Requirement overview

➔ Front-end user interfaces are needed for our website that is integrated with **PHP 5.6, MySQL 5.6, Codeigniter 3.1.2, bootstrap 3.3.7, jquery 2.2.3, AdminLTE 2.3.8** allowing a user, even with limited expertise, to add, modify, and remove content (images too) from a website for creation and modifying the digital content.
➔ The interface can be assessed by only one type of user(admin).
➔ The interface data must be saved in database in their appropriate fields.
➔ The interfaces will resemble **Drupal CMS.**

Different interfaces with input-boxes

WEB-PAGES	INPUT-BOX name(input type)
Press Coverage	1. Cover photo(images) 2. Photo caption(text) 3. News-title(text) 4. Source(text) 5. Published-date(datetime) 6. News-content(text) -> using ckeditor(store html source) 7. Status(char)->(D : draft, P : published, W : withheld) 8. Category(text) 9. Website(text)
Vision	1. Vision Content(text) -> using ckeditor(store html source) 2. Category(text) 3. Website(text)
Team & Advisors	1. Members (Images) 2. Member-type(Team-member/Advisors) 3. Name (text) 4. Description (text) 5. Fun facts (text) 6. Fb-link (text) 7. Twitter link (text) 8. Linkedin link (text) 9. Display-order(integer) 10. Category(text) 11. Website(text)
Career	1. Career content(text) -> using ckeditor(store html source)

	2. Job openings(text) 3. Job Title(text) 4. Job description(text) -> using ckeditor(store html source) 5. Category(text) 6. Website(text)
Life@agrud	1. Life images(images) 2. Image title(text) 3. Image description(text)-> using ckeditor(store html source) 4. Life content (text) -> using ckeditor(store html source) 5. Category(text) 6. Website(text)
FAQ	1. Questions(text) 2. Answers(text and images) -> using ckeditor(store html source) 3. Category(text) 4. Website(text)
Blogs	1. Cover photo(images) 2. Photo caption(text) 3. Blog-title(text) 4. Author(text) 5. Topic(text) 6. Published-date(date) 7. Blog-content(text) -> using ckeditor(store html source) 8. Search-tags(text) 9. Status 10. Category(text) 11. Website(text)
Guides	1. Cover-photo(images) 2. Guide-title(text) 3. Topic(text) 4. Guide-content(text) -> using ckeditor(store html source) 5. Search-tags(text) 6. Category(text) 7. Website(text)
Technology	1. Content(text)-> using ckeditor(store html source) 2. Category(text)

	3. Website(text)
Terms of Use	1. Terms content (text) -> using ckeditor(store html source) 2. Category(text) 3. Website(text)
Privacy & Cookies Policy	1. Privacy content (text) -> using ckeditor(store html source) 2. Category(text) 3. Website(text)

COMMON DATABASE TABLES

Table: Website_master

Id	Int	Unsigned auto_increment
Website_name	Varchar	The websites for which web-pages will be created

Table: Category_master

Id	Int	Unsigned auto_increment
Website_id	int	FK: website_master (id)
Name	varchar	Name of the web-page

