCS1427

Under the Guidance of

Prof A. Ashraf Ali M.Sc., M.Phil.,

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE

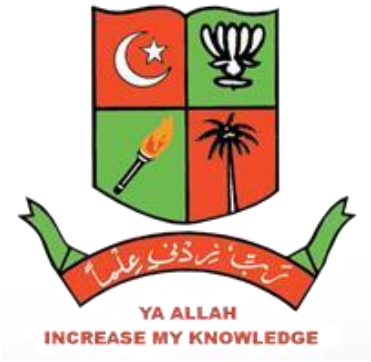
THE NEW COLLEGE – (AUTONOMOUS)

CHENNAI-600014

DEPARTMENT OF COMPUTER SCIENCE

THE NEW COLLEGE – (AUTONOMOUS)

CHENNAI-600014



BONAFIDE CERTIFICATE

This is to certify that the mini project work entitled "EAGLE SEARCH" is the bonafide record work done by Mr. K BASLUR RAHMAN, Reg. No: 15UCS1427, a student of III B.Sc., COMPUTER SCIENCE in partial fulfilment of the requirement for the Degree of Bachelor of Computer Science during the year 2017 – 2018.

Project Guide **Head of the Department**

(A. Ashraf Ali) **(P.Hakkim Divan Mydeen)**

Submitted for the Viva Voice Examinations held on April 2018 at The New College – (Autonomous), Chennai – 600 014.

Internal Examiner **External Examiner**

PLACE : Chennai

DATE :

ACKNOWLEDGEMENT

"In the Name of ALLAH, the most Beneficent, the most Merciful"

Almighty, the Most Merciful, without whom I would not have started this work, I thank first.

I would like to thank our principal **Dr. Major. Zahid Husain M.A., M.Phil., Ph.D.,** for his guidance.

I thank Prof. **P. HAKKIM DIVAN MYDEEN M.Sc., M.Phil.,** Head of the Department, for his guidance and helpful mind to complete this project.

I am extremely grateful to my project guide **A. Ashraf Ali M.Sc., M.Phil., Assistant Professor of Computer Science,** for suggesting me the guidance at the right time and discussions on very useful topic needed in the course of the work.

I wish to record my sincere thanks to **MY PARENTS and FRIENDS** for giving their valuable suggestions and support to complete my project successfully.

(K BASLUR RAHMAN)

TABLE OF CONTENTS

ABOUT THE PROJECT

The main scope of mini-project titled "EAGLE SEARCH" is to enable the user to search web with keywords.

The project contains search engine to make result accurate in search result for keyword and eagle search user friendly design and simplified one

The input keyword is more than one word it will be split into single term to search the keywords and description sometimes it also search keyword in url in the database of eagle search to provide accurate search result for the user input keywords

Form I: User Login

In this form the user will maintain his account information such as username and password. Infinite number of users can create their account and give their specific username and passwords.

Form II: Registration

You need to register a new user whenever you have first visited or site then for future it will be stored in our database permanently and you can book you movie ticket at any time you want with this username and password.

Form III: Search form

In this form the user can give input keyword to find the search result .in the form is act as a basement for the eagle search project .the input keyword is more than one word it will be split into single term to search the keywords and description in the database

INTRODUCTION

INTRODUCTION

This project is aimed at developing a Search Engine application for web search. The Search Engine is an Internet based application that can be accesses throughout the Net and can be accessed by anyone who has a net connection. This application will automate the result for the keyword from the user. This application includes Search result in type of both image and web based result . This is one of the important features of our system. By examining the existing system on the internet, we found that there is no such function available at this moment, so we try to include it in our system. To protect the benefit of the User, we will not spam web result and there result will be accurate to the keyword it will not based on user thought it based on majority of search result queried by previous users

SYSTEM ANALYSIS

SYSTEM ANALYSIS

Existing system was manual. Manual system involves lot of man power and paper work. The manual system had problems in maintenance.

The existing system requires a large storage space for maintaining the records. And they were hard to maintain and keep track of all the record. Manual searching of records from the pile of papers were very hard and it was a time consuming.

EAGLE SEARCH is used to the search result details are accessed and used safely by users within the environment

EAGLE SEARCH involves information processing, which means retrieving information from one file and using it to compare ,update, or display information from another file. The employee could play an essential role in providing accurate data for managing customer care. Information systems are now necessary to help employee to perform their expanding list of daily tasks efficiently.

DEMERITS OF SYSTEM:

- There was chance of collisions due to the inconvenient in knowing block out date / time of a provider as well as previous appointment on the same date / time.
- Records cannot be searched immediately.
- Manual maintenance of data is time consuming process.
- Stationeries and maintenance are highly cost expensive.
- Data may be stole by the unauthorized users.
- Storage of huge data with less cost is impossible.

NEED FOR PROPOSED SYSTEM:-

As it is a mini project its area are used to book up to required tickets. But later it can use for a major project when it is developed.

EAGLE SEARCH ENGINE is to ensure that the result accessed and used safely by users and professionals both within the environments of a web application and beyond. EAGLE SEARCH involves information processing, which means retrieving information from one file and using it to compare, update, or display information from another file.

Information systems are now necessary to help employees to perform their expanding list of daily tasks efficiently. The EAGLE SEARCH collects stores and manages information related to search keywords in the search engine are the most widely used information systems today.

MERITS OF SYSTEM:

- Task becomes easy and relevant when one uses this management system.
- It reduces the time consumption to provide medicine.
- It reduces the number of assistants, because PC's can do multi task at a time.
- This system is easy to operate. Anyone who has the basic knowledge of computers can do this task.
- If there are many user in the web application the web application manage to provide result as quick as possible to make good user experience .

REQUIREMENT SPECIFICATION

Hardware requirements

- PROCESSOR : AMD PHENOM™ 2.90 GHZ
- RAM : 2.00 GB
- MONITOR : STANDARD LCD MONITOR
- HARD DISK : 80GB
- KEYBOARD : STANDARD KEYBOARD
- MOUSE : OPTICAL MOUSE

Software Requirements

- FRONT END : HTML,CSS,JS
- BACK END : PHP,MYSQL
- SERVER : XAMPP SERVER
- OPERATING SYSTEM : WINDOWS 7/8/10

SYSTEM DESIGN

Introduction:

System design is a transition which goes through logical and physical design with emphasis on preparing input/output specification .Specify implementation plan preparing a logical design before implementation. After this, the logical design, the requirements are to be translated in terms of hardware, software, equipment that is program software, database files, text files and the working files were produced, the physical design.

Physical design:

Physical design deals with reviewing the correct physical system and its requirements. Once the requirements are studied then an output specification is prepaid from that of the problem is designed. Its feasibility is studied. The application is designed logically based on the requirement and feasibility is accessed.

Logical Design:

Logical design is nothing but system specification. Logical designs, they specify the user needs at a level of details that virtually determines the information flow into and out of the system.

- Reviews the correct physical system: it's volumes, frequencies, and designs.
- Prepare output screen specifications: it's determines the format, connect, and frequency of results,including terminal specification and locations.
- Specifies the implementation plan
- Prepare a logical design walkthrough of the input screen, and implementation plan.
- Reviews benefits,costs,target results and system constraints.
- The logical design also specifies output screen,input screen.

Input Design:

In accurate input data are most common cause of errors in data proessing. Input design is the process of converting user originated inputs into a computer-based format. The goal of designing is to data entry as easy, logical and free errors as possible.

In the manual system, paper forms are used to collect information. So to have compatibility, the electronic from is designed to be like the original form for pacing requests. The input design incorporates are much automation as possible. The inputs are validated when required. An important part in input design is the checking for incomplete forms. Blank fields do not cause wrong data but incomplete data is equally harmful as the bad data. Before the document is saved it is checked for the existence of any blank fields.

The input forms are designed to be user friendly. Meaningful and labels are given to the input fields. Immediate validation is done for each input form the user that the invalid inputs are not sent to database. The input forms are designed so that they provide the proper links to other forms. Provisions have been given to go back to the main menu as and when required.

Output Design:

Computer outputs are the most important and direct of information to the administrator. They are encountered everywhere in everyone's day -to -day life. The usefulness and clarity of such common place outputs depend a great deal on that care with which an analyst designed them, keeping their major purpose in mind. Every kind of business produces some kind of reports, and many systems produce a lot.

The screens for the display of the action of customer report, ticket report,movies report,billing report and complete report that the output are provided to the administrator can view the form. The design of the output screens include that they are complete with all necessary information requested for. The administrator displays the complete summary of the output depending on the option.

Code Design:

once the output requirements are determined, the system designer can decide what to include in the system and how to structure it and the requirements output can be produced. The output forms or the screen used in the system are required for viewing and reporting the system.

Types of code

The common purpose of code is to classify object for analysis. They are also used for feed the user details and the user can browse the details in home page. Login code type is also available in this project.

SOFTWARE DESCRIPTION

ABOUT THE FRONT-END:

PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. It is originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for *Personal Home Page*, but it now stands for the recursive acronym *PHP: Hypertext Pre processor*

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014 work has gone on to create a formal PHP specification.

During the 2010s there have been increased efforts towards standardisation and code sharing in PHP applications by projects such as PHP-FIG in the form of PSR-initiatives as well as Composer dependency manager and the Packagist repository. PHP hosts a

diverse array of web frameworks requiring framework-specific knowledge, with Laravel recently emerging as a popular option by incorporating ideas made popular from other competing non-PHP web frameworks, like Ruby on Rails.

