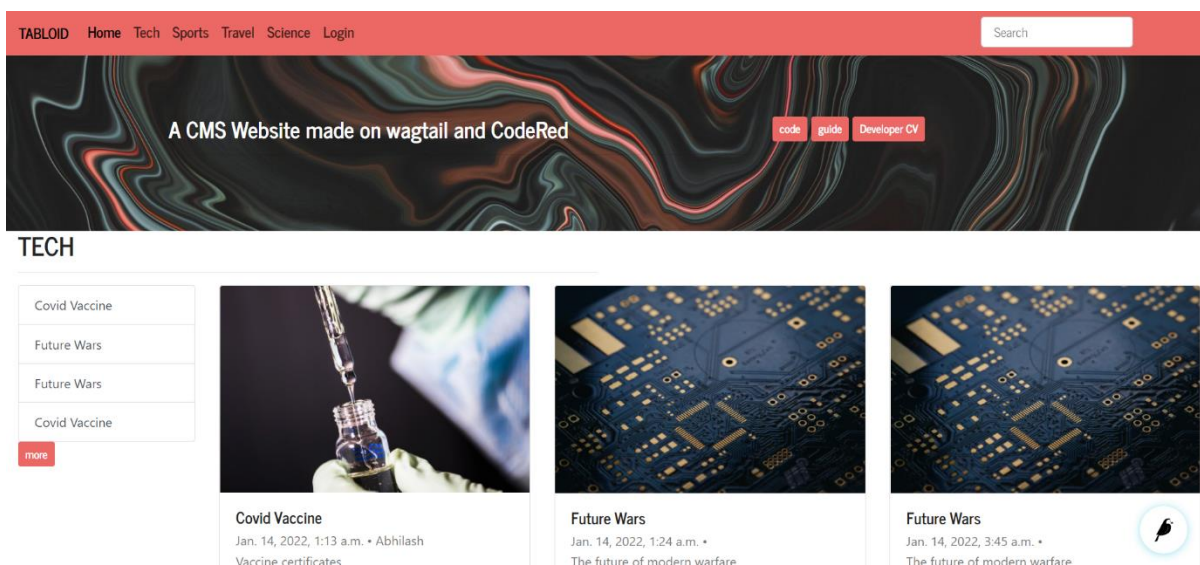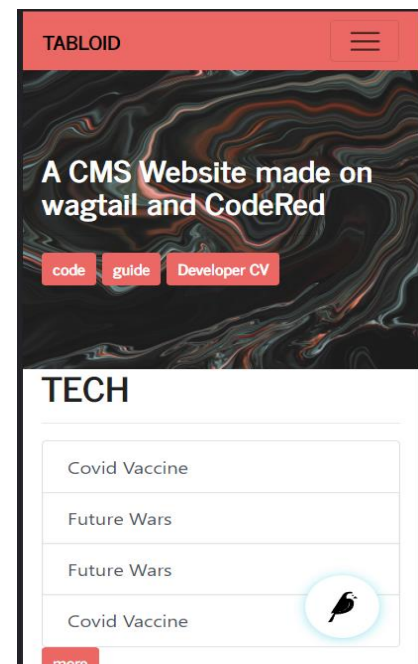# TABLOID CMS

The website is developed to overcome the problems facing in a regular cms website with core frame work of built on top of django and bootstrap.