VISION PAGE

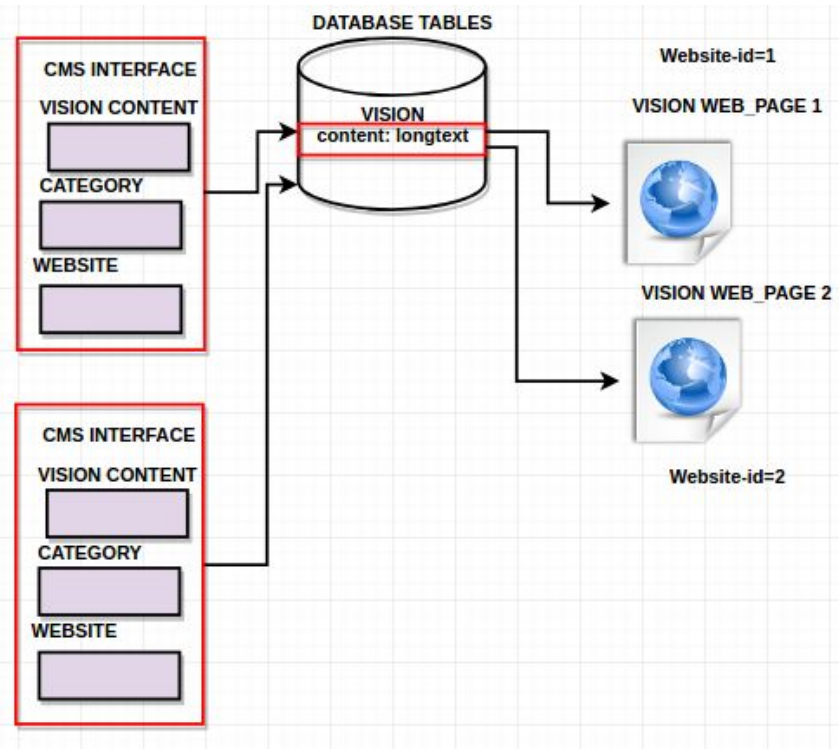1. The following image is the table structure for the vision-page.

Table: Vision

Id	Bigint	Unsigned auto increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)

AGRUD

Content	Longtext	Content of the web-page

2. There can be multiple occurrences of the same web-page with different content in the same website, i.e Agrud.com can have 2 vision pages.
3. There can be multiple occurrences of the same web-pages in different websites as well, i.e agrud.com and xyz.com both can have vision pages.
4. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
5. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
6. The interface will be created as such, that the admin will be able to enter just text.
7. It should contain one 'vision_content' input box in its interface.
8. The html source of the content should be saved instead of the content itself.
9. If the admin wishes to have 2 vision web-pages, in the category_master table, the vision page could be saved as name=vision in different rows for same/or different websites.
10. In the vision table, in whatever way the admin writes and saves the content, the html source code will be saved in the content field along with the website and category.
11. Please visit the url of the vision page for your better understanding: http://staging.agrud.com/vision
12. Here is the overview diagram of how cms interface and web-page are integrated.

TEAM & ADVISORS PAGE

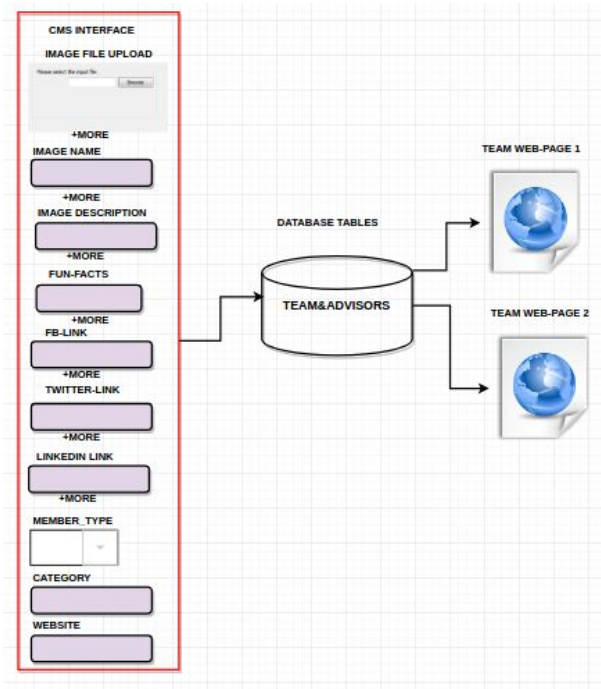1. The following image is the table structure for the team page.