PHP development began in 1994 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could be used to build simple, dynamic web applications. To accelerate bug reporting and improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group *comp.infosystems.www.authoring.cgi* on June 8, 1995. This release already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler, more limited and less consistent.

Early PHP was not intended to be a new programming language, and grew organically, with Lerdorf noting in retrospect: "I don't know how to stop it, there was never any intent to write a programming language […] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way." A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

The fact that PHP was not originally designed but instead was developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping"while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

Data types

PHP stores integers in a platform-dependent range, either a 64-bit or 32-bit signed integer equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages.[99] Integer variables can be assigned using decimal (positive and negative), octal, hexadecimal, and binary notations.

Functions

PHP defines a large array of functions in the core language and many are also available in various extensions; these functions are well documented in the online PHP documentation.[103] However, the built-in library has a wide variety of naming conventions and associated inconsistencies

PHP Objects

Basic object-oriented programming functionality was added in PHP 3 and improved in PHP 4. This allowed for PHP to gain further abstraction, making creative tasks easier for programmers using the language. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like value types. The drawback of this method was that code had to make heavy use of PHP's "reference" variables if it wanted to modify an object it was passed rather than creating a copy of it. In the new approach, objects are referenced by handle, and not by value.

ABOUT THE back-END:

Mysql 5.0

MySQL (officially pronounced as /maɪ ˌɛskjuːˈɛl/ "My S-Q-L") is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

Mysql is a popular choice of database for use in web applications, and is a central component.

RELATIONS,DOMAINS & ATTRIBUTES

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

RELATIONSHIPS

Table relationships are established using key. The two main keys of prime importance are Primary key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys. Entity Integrity enforces that no Primary Key can have null values. Referential Integrity enforces that no Primary Key can have null values.
Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super key and Candidate Keys.
Relationships have been set between every table inn the database. This ensures both Referential and Entity Relationship Integrity.

Normalization

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

The basic objectives of normalization are to reduce redundancy, which means that information is to be stored only once. Storing information several times leads to waste of storage space and increase in the total size of data stored. Relations are normalized so that when relation in the database is to be altered during the lifetime of the database, we don't lose information or introduce inconsistencies. The type of alterations normally needed for relation is:

- Insertion of new data values to relation. This should be possible without being forced to leave blank fields for some attributes.

- Deletion of a tuple namely, row of a relation. This should be possible without losing vital information technology.

The Normal Forms

The database community has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through five (fifth normal form or 5NF). In practical applications, you'll often see 1NF, 2NF, and 3NF along with the occasional 4NF.

Fifth normal form is very rarely seen and won't be discussed in this article.

Before we begin our discussion of the normal forms, it's important to point out that they are guidelines and guidelines only. Occasionally, it becomes necessary to stray from them to meet practical business requirements. However, when variations take place, it's extremely important to evaluate any possible ramifications they could have on your system and account for possible inconsistencies.

That said, let's explore the normal forms.

First Normal Form (1NF)

First normal form (1NF) sets the very basic rules for an organized database:

- Eliminate duplicative columns from the same table.

- Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).
-

Second Normal Form (2NF)

Second normal form (2NF) further addresses the concept of removing duplicative data:

- Meet all the requirements of the first normal form.
- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
- Create relationships between these new tables and their predecessors through the use of foreign keys.

Third Normal Form (3NF)

Third normal form (3NF) goes one large step further:

- Meet all the requirements of the second normal form.
- Remove columns that are not dependent upon the primary key.

Boyce-Codd Normal Form (BCNF or 3.5NF)

The Boyce-Codd Normal Form, also referred to as the "third and half (3.5) normal form", adds one more requirement:

- Meet all the requirements of the third normal form.
- Every determinant must be a candidate key.

Fourth Normal Form (4NF)

Finally, fourth normal form (4NF) has one additional requirement:

- Meet all the requirements of the third normal form.
- A relation is in 4NF if it has no multi-valued dependencies.

Remember, these normalization guidelines are cumulative. For a database to be in 2NF, it must first fulfill all the criteria of a 1NF database.

SYSTEM TESTING AND IMPLEMENTATION

IMPLEMENTATION:

The crucial phase in the system life cycle is successful completion of the new system design. Implementation simply means converting a new system design in to operation. This involves creating files, training the operation staff and installing the hardware and telecommunication network before the system is up and running. A Crucial factor in the conversion is not disrupting the functioning of the organization. In system implementation user training is essential.

The user manual is prepaid reflexively because it is an item that must accompany every system. The manual is necessary when the user us geographically, remote from the project team or what they cannot attend all training sessions.

TESTING:

Testing is the major quality measure employed during the software development. After the coding phase, computer programs are available that can be executed for testing purpose. Testing is not only has to uncover errors introduced during coding, but also locates errors during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

System testing is an expensive but critical process that can take us much as fifty percent of the budget for the programs development. Consequently, different levels of testing are employed. In fact, a successful test is one that finds as error .The system performance criteria deal with turnaround time, back up, file

protection and human factor. A test for the user acceptance should also be carried out.

TESTING STRATEGY:

This is the phase where bugs in the programs were to be found and corrected. One of the goals during dynamic testing is to produce a test suite, where the salary calculated with the desired outputs such as reports in the case. This is applied ensure that the modification of the program does not have any side effects. This type of testing is called regression testing. Testing generally reeves all the residual bugs and removes the bugs and improves the reliability of the program.

TYPES OF TESTING:

The basic types of testing that can done in this project is

- Unit testing
- Integration testing
- Validation testing
- Output testing
- User acceptance testing

I. UNIT TESTING:

- This is the first level of testing. In this different modules are tested against the specifications produced the design of the modules. Unit testing is done for the verification of the code produced during the coding of single program module is an

isolated environment. Unit testing first focuses on the modules independently of one another to locate errors.

- After coding each dialog is tested and run individually. All necessary coding was removed and ensured that all the modules worked, as the programmer would expect. Logical errors found were corrected. So by working all the modules independently and verifying the outputs of each module in the presence of staff was concluded that the program was functioning as expected.

II. INTEGRATION TESTING:

- Data can be lost access an interface, one module can have as adverse effort on the other sub function when combined may not produce the desired major functions. Integration testing is systematic testing for constructing the program structure, while at the same time conducting tests to uncover errors associated with the interface. The objectives are to take a unit tested as a whole. Here the correction is difficult because vast expenses of the entire program complicate the isolation of causes. Thus in the integration testing step, all the errors are uncovered are corrected for the next testing steps.

III. VALIDATION TESTING:

- This provides the final assurance that the software meets all functional, behavioral and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in manner in which the user expects. Validation refers to processor using software in alive environment in order to find errors. During the course of validating the system, failures may occur and sometimes the

coding has to be changed according to the requirement. Thus the feedback from the validation phase generally produces changes in the software. Once the application was made free all of logical and interface errors, inputting dummy data ensured that the software developed satisfied all the requirements of the user.

IV. OUTPUT TESTING:

- After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered into two ways; one is on screen and another is printed format.
- The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also the output comes out as the specified requirements by the user hence output testing does not result in any correction in the system.

V. USER ACCEPTANCE TESTING:

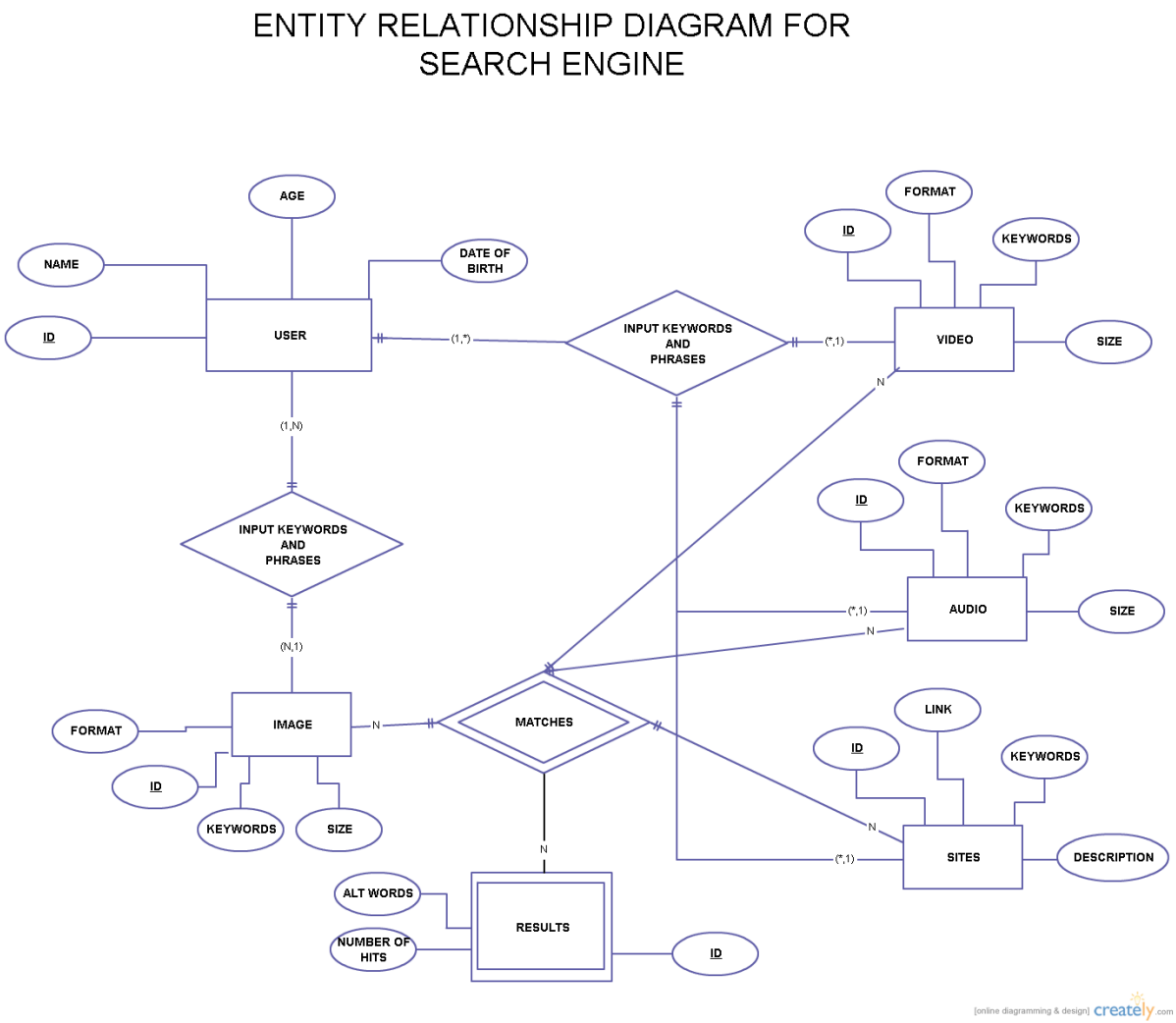
- User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

- Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using the test data. While testing the system

by using test data errors are again uncovered and the corrections are also noted for future use.

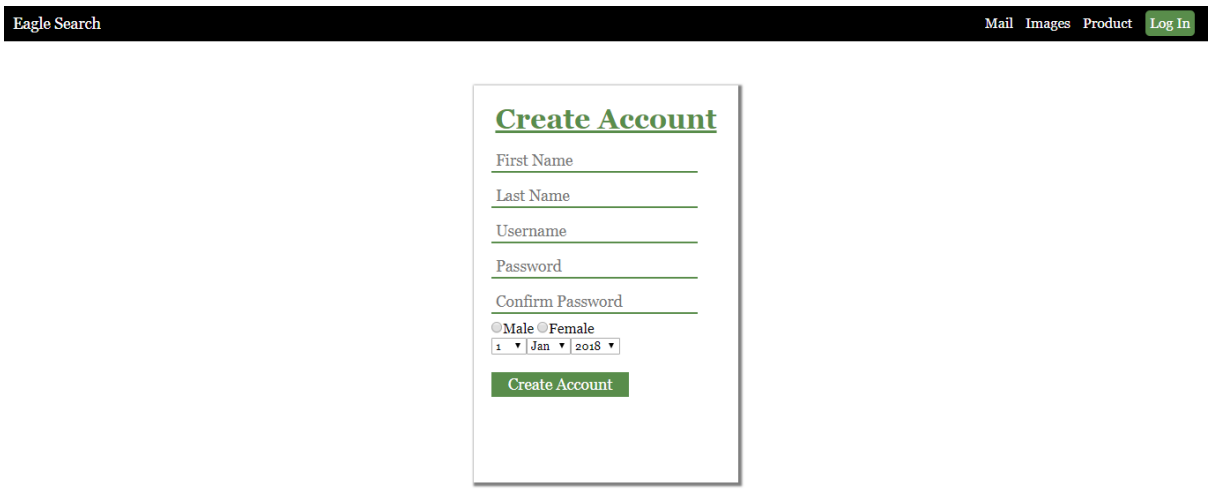
ER DIAGRAM

ER DIAGRAM:

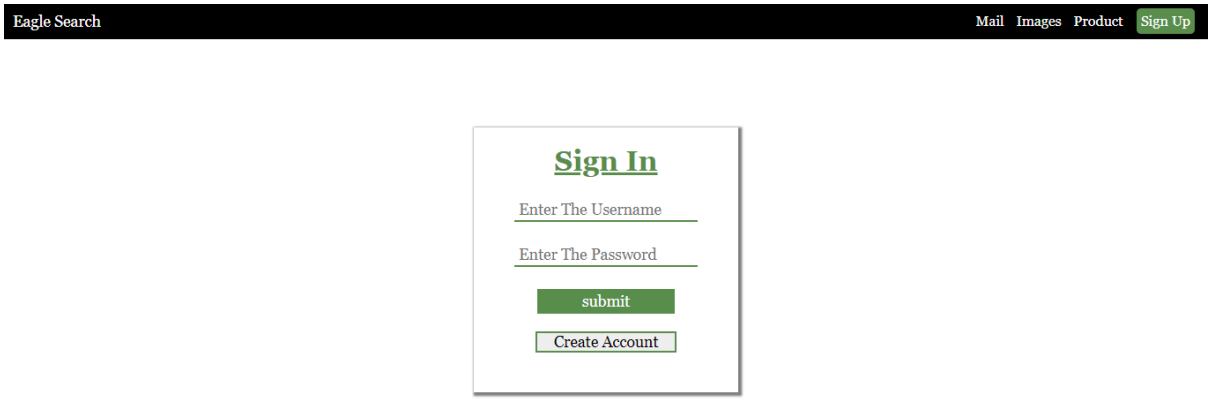

ENTITY RELATIONSHIP DIAGRAM FOR
SEARCH ENGINE
AGE
NAME
DATE OF BIRTH
ID
USER
(1,*)
INPUT KEYWORDS AND PHRASES
(*,1)
FORMAT
ID
KEYWORDS
VIDEO
SIZE
N
(1,N)
INPUT KEYWORDS AND PHRASES
(N,1)
FORMAT
ID
KEYWORDS
AUDIO
SIZE
(*,1)
N
FORMAT
IMAGE
N
MATCHES
ID
KEYWORDS
SIZE
LINK
ID
KEYWORDS
N
SITES
DESCRIPTION
(*,1)
N
ALT WORDS
NUMBER OF HITS
RESULTS
ID
[online diagramming & design] creately.com

FORM LAYOUT

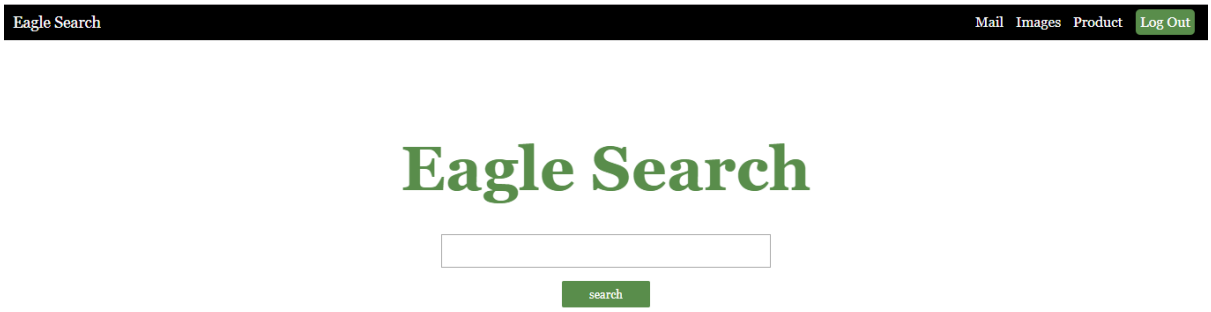
FORM I:



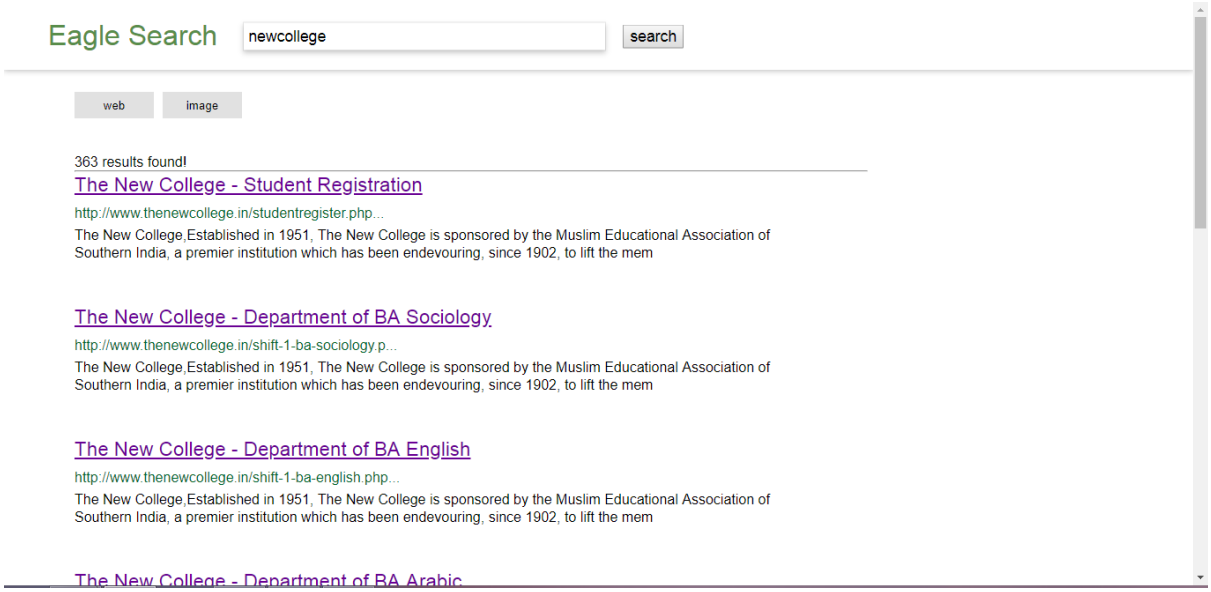
FORM II:



