ACKNOWLEDGEMENT

"A project is completed when it starts working for you, rather you working for it"

……Scott Allen

The satisfaction that accompanies on the successful completion of any task would be incomplete without the mention of people, whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I am grateful to my project guide, the **Head of Department of CSE**, **Mr. S. RAJESHWAR** for the guidance, inspiration and constructive suggestions that helped me in the preparation of this project.

In addition, I would like to thank all the faculties of Department of Computer Science and Engineering and my colleagues who have helped in successful completion of this project.

Last, but not the least, I would like to thank my parents for their love, affection and moral support for the care they took in this engineering life.

ABSTRACT

The exponential increasing Internet has a lot of websites, and still requires a large amount of websites for various purposes like social, commercial, educational, etc. The development cycle of any website is a complex task and requires a lot of human effort by its developers. This *Website Maker,* will let the developers make any websites on the go.

The main purpose of the project is to build a web-app that could help to make any website with little effort and will also let the administrator to manage the website which includes adding, deleting and editing any webpage of the website.

Through the existing traditional system, website making processes which is purely manual process becomes a time taking process. The designing phase especially takes a lot of time and there are chances of errors. The existing system is not much secured and could be tampered. The existing system also does not have user-friendly nature during the development.

However there exist many online sources that let the developers choose templates and start building there websites but are costlier.

This project is mainly designed to provide a simple, user-interactive, one-roof platform to make the tasks performed a web-developer and web administrator more comfortable.

This system is capable to be used in by various web developers, who want to make the web development phase simple.

This system is highly reliable, secure, user friendly, durable, flexible, inter-operable, cross-platform and scalable.

CONTENTS

INTRODUCTION

The exponential increasing Internet has a lot of websites, and still requires a large amount of websites for various purposes like social, commercial, educational, etc. The development cycle of any website is a complex task and requires a lot of human effort by its developers. This *Website Maker*, will let the developers make any websites on the go.

Problems with Existing System

Through the existing traditional system, website making processes which is purely manual process becomes a time taking process. The designing phase especially takes a lot of time and there are chances of errors. The existing system is not much secured and could be tampered. The existing system also does not have user-friendly nature during the development.

However there exist many online sources that let the developers choose templates and start building there websites but are costlier.

Solutions to the Problems

- Reliability: The project performs intended function with required precision; hence this project is very reliable.
- Feasibility & Scalability: The project maintenance is very easy and modifications can be made in the existing system in future. The system can easily be scaled up/down as per the requirements.
- Simplicity: The processing of the system is simple following the existing manual method without changes and suitable validation is provided for the easy and correct access of data
- Security: Security measures are taken to avoid mishandling of database. Password restrictions are provided to enter into database. A correct password only will access to the database. Moreover, privileges to various users have also been considered at the database level restricting access of unauthorised users.
- Non Redundant Database: The database of the project is designed in way that most of the redundant data is removed, using Normalization techniques.
- High User-Friendly: The interface for the system is highly user-friendly, that is able to attract users and make their response with the system efficient.

PURPOSE

The main purpose of the project is to build a web-app that could help to make any website with little effort and will also let the administrator to manage the website which includes adding, deleting and editing any webpage of the website.

PRELIMINARY IDEAS

The idea for this project begins from the time when I was trying to make a web-blog. The development task was complex. So thought for a cost-effective tool which could let development and managing of website easier.

PROJECT SCOPE

This project is mainly designed to provide a simple, user-interactive, one-roof platform to make the tasks performed a web-developer and web administrator more comfortable.

This system is capable to be used in by various web developers, who want to make the web development phase simple.

This system is highly reliable, secure, user friendly, durable, flexible, inter-operable, cross-platform and scalable.

SOFTWARE REQUIREMENT & SPECIFICATION

FUNCTIONAL REQUIREMENTS

- The visitor is only able to view the website, its related webpages and its contents.
- The administrator is able to login to the *'website maker'*, by its respective login id and password combination.
- The administrator is able change the timeline of the website.
- The administrator is also able to list down all the pages in the manage page section.
- In manage page section, the administrator can add new pages or delete existing pages or even modify the details within each page.

CAPABILITY REQUIREMENTS

- The system must be capable to handle a huge number of requests at a time.
- The system must be secure, feasible, scalable with a high interactive UI.
- The password saved must be encrypted.
- Only one user per user-id is capable to login at any moment.

PERFORMANCE REQUIREMENTS

- The requests processing time must be low, for this the overhead in the requests must be minimum.
- The webpage loading must be dynamic and this time must be minimum.

USER INTERFACE REQUIREMENTS

- The UI must be interactive to let user perform task simply & easily.
- For all modules there must be a separate UI entity (web page), to let users easily flow through the system as per their requirement.
- Use of JavaScript, CSS, AJAX, Boot-strapping, etc., will provide easy searching, listing, entering the information as well as provides dynamic UI.

Number of Modules

1. Login page for Administrator
2. Edit Timeline
3. Modify Pages
4. Logout
5. Website View for guest visitors

Modules Description

1. Administrator:

Administrator is the owner of the system and has the privilege to view, add, delete or modify webpages. Administrator is also able to modify the website timeline.

Administrator can even change own user-id and password combination.

2. Guest User:

A guest user to the system is capable to just view website and its webpages with contents through the hyperlinks available.

3. Login page for Administrator:

This page allows the administrator to enter the user-id & password, then these credentials are verified against the database. If found, to be correct the administrator is redirected to the admin page.

4. Edit timeline:

This allows the administrator to edit the timeline of the website.

5. Modify pages:

This section allows the administrator to perform the following operations related to the webpages of the website:-

i. Add new webpage.

ii. Delete an existing webpage.

iii. Modify an existing webpage along with its contents.

4. Logout: By clicking the *"logout"* button available on the Administrator Webpage, the administrator can logout from the system. By this operation, all the privileges will be revoked from the user and will be redirected again to the *"login"* page.

CONSTRAINT REQUIREMENTS

- One must be forwarded to their secret pages only after successful verification of their user-id and password combinations.
- There must not exist any web-pages with similar alias.
- A user must be able to change his/her password, only after their successful login.
- An alert to be provided before any deletion of data entry from database.

INPUT & OUTPUT REQUIREMENTS

The major inputs and outputs functions of the system are follows:

Inputs:

- Administrator enters his user id, username and password for login to authenticate in this system.
- The administrator if willing to add / remove / modify any webpage, then he/she has to click over the respective button-link available.
- To set or modify the timeline of the website, the administrator must enter the tag-line heading and tag-line contents.
- To add any new page or modify existing webpages the administrator has to enter the following details:-
    - i. Title of the page
    - ii. Parent class of the page (if any).
    - iii. Page alias (must be unique).
    - iv. Description or content of the webpage.
    - v. Meta keywords and meta description to identify the webpage.

vi. Sort order number of the webpage.

vii. Status of the webpage (Active/Inactive).

- Administrator must click on the confirmation message before deleting the webpage.

Outputs:

- Every guest user is able to see the changes made by the administrator for the website and its web-pages.
- For any changes to the database of the website, a successful confirmation must be displayed on success or else an error message must be displayed.
- The output must be displayed on the same page where is it searched, usually in tabular form.
- After successful login the user must be redirected to proper logged-in page else the flow must remain there itself.
- Before deleting any webpage, there must be a confirmation to be displayed to alert the administrator for proper deletion.

SOFTWARE REQUIREMENTS

- APACHE TOMCAT
- MYSQL
- NETBEANS IDE
- MYSQL WORKBENCH

HARDWARE REQUIREMENTS

- CPU [1 GHz]
- SECONDARY MEMORY [200 MB]
- PHYSICAL MEMORY [512 MB]

SYSTEM ANALYSIS AND DESIGN

System analysis and design is the phase of software development during which the team studies the requirements specified in the first phase with respect to available hardware and software technologies and prepares a system design document.