Table: Team&advisors

Id	Bigint	Unsigned auto increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Member-image	varchar	Image urls
Member-type	varchar	Type of member
Name	varchar	Name of the member
Description	text	Description about the person
Fun facts	text	Funny facts about the person
Fb link	varchar	Social-media link
Twitter link	varchar	Social-media link
Linkedin link	varchar	Social-media link
Display order	int	The order in which the images will be displayed

2. The interface should be created as such the admin can be able to enter images as well as text.
3. The interface should be containing the input boxes as mentioned above.
4. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
5. If the user chooses to have 2 similar pages, he should choose which category it belongs to.

6. One image upload input box should be there for uploading the images of the people and the corresponding urls must be put in the in the 'member-image' box.
7. Please visit the url of the team page for your better understanding: http://staging.agrud.com/team
8. There can be multiple occurrences of the same web-page with different content in the same website, i.e Agrud.com can have 2 team pages.
9. There can be multiple occurrences of the same web-pages in different websites as well, i.e agrud.com and xyz.com both can have team pages.
10. Here is the overview diagram of how cms interface and web-page are integrated.

CAREER PAGE

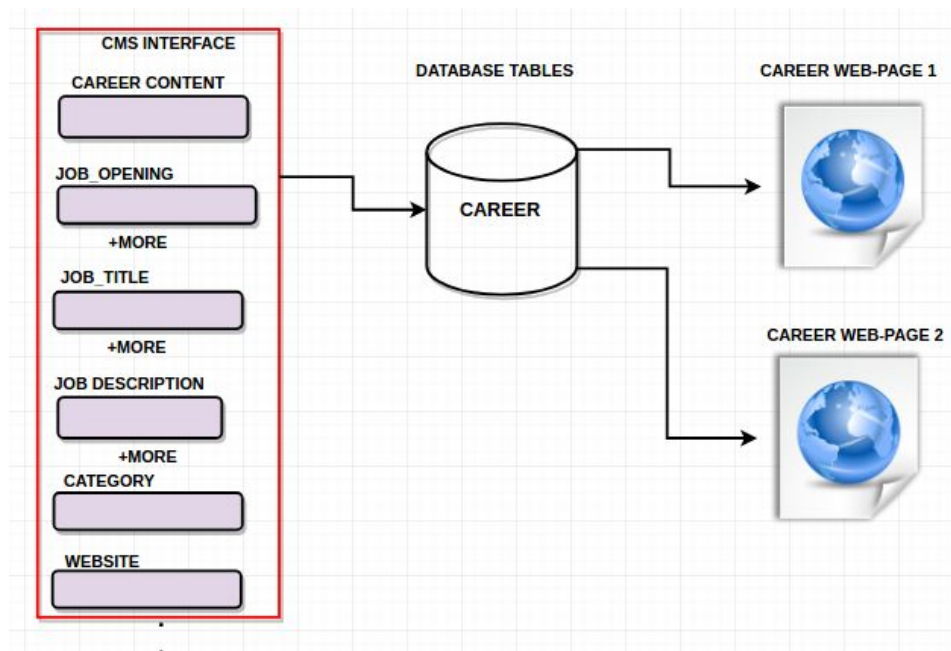1. The following image is the table structure for the career page.

Table: career

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Content	Longtext	Content of career

Job-openings	varchar	Job openings for our company
Job-title	varchar	Title for the job-details popup
Job-description	text	Description about the job

2. The interface should be created as such that the admin will be enter text only.
3. Please visit the url of the career page for your better understanding:
   http://staging.agrud.com/career
4. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
5. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
6. There can be multiple occurrences of the same web-page with different content in the same website, i.e Agrud.com can have 2 career pages.
7. There can be multiple occurrences of the same web-pages in different websites as well, i.e agrud.com and xyz.com both can have career pages.
8. There can be more than one job openings so the job details will enter the table separately in multiple rows.
9. The job details must be entered as next rows, not with the career content row.
10. Here is the overview diagram of how cms interface and web-page are integrated.