FORM III:


Eagle Search
Mail Images Product Log Out
Eagle Search
search

FORM IV:


Eagle Search
newcollege
search
web
image
363 results found!
The New College - Student Registration
http://www.thenewcollege.in/studentregister.php...
The New College,Established in 1951, The New College is sponsored by the Muslim Educational Association of Southern India, a premier institution which has been endevouring, since 1902, to lift the mem
The New College - Department of BA Sociology
http://www.thenewcollege.in/shift-1-ba-sociology.p...
The New College,Established in 1951, The New College is sponsored by the Muslim Educational Association of Southern India, a premier institution which has been endevouring, since 1902, to lift the mem
The New College - Department of BA English
http://www.thenewcollege.in/shift-1-ba-english.php...
The New College,Established in 1951, The New College is sponsored by the Muslim Educational Association of Southern India, a premier institution which has been endevouring, since 1902, to lift the mem
The New College - Department of BA Arabic

FORM IV:

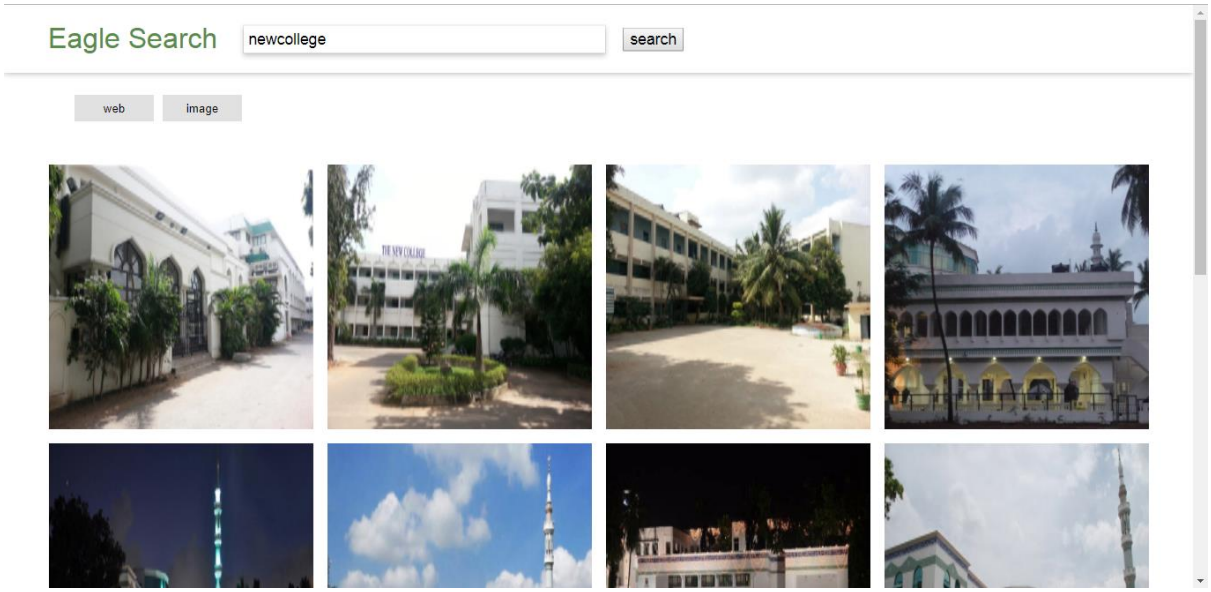
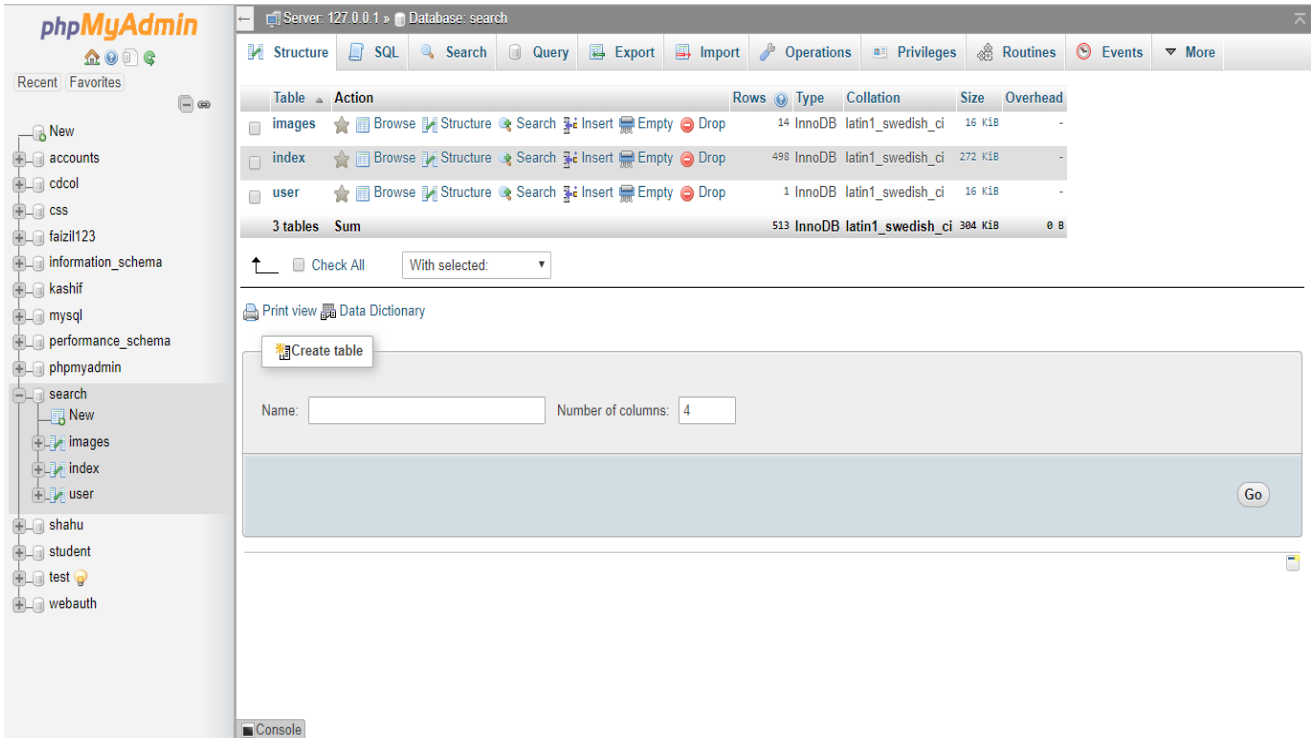
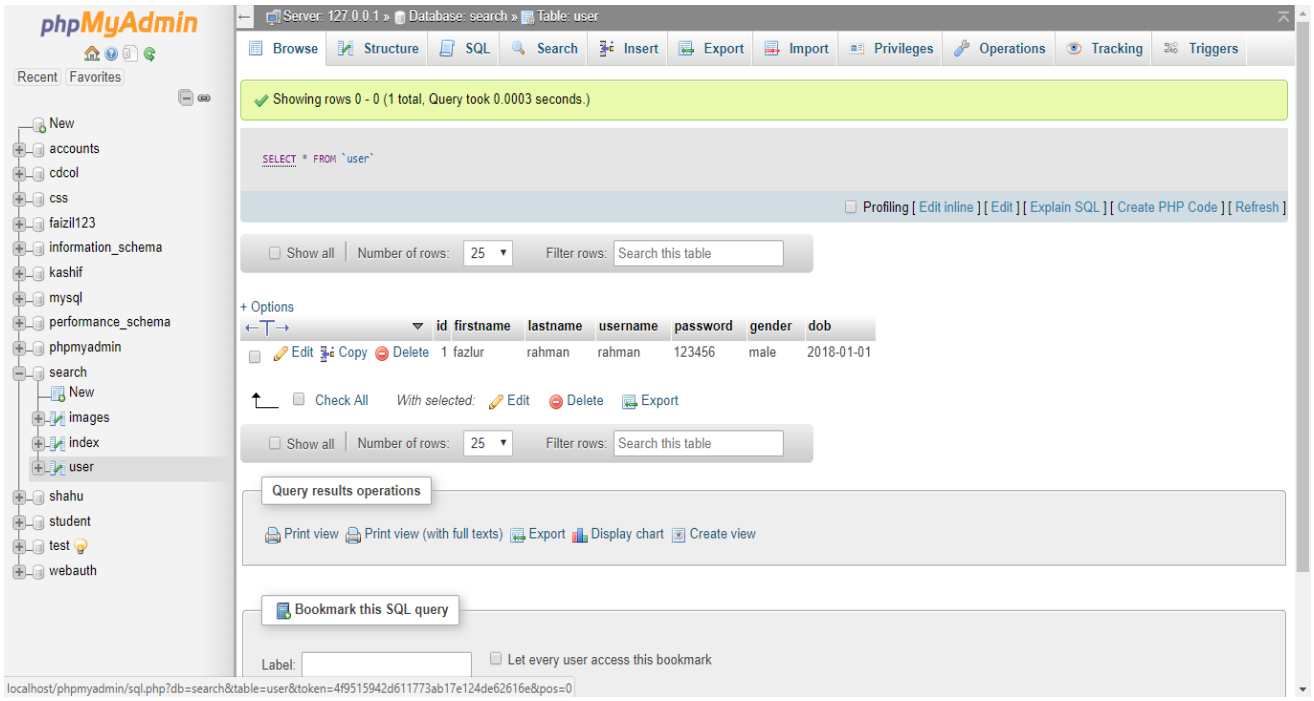