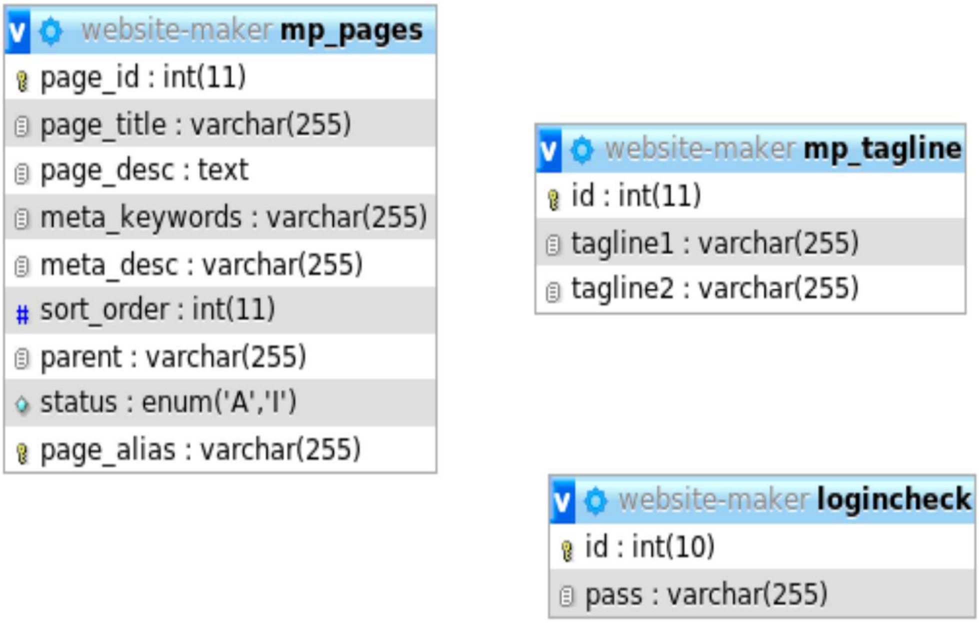
The system design document defines the overall system architecture.

It specifies the various hardware and software components of the system and interfaces between the components.

It includes the hierarchy of software components, rules for component selection, and interfaces between components. The design document serves as input to the next phase of the model.

* The design of this system, has been made by the use of tools like MySQL Workbench (for database), Microsoft One-Note (for flow of controls & data in website), Dia (for UML diagrams), etc.

* For proper analysis of designs, the data from MySQL Workbench was exported to PhpMyAdmin which also enabled to construct the database effectively in later stages.



website-maker **mp_pages**
page_id : int(11)
page_title : varchar(255)
page_desc : text
meta_keywords : varchar(255)
meta_desc : varchar(255)
sort_order : int(11)
parent : varchar(255)
status : enum('A','I')
page_alias : varchar(255)

website-maker **mp_tagline**
id : int(11)
tagline1 : varchar(255)
tagline2 : varchar(255)

website-maker **logincheck**
id : int(10)
pass : varchar(255)

fig: ER Diagram for Web-Site Maker

TESTING & DEBUGGING

The developed system has undergone the following testing methods in ordered steps as:

1. Unit Testing
   a) Ad-hoc
   b) Black-box
   c) White-box
2. Integration Testing
   a) Top-down
   b) Bottom-up
   c) Regression
   d) Smoke
3. Validation Testing
4. System Testing
   a) Performance
   b) Stress
   c) Security
   d) Recovery

After testing of the system, the following results & conclusions are:

1. Unit Testing: All the errors like erroneous computation, incorrect computation & improper control flow has been tested and debugged.
   a) Ad-hoc: The random testing was on its way from the time it started and all the bugs found has been rectified.
   b) Black-box: Every functionalities of each module has been tested for boundary values, domain values & with various command line.
   c) White-box: Every segment of each module has been tested here for branch coverages, paths, data flow test, loop tests & compound condition coverage.

2. Integration Testing: This testing methodology was used to remove bugs like data losses across interfaces, improper results by sub-functions to major functions, individual imprecisions at unacceptable levels & global data structure problems.

    a) Top-down: Starting from the root level (i.e., the index of root directory) to the leaves of various path or subdirectories, this testing was used. Many logistics bugs were found however has been removed.

    b) Bottom-up: This approach was used to eliminate the need for any complex stubs. However none were found during testing.

    c) Regression: This testing reduces 'side effects' during module-module communication. Each time any changes were made this testing was used, including the integration of new components.

    d) Smoke: Each time the software is rebuilt (with new components), the smoke testing was performed by testing each & every module & function with the existing project.

3. Validation Testing: This testing was performed by focussing over the user-visible actions and user recognizable output of the system. This was successfully compromised when the system functioned in a manner that can be reasonably expected a the customer (as per the Software Requirements & Specifications).

    a) Alpha Testing: This system has undergone all testing ways at alpha level, i.e., at the developer's site by various colleagues, end-users, etc.

    b) Beta Testing: The system has been opened public to undergo the Beta Testing, by its various interested users.

4. System Testing: A classic system testing - "The Finger Pointing" is been used to test the system.

    a) Performance: The system has met all the non-functional requirements as defined in SR&S documents and has been tested.

    b) Stress: This testing helped to evaluate the system performance when it is stressed for a short period of time.

    c) Security: This test was allowed to protect the system from improper testing and to protect the system from vulnerabilities of frontal attack.

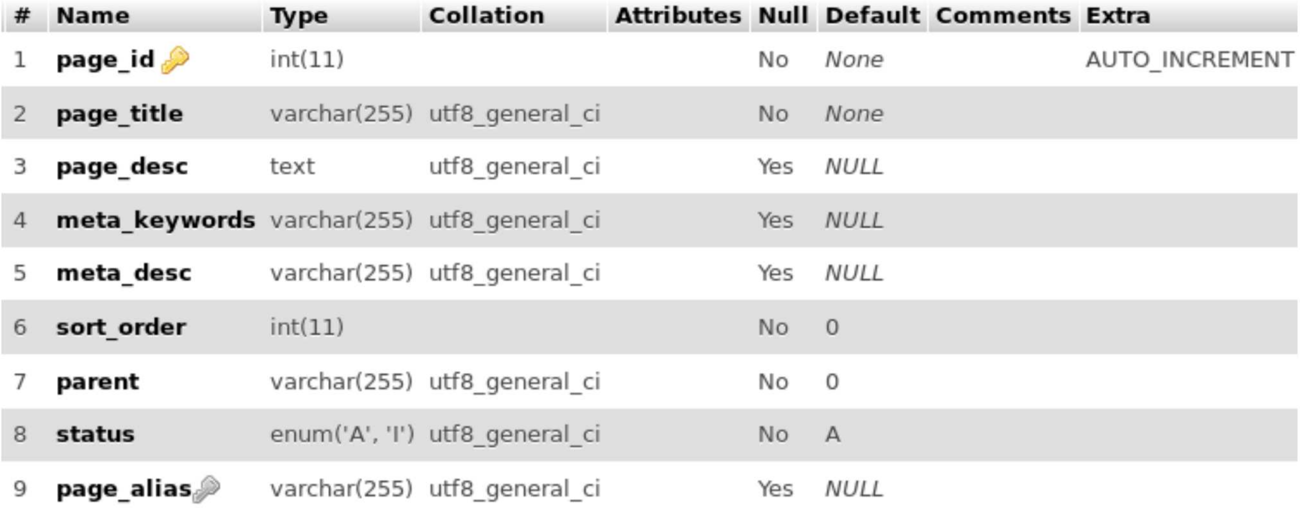
    d) Recovery: During this, the system was forced to fail in various ways. The use of Database like MySQL & scripting language like php has let this system to recover in all the cases. The recovery processes are automatic, however may require the administrator to restart the system services after the fail but the data integrity is maintained.

