**Small Search Engine using PHP**

**A Project Submitted to**



Project Based Learning

in

Data Structure and Algorithm

By:

**Atul Anand(BETn1cs15028),**

**Bharti Parmar(BETN1cs15032),**

**Kamlesh Sharma(BETN1CS15048)**

Under the supervision of

Mrs. Ritu Chauhan

Asst. Prof. CS/IT Dept.

ITM UNIVERSITY

Table of Contents:

1. Introduction or a tale of how easy a search engine could work...................................................................................................................2

2. Objective....................................................................................................2

3. Introduction.............................................................................................2

4. Hardware Requirements.....................................................................2

5. Software Requirements........................................................................3

6. Need of Undertaking Current Project.............................................3

7. Our Task

7.1. Preparation..........................................................................................3

7.2. Database Stored in MYSQL...............................................................5

8. Source Code

8.1. Index.php.............................................................................................8

8.2. Search.php...........................................................................................9

9. Keywords Stored in Database..........................................................11

10. More Screenshots...............................................................................12

11. References.............................................................................................15

Small Search Engine using PHP

Introduction or a tale of how easy a search engine could work

Five years ago your site may have had five or six pages. However, over the time it has grown and one day you wake up with 2 GB of content, more than a thousand links and no way to sort them by relevance. You need a way for you and your visitors to get information about your services and products faster. A search engine could help you and your site visitors find that precious piece of information on your extensively growing site.

Objective

Our main Objective is to create a working model/prototype of a search engine on a small scale. It takes keywords from users and matches it with the pre-defined keywords stored in the database,then shows the search result.

Introduction

In this project, we will be designing and implementing a mini search engine. We are probably familiar with Google, Altavista or Yahoo, which are some of the most popular search engines that you can find on the Web. The task performed by a search engine is, as the name says, to search through a collection of documents. Given a set of keywords and query, the search engine will locate all documents that contain the keywords in the query. The problem may be therefore reduced to a search problem, which can be efficiently solved with the data structures we have studied in this class. It mostly implements the “Tree” & “Array/String” Data structure.

Hardware Requirements

🡺A pc with:

* 512 MB RAM or above
* Proper Sreen size
* 32 GB HDD or above
* Any INTEL processor

Software Requirements

* XAMPP/WAMP server(LocalHost Server)
* PhpMyadmin(installed in XAMPP/WAMP)
* WEB BROWSER(CHROME recommended)
* Any text Editor(NOTEPAD++,Brackets)
* MYSQL database

Need of Undertaking Current Project

simple yet very powerful and fast PHP website search engine. MYSEARCH-ENGINE is built to index your site so it can be searched later within seconds. Features: stopwords, logging, MySQL boolean search, localized, CSS formatting.

Small MySQL Efficient Lightweight PHP Search Engine This is a PHP and MySQL search engine, coded with High Efficiency coding, designed to be a lightweight OpenSource search engine project.

**Our Task**

Preparation

You should have Apache, MySQL and PHP installed and running of course (you can use [XAMPP](http://www.apachefriends.org/en/xampp.html) for different platforms  or  [WAMP](http://www.wampserver.com/en/) for windows, [MAMP](http://www.mamp.info/en/index.html) for mac) or a web server/hosting that supports PHP and MySQL databases.

Let's create database, table and fill it with some entries we can use for search:

* Go to phpMyAdmin, if you have server on your computer you can access it at <http://localhost/phpmyadmin/>
* Create database, I called mine tutorial\_search
* Create table I used 3 fields, I called mine articles.
* Configuration for 1st field. Name: id, type: INT, check AUTO\_INCREMENT, index: primary

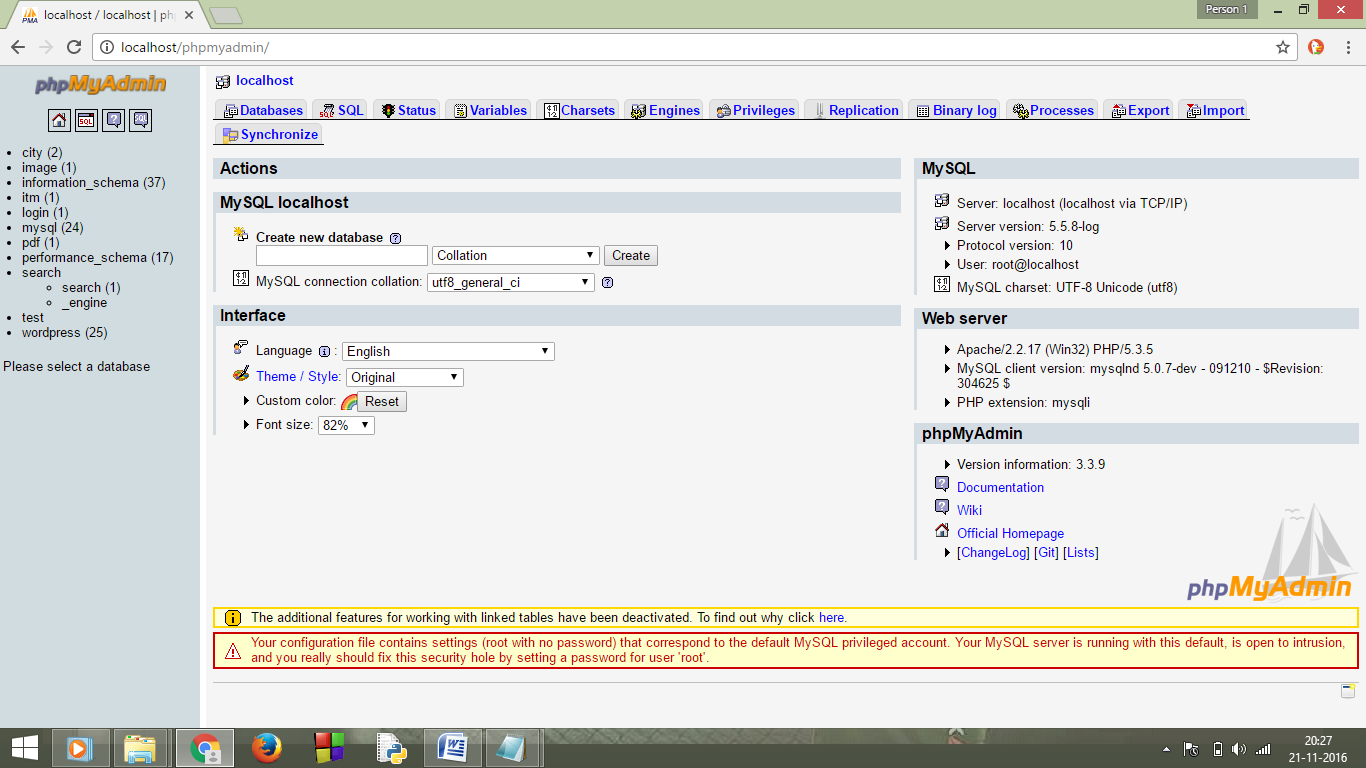
INT means it's integer  
AUTO\_INCREMENT means that new entries will have other(higher) number than previous  
Index: primary means that it's unique key used to identify row