TABLE STRUCTURE

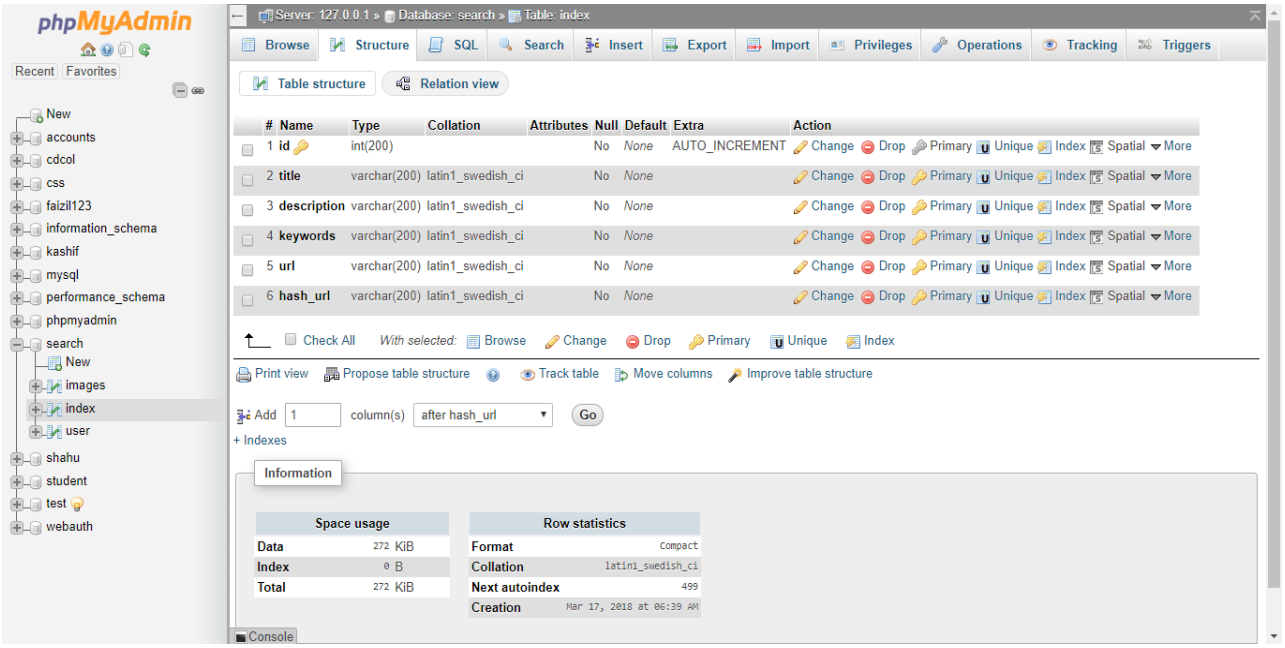
EAGLE SEARCH DATABASE:



USER TABLE



WEBRESULT TABLE STUCTURE:



WEBRESULT CRAWLED RESULT:

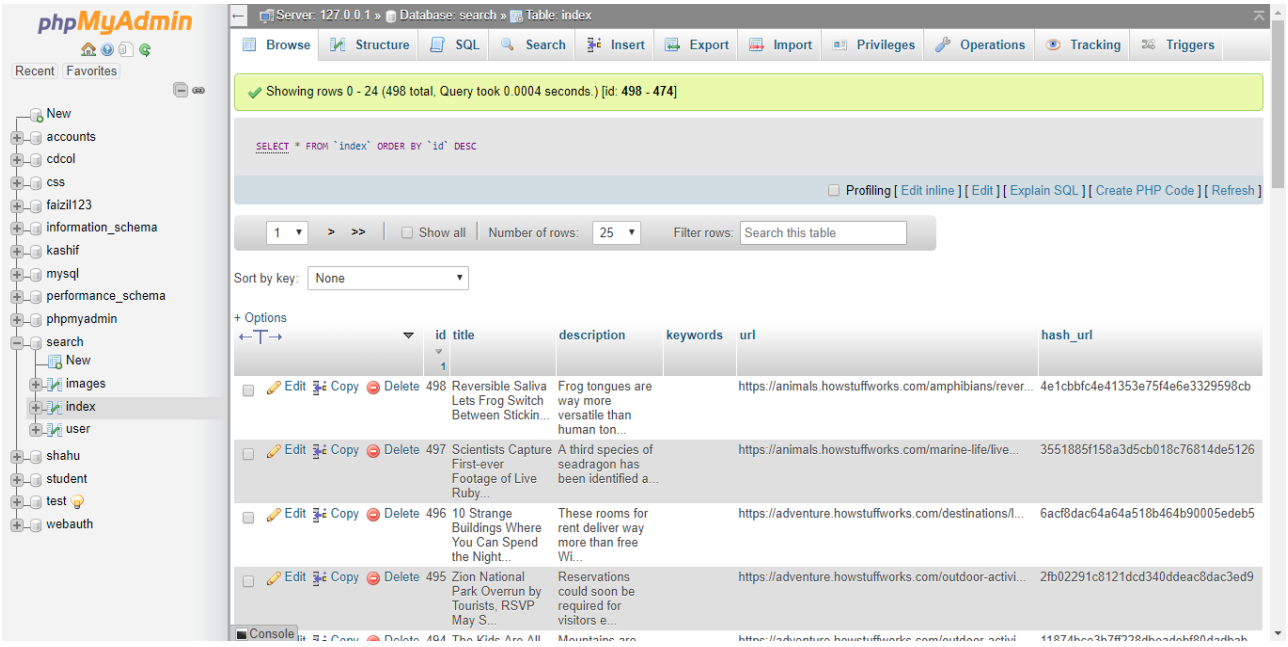


IMAGE RESULT TABLE STUCTURE:

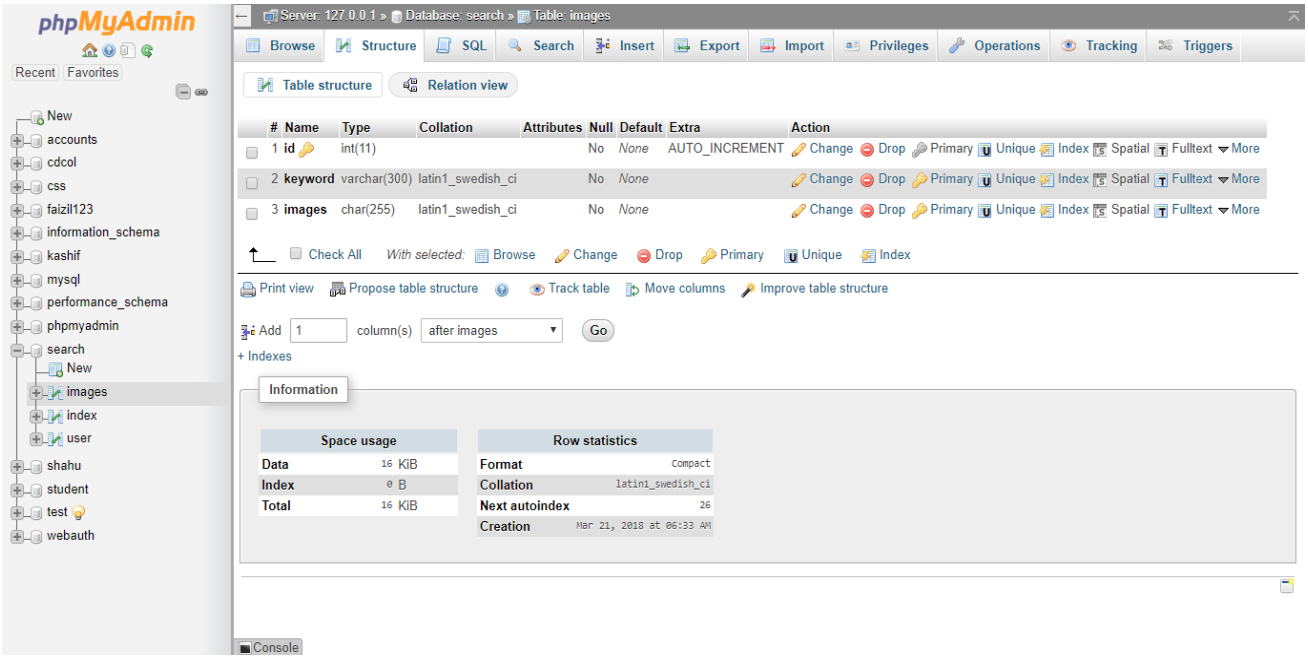
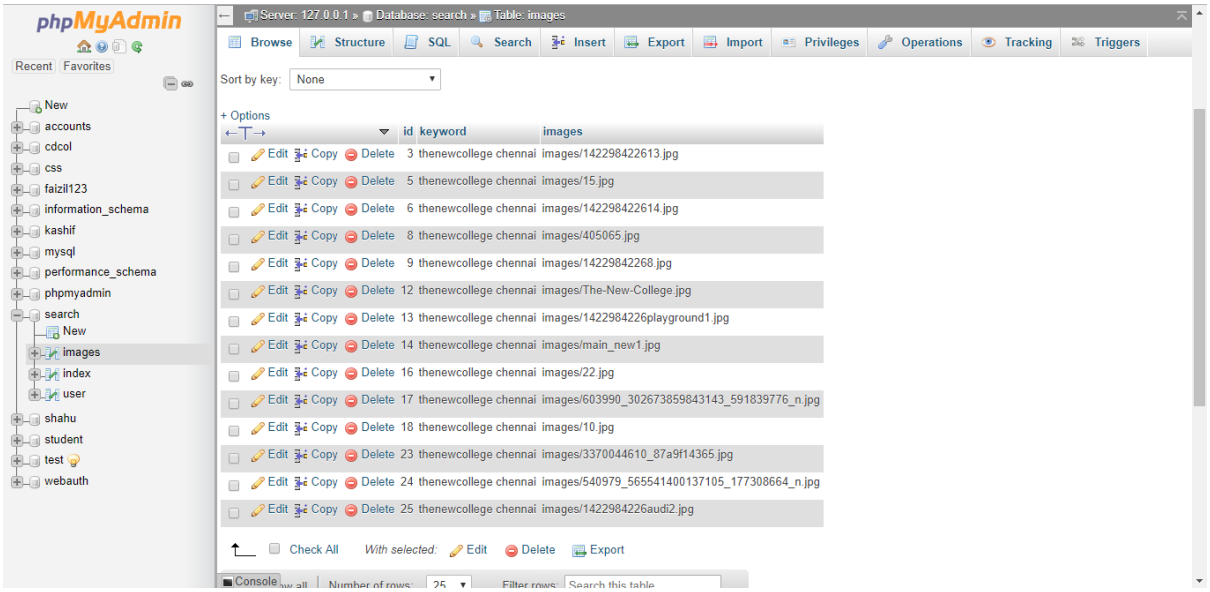


IMAGE RESULT CRAWLED RESULT:



SOURCE CODE

DBCONFIG.PHP

```
<?php
$con=mysqli_connect("localhost","root","") or die("unable to
connect");
mysqli_select_db($con,"search");
?>
```

CRAWLER.PHP

```
 <?php
$start = "http://localhost/se/test.html";
$pdo = new
PDO('mysql:host=127.0.0.1;dbname=search','root','');
$already_crawled = array();
$crawling = array();
function get_details($url) {
	$options = array('http'=>array('method'=>"GET",
'headers'=>"User-Agent: howBot/0.1\n"));
	$context = stream_context_create($options);
	$doc = new DOMDocument();
	@$doc->loadHTML(@file_get_contents($url, false,
$context));
	$title = $doc->getElementsByTagName("title");
	@$title = $title->item(0)->nodeValue;
	$description = "";
	$keywords = "";
	$metas = $doc->getElementsByTagName("meta");
	for ($i = 0; $i < $metas->length; $i++) {
		$meta = $metas->item($i);
```

```
if (strtolower($meta->getAttribute("name")) ==
"description")
                $description = $meta-
>getAttribute("content");
            if (strtolower($meta->getAttribute("name")) ==
"keywords")
                $keywords = $meta->getAttribute("content");
        }
        return '{ "Title": "'.str_replace("\n", "", $title).'",
"Description": "'.str_replace("\n", "", $description).'",
"Keywords": "'.str_replace("\n", "", $keywords).'", "URL":
"'.$url.'"}';
}
function follow_links($url) {
        global $already_crawled;
        global $crawling;
        global $pdo;
        $options = array('http'=>array('method'=>"GET",
'headers'=>"User-Agent: howBot/0.1\n"));
        $context = stream_context_create($options);
        $doc = new DOMDocument();
        @$doc->loadHTML(@file_get_contents($url, false,
$context));
        $linklist = $doc->getElementsByTagName("a");
        foreach ($linklist as $link) {
            $l =  $link->getAttribute("href");
            if (substr($l, 0, 1) == "/" && substr($l, 0, 2) != "//") {
                $l =
parse_url($url)["scheme"]."://".parse_url($url)["host"].$l;
            } else if (substr($l, 0, 2) == "//") {
```

```
        $l = parse_url($url)["scheme"].":".$l;
            } else if (substr($l, 0, 2) == "./") {
                $l =
parse_url($url)["scheme"]."://".parse_url($url)["host"].dirna
me(parse_url($url)["path"]).substr($l, 1);
            } else if (substr($l, 0, 1) == "#") {
                $l =
parse_url($url)["scheme"]."://".parse_url($url)["host"].parse
_url($url)["path"].$l;
            } else if (substr($l, 0, 3) == "../") {
                $l =
parse_url($url)["scheme"]."://".parse_url($url)["host"]."/".$l;
            } else if (substr($l, 0, 11) == "javascript:") {
                continue;
            } else if (substr($l, 0, 5) != "https" && substr($l, 0,
4) != "http") {
                $l =
parse_url($url)["scheme"]."://".parse_url($url)["host"]."/".$l;
            }
            if (!in_array($l, $already_crawled)) {
                    $already_crawled[] = $l;
                    $crawling[] = $l;
                    $details=json_decode(get_details($l));
                    echo $details->URL."\n";
                    $rows=$pdo->query("SELECT * FROM
`index` WHERE hash_url='".md5($details->URL)."'");
                    $rows=$rows->fetchColumn();

                    $params=array(':title'=> $details-
>Title,':description'=>$details-
```

```
>Description,':keywords'=>$details-
>Keywords,':url'=>$details->URL,':hash_url'=>md5($details-
>URL));

                        if($rows>0){
                            if(!is_null($params[':title']) &&
!is_null($params[':description']) && $params[':title'] != ''){
                            $result=$pdo-
>prepare("UPDATE`index` SET
title=:title,description=:description,keywords=:keywords,url=
:url,hash_url=:hash_url WHERE hash_url=:hash_url");
                            $result=$result->execute($params);
                            }
                        }else{
                            if(!is_null($params[':title']) &&
!is_null($params[':description']) && $params[':title'] != ''){
                            $result=$pdo->prepare("INSERT
INTO `index`
VALUES('',:title,:description,:keywords,:url,:hash_url)");
                            $result=$result->execute($params);
                            }
                        }
                }

        }
        array_shift($crawling);
        foreach ($crawling as $site) {
                follow_links($site);
        }
```

```
}
follow_links($start);
```

HEADER.PHP

```
<!DOCTYPE html>
<html>
<head>
	<title></title>
	<style type="text/css">
	*{
		margin:0px;
		font-family:sans-serif;
	}
	.buttons{
		display: flex;
			margin:0 75px;
			box-sizing: border-box;
	}
	.buttons>form>button{
			border:none;
			margin:5px;
			width:90px;
			height:30px;
			background: #e1e1e1;
		}
		.flex-container {
			width:100%;
			display: flex;
			flex-wrap: wrap;
			justify-content: center;
}
```

```
            .flex-container img{
                margin:8px;
            }
            .logo{
                font-size:2em!important;
                color:#5f8d43;
            }
            .si{
                margin-left:30px;
                position: absolute;
                top:50%;
                transform: translateY(-50%);
            }
            .si .search{
                border:none;
                box-shadow:0 3px 8px 0 rgba(0,0,0,0.2), 0 0 0
1px rgba(0,0,0,0.08);
                padding:5px ;
            }
            .pages{
                over
            }
            .pages a{
                text-decoration: none;
                margin-right:10px;
            }

        </style>
</head>
<body>
```

```
<div class="head" style="width:100%;padding:20px 50px;box-sizing: border-box;background:white;box-shadow: 0 0 10px #aaa;margin-bottom:20px;position: relative;">
	<div class="logo" style="display: inline-block;display:inline-block;font-size:1.5em;">Eagle Search</div>
	<div class="si" style="display: inline-block;">
		<form action="webresult.php?q=" method="get" >
		<?php
		echo '<input class="search" type="text" style="width:400px;font-size:1.1em;" name="q" value="'.$_GET['q'].'">';
		echo '<input type="submit" value="search" name="search" style="margin-left:20px;font-size:1.1em;">';
		?>
</div>
	</form>
	</div>
	<div class="buttons" style="">
		<form action="webresult.php">
			<input type="hidden" name="q" value="<?php echo $_GET['q'] ?>">
		<button type="submit">web</button>
		</form>
		<form method="get" action="imgresult.php">
		<?php
		echo "<input type='hidden' name='q' value='".$_GET['q']."'>";
		echo "<button type='submit'>image</button>";
		?>
```

```
    </form>   </div> <br> <br/>
```

