Custom blog design

Front end technologies:

- CSS
- HTML
- Jinja2
- jQuery
- Ajax
- Javascript

Backend technologies:

- Python
- Django web framework:

Deployment technologies:

- Heroku – used to deploy the website
- GitHub – used as a repository

Git hub repository - https://github.com/inacceptable/custom-blog-design

Heroku link - https://custom-blog-design.herokuapp.com/

Website breakdown description

This website was created to act as a custom website design. The website features 2 main parts: The first part is a home page which contains a header, additional sections, clickable topics, and posts related to the topic. The second part is a custom admin page which lets you change the colors and content of the main website (it also includes features to allow you to create new sections, new topics and new posts as well as allow you to update existing entries). The blog posts on the website are HTML markup which allows endless designs to a content in the posts.

Feel free to play around with the website. I've included 2 links below. The first link is the link to the websites home page (which would be the interactive part for users) and the second link is the admin page which will let you customise and add additional topics, sections and posts to the website. As well as change the colors for objects on the website. I'm sure features have been missed and this could use more work, however this was a quick way to demonstrate integration of jQuery, CSS, html, and Ajax to interact with my python Django database.

I personally don't think that the UI of the website is the greatest and the layout could be changed to be more user friendly and a lot more customised. My goal with this website was to create an interface which allows other sections within the website to be customised based off an admin page within the website. Using jinja embedded into html files this allows features such as the color changing to occur. Which I think could be a useful way to create a custom solution for different websites (not just this one).

https://custom-blog.herokuapp.com/

https://custom-blog.herokuapp.com/controladmin

Home page

The home page is broken down with the below sections:

- Navigation bar
- Page header (includes page header name and slogan)
- Additional sections: These sections have been created to allow you to upload an image, change the section title and the section description. There is also a button which links to a post (this part acts as almost a feed for the website).

The home page extends a template.html file I created. This file just contains some metadata information as well as links to CSS, bootstrap, and jQuery files. It extends the navigation.html file I created which would be at the top of each webpage.

Home page header:

The home page header is broken down into three sections (mainly for colour customisation). This includes:

- Top header (you can change the colour for this section)
- A header title section: this just includes the header title and a background colour (both can be changed)
- A header slogan section: This includes the header slogan which you can change as well as the background colour for the slogan

HTML and CSS were used to display the header on the page and Jinja was used to pull information from the model database to have the user selected colour displayed.

Additional sections

An additional section object is broken down into the below items:

- A border: this border's colour changes based on user selection using Jinja
- Background colour: This changes using Jinja based on user input
- Section title: This can be changed based on what the user wants it to say, and the colour of this section can also be changed
- Section content: This will be part of the additional section object. It is a small description within the section which will also be changed based on what the user wants it to say in the control admin page of the website
- Link to a post: this is a button which the user will be allowed to change both the colour of the button as well as the font colour of the button. This is an easy way to integrate into the main post items from the home page and the user

You can add multiple additional items to the website as I have used a CSS grids to make the them look better when a new one is added to the page, rather than having to style a new section repeatedly.

Navigation bar

The navigation bar is broken down into 2 sections (a menu that when clicked on shows topics, and a social section which will allow the user to click on Facebook, Reddit, and Twitter icons to share the website to social media)

- The social buttons are static, so they just allow you to share the main website as this is the main sharable item on the website (this was created using html and CSS code to).
- There is a home button on the navigation bar which uses html and Jinja to go back to the websites home page
- Topics: topics are revealed when the user clicks on the menu drop down, new ones can be added and the name for all of these can be changed.
    - When a topic is clicked on it uses jQuery to reveal a drop down with the topic description as well as 2 navigation buttons to check out more posts from the topic as well as a close button to close the drop down
- Items that can be colour changed in the navigation bar: menu drop down button, the background of the navigation bar container, the check out more button and close button background colour and font colour.

View topic page

The view topic page is broken down into the below sections:

- A navigation bar
- A page header
- Posts related to the topic

The view topic page extends the template.html so the navigation bar hasn't changed from the home page as well as the page header (these are the same as the home page). The main item that is different are the posts related to the topic which contain the below items:

- A post heading
- A post description
- A link to read the rest of the post

CSS was used to style each of the posts for the topic, so it looks fine on the page.

View post page

The view post page is broken down into the below sections:

- The navigation menu
- Post container

The post container is based from the user input in the control admin page of the website. The container for the post uses CSS styling to keep items with the page. The built in Django safe function was used to render the user input as html. This will allow posts to be easy customisation and allow control for users to change what appears in the post (i.e., if they want to add an image they can).

Control admin page

The control admin page is broken down into the below sections:

- Customise page colours sections. This is further broken down into:
    - Navigation menu
    - Header
    - Additional sections

- Customise header section. This includes
    - Header title
    - Header slogan
- Customise additional sections. This includes:
    - Add a new section
    - Update existing sections
- Manage topics. This includes:
    - Add a new topic
    - Update existing topics
- Add and manage posts. This includes:
    - Add a new post
    - Update existing posts

The items in this page use a mix of jQuery for the functional clicking of buttons (i.e., expand/closing dropdowns). Jinja to represent the items from Django database and Ajax to stop the page from refreshing when you upload an image.

Small conclusion

When I deployed this website to Heroku, unfortunately it doesn't let me keep media that has been uploaded for long (i.e., unfortunately I didn't page for the hosting so it's free and they will disappear after a little bit of time).

There are other design elements of the website and features that can be added. However, in this website I just wanted to demonstrate the use of the technologies to make it functional.