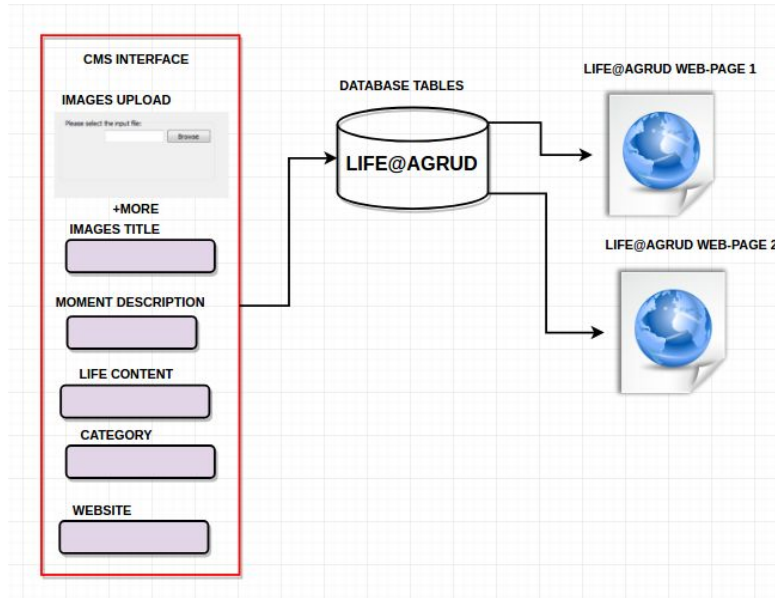# LIFE@AGRUD PAGE

1. The following image is the table structure for the career page.

Table : life@agrud

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Carousel images	Varchar	Images shown in web-page
Carousel title	Varchar	Title show beside the carousel
Carousel description	text	Description of the carousel
Life content	longtext	Content of the web-page

2. The interface should be consisting of text as well as images.
3. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
4. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
5. There can be multiple occurrences of the same web-page with different content in the same website, i.e Agrud.com can have 2 life pages.
6. There can be multiple occurrences of the same web-pages in different websites as well, i.e agrud.com and xyz.com both can have life pages.
7. The number of file uploads can be more than one for adding in the carousal facility in the web-page but the title and description should be a single.
8. It should be added to the placeholder of the image description box that the admin can add hashtags as well.
9. Please visit the url of the life page for your better understanding: http://staging.agrud.com/life
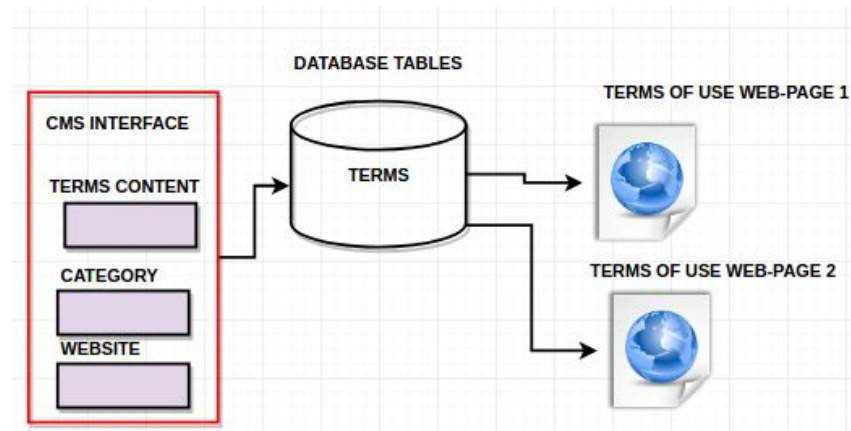10. Below is the overall integration diagram of the cms interface and the web-page.