* 2nd field: Name: title, type: VARCHAR, length: 225

VARCHAR means it string of text, maximum 225 characters(it is required to specify maximum length), use it for titles, names, addresses  
length means it can't be longer than 225 characters(you can set it to lower number if you want)

* 3rd field: Name: text, type: TEXT

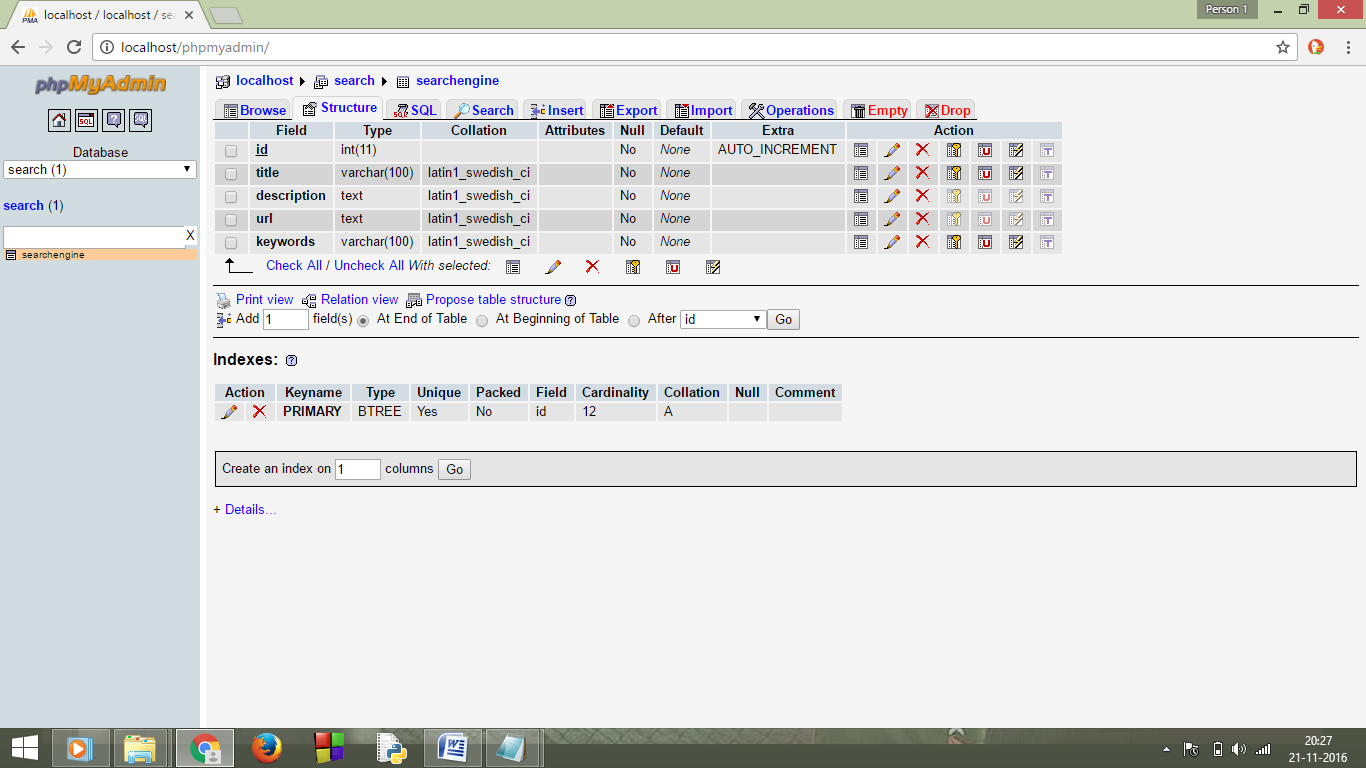
TEXT means it's long string, it's not necessary to specify length, use it for long text.



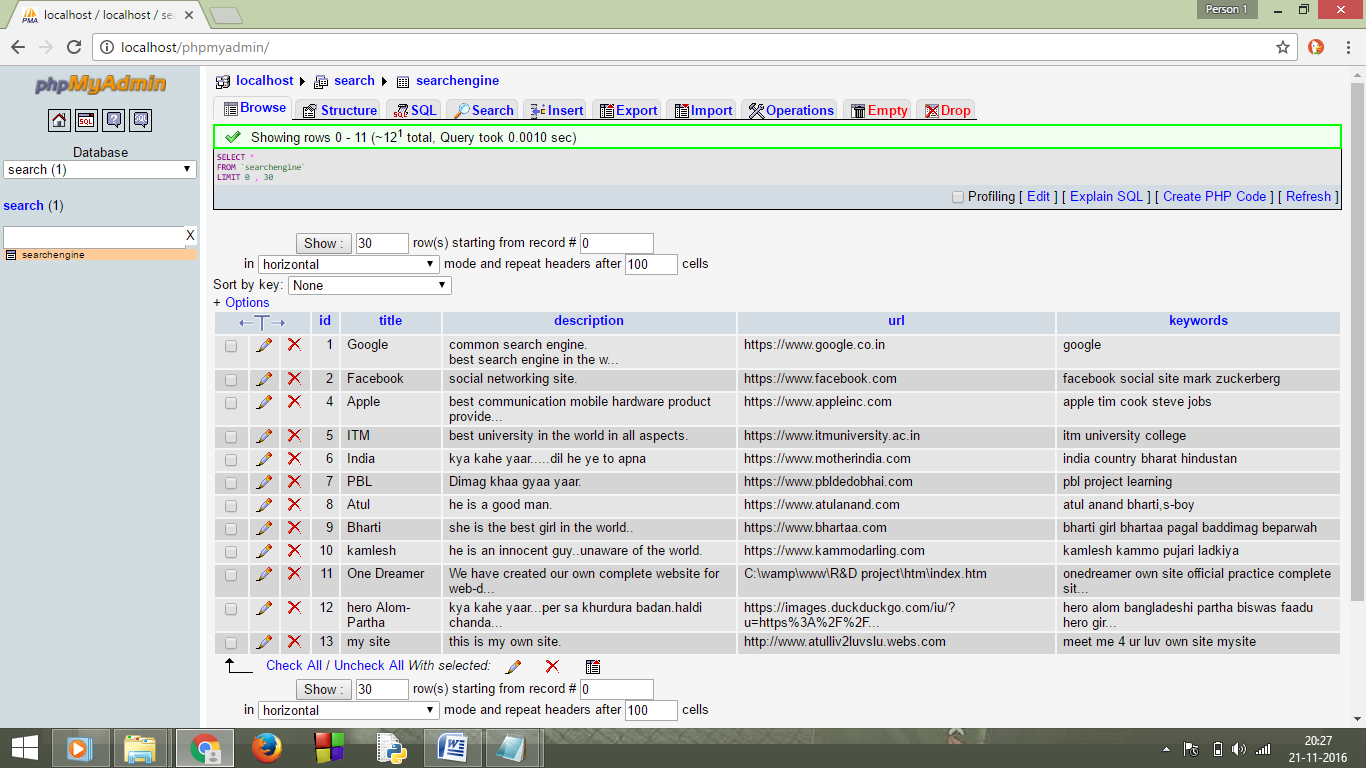
* Fill the table with some random articles(you can find them on news websites, for example: CNN, BBC, etc.). Click insert on the top menu and copy text to a specific fields. Leave "id" field empty. Insert at least three.