The application takes care of the editing of your cms content by the use of richtextfield(in built text editor) and a system of user groups involving Editors and moderators to manage the content publication and management.
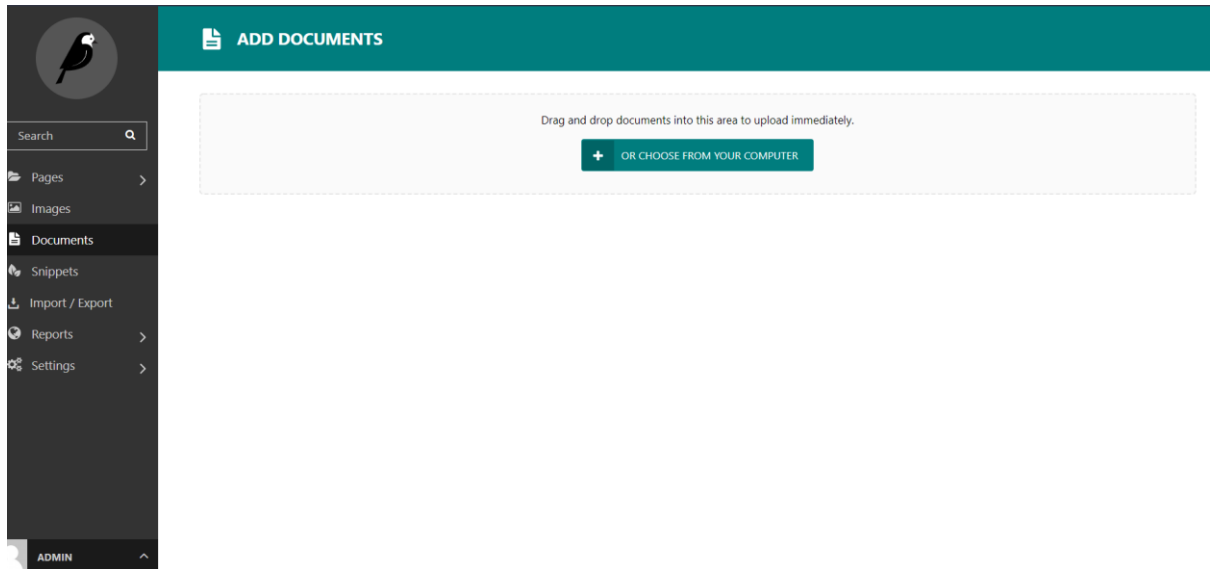
Software involved in the project:

- Wagtail cms
- Codered cms
- Django
- SQL
- Bootstrap4
- Javascript
- HTML5
- CSS
- Hosting AWS EC2 instance, ngnix, gunicorn

The Features of the Website

Download Block

The download block enables users to add documents to the CMS which website visitors can download from the site.

Embed Media Block

The embed media block allows users to embed media like YouTube videos or SoundCloud with just a link to the external media source.

Field Reference

Fields and purposes:

- **URL** - The URL for the external media, whether it's a tweet, Facebook post, YouTube video, etc.

Display publish date:

Google Map Block

Allows the user to add a Google Map location and display the location on the map.

Field Reference

Fields and purposes:

- **Search Query** - Address or search term used to find your location on the map.
- **Map Title** - Map title for screen readers, ex: "Map to Simple Sweet Desserts"
- **Google Place ID** - Place IDs uniquely identify a place in the Google Places database and on Google Maps
- **Map Zoom Level** - Requires API key to use zoom. ex: 1: World, 5: Landmass/continent, 10: City, 15: Streets, 20: Buildings
- **Advanced Settings** - Add custom CSS classes or a CSS ID to style the block with your custom CSS

Google Map

ADVANCED SETTINGS

Search query

Public Square, Cleveland, OH

Address or search term used to find your location on the map.