TERMS OF USE PAGE

1. The following image is the table structure for the terms page.

Table : terms

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Content	longtext	Content of the terms page

2. The interface will be created as such, that the admin will be able to enter just text.
3. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
4. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
5. There can be multiple occurrences of the same web-page with different content in the same website, i.e Agrud.com can have 2 terms pages.
6. There can be multiple occurrences of the same web-pages in different websites as well, i.e agrud.com and xyz.com both can have terms pages.
7. Please visit the url of the vision page for your better understanding: http://staging.agrud.com/terms
8. Here is the overview diagram of how cms interface and web-page are integrated.
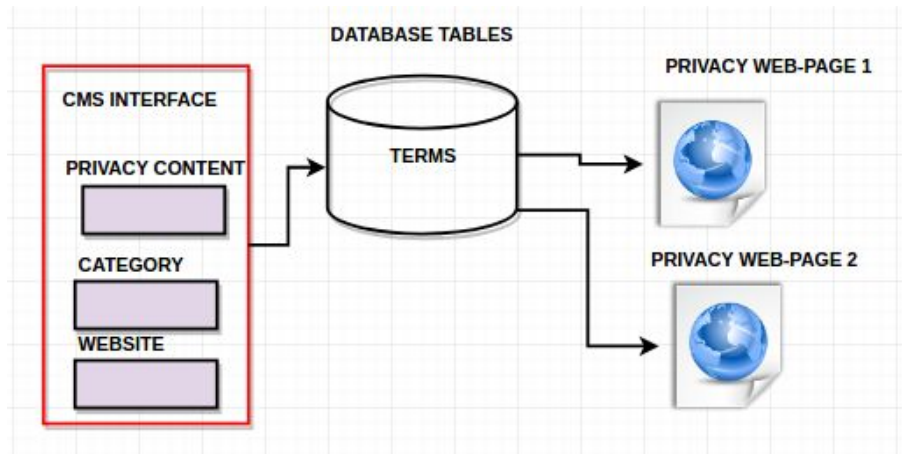
PRIVACY & COOKIES POLICY PAGE

1. The following image is the table structure for the privacy page.

Table : terms

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Content	longtext	Content of the privacy page

2. The interface will be created as such, that the admin will be able to enter just text.
3. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
4. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
5. There can be multiple occurrences of the same web-page with different content in the same website, i.e Agrud.com can have 2 privacy pages.
6. There can be multiple occurrences of the same web-pages in different websites as well, i.e agrud.com and xyz.com both can have privacy pages.
7. Please visit the url of the privacy page for your better understanding: http://staging.agrud.com/privacy
8. Here is the overview diagram of how cms interface and web-page are integrated.

PRESS-COVERAGE PAGE

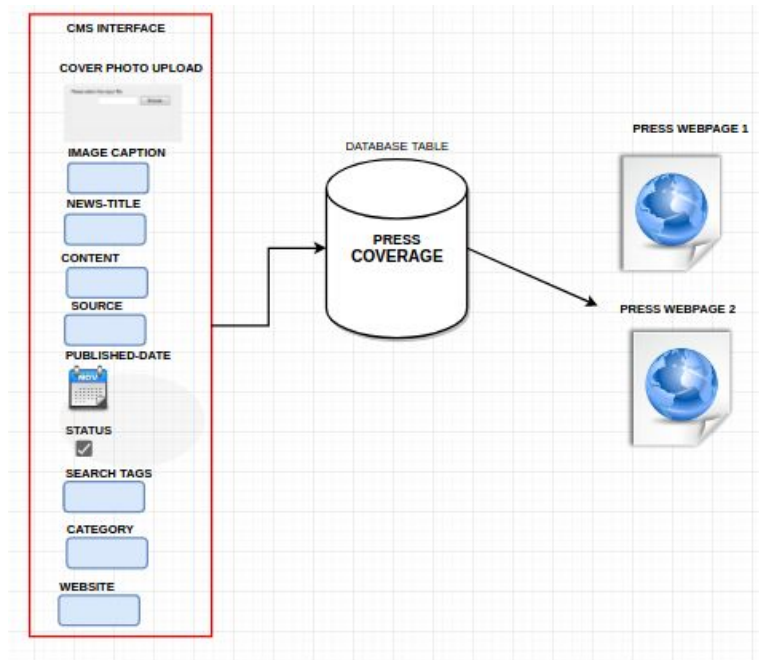1. The following image is the table structure for the press page.