It should look something like this:

* Create a folder in your server directory and two files: index.php and search.php (actually we can do all this just with one file, but let's use two, it will be easier)
* Fill them with default html markup, doctype, head, etc.

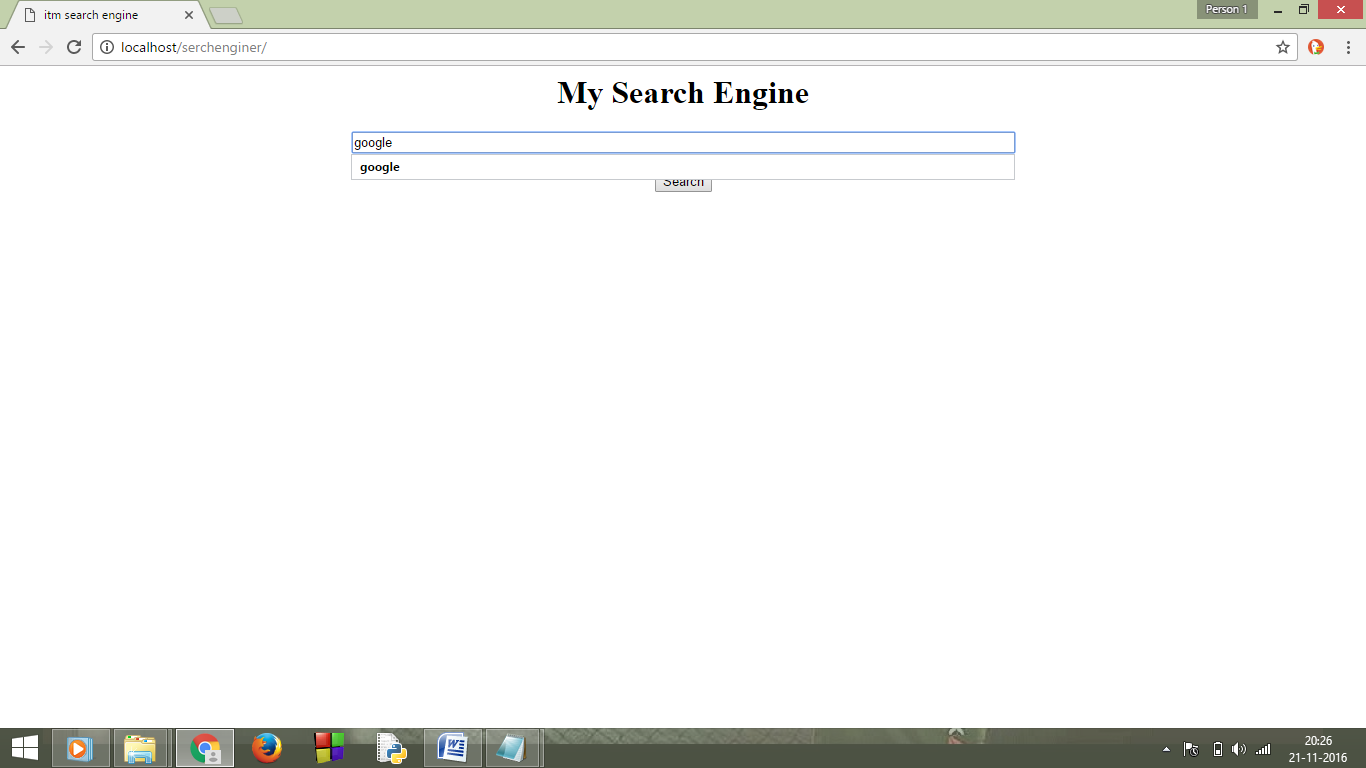


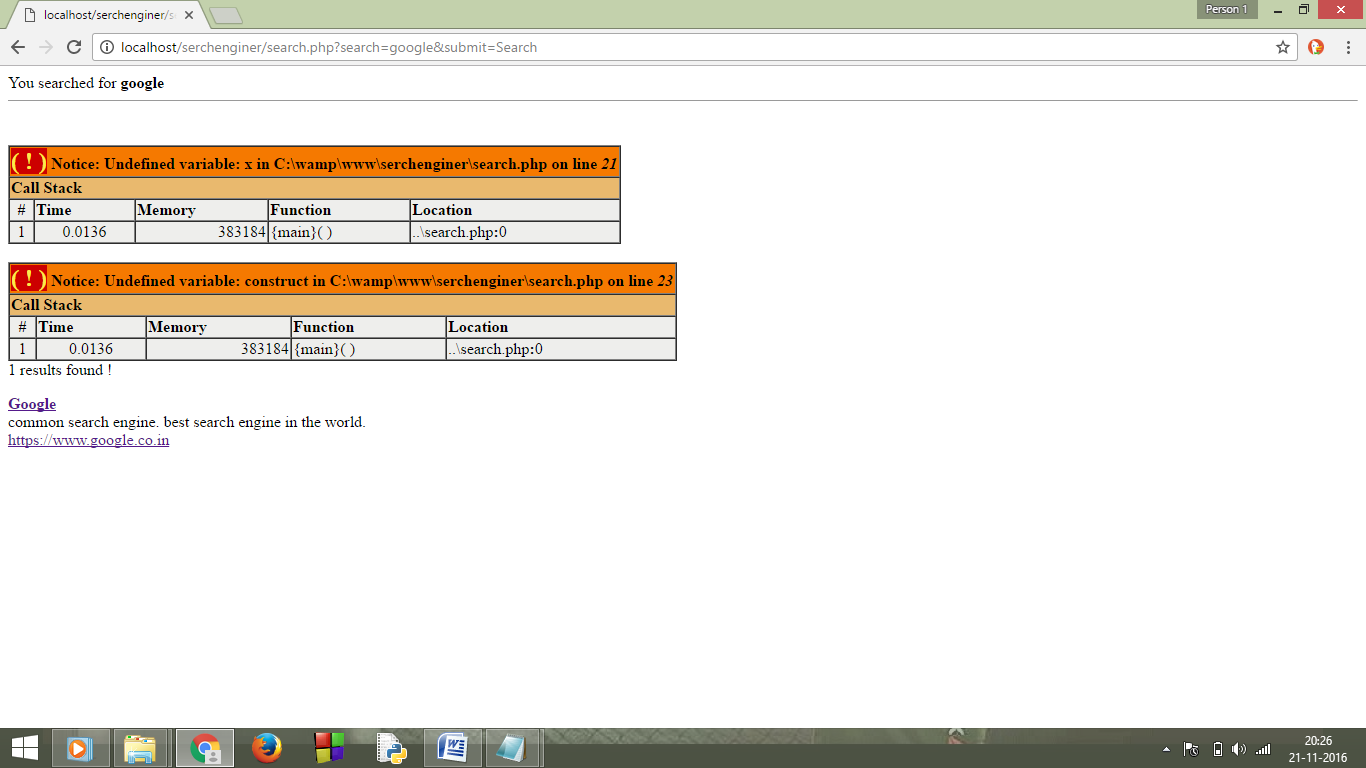
Database stored in MYSQL:



* Create a form with search field and submit button in index.php, you can use GET or POST method, set action to search.php. I used "query" as name for text field

GET - means your information will be stored in url ([http://localhost/tutorial\_search/search.php?query=**yourQuery**](http://localhost/tutorial_search/search.php?query=yourQuery))

  
POST - means your information won't be displayed it is used for passwords, private information, much more secure than GET



[?](https://owlcation.com/stem/Simple-search-PHP-MySQL)

|  |  |
| --- | --- |
| 1  2  3  4 | <form action="search.php" method="GET">      <input type="text" name="query" />      <input type="submit" value="Search" />  </form> |

Ok, let's get started with php.

* Open search.php
* Start php (<?php ?>)
* Connect to a database(read comments in following code)

[?](https://owlcation.com/stem/Simple-search-PHP-MySQL)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <?php      mysql\_connect("localhost", "root", "") or die("Error connecting to database: ".mysql\_error());      /\*          localhost - it's location of the mysql server, usually localhost          root - your username          third is your password            if connection fails it will stop loading the page and display an error      \*/        mysql\_select\_db("tutorial\_search") or die(mysql\_error());      /\* tutorial\_search is the name of database we've created \*/        ?> |

You can go and check if there is no errors.

* Now go to the <body></body> part of the page
* I'm using GET method, if you want to use POST, just use $\_POST instead of $\_GET
* Also some functions to make it more secure. Read comments in the code
* Send query to database
* Check if there is any results
* If there is any, post them using while loop

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | <?php      $query = $\_GET['query'];      // gets value sent over search form        $min\_length = 3;      // you can set minimum length of the query if you want        if(strlen($query) >= $min\_length){ // if query length is more or equal minimum length then            $query = htmlspecialchars($query);          // changes characters used in html to their equivalents, for example: < to &gt;            $query = mysql\_real\_escape\_string($query);          // makes sure nobody uses SQL injection            $raw\_results = mysql\_query("SELECT \* FROM articles              WHERE (`title` LIKE '%".