

Some of the Django projects I worked on

1	Music on repeat	1
2	WebSocket stream with binance api	3
3	Django app to upload and download files to/from azure storage	4
4	Fika store, django web shop	4
5	Blog app	6
6	Couple of projects on Codepen	7

1 Music on repeat

Github repo : Private repository, still working on it when have spare time.

Hosting : <https://bojan.pythonanywhere.com/>

This is the most complex django app that I created so far. It has a lot of components. As a user you can search for a song (using youtube api), from a given search result you can add a song to your playlist and then play it on repeat mode. I built this for my personal user because when I listen some music I like to listen the same composition over and over again. The project is still in development even though the main components are finished. It is very popular, at least within my family. 😊

There are several django apps within this project.

Main app

This app handles all the python logic regarding user login, logout, creation of an account, having the base template for the whole project. I used just a tiny bit of bootstrap and build things mostly on my own using plain CSS.

Signup form: Users can create an account and for passwords I'm using hashing algorithms so that the password is never stored as a plain text in the db. I also defined some password/username validations.

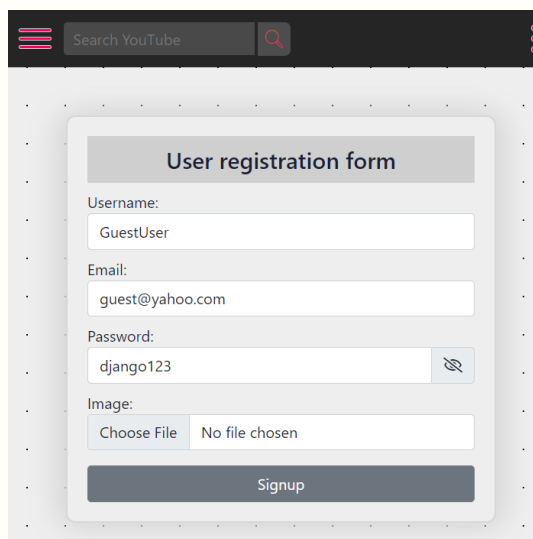
A screenshot of a web browser showing a 'User registration form'. The form is titled 'User registration form' and contains fields for 'Username:' (with 'GuestUser' entered), 'Email:' (with 'guest@yahoo.com' entered), 'Password:' (with 'django123' entered and a toggle icon), and 'Image:' (with a 'Choose File' button and 'No file chosen' text). A 'Signup' button is at the bottom.

Figure 1: Signup form

Login form: For the password show/hide button I wrote a short JS function that will basically toggle image src for those two icons.

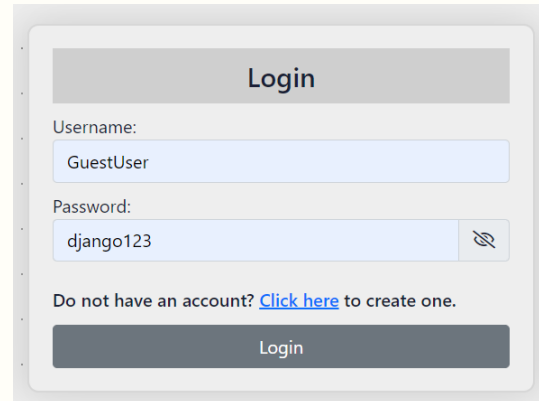
A screenshot of a web browser showing a 'Login' form. The form is titled 'Login' and contains fields for 'Username:' (with 'GuestUser' entered) and 'Password:' (with 'django123' entered and a toggle icon). A link 'Do not have an account? Click here to create one.' is below the password field. A 'Login' button is at the bottom.