**The use of NetBeans IDE, has let to overcome most of the specific errors.

IMPLEMENTATION

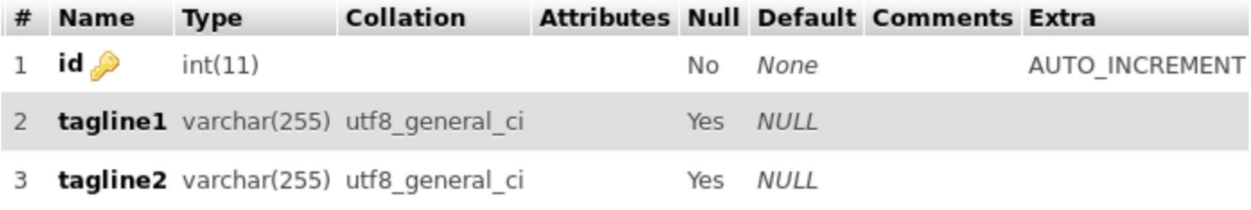
From the System Analysis and Design, Database Tables are implemented on MySQL using PhpMyAdmin tool, that are capable to fulfil the requirements.

Following Figures and Tables depicts some samples of it.



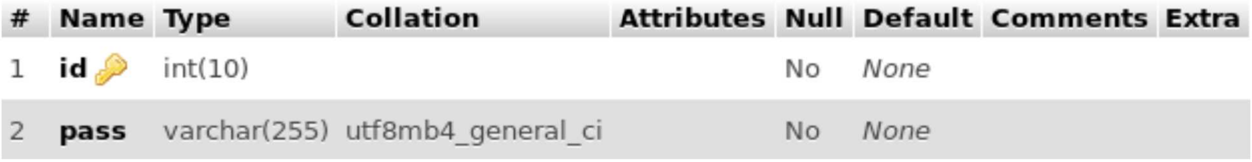
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**page_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**page_title**	varchar(255)	utf8_general_ci		No	*None*		
3	**page_desc**	text	utf8_general_ci		Yes	*NULL*		
4	**meta_keywords**	varchar(255)	utf8_general_ci		Yes	*NULL*		
5	**meta_desc**	varchar(255)	utf8_general_ci		Yes	*NULL*		
6	**sort_order**	int(11)			No	0		
7	**parent**	varchar(255)	utf8_general_ci		No	0		
8	**status**	enum('A', 'I')	utf8_general_ci		No	A		
9	**page_alias** 🔑	varchar(255)	utf8_general_ci		Yes	*NULL*		

Table: pages



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**tagline1**	varchar(255)	utf8_general_ci		Yes	*NULL*		
3	**tagline2**	varchar(255)	utf8_general_ci		Yes	*NULL*		

Table: tagline



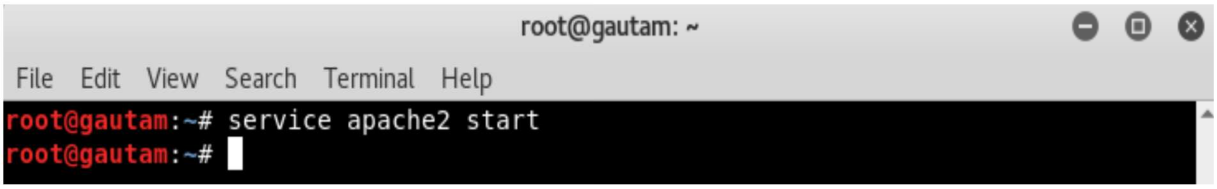
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(10)			No	*None*		
2	**pass**	varchar(255)	utf8mb4_general_ci		No	*None*		

Table: login

DEPLOYMENT

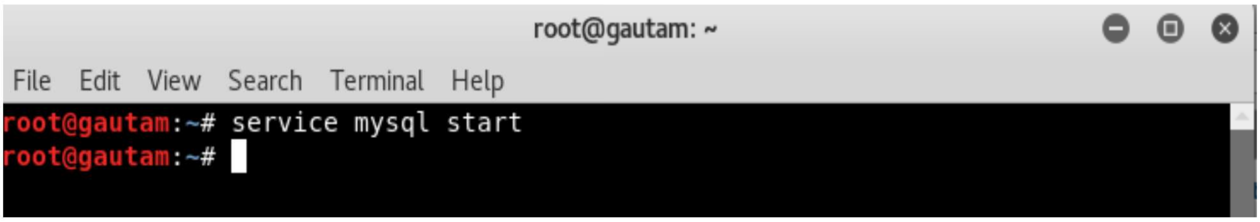
The deployment of the system has been performed using some specific tools by the following ways:

1. Deployment of Web Server (Apache-2 or later)



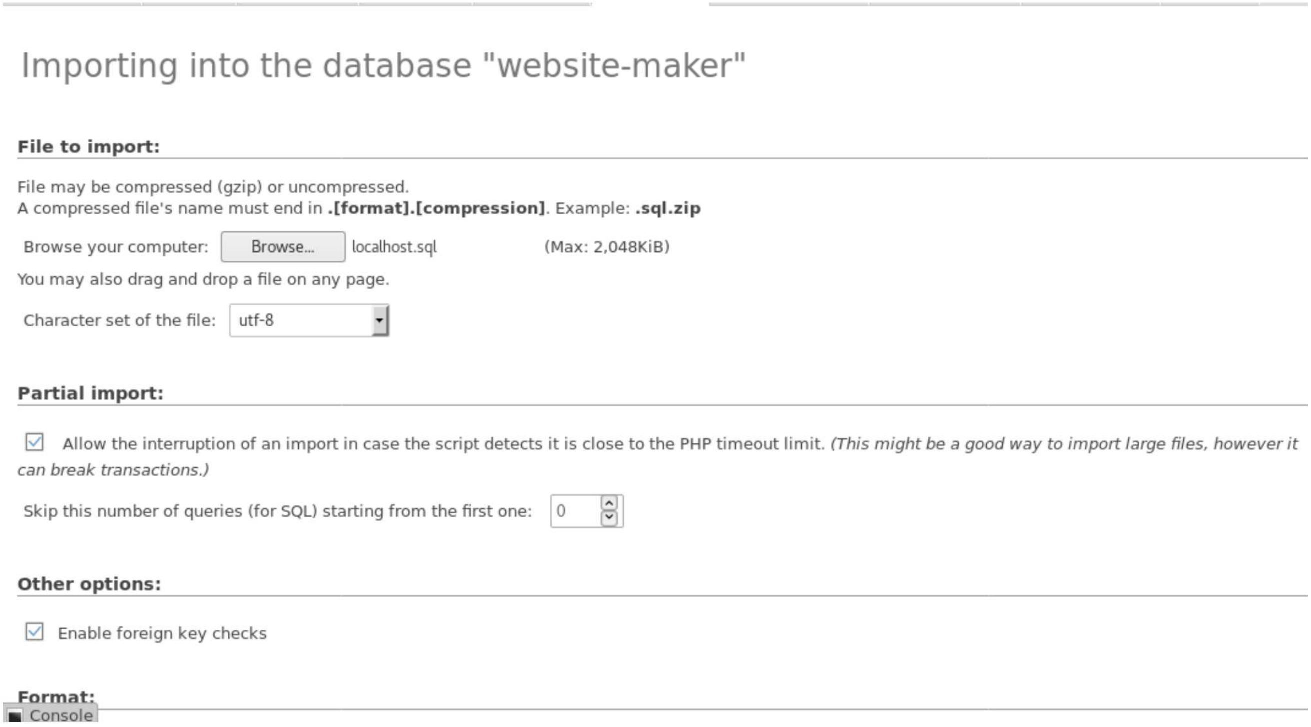
```
root@gautam:~# service apache2 start
root@gautam:~# 
```

2. Deployment of MySQL Server.



```
root@gautam:~# service mysql start
root@gautam:~# 
```

3. Importing the *'website maker'* database to MySQL using PhpMyAdmin.



Importing into the database "website-maker"

File to import:

File may be compressed (gzip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Browse... localhost.sql (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