$query."%') OR (`text` LIKE '%".$query."%')") or die(mysql\_error());            // \* means that it selects all fields, you can also write: `id`, `title`, `text`          // articles is the name of our table            // '%$query%' is what we're looking for, % means anything, for example if $query is Hello     // it will match "hello", "Hello man", "gogohello", if you want exact match use `title`='$query          // or if you want to match just full word so "gogohello" is out use '% $query %' ...OR ... '$query %' ... OR ... '% $query'            if(mysql\_num\_rows($raw\_results) > 0){ // if one or more rows are returned do following                while($results = mysql\_fetch\_array($raw\_results)){              // $results = mysql\_fetch\_array($raw\_results) puts data from database into array, while it's valid it does the loop                    echo "<p><h3>".$results['title']."</h3>".$results['text']."</p>";                  // posts results gotten from database(title and text) you can also show id ($results['id'])              }            }          else{ // if there is no matching rows do following              echo "No results";          }        }      else{ // if query length is less than minimum          echo "Minimum length is ".$min\_length;      }  ?>  **Done!** |

**Source Code**

Index.php

<html>

<head>

<title>itm search engine</title>

</head>

<body>

<form action='search.php' method='GET'>

<center>

<h1>My Search Engine</h1>

<input type='text' size='90' name='search'></br></br>

<input type='submit' name='submit' value='Search' ></br></br></br>

</center>

</form>

</body>

</html>

**Search.php**

<?php

$button = $\_GET ['submit'];

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

if(!$button)

echo "you didn't submit a keyword";

else

{

if(strlen($search)<=1)

echo "Search term too short";

else{

echo "You searched for <b>$search</b> <hr size='1'></br>";

mysql\_connect("localhost","root","");

mysql\_select\_db("search");

$search\_exploded = explode (" ", $search);

foreach($search\_exploded as $search\_each)

{

$x++;

if($x==1)

$construct .="keywords LIKE '%$search\_each%'";

else

$construct .="AND keywords LIKE '%$search\_each%'";

}

$construct ="SELECT \* FROM searchengine WHERE $construct";

$run = mysql\_query($construct);

$foundnum = mysql\_num\_rows($run);

if ($foundnum==0)

echo "Sorry, there are no matching result for <b>$search</b>.</br></br>1.