Figure 2: Signup form

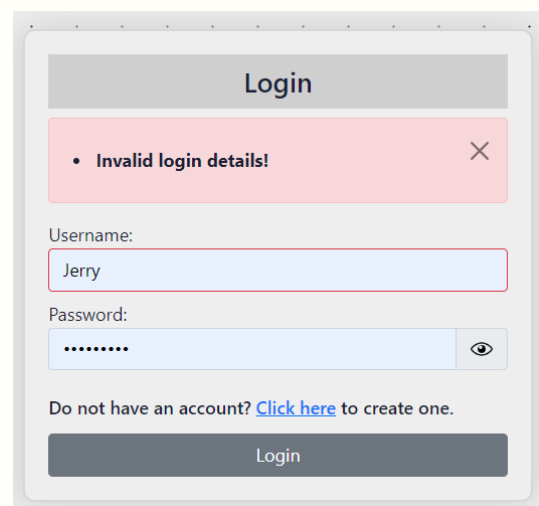
A screenshot of a web browser showing a 'Login' form with an error message. The form is titled 'Login' and contains fields for 'Username:' (with 'Jerry' entered) and 'Password:' (with '.....' entered and a toggle icon). A red error message 'Invalid login details!' is displayed above the password field. A link 'Do not have an account? Click here to create one.' is below the password field. A 'Login' button is at the bottom.

Figure 3: Handling error.

After login, user is redirected to the index page. As you can see the index page still requires some work, but that is practically only some styling, nothing more.

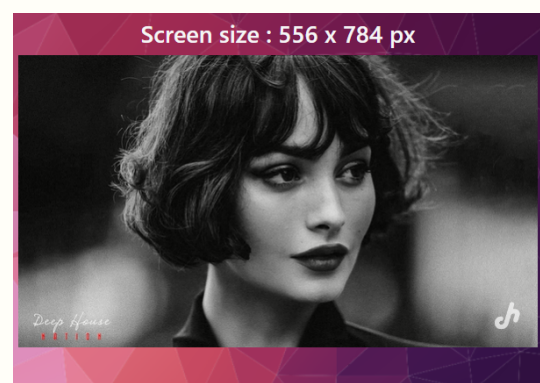


Figure 4: Snippet of the index page

Updates

This app is only for my own use and I created it just to keep track of some updates. Whenever I add certain functionality to the project I would add an item in this part of the database. It is fully functional with complete CRUD operations and for this I've been leveraging AJAX with HTMX, no page refresh is needed. 😊

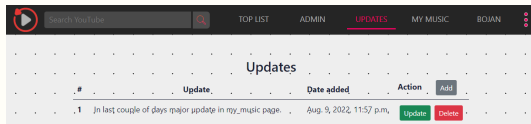


Figure 5: Updates page

YouTube app

This is the major app of the project. It handles all the python logic and templates regarding the user experience.

Once user add a song/video to their playlist, they can play it, add it to the list of favorites, remove completely from the playlist.

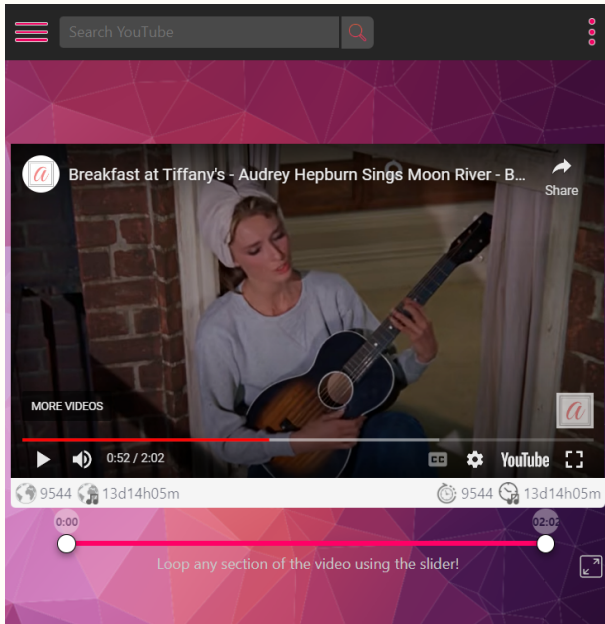


Figure 6: Snippet of the My Music page.

The My Music page contains a list of videos that user has added to the playlist.