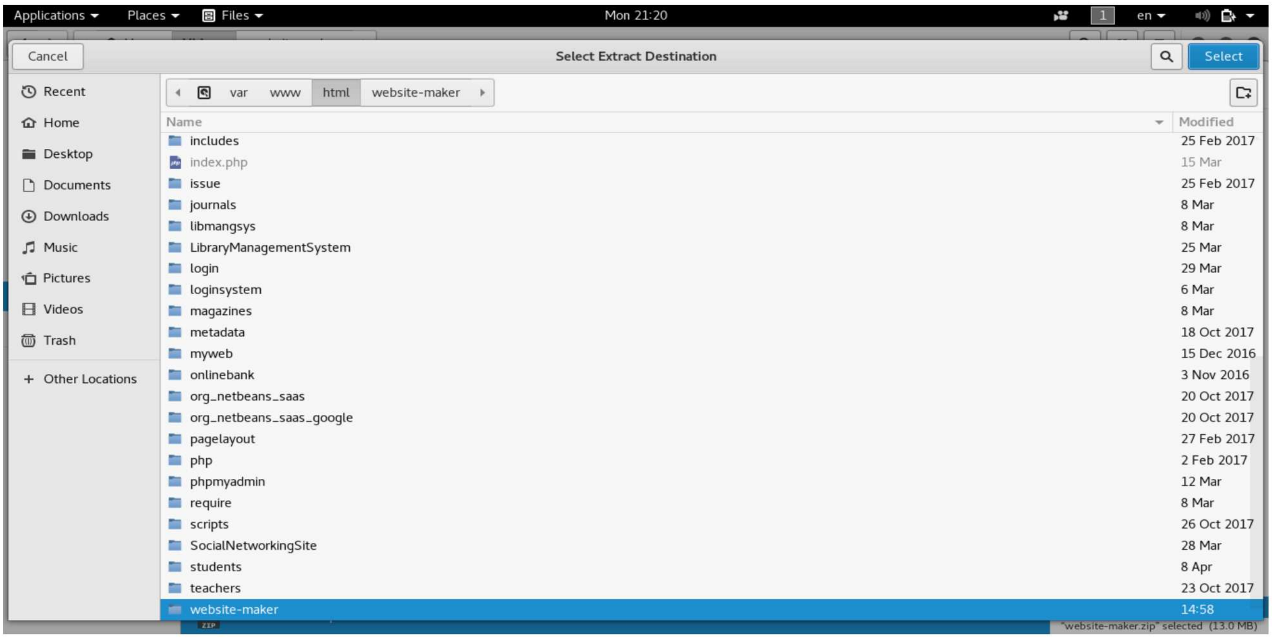
Skip this number of queries (for SQL) starting from the first one: 0

Other options:

☑ Enable foreign key checks

Format:

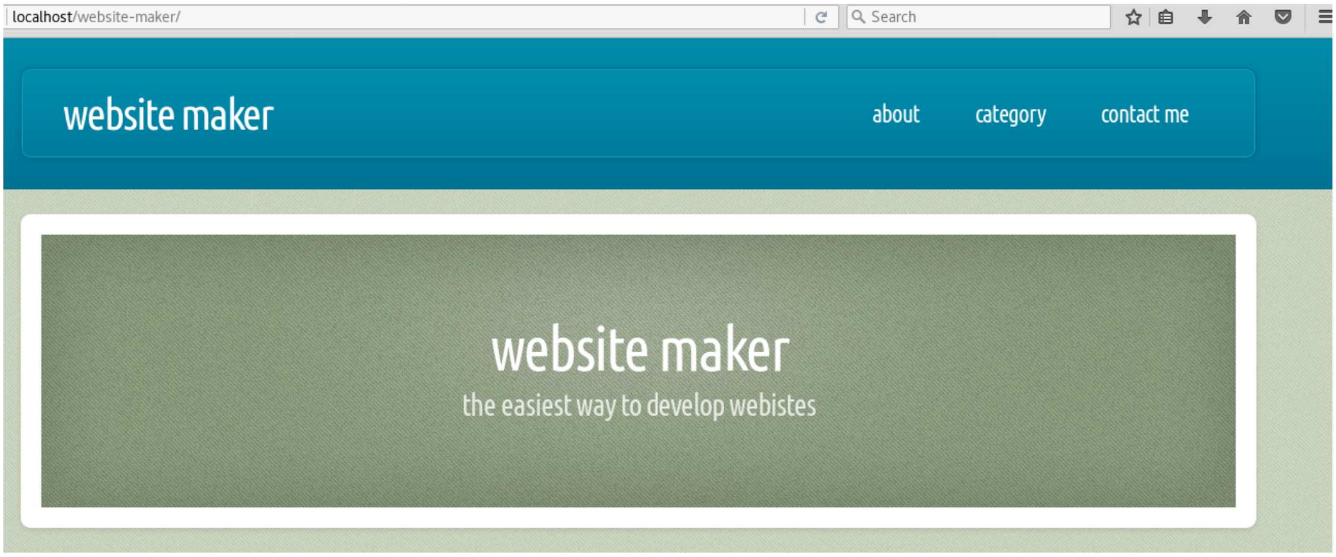
4. Extracting the ‘*Website Maker*’ webpages to the web-server’s root folder.



Name	Modified
includes	25 Feb 2017
index.php	15 Mar
issue	25 Feb 2017
journals	8 Mar
libmangsys	8 Mar
LibraryManagementSystem	25 Mar
login	29 Mar
loginsystem	6 Mar
magazines	8 Mar
metadata	18 Oct 2017
myweb	15 Dec 2016
onlinebank	3 Nov 2016
org_netbeans_saas	20 Oct 2017
org_netbeans_saas_google	20 Oct 2017
pagelayout	27 Feb 2017
php	2 Feb 2017
phpmyadmin	12 Mar
require	8 Mar
scripts	26 Oct 2017
SocialNetworkingSite	28 Mar
students	8 Apr
teachers	23 Oct 2017
website-maker	14:58

“website-maker.zip” selected (13.0 MB)

5. Verifying the system

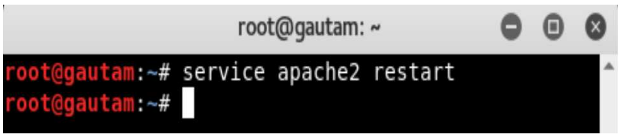


MAINTENANCE

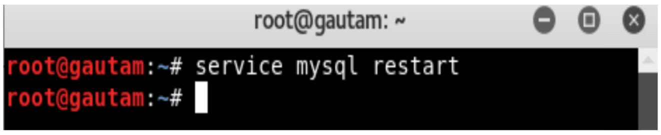
The system has undergone testing and debugging in most of the possible ways, but as specified by experts, the Debugging always have 2 outcomes:-

1. The cause will be found and corrected.
2. **The cause will never be found but will appear magically anytime & anywhere, which no one could predict.**

So there must be a Maintenance System, for any system. The *"Website Manager"*, has been developed in such a way that requires minimum maintenance. Most of the time if failure occurs, the system can be restarted by restarting the database service and web-server service.



```
root@gautam:~# service apache2 restart
root@gautam:~# 
```



```
root@gautam:~# service mysql restart
root@gautam:~# 
```

Fig: To restart services of Apache and MySQL.