Map title

Public Square Offices in Cleveland

Map title for screen readers, ex: "Map to Goodale Park"
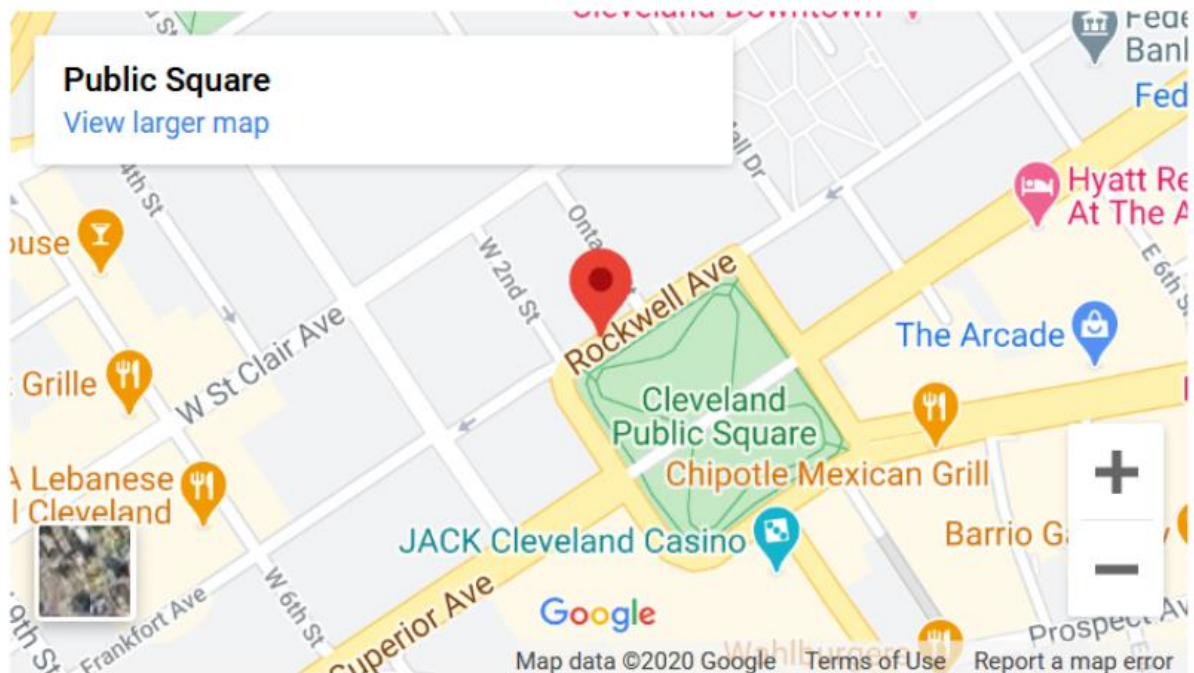
Google place ID

Google place ID

Requires API key to use place ID.

Map zoom level

14

Requires API key to use zoom. 1: World, 5: Landmass/continent, 10: City, 15: Streets, 20: Buildings

Article Pages

Create a blog

Usage

First start by creating an "Article Landing Page" and then add any number of "Article Page"s as children of the "Article Landing Page". Each child page here represents an individual article. Add content to these pages as you would for a normal "Web Page".
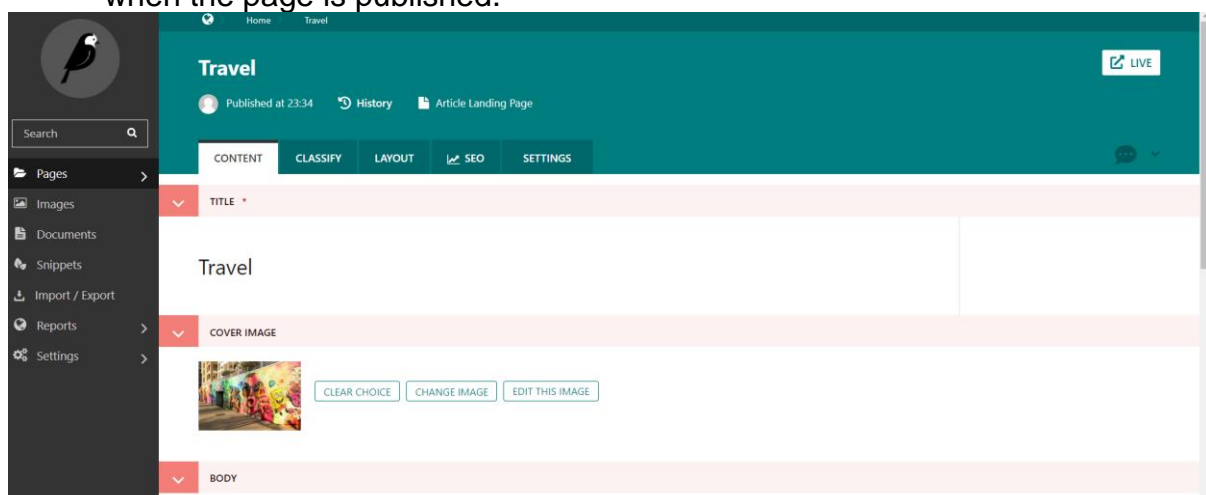
Article Landing Page