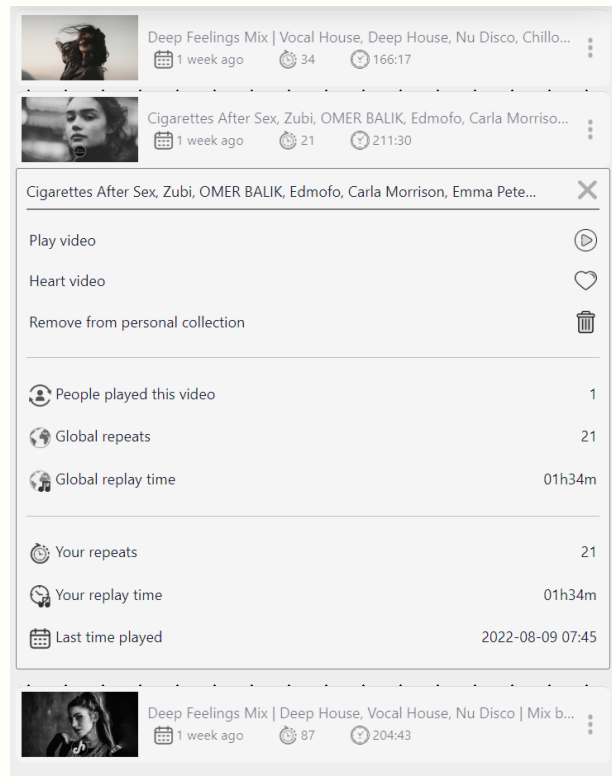


Figure 7: Snippet of the music list.

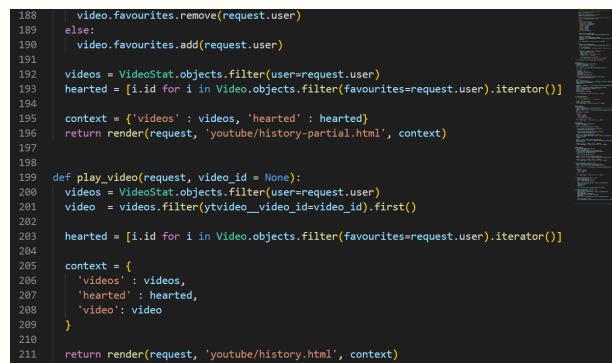


Figure 8: Code snippet.

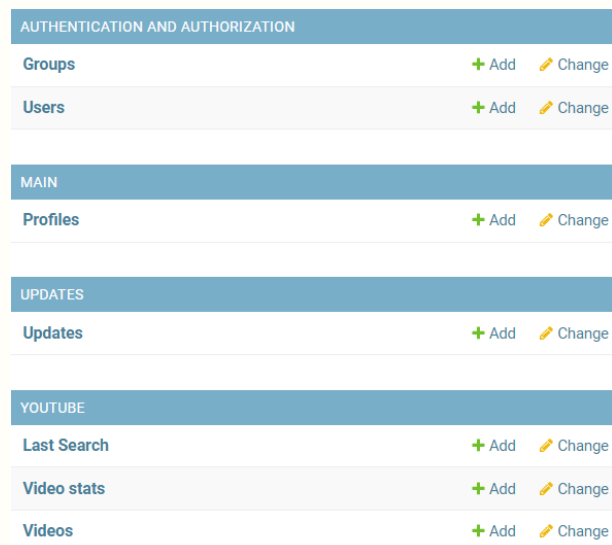


Figure 9: Admin panel.

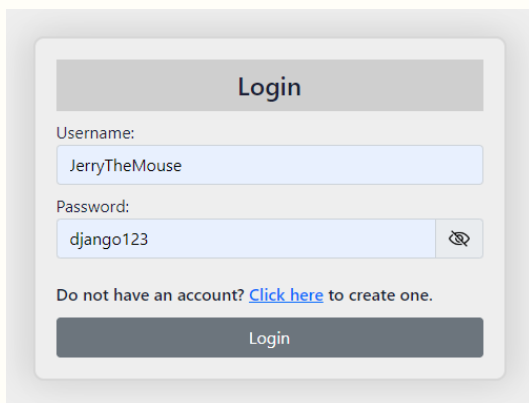
2 WebSocket stream with binance api

Github repo : https://github.com/bojan-vujic/WebSocket_Binance

Hosting : <https://binancestream.pythonanywhere.com/>

In this project I build a django app that will monitor real time price of cryptocurrencies. It consists of two apps, one called main that handles all the user logic and another one bnc that handles the rest of the work. Some django models have been created as well. It was a nice challenge since I haven't been working with finance data before, but at the end of the day these are just numbers as any other and I'm very comfortable with it. In my humble opinion this app is only a couple steps away from an autonomous trading bot and I will continue working on it.