** The administrator also has the privilege to login to the database using UI tool like PhpMyAdmin, and may correct the database if required at the time of maintenance.*

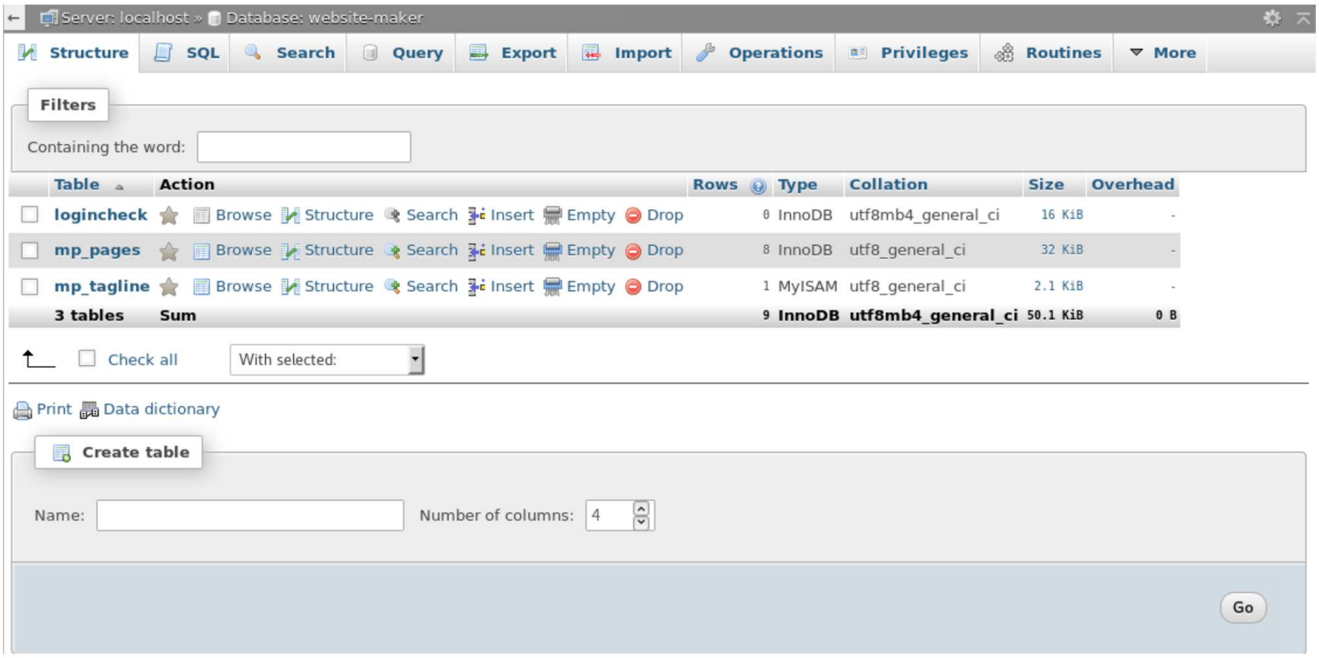


	Table	Action	Rows	Type	Collation	Size	Overhead
□	logincheck	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16 KiB	-
□	mp_pages	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci	32 KiB	-
□	mp_tagline	Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_general_ci	2.1 KiB	-
	3 tables	**Sum**	**9**	**InnoDB**	**utf8mb4_general_ci**	**50.1 KiB**	**0 B**

RESULT & ANALYSIS

"The Result can be obtained and analysed in its best way by Feasibility Study..."

Feasibility Study

This study is to get better understanding of problems and reasons by studying existing system, if available.

- are there feasible solutions? - is the problem worth solving?

This in detail includes:

- Consider different alternatives.
- Estimate costs and benefits for each alternatives.
- Essentially covers other steps for methodology (analysis, design, etc.) in a 'c*apsule'* form.
- Make a formal report and present it to management and users, review here confirms the following:
  - will alternatives be acceptable
  - are we solving the right problems
  - does any solution promise a significant return
- users/management select the alternative

1. Technical Feasibility:
   a) Understand the different technologies involved in the proposed system.

      Ans: All the technologies involved are open source.

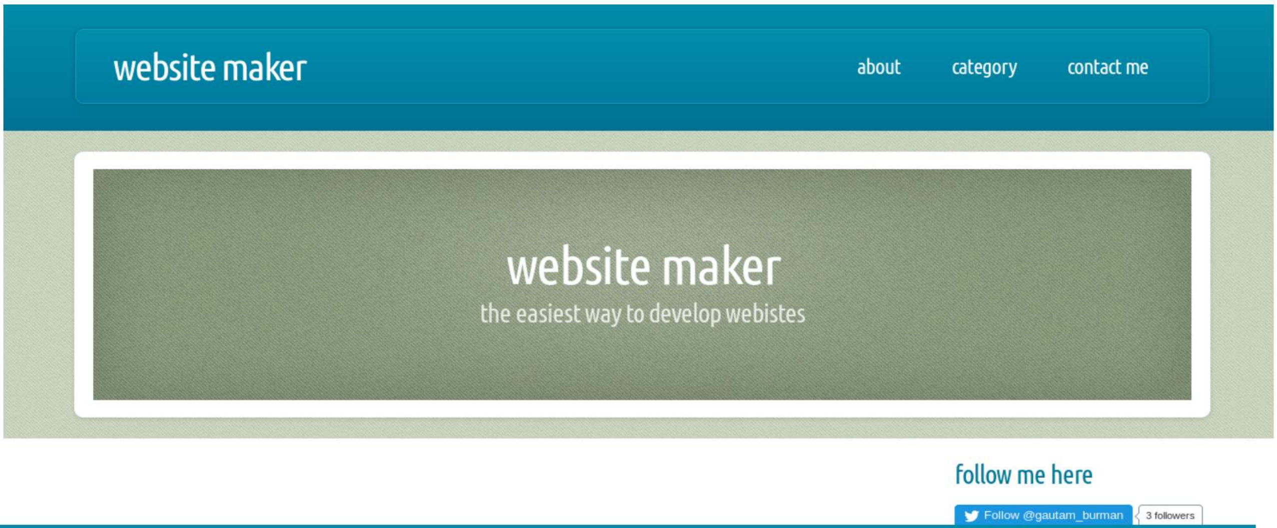
   b) Find out whether the organization currently possesses the required technologies.

      Ans: This requires a server enabled system, which any developer/organization has who want to host any website.
2. Operational Feasibility:
   a) Is there sufficient support for the project from management from users?

      Ans: Every one want there processes to run smoothly in a simple automated way having security, integrity & durability over their data.

   b) Are the current business methods acceptable to the user?

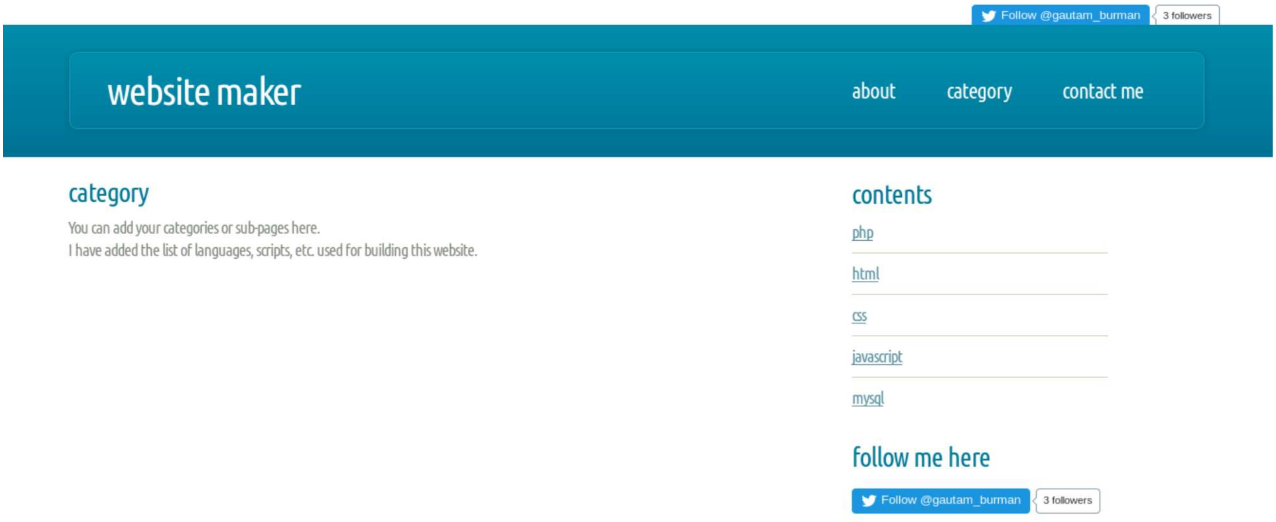
      Ans: This is one of the cheapest mode.