Layout Tab
- **Show Images**: If toggled on, this landing page will show the cover images for its children articles.
- **Show Captions**: If toggled on, this landing page will show the captions for its children articles.
- **Show Author and Date Info**: If toggled on, this landing page will show the captions for its children articles.
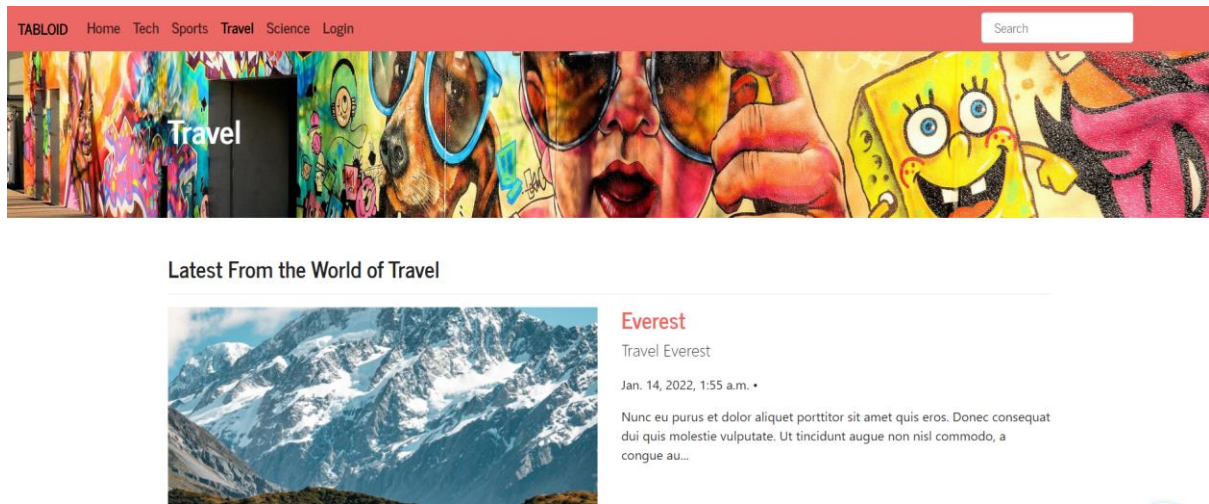- **Show Preview Text**: If toggled on, this landing page will show a preview of its children's content.