HOME.PHP

```
<?php
require'dbconfig/config.php';
session_start();
if(!isset($_SESSION['$username']))
{
   header("Location:login.php");
}
?>
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta charset="utf-8" />
   <style>
      * {
         margin: 0;
         padding: 0;
         font-family: Georgia;
      }

      .body {
         background: #808080;
         height: 100%;
         width: 100%;
      }
      .nav a {
         text-decoration: none;
         color: #ffffff;
```

```
font-size: 1em;
  }

  .nav {
    background-color: #000000;
    overflow: hidden;
    box-sizing: border-box;
    padding: 5px;
  }

  .title {
    color: #ffffff;
    float: left;
    padding: 5px;
    font-size: 1.1em;
  }

    .title:hover {
      color: #5f8d43;
      cursor: pointer;
    }

  .nav-acnt, .nav-menu {
    display: inline-block;
    float: right;
    margin-right: 10px;
  }


    .nav-acnt a, .nav-menu a:hover {
```

```
    background-color: #5f8d43;
   border-radius: 5px;
}

.nav-acnt a, .nav-menu a {
   display: inline-block;
   padding: 5px;
}
.si {
   height:200px;
   width:1000px;
   margin:100px auto;
}
.main-wrapper{
   height: 50px;
   width:500px;
   margin:auto;
   line-height: 90px;
}
.sh{
   text-align: center;
   font-family: helvatica;
   font-size: 2.2em;
   color:#5f8d43;
}
.si .sit input[type="text"],.si .sit input[type="submit"]{
   display: block;

}
.si .sit input[type="text"]{
```

```
        margin:30px auto 0px;
            height:30px;
            width:350px;
            max-width:445px;
            padding:2px 10px;
        }
        .si .sit input[type="submit"]{
            border:none;
            height:30px;
            width:100px;
            background:..

            ;
            color:white;
            border-radius:2px;
            margin:15px auto;
        }
    </style>
    <title></title>
</head>
<body>
    <!--navigation-->
    <div class="nav">
        <p class="title">Eagle Search</p>
        <div class="nav-acnt">

<a href="logout.php">Log Out</a>
        <!-- <?php
            // if(isset[])
        ?> -->
```

```
</div>
      <div class="nav-menu">
        <a href="#mail-page">Mail</a>
        <a href="#image-page">Images</a>
        <a href="#product-page">Product</a>
      </div>
   </div>
   <!--search engine Form-->
   <div class="si">
      <div class="main-wrapper">
      <div class="sh"><h1>Eagle Search</h1></div>
      <div class="sit"><form method="GET"
action="webresult.php">
        <input type="text" name="q">
        <input type="Submit" name="search" value="search">
      </div>
      </form>
      </div>
   </div>
</body>
</html>
```

IMGCRAWLER.PHP

```
<?php
 require 'dbconfig/config.php';

$keyword="thenewcollege chennai";
$url="http://www.bing.com/images/search?q=".str_replace(
" ","+",$keyword)."&qs=n&form=QBILPG&sp=-
```

```
1&pq=".str_replace(" ","+",$keyword)."&sc=8-
9&sk=&cvid=B2C423E3F2D94410AB758636BF7FB7E9";
$output=get($url);
// echo $output;

function get($url){
$curl=curl_init();
curl_setopt($curl,CURLOPT_URL,$url);
curl_setopt($curl,CURLOPT_RETURNTRANSFER,true);
// curl_exec($curl,CURLOPT_SSL_VERIFYPEER,false);

$output=curl_exec($curl);
curl_close($curl);
return $output;
}
preg_match_all('!<a class="thumb" target="_blank"
href="(.*?)"!',$output,$url_matches);
// print_r($url_matches[1]);

$local_path='C:\XAMPP\htdocs\se\images\\';

for($i=0;$i<count($url_matches[1]);$i++){

    preg_match_all("!.*?/!",$url_matches[1][$i],$matches);
    $last_part=end($matches[0]);

    preg_match("!$last_part(.*?.jpg)!",$url_matches[1][$i],
$match);
    $image_name=str_replace("+","-",$match[1]);
    $image_pat='images/'.$image_name;
```

```
$image_url=$url_matches[1][$i];
        $image_data=get($image_url);

        $file=fopen($local_path.$image_name,'w');
        fwrite($file,$image_data);
        fclose($file);
 mysqli_query($con,"INSERT into images
VALUES('','$keyword','$image_pat')");
}
```

IMGRESULT.PHP

```
<?php
require 'dbconfig/config.php';
require 'header.php';
$key = $_GET['q'];
$results=mysqli_query($con,"SELECT * FROM images WHERE
keyword LIKE '%$key%'");
echo '<div class="flex-container">';
foreach ($results as $result) {
        echo "<img src='".$result['images']."' height='300px'
width='300px'>";
}
echo '</div>';
?>
```

LOGIN.PHP

```
<?php
require'dbconfig/config.php';
```

```
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="utf-8" />
   <title></title>
   <style>
      *{
         margin:0;
         padding:0;
         font-family:Georgia;
      }
      .body{
         background:#808080;
         height:100%;
         width:100%;
      }
      .nav a{
         text-decoration:none;
         color:#ffffff;
         font-size:1em;
      }
      .nav {
         background-color: #000000;
         overflow: hidden;
         box-sizing:border-box;
         padding:5px;
      }
      .title{
```

```
  color:#ffffff;
   float:left;
   padding:5px;
   font-size:1.1em;
}
.title:hover{
   color: #5f8d43;
   cursor:pointer;
}
.nav-acnt, .nav-menu {
   display: inline-block;
   float: right;
   margin-right: 10px;
}
.nav-acnt,.nav-menu {
   display: inline-block;
   float: right;
   margin-right: 10px;
}
   .nav-acnt a,.nav-menu a:hover{
      background-color:#5f8d43;
      border-radius:5px;

   }
   .nav-acnt a, .nav-menu a {
      display: inline-block;
      padding: 5px;
   }
.login-ol {
   background: #ffffff;
```

```
 position: relative;
    width: 300px;
    height:300px;
    margin:100px auto;
    padding:20px;
    box-shadow:2px 2px 2px 2px rgba(0,0,0,0.5);
    box-sizing:border-box;
}
    .login-ol input{
        display:block;
    }
h1 {
    text-align: center;
    text-decoration: underline;
    color: #5f8d43;
}
.loginp input[type=text], .loginp input[type=password] {
    margin: 25px auto;
    border: none;
    border-bottom: 2px solid #5f8d43;
    padding: 2px 5px;
    width: 80%;
    font-size: 1.1em;
    box-sizing: border-box;
}
.loginp input[type=submit] {
    margin: 20px auto;
    border: none;
    width:60%;
    color:white;
```

```
            font-size:1.1em;
            padding:4px 4px;
            border: 2px thin #5f8d43;
            background: #5f8d43;
            cursor:pointer;
        }
        .signup {
            border: 2px solid #5f8d43;
            background: #eee;
            display: block;
            text-align: center;
            width: 60%;
            margin:6px auto;
            color:black;
        }
            .signup a {
                text-decoration: none;
                color: white;
                font-size: 1.1em;
                color: black;
            }
            .signup :hover {
                background: #5f8d43;
                color:white;
            }
    </style>
</head>
<body>
    <!--navigation-->
    <div class="nav">
```

```
    <p class="title">Eagle Search</p>
      <div class="nav-acnt">
        <a href="signup.php">Sign Up</a>
      </div>
      <div class="nav-menu">
        <a href="#mail-page">Mail</a>
        <a href="#image-page">Images</a>
        <a href="#product-page">Product</a>
      </div>
    </div>
    <!--Login Form-->
    <div class="login-ol">
      <h1>Sign In</h1>
      <div class="loginp">
        <form method="post" action="login.php">
          <input name="username" type="text"
placeholder="Enter The Username">
          <input name="password" type="password"
placeholder="Enter The Password">
          <input type="submit" name="submit"
value="submit">
          <div class="signup"><a href="signup.php">Create
Account</a></div>
          <?php
            if(isset($_POST['submit']))
            {
              $username=$_POST['username'];
              $password=$_POST['password'];
              $query="select * from user WHERE
username='$username' AND password='$password'";
```

```
                $query_run=mysqli_query($con,$query);
                  if(mysqli_num_rows($query_run)>0)
                     {
                        $_SESSION['$username']=$username;
                        header('location:home.php');
                     }
                     else
                     {
                         echo '<script
type="text/javascript">alert("Invalid username or
password");</script>';
                     }
                }
            ?>
          </form>
       </div>
   <!--</div><div class="">-->
   </div>
</body>
</html>
```

LOGOUT.PHP

```
<?php
session_start();

if(session_destroy()){
       header("location:login.php");
}
```

```
else{
        header("location:home.php");
}
```