website maker

the easiest way to develop webistes

follow me here



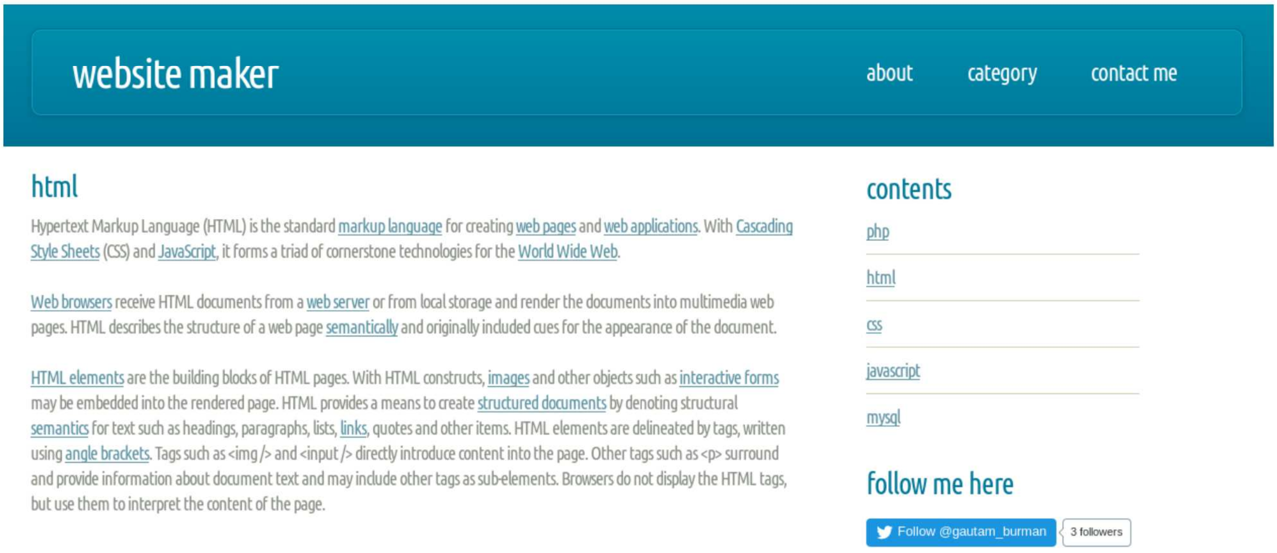
category

You can add your categories or sub-pages here.
I have added the list of languages, scripts, etc. used for building this website.

contents

- php
- html
- css
- javascript
- mysql

follow me here



html

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

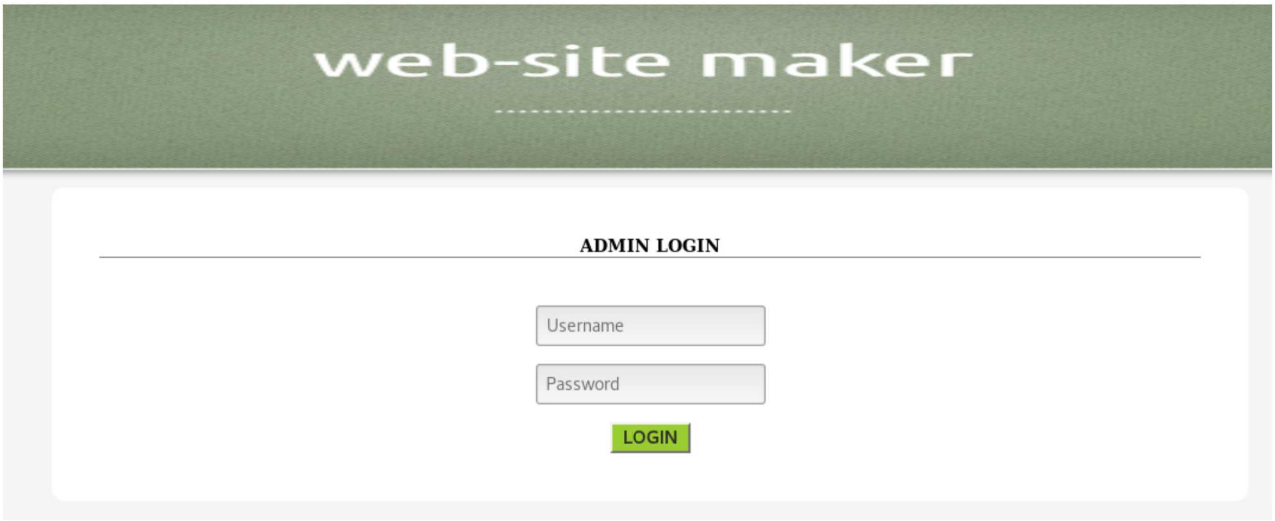
Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

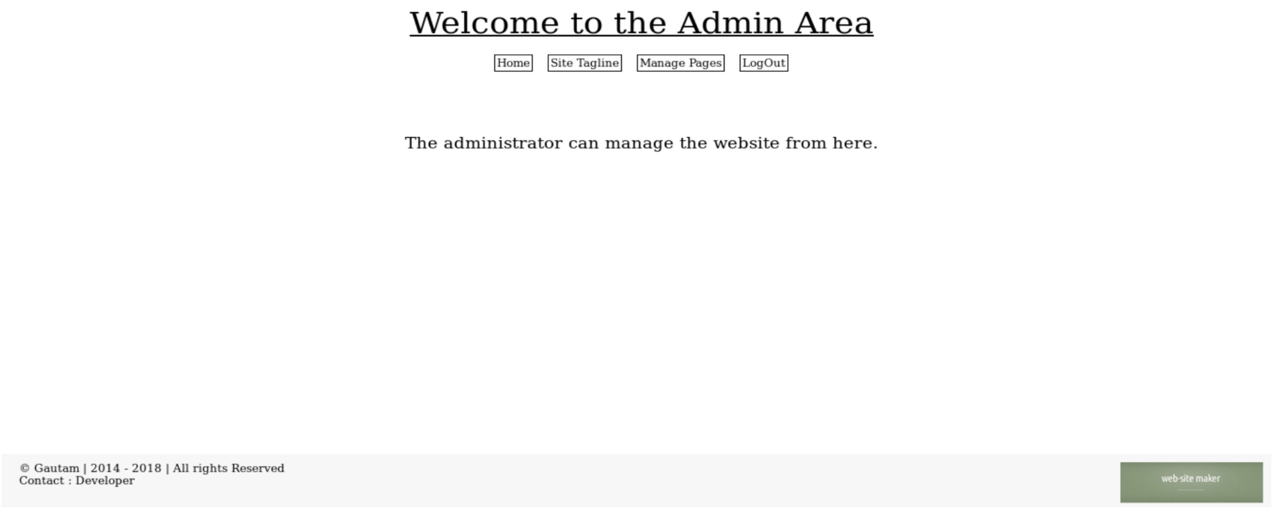
HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