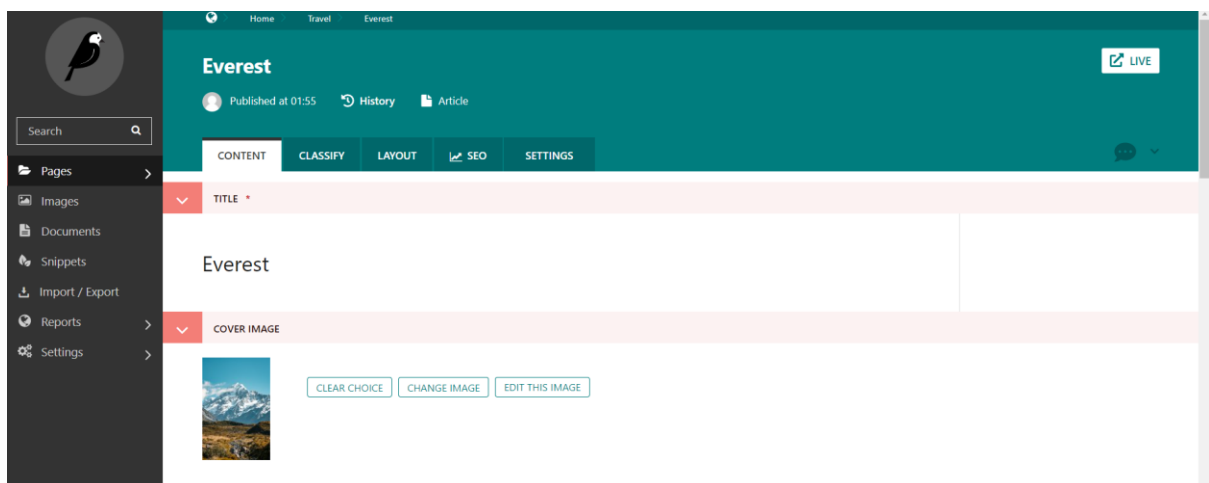## Article Page

Content Tab
- **Caption**: The caption you want to be displayed under your article's title.
- **Author**: The user you want to tie the article to.
- **Display Author As**: The name you want the author to appear as. This will default to the author's name if not defined.
- **Display Publish Date**: The publish date you want to be shown on your article. This is separate from the actual publish date which is determined when the page is published.
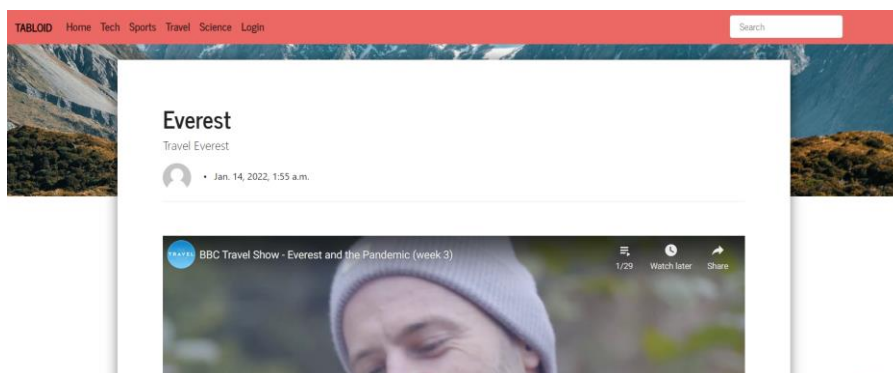
The fontend of the the article landing page above

Article Page

Front end

Below is a table of the current possible "Parent-Child" relationships.

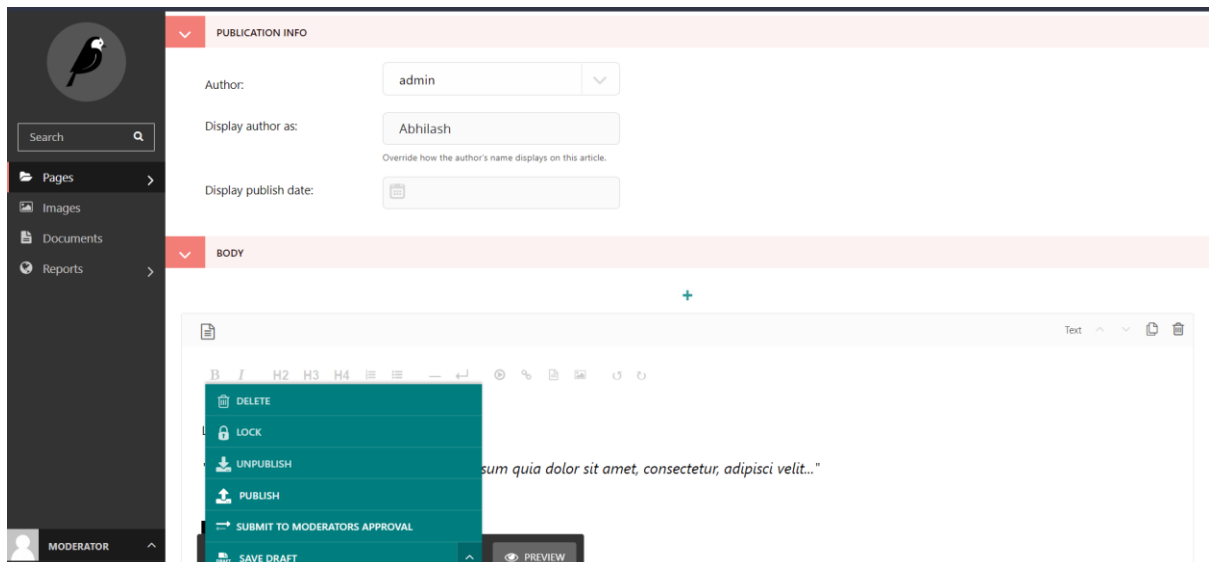| Parent Page Type	Child Page Types
Web Page	Web Page, Article Landing Page, Event Landing Page, Location Landing Page, Form Page
Article Landing Page	Article Page
Event Landing Page	Event Page
Location Landing Page	Location Page