Table : press

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Cover photo	varchar	Cover photo of the news
Photo caption	varchar	Caption of the image
News-title	varchar	Title of the news
Source	varchar	Source of the news
Published-date	datetime	Publishing date of the news
Content	longtext	Content of the news
Title_segment	varchar	Part of url to identify news
Status	char	D : draft, P : published, W : withheld

2. There will be two interfaces- one for entering new news and one for editing news.

3. The interface will be containing texts as well as images.
4. There can be 2 news with same title so the title segment should be such that both title-segments are not same.
5. The title_segment field should be defined as stripping all the spaces from the title and adding hyphen to its place, removing all special characters and adding a random character at the end. For example, if the title is 'The Difference Between Cash and Profit' then the title-segment becomes 'the-difference-between-cash-and-profits-01' and so on.
6. The title_segment should be generated and saved in the table when the news is saved.
7. When the news is edited, the title_segment should not change once created.
8. The status denotes the condition of blog. For example, when the admin is ready to publish the blog then he should select 'P', if the admin is yet to complete the blog then he should select 'D', if admin decides to not post the blog even though it's completed then he should select 'W'.
9. Below is the overall integration diagram of the cms interface and the web-page.

FAQ PAGE

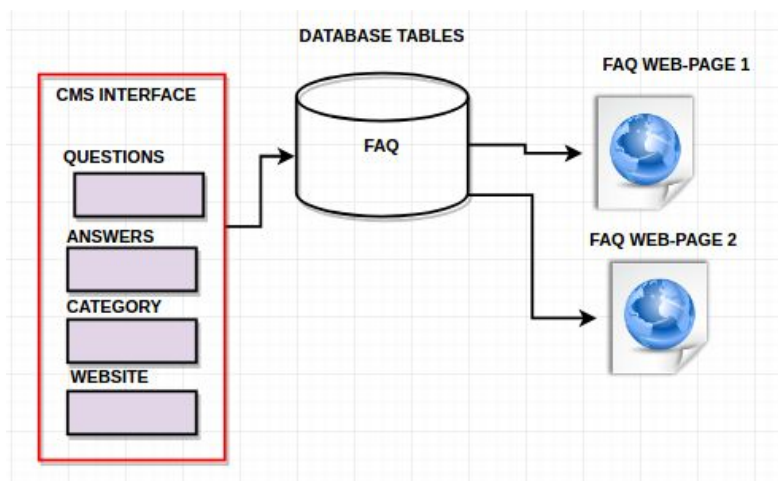1. The following image is the table structure for the FAQ-page.

Table: FAQ

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)

Category_id	Bigint	FK: category_master (id)
Questions	varchar	
Answers	longtext	Can contain text, images
Display-order	bigint	Display order of the questions and answers

2. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
3. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
4. In the answers input box, the admin can add images as well but those images won't be saved.
5. The admin will upload the image and get the created url and use that in the answer input-box.
6. So with the answer text, the image url gets saved as well in the answers field.
7. There will be two interfaces- one for creating FAQs and one for editing existing FAQs.
8. Below is the overall integration diagram of the cms interface and the web-page.