contents

- php
- html
- css
- javascript
- mysql

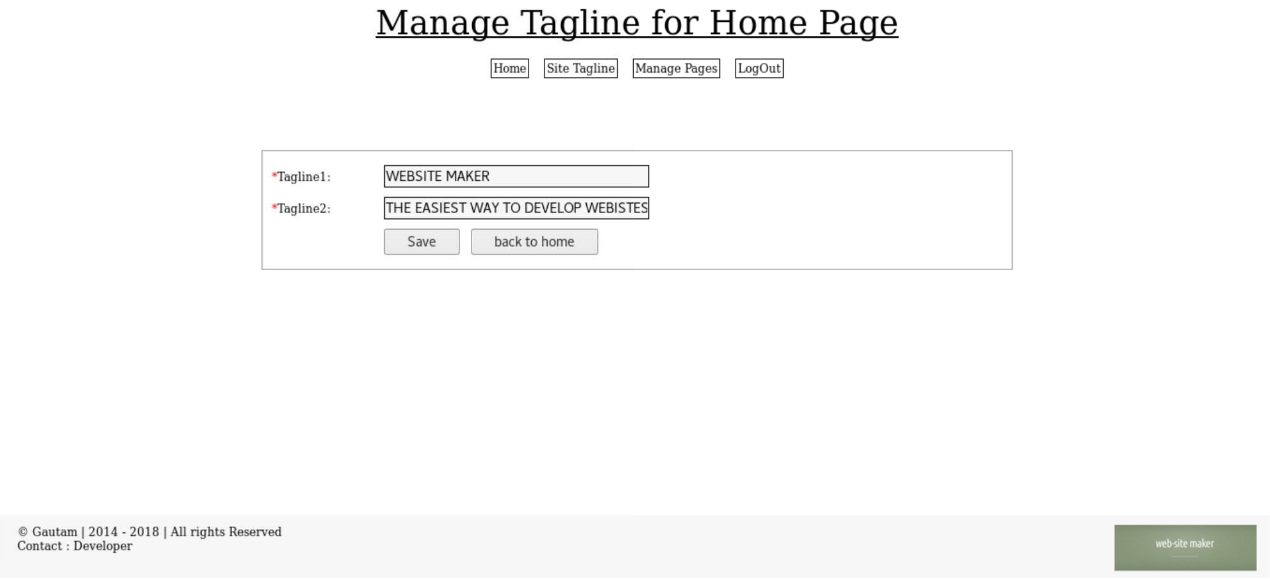
follow me here





The administrator can manage the website from here.

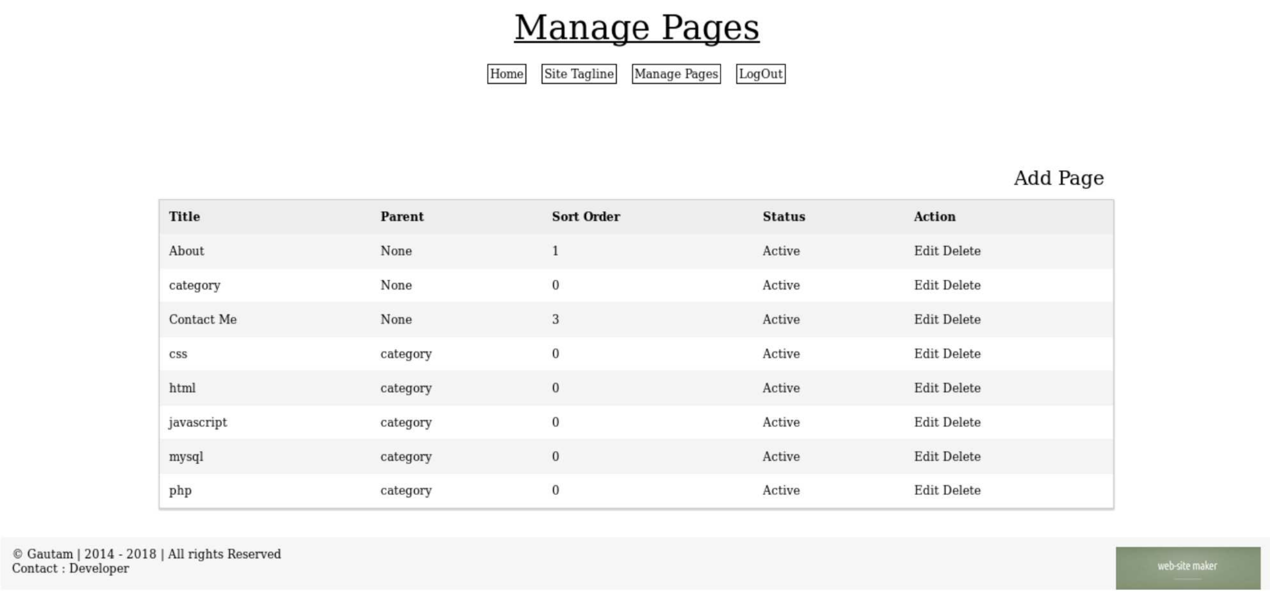
© Gautam | 2014 - 2018 | All rights Reserved
Contact : Developer



*Tagline1: WEBSITE MAKER

*Tagline2: THE EASIEST WAY TO DEVELOP WEBISTES

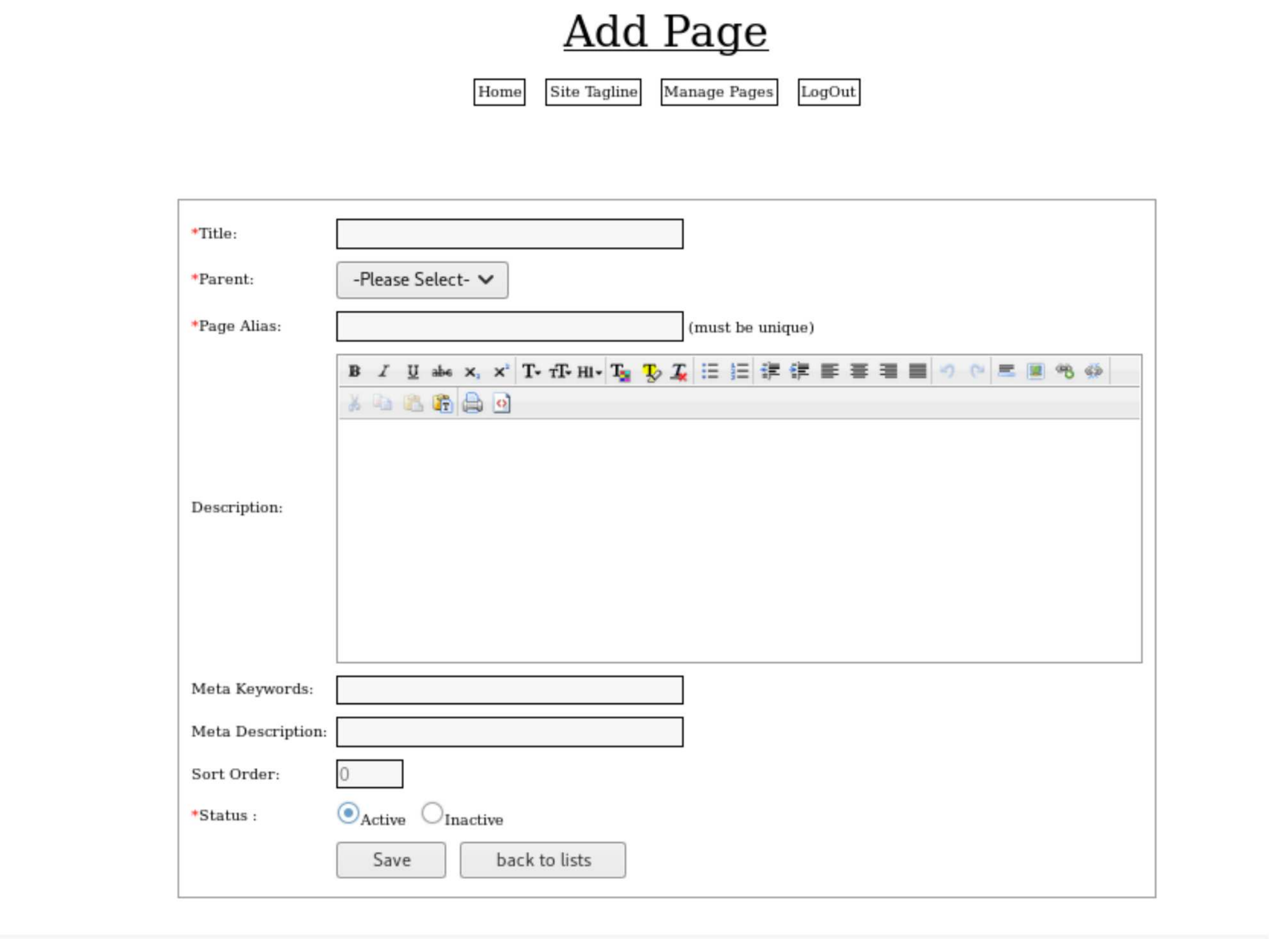
© Gautam | 2014 - 2018 | All rights Reserved
Contact : Developer