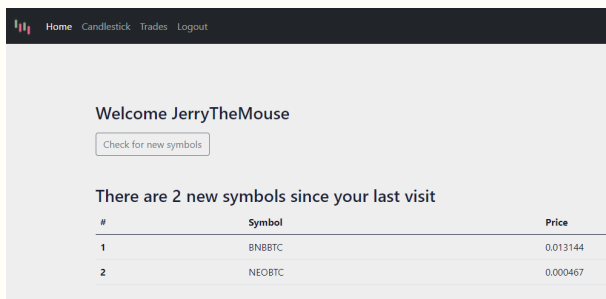
As a user you can check for new cryptocurrency symbol:



The login form has a title 'Login' in a grey box. Below it are fields for 'Username:' with the value 'JerryTheMouse' and 'Password:' with the value 'django123'. There is a 'Login' button at the bottom. A link 'Click here' is present next to the text 'Do not have an account?'.

Figure 10: Login system

Once you are logged in it will redirect you to the index page where you can check for a new symbols.



The index page shows a welcome message 'Welcome JerryTheMouse' and a button 'Check for new symbols'. Below it, a message says 'There are 2 new symbols since your last visit'. A table lists these symbols:

#	Symbol	Price
1	BNBBTC	0.013144
2	NEOBTC	0.000467

Figure 11: Check for new symbols

On the Candlestick link user can monitor the real time price for a given cryptocurrency. For

these charts I used lightweight JavaScript library and it's very nice and intuitive thing.

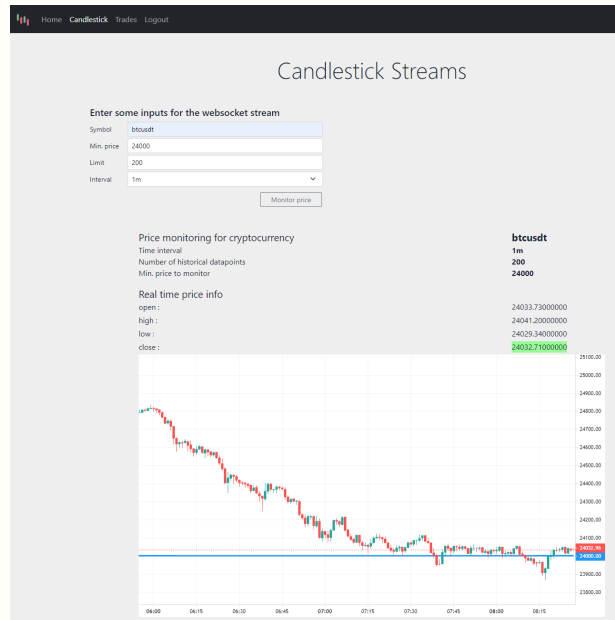


Figure 12: Candlestick stream

Finally, user can monitor price in a real time and also compare it with user defined minimum price. For instance if the current price is above the minimum, we can start trading.