User Structure in the website

ADMIN--→ Has all the root access can create all the above mentioned pages from the admin interface and create groups for users to grant and deny access to the website

Editors groups --→ user under this can create Article pages can draft it or send it to moderation to moderators in this particular website for publishing

Moderators-→ Group can look into the workflow of the website and reports to see any article moderation and set a time for the article to go live in Web

In the above is the admin panel for the editor you could see that the editor can only save the page and send for moderation towards the moderator. This feature can be edited by the admin if necessary This feature allows the admin to manage multiple users and custom user groups through out the website with ease below is the image of moderator's admin panel you can notice the publish and lock feature which editors doesn't have.
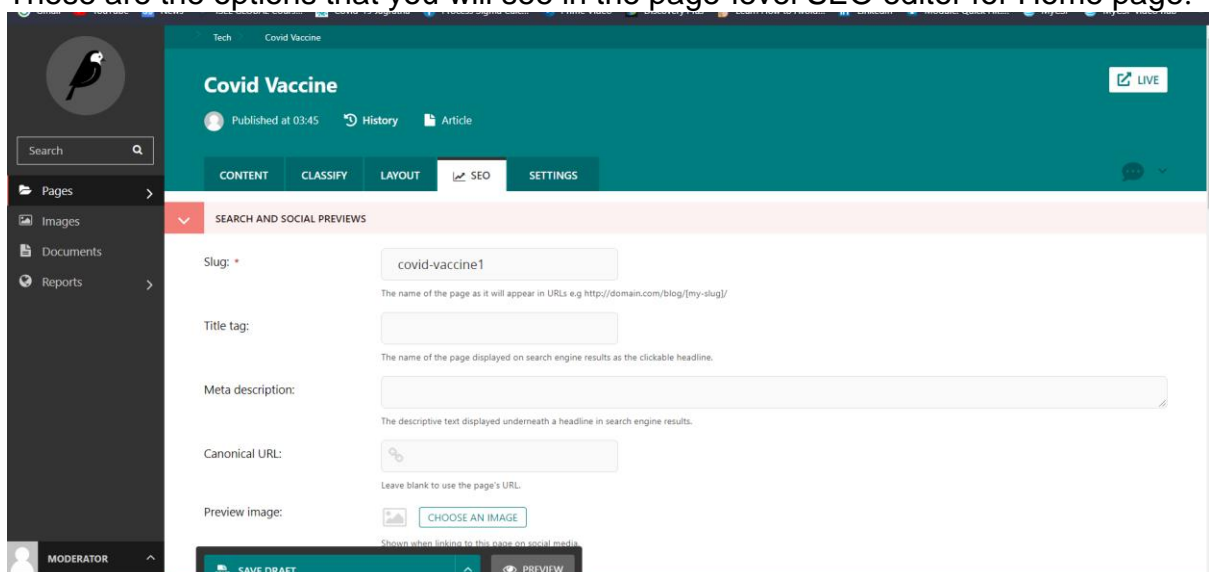

SEO enabled

SEO settings for best practices are already selected for you, but you can change this if you would like.

Adding SEO metadata to your pages

Now you will want to add SEO metadata to your pages in the CMS. This can be easily done from the page editor. In the admin, navigate to the page that you want to add SEO, click the Edit option, and go to the SEO tab at the top of the page editor. These are the options that you will see in the page-level SEO editor for Home page:

REFERENCES

https://docs.wagtail.io/en/stable/
https://docs.djangoproject.com/en/4.0/