Try more general words. for example: If you want to search 'how to create a website'

then use general keyword like 'create' 'website'</br>2. Try different words with similar

meaning</br>3. Please check your spelling";

else

{

echo "$foundnum results found !<p>";

while($runrows = mysql\_fetch\_assoc($run))

{

$title = $runrows ['title'];

$desc = $runrows ['description'];

$url = $runrows ['url'];

echo "

<a href='$url'><b>$title</b></a><br>

$desc<br>

<a href='$url'>$url</a><p>

";

}}

}}

?>

Keywords Stored in Database

**TITLE KEYWORDS**

1. Google- google

2. Facebook- facebook social site mark zuckerberg

3. Apple- apple tim cook steve jobs

4. ITM- itm university college

5. INDIA- india country bharat hindustan

6. PBL- pbl project learning

7. Atul- atul anand boy

8. Bharti- bharti girl bhartaa

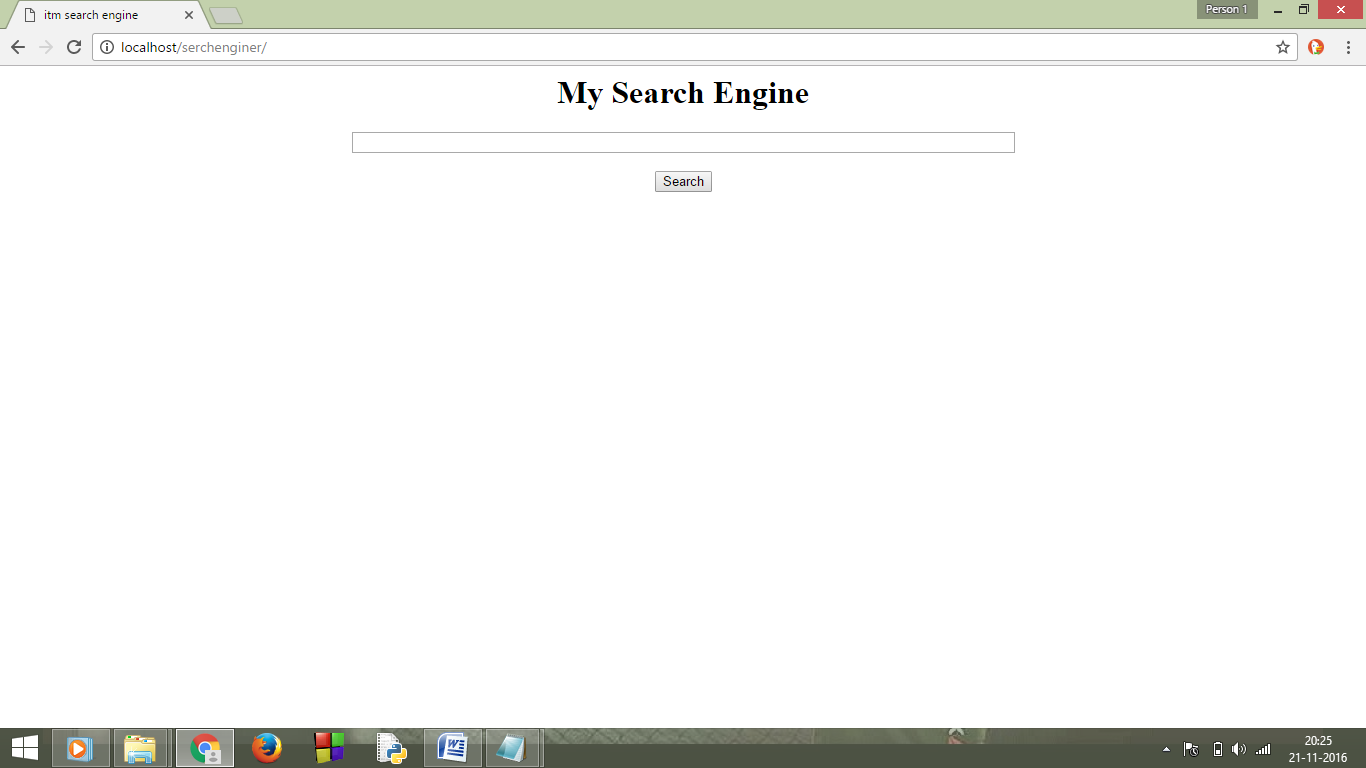
9. Kamlesh- kamlesh kammo pujari

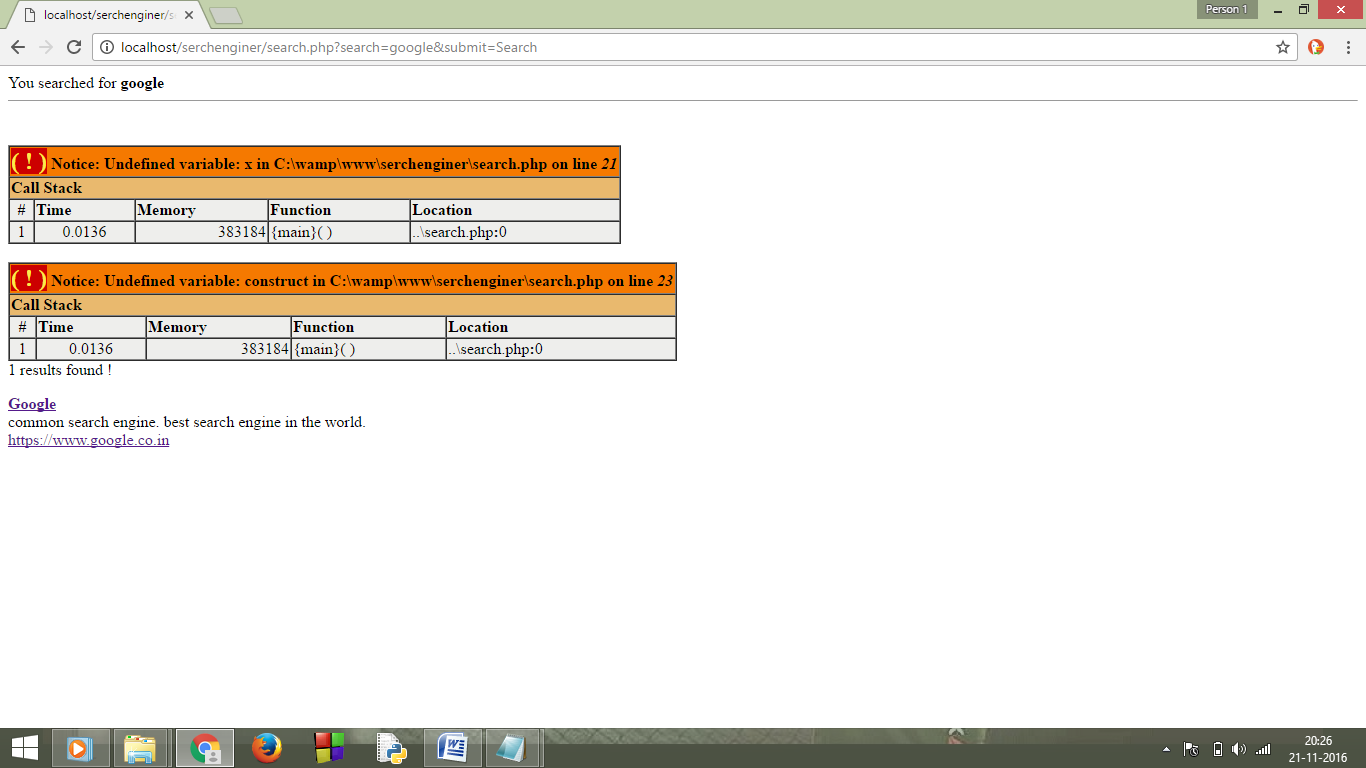
10. One Dreamer- onedreamer own site official practice complete sit...