BLOG PAGE

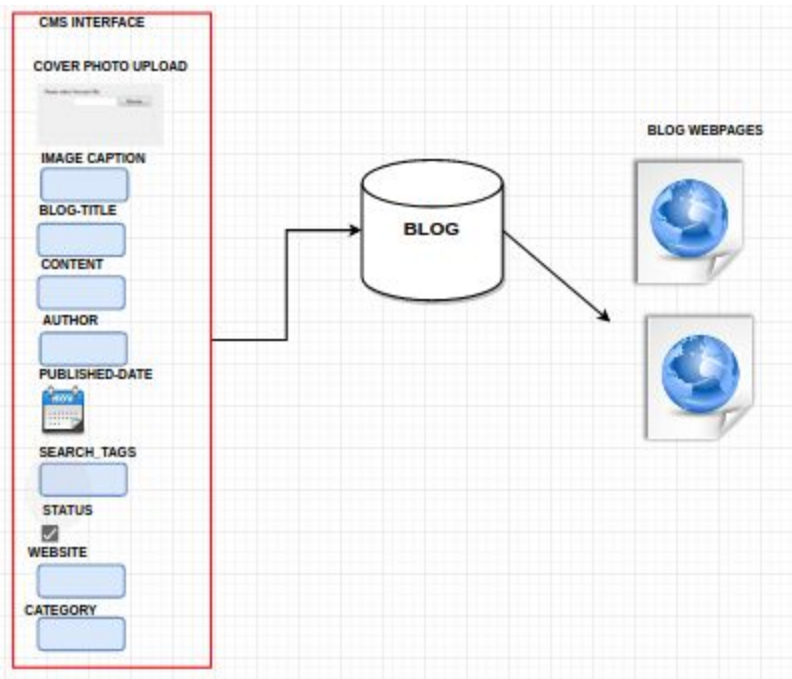1. The following image is the table structure for the blog-page.

Table : blog

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)

Category_id	Bigint	FK: category_master (id)
Cover photo	varchar	Cover photo of the blog
Photo caption	varchar	Caption of the blog
Blog-title	varchar	Title of the blog
Author	varchar	Author of the news
Published-date	datetime	Publishing date of the news
Modified-date	datetime	Date when modification is done
Content	longtext	Content of the blog
Search-tags	varchar	Tags to find the blog
Title_segment	varchar	Part of url to identify blog
Status	char	D : draft, P : published, W : withheld

2. There will be two interfaces- one for creating blogs and one for editing existing blogs.
3. There can be 2 blogs with same title so the title segment should be such that both title-segments are not same.
4. The title_segment field should be defined as stripping all the spaces from the title and adding hyphen to its place, removing all special characters and adding a random character at the end. For example, if the title is 'The Difference Between Cash and Profit' then the title-segment becomes 'the-difference-between-cash-and-profits-01' and so on.
5. The title-segment should be created once while creating new blogs but it should not change when those blogs are modified.
6. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
7. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
8. In the content input box, the admin can add images as well but those images won't be saved.
9. The admin will upload the image and get the created url and use that in the content input-box.
10. So with the blog content text, the image url gets saved as well in the content field.
11. The status denotes the condition of blog. For example, when the admin is ready to publish the blog then he should select 'P', if the admin is yet to complete the blog then he should select 'D', if admin decides to not post the blog even though it's completed then he should select 'W'.

12. Please visit these urls for your reference: http://sramana.news.agrud.com/blogs/page, http://sramana.news.agrud.com/blog/how-i-pivoted-my-startup-3-times-to-find-success
13. Below is the overall integration diagram of the cms interface and the web-page.

GUIDE PAGE

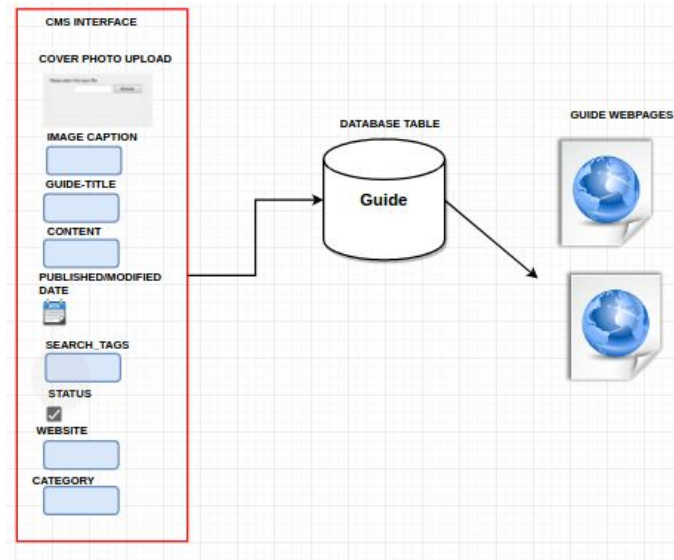1. The following image is the table structure for the guide-blog.