SIGNUP.PHP

```
<?php
 require'dbconfig/config.php';
?>
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta charset="utf-8" />
   <style>
      * {
         margin: 0;
         padding: 0;
         font-family: Georgia;
      }

      .body {
         background: #808080;
         height: 100%;
         width: 100%;

      }

      .nav a {
         text-decoration: none;
```

```
  color: #ffffff;
   font-size: 1em;
}

.nav {
   background-color: #000000;
   overflow: hidden;
   box-sizing: border-box;
   padding: 5px;
}

.title {
   color: #ffffff;
   float: left;
   padding: 5px;
   font-size: 1.1em;
}

   .title:hover {
      color: #5f8d43;
      cursor: pointer;
   }

.nav-acnt, .nav-menu {
   display: inline-block;
   float: right;
   margin-right: 10px;
}

.nav-acnt, .nav-menu {
```

```
    display: inline-block;
    float: right;
    margin-right: 10px;
}

    .nav-acnt a, .nav-menu a:hover {
        background-color: #5f8d43;
        border-radius: 5px;
    }

    .nav-acnt a, .nav-menu a {
        display: inline-block;
        padding: 5px;
    }
.signup-ol {
    background: #ffffff;
    position: relative;
    width: 300px;
    height: 450px;
    margin: 50px auto;
    padding: 20px;
    box-shadow: 2px 2px 2px 2px rgba(0,0,0,0.5);
    box-sizing: border-box;
}
h1{
    text-align:center;
    color:#5f8d43;
    text-decoration:underline;
    margin-bottom:10px;
}
```

```
.
 signupp{
   width:100%;
   margin:auto;
   box-sizing:border-box;
 }
  .signup-ol .signupp input[type=text], .signupp
input[type=password]  {
      margin: 7px auto;
      border: none;
      border-bottom: 2px solid #5f8d43;
      padding: 2px 5px;
      width: 90%;
      margin-right:10px;
      font-size: 1.1em;
      box-sizing: border-box;
   }
   .signupp .bday{
      width:80%;
   }

  .signupp input[type=date] {
      margin: 5px auto;
      border: none;
      border-bottom: 2px solid #5f8d43;
      padding: 2px 5px;
      font-size: 1.1em;
      box-sizing: border-box;
   }
 .inna{
   color:#5f8d43;
```

```
        font-size:1.1em;
      }
      .signupp input[type=submit]{
         margin: 20px auto;
         border: none;
         width: 60%;
         color: white;
         font-size: 1.1em;
         padding: 4px 4px;
         border: 2px thin #5f8d43;
         background: #5f8d43;
         cursor: pointer;
      }
   </style>
   <title></title>
</head>
<body>
   <!--navigation-->
   <div class="nav">
      <p class="title">Eagle Search</p>

  <div class="nav-acnt">

   <a href="login.php">Log In</a>
      </div>
      <div class="nav-menu">
         <a href="#mail-page">Mail</a>
         <a href="#image-page">Images</a>
         <a href="#product-page">Product</a>
      </div>
```

```
</div>
  <!--Login Form-->
  <div class="signup-ol">
    <h1>Create Account</h1>
    <div class="signupp">
      <form method="post" action="signup.php">
        <!--<div class="inna">Enter The First Name</div>-->
        <input type="text" name="firstname"
placeholder="First Name">
        <!--<div class="inna">Enter The last Name</div>-->
        <input type="text" name="lastname"
placeholder="Last Name">
        <!--<div class="inna">Pick a Username</div>-->
        <input type="text" name="username"
placeholder="Username">
        <!--<div class="inna">Enter The Password</div>-->
        <input type="password" name="password"
placeholder="Password">
        <!--<div class="inna">confirm The Password</div>--
>


   <input type="password" name="cpassword"
placeholder="Confirm Password">

  <!--<div class="inna">Select Gender</div>-->
        <div>

  <input type="radio" value="male"
name="gender"><span>Male</span>
```

```
   <input type="radio" value="female"
name="gender"><span>Female</span>
            </div>
            <!--<div class="inna">Birth Day</div>-->
            <div class="bday">
               <select name='day'>
               <?php

                  foreach(range(1,31,1) as $day){
                     echo "<option value='$day'>$day</option>";
                  }
                  echo "</select>";

                  $months =
['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','D
ec'];
                  echo "<select name='month'>";
                  foreach ($months as $month) {
                     echo "<option
value='$month'>$month</option>";
                  }
                  echo "</select>";

      echo "<select name='year'>";
                   $cyear=date('Y');
                  foreach (range($cyear,1920,1) as $year) {
                     echo "<option
value='$year'>$year</option>";
                  }
                  echo "</select>";
```

```
            ?>
          </div>
          <input name="sbt_btn" type="submit"
value="Create Account" />
          <?php
           if(isset($_POST['sbt_btn'])){
             $firstname=$_POST['firstname'];
             $lastname=$_POST['lastname'];
             $username=$_POST['username'];
             $password=$_POST['password'];
             $cpassword=$_POST['cpassword'];
             $gender=$_POST['gender'];
             $day=$_POST['day'];
             $month=$_POST['month'];
             $year=$_POST['year'];
             $dob="{$day}-{$month}-{$year}";
             $dob=date('Y-m-d',strtotime($dob));
             if($password==$cpassword)
               {
               $query="select * from user WHERE username
='$username'";
               $query_run=mysqli_query($con,$query);

    if(mysqli_num_rows($query_run)>0)
                        {
                           echo '<script
type="text/javascript">alert("user already exist");</script>';
                        }
                        else
                        {
```

```
    $query="INSERT INTO `user`(`firstname`, `lastname`,
`username`, `password`, `gender`, `dob`) VALUES
('$firstname','$lastname','$username','$password','$gender','
$dob')";
                        $query_run=mysqli_query($con,$query);
                        if($query_run)
                         {
                         echo '<script
type="text/javascript">alert("account created");</script>';
                         }
                         else
                         {
                        echo '<script
type="text/javascript">alert("ERORR!");</script>';
                         }
                         }
                }
                else
                {



echo '<script type="text/javascript">alert("Confirm Password
and Password are not Matched");</script>';
                }
            }
            ?>
          </form>
       </div>
```

```
 <!--</div><div class="">-->
   </div>
</body>
</html>
```

WEBRESULT.PHP

```
<?php
$pdo =new
PDO('mysql:host=127.0.0.1;dbname=search','root','');
$search=$_GET['q'];
@$sear=$_GET['search'];
$urlSearch=urlencode($search);


$searche=explode(" ",$search);
$x=0;
$construct="";
$params=array();
foreach ($searche as $term){
	$x++;
	if($x==1){
		$construct .= "title LIKE CONCAT('%',:search$x,'%')
OR description LIKE CONCAT('%',:search$x,'%') OR keywords
LIKE CONCAT('%',:search$x,'%')";
	}else{
		$construct .= "AND title LIKE
CONCAT('%',:search$x,'%') OR description LIKE
```

```
CONCAT('%',:search$x,'%') OR keywords LIKE
CONCAT('%',:search$x,'%')";

}
	$params[":search$x"]=$term;
}
require 'header.php';
echo "<div style='width:900px;margin-left:80px;box-
sizing:border-box;'>";
$resu=$pdo->prepare("SELECT * FROM `index` WHERE
$construct");
$resu->execute($params);
if($resu->rowCount()==0){
echo "0 results found! <hr/>";
}else{
echo $resu->rowCount()." results found! <hr/>";
}
$results=$pdo->prepare("SELECT * FROM `index` WHERE
$construct");
$results->execute($params);
$result_per_page=10;
$num_rows=$resu->rowCount();
$num_of_pages=ceil($num_rows/$result_per_page);
if(!isset($_GET['page'])){
	$page=1;
}else{
	$page=$_GET['page'];
}
$this_page_first_result=($page-1)*$result_per_page;
```

```
$results=$pdo->prepare("SELECT * FROM `index` WHERE
$construct LIMIT $this_page_first_result,$result_per_page");

$results->execute($params);

foreach($results->fetchAll() as $result) {
	echo "<div style='width:800px;box-sizing:border-
box;line-height:30px'>";
	echo "<a href=".$result["url"]." style='font-
size:1.4em;color:#609;'>".substr($result["title"],0,60)."</a>";
	echo '<p
style="color:#006621">'.substr($result["url"],0,50)."...".'</p>'
;
	if($result["description"]==""){
		echo "<p>"."No description
available."."</p></br>";
	}else{
		echo "<p style='line-
height:20px'>".$result["description"]."</p></br>";
	}
	echo "</div>";
	echo "</br>";
}
echo "<div class='pages'>";
for($page=1;$page<=$num_of_pages;$page++){
echo '<a
href="webresult.php?q='.$urlSearch.'&search=search&page=
'.$page.'">'.$page.'</a>'."  ";}
echo "</div>";
echo "</div>";
// print_r($results->fetchAll());
```

CONCLUSION

CONCLUSION:

I started this project “EAGLE SEARCH” and completed to my satisfaction with my limited knowledge of PHP. Through this sample project it is possible to do the file and folder option very easily. The encryption and decryption concepts are also made through project manually. Although this project performs some operations, in future we will add some more features in this software to develop a new version.

BIBLIOGRAPHY

BIBLIOGRAPHY:

BOOK NAME - "*WROX BEGINNING PHP*"

AUTHOR - TIMOTHY BORONCZYK

BOOK NAME - "*STY PHP MYSQL AND APACHE*"

AUTHOR - JULICE C.MELONI

BOOK NAME - "*MURACH'S PHP & MYSQL*"

AUTHOR - JOEL MURACH