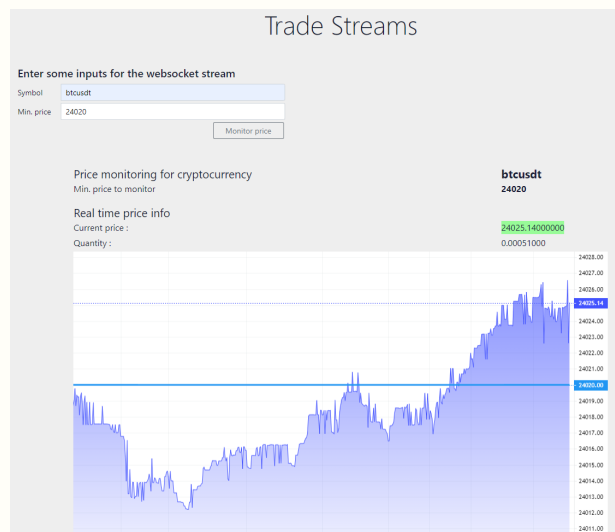


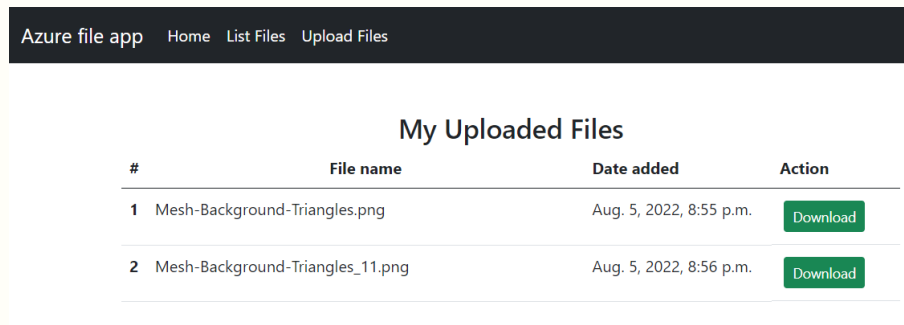
Figure 13: Trade stream

3 Django app to upload and download files to/from azure storage

Github repo : <https://github.com/bojan-vujic/DjangoAzure>

Hosting : <https://djangoazure.pythonanywhere.com/>

I started this app originally as a service to upload and download the files from azure web storage. However, since I no longer have an Azure account, I modified the app, so it only works as a service to upload and download files to/from web server where the app is being hosted. It was a nice challenge, but at the end of the day rather simple thing to do.



Azure file app			
Home List Files Upload Files			
My Uploaded Files			
#	File name	Date added	Action
1	Mesh-Background-Triangles.png	Aug. 5, 2022, 8:55 p.m.	Download
2	Mesh-Background-Triangles_11.png	Aug. 5, 2022, 8:56 p.m.	Download

Figure 14: Snippet of the app

4 Fika store, django web shop

Github repo : https://github.com/Lexicon-Group-2/Coffee_shop

Hosting : <https://fikaproject.pythonanywhere.com/home/>

In this app I worked as a group coordinator in a group of 4 people. My task was to write all the python functionalities and also a front-end css styling of the Store, Cart and User pages.

You can login using these credentials:

Username : **Jerry**,

Password : **django123**

The store page contains four categories and upon click it will redirect to a particular category, coffee machines, coffee beans, coffee mugs and accessories.

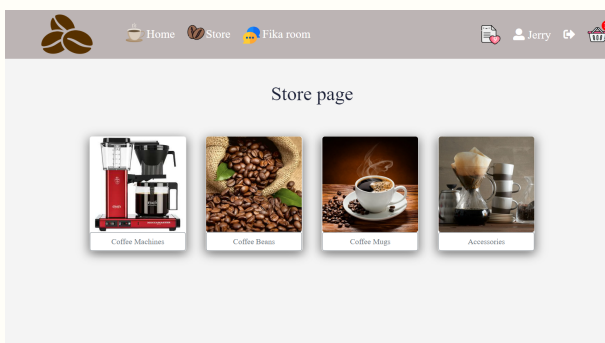


Figure 15: Store page

Each category contains several items which are stored in a database under particular model and

then using displayed in a template using django functionalities.