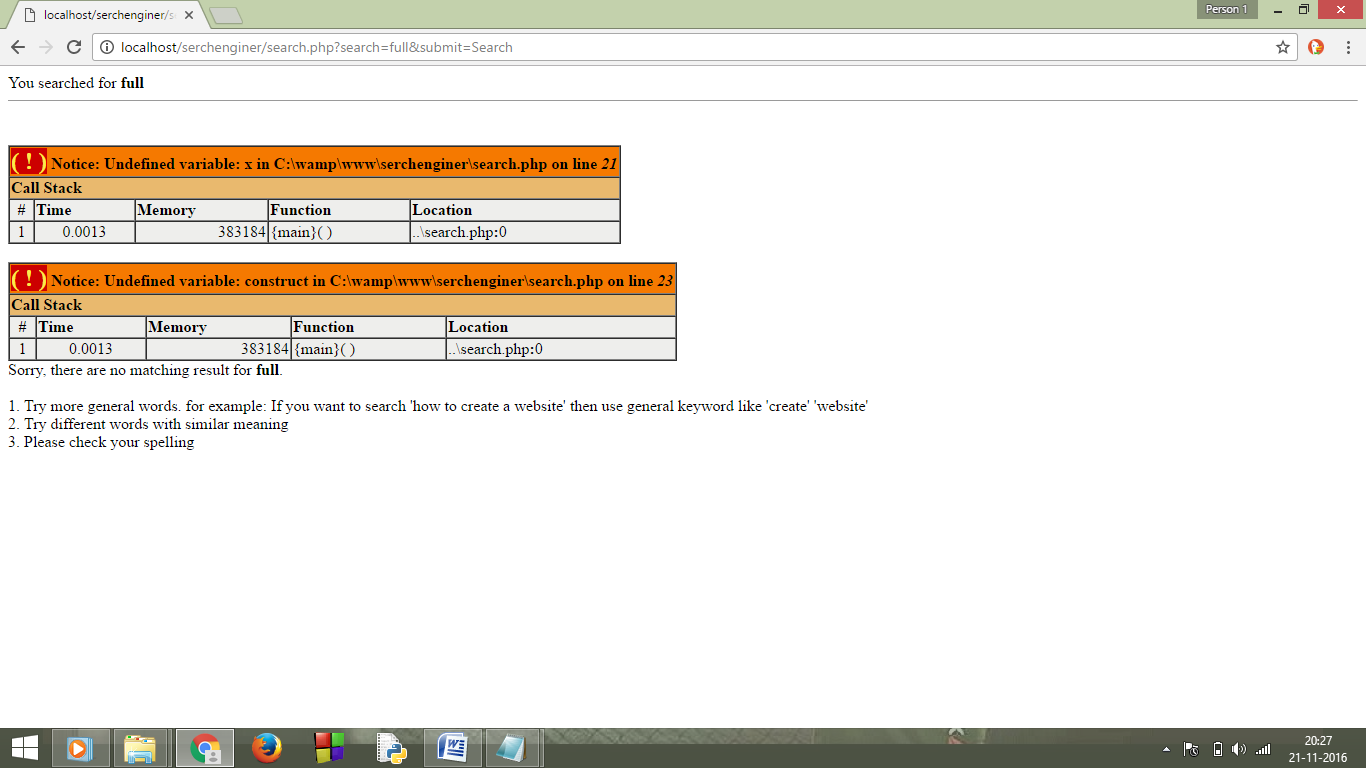
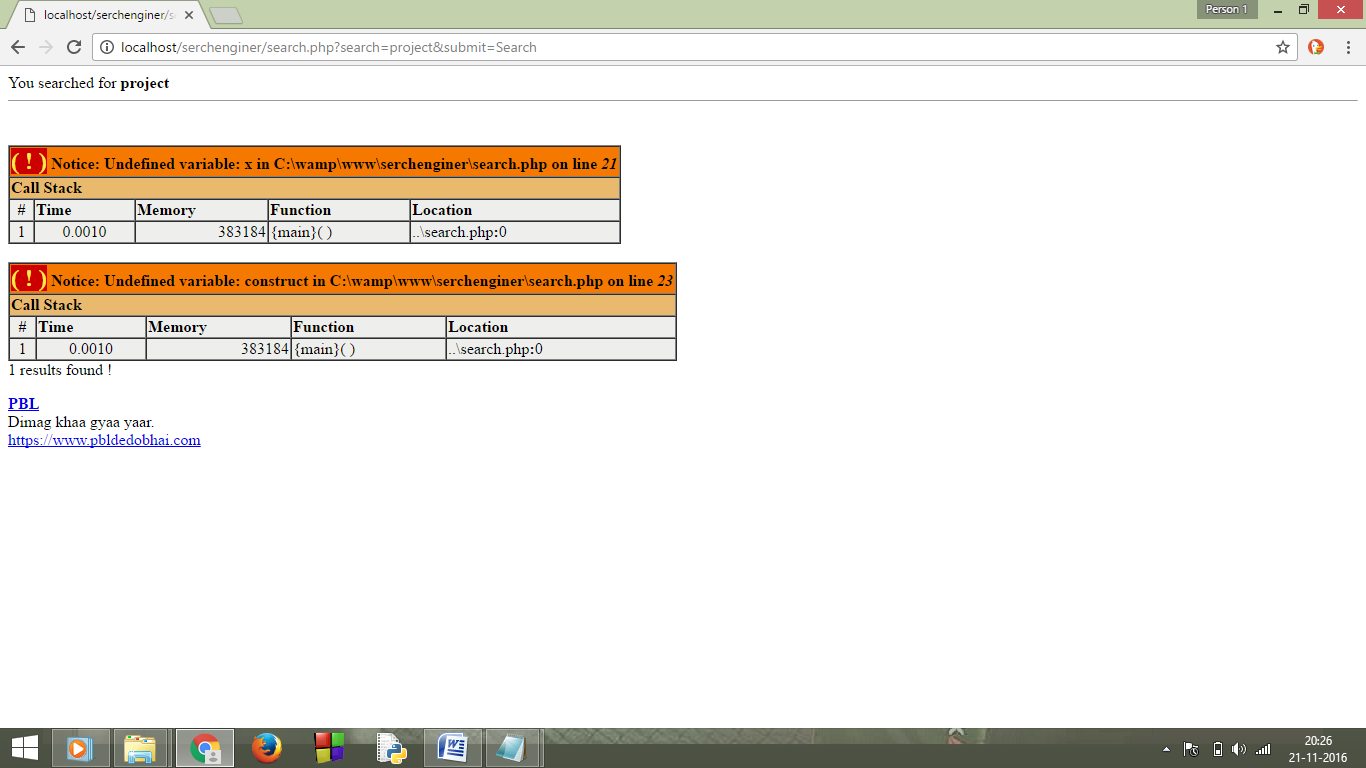
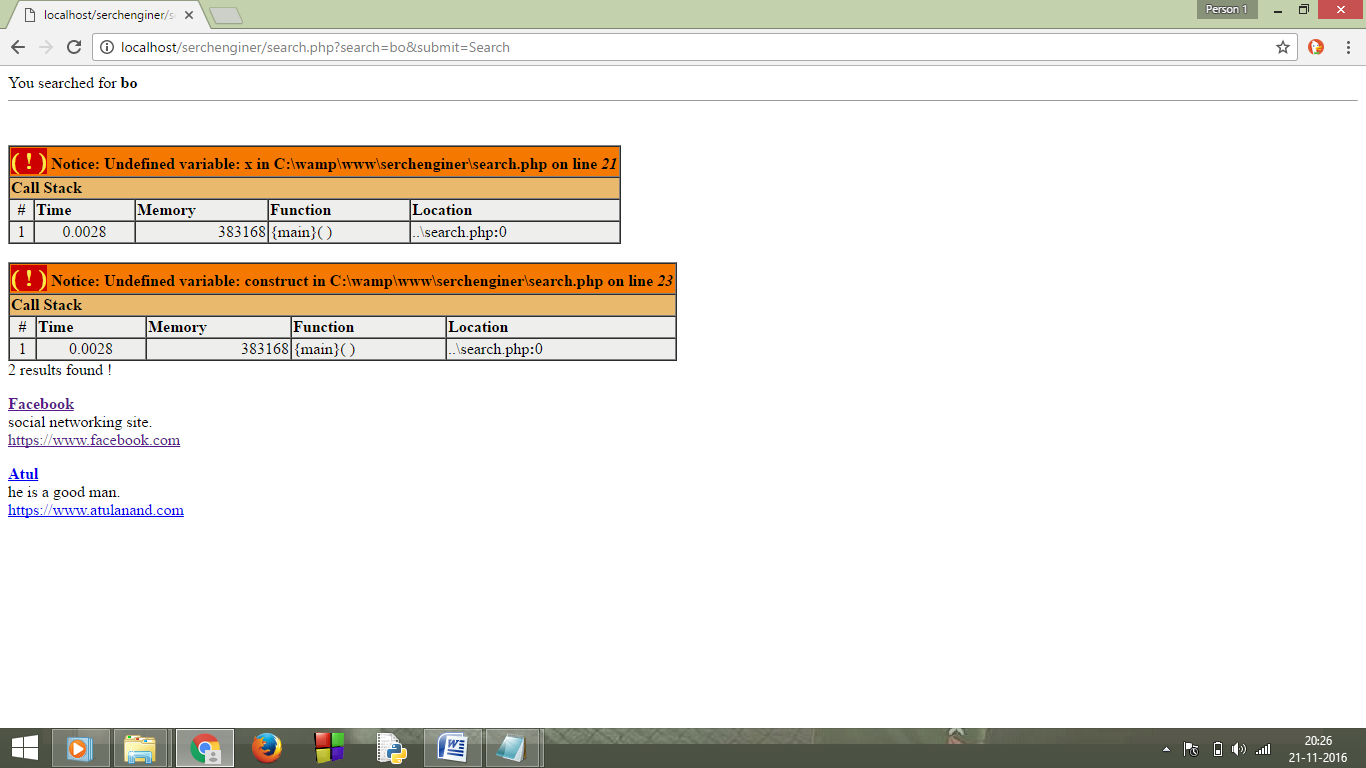
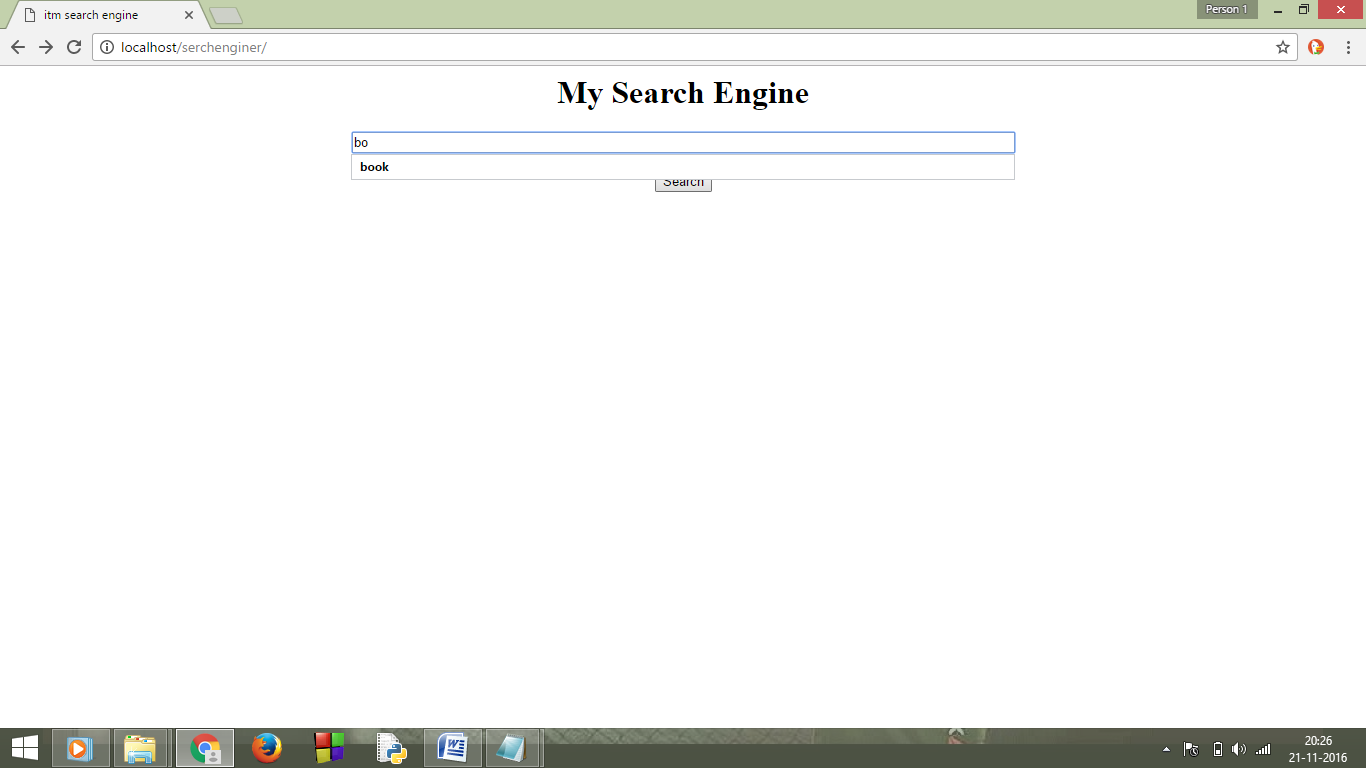