Table : guide

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Cover photo	varchar	Cover photo of the guide
Photo caption	varchar	Caption of the guide
Guide-title	varchar	Title of the guide
Published-date	datetime	Publishing date of the news
Modified-date	datetime	Date when modification is done
Search-tags	varchar	Tags to find the guide

Content	longtext	Content of the guide
Title_segment	varchar	Part of url to identify blog
Status	char	D : draft, P : published, W : withheld

2. There will be two interfaces- one for creating blogs and one for editing existing blogs.
3. There can be 2 blogs with same title so the title segment should be such that both title-segments are not same.
4. The title_segment field should be defined as stripping all the spaces from the title and adding hyphen to its place, removing all special characters and adding a random character at the end. For example, if the title is 'The Difference Between Cash and Profit' then the title-segment becomes 'the-difference-between-cash-and-profits-01' and so on.
5. The title-segment should be created once while creating new blogs but it should not change when those blogs are modified.
6. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
7. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
8. In the content input box, the admin can add images as well but those images won't be saved.
9. The admin will upload the image and get the created url and use that in the content input-box.
10. So with the guide content text, the image url gets saved as well in the content field.
11. The status denotes the condition of guide. For example, when the admin is ready to publish the blog then he should select 'P', if the admin is yet to complete the blog then he should select 'D', if admin decides to not post the blog even though it's completed then he should select 'W'.
12. Please visit these urls for your reference: http://sramana.news.agrud.com/blogs/page, http://sramana.news.agrud.com/blog/how-i-pivoted-my-startup-3-times-to-find-success
13. Below is the overall integration diagram of the cms interface and the web-page.

TECHNOLOGY PAGE

1. The following image is the table structure for the blog-page.

Table : Technology

Id	Bigint	Unsigned auto_increment
Website_id	Bigint	FK: website_master (id)
Category_id	Bigint	FK: category_master (id)
Cover photo	varchar	Cover photo of the technology page
Photo caption	varchar	Caption of the technology page
Technology-title	varchar	Title of the technology page
Author	varchar	Author of the technology page
Published-date	datetime	Publishing date of the technology page
Modified-date	datetime	Date when modification is done
Content	longtext	Content of the technology page

Search-tags	varchar	Tags to find the parts of contents
Title_segment	varchar	Part of url passed to identify parts of content
Status	char	D : draft, P : published, W : withheld

2. There will be two interfaces- one for creating technology content and one for editing existing technology page.
3. There can be 2 technology contents with same title so the title segment should be such that both title-segments are not same.
4. The title_segment field should be defined as stripping all the spaces from the title and adding hyphen to its place, removing all special characters and adding a random character at the end. For example, if the title is 'The Difference Between Cash and Profit' then the title-segment becomes 'the-difference-between-cash-and-profits-01' and so on.
5. The title-segment should be created once while creating new technology contents but it should not change when those technology contents are modified.
6. The category_id comes into play when the admin wishes to have many similar web-pages in same website.
7. If the user chooses to have 2 similar pages, he should choose which category it belongs to.
8. In the content input box, the admin can add images as well but those images won't be saved.
9. The admin will upload the image and get the created url and use that in the content input-box.
10. So with the technology contents text, the image url gets saved as well in the content field.
11. The status denotes the condition of technology contents. For example, when the admin is ready to publish the blog then he should select 'P', if the admin is yet to complete the blog then he should select 'D', if admin decides to not post the blog even though it's completed then he should select 'W'.
12. Below is the overall integration diagram of the cms interface and the web-page.

CMS INTERFACE

COVER PHOTO UPLOAD

IMAGE CAPTION

TECH-TITLE

CONTENT

AUTHOR

PUBLISHED-DATE

SEARCH_TAGS

STATUS

WEBSITE

CATEGORY

DATABASE TABLES

TECH-NOLOGY

TECHNOLOGY WEBPAGES