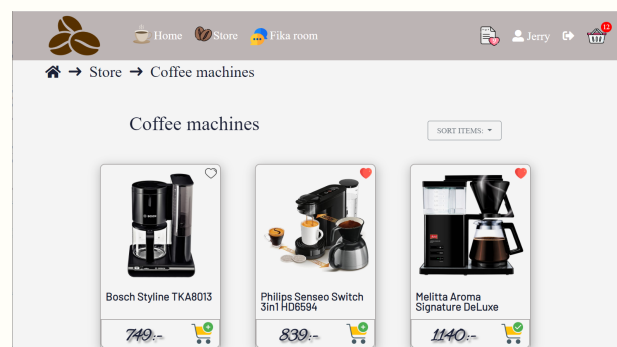


Figure 16: Coffee machines

I also implemented option to add an item into the list of favorite items and user can see all the favorites under favorite page. In the navbar section I added a small div containing number of items in the favorite page and also another one showing number of items in the shopping cart.

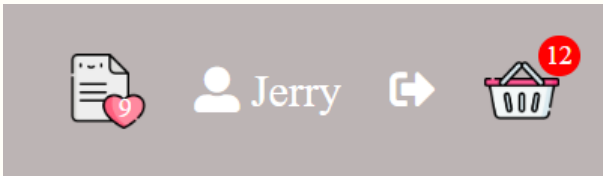


Figure 17: Section of the navbar

Shopping cart page contains all the items that user has added. There are standard options to increase/decrease number of items or remove items from the cart. The price is updated accordingly. Due to time constraint we stopped our project here, but later on I continued working on it, implementing various sort options in the cart page and also implementing PayPal paying options.

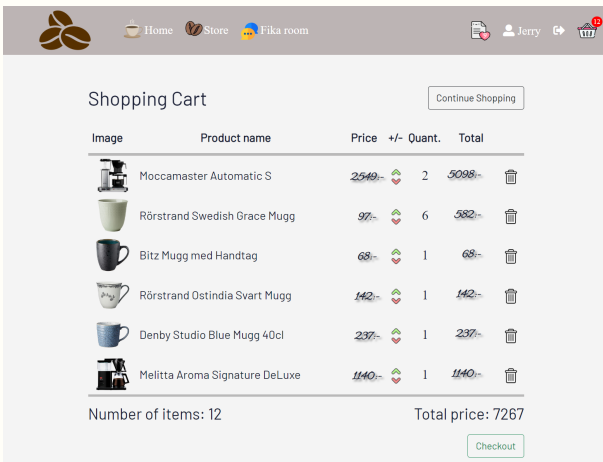


Figure 18: Store page

User page contains all the shopping history of the current user.

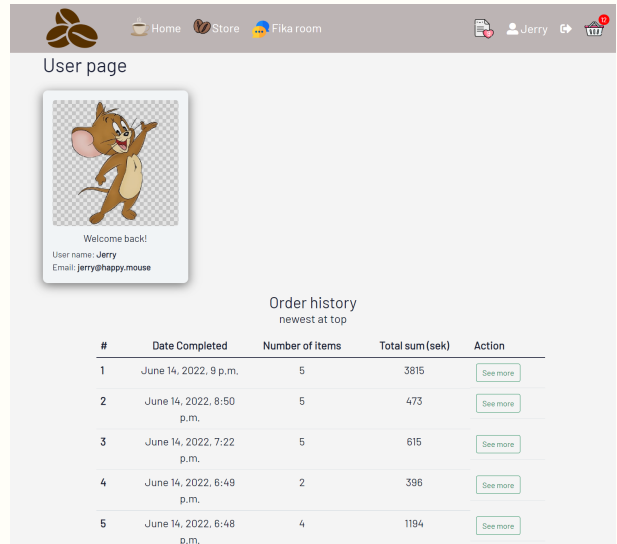


Figure 19: User page

Upon clicking onto button **Show more**, a new page will appear. It contains all the product that user has purchased in that particular session. My job was to write all the python logic, html templates and css styling for this pages.

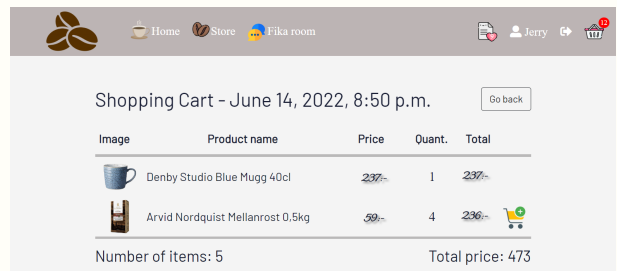


Figure 20: User page

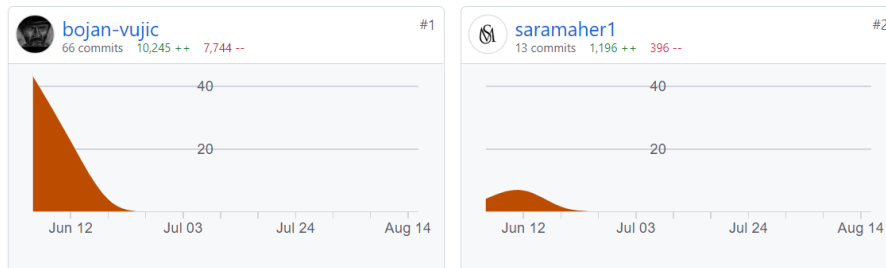


Figure 21: Github commits for this project

5 Blog app

Github repo : <https://github.com/Lexicon-Group-2/Blog-project>

Hosting : No hosting is provided

This django application has been done in a collaboration with four people and I was the coordinator of the group. We had only 3 days to work on it and it was a nice exercise. My task was to write all the python logic for the project and organize the things in a templates. I also did some css styling.

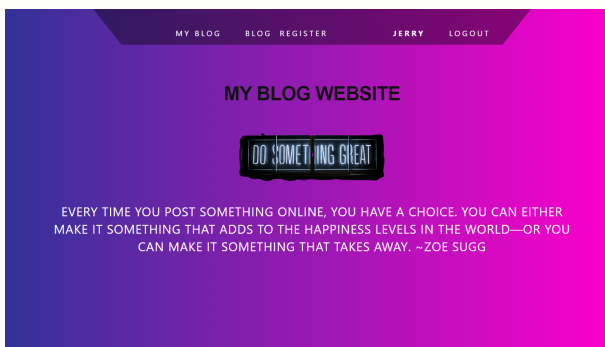


Figure 22: Landing page

User have options to write a post:



Figure 23: Blog page

Upon clicking onto link **Read more**, a detail page is opened from where user can comment the post.

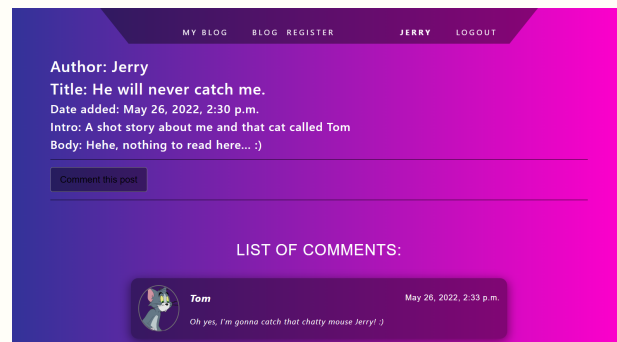


Figure 24: Blog page

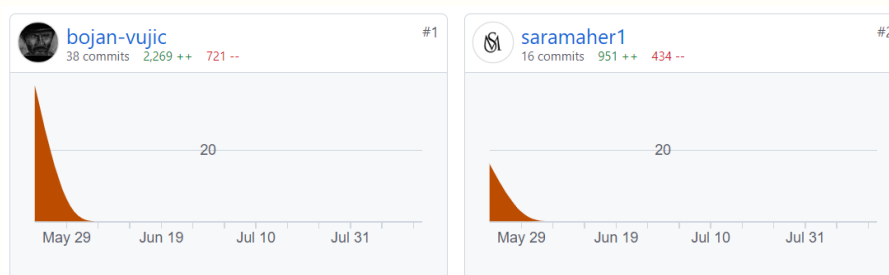


Figure 25: Github commits for this project

6 Couple of projects on Codepen

Github repo : No github repository

Hosting : <https://codepen.io/bojan-vujic>

There are several projects and tests here on my codepen profile.

JavaScript calculator:

<https://codepen.io/bojan-vujic/full/rNJVmGq>