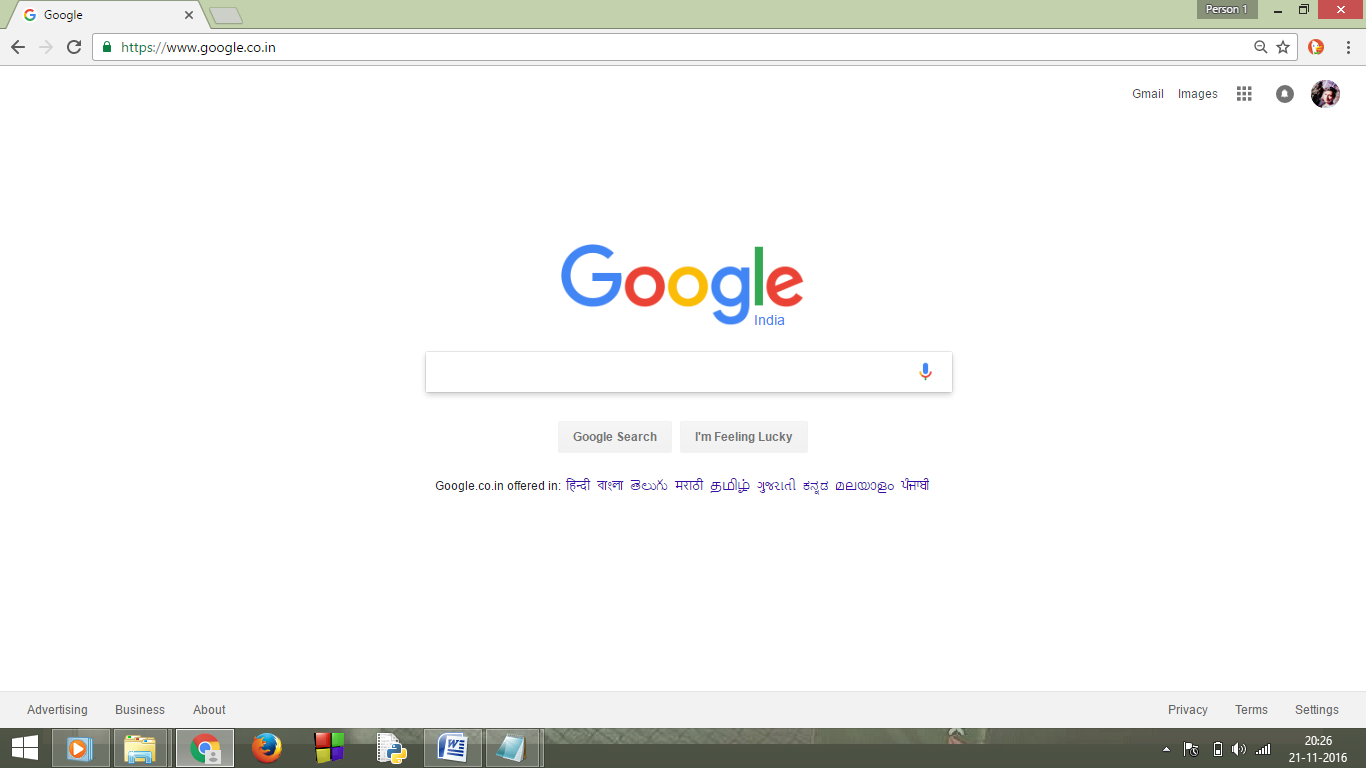
11. Hero Alom-PARTHA- hero alom bangladeshi partha biswas faadu hero girl

12.my site- meet me 4 ur luv own site mysite

**More Screenshots:**





****

**References**

* W3School-PHP,MYSQL
* Google-custom Search
* Stackoverflow
* And Books on PHP