Manage Pages

Title	Parent	Sort Order	Status	Action
About	None	1	Active	Edit Delete
category	None	0	Active	Edit Delete
Contact Me	None	3	Active	Edit Delete
css	category	0	Active	Edit Delete
html	category	0	Active	Edit Delete
javascript	category	0	Active	Edit Delete
mysql	category	0	Active	Edit Delete
php	category	0	Active	Edit Delete

© Gautam | 2014 - 2018 | All rights Reserved
Contact : Developer



Add Page

*Title:

*Parent: -Please Select-

*Page Alias: (must be unique)

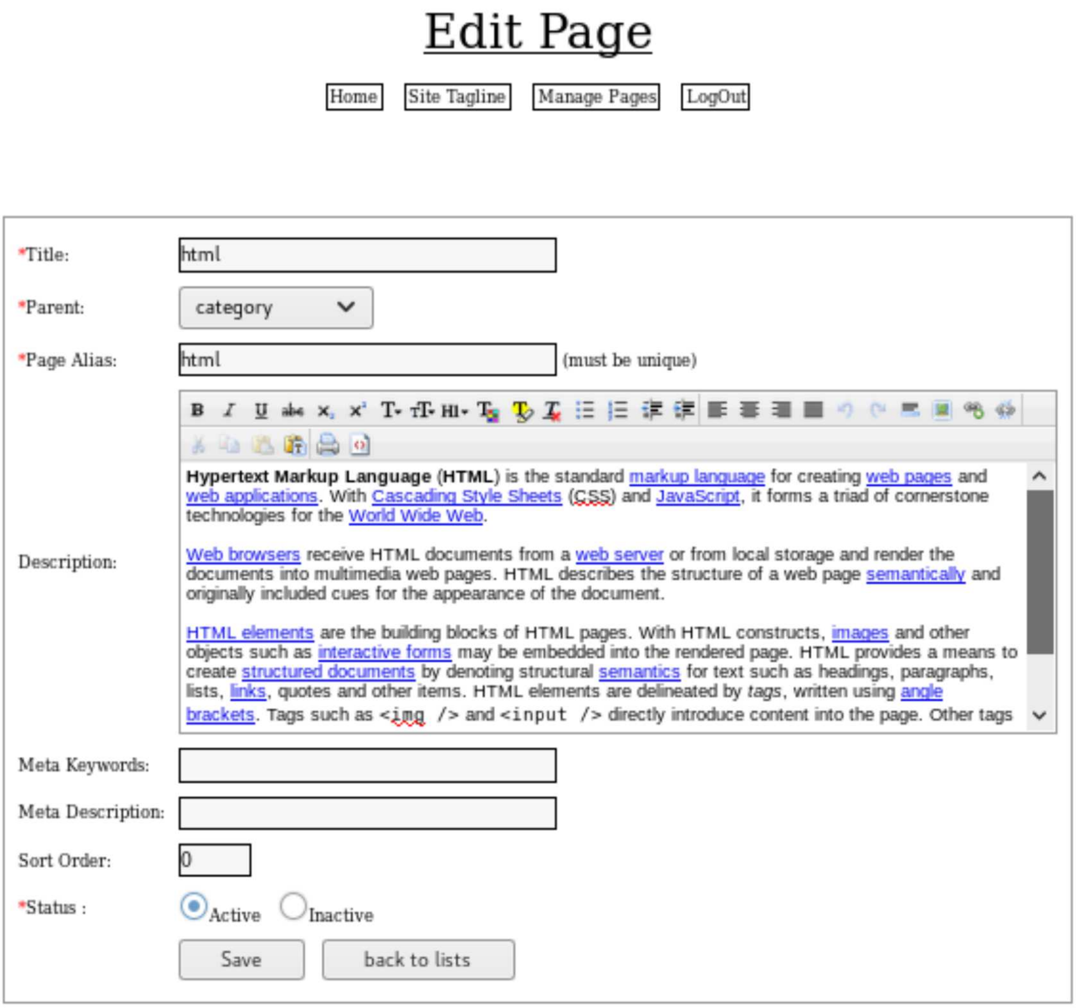
Description:

Meta Keywords:

Meta Description:

Sort Order: 0

*Status : ● Active ○ Inactive



Edit Page

*Title: html

*Parent: category

*Page Alias: html (must be unique)

Description:

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

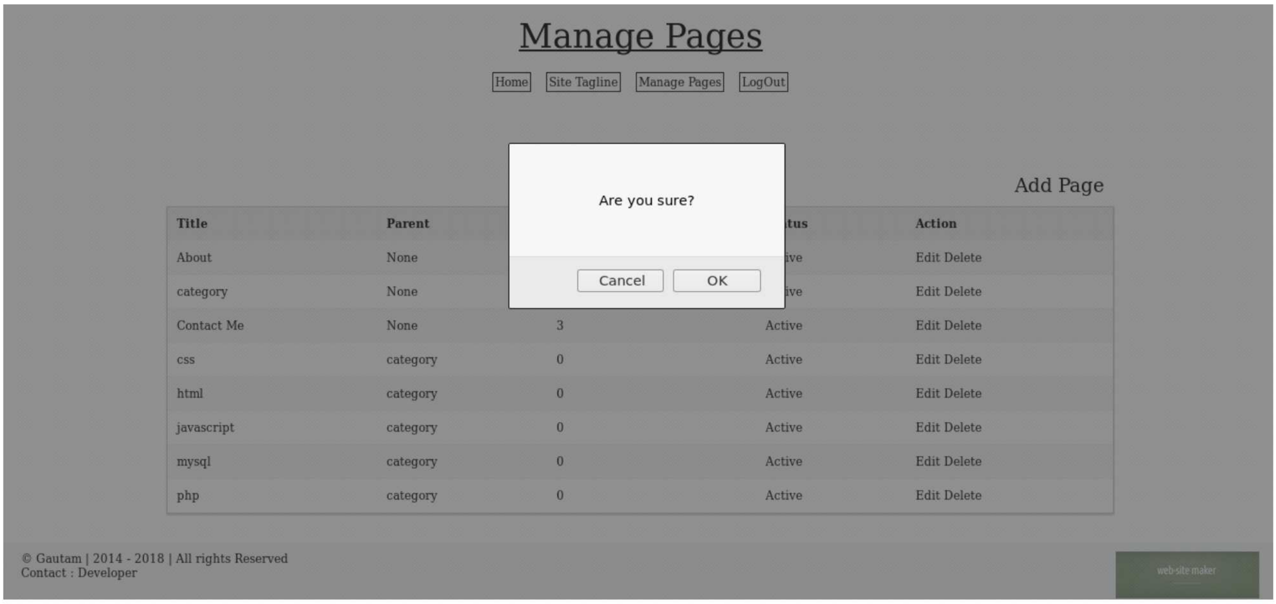
HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags

Meta Keywords:

Meta Description:

Sort Order: 0

*Status : ● Active ○ Inactive



Manage Pages

Are you sure?

Title	Parent		tus	Action
About	None		ive	Edit Delete
category	None		ive	Edit Delete
Contact Me	None	3	Active	Edit Delete
css	category	0	Active	Edit Delete
html	category	0	Active	Edit Delete
javascript	category	0	Active	Edit Delete
mysql	category	0	Active	Edit Delete
php	category	0	Active	Edit Delete

© Gautam | 2014 - 2018 | All rights Reserved
Contact : Developer

CONCLUSION & BIBLIOGRAPHY

CONCLUSION

This Website Maker is one in all tool that is capable to overcome all the failures of previous systems, with a high economic, cost & technology feasibility. That could be easily implemented and maintained. This system is highly scalable and can be turned to different views with minor changes in no time. This system also provides a high level of security over data. This system is very must resistant to failures and is highly durable with a lot of options of recovery.

This is one of the best system tool for website development till date.

BIBLIOGRAPHY & REFERENCES

- Web Technologies (Oxford University Press)– Uttam K Roy
- Data Base Management System – Abraham Silberschatz, Henry F. Kroth, S. Sudarshan
- https://dev.mysql.com/
- http://www.php.net/
- https://www.w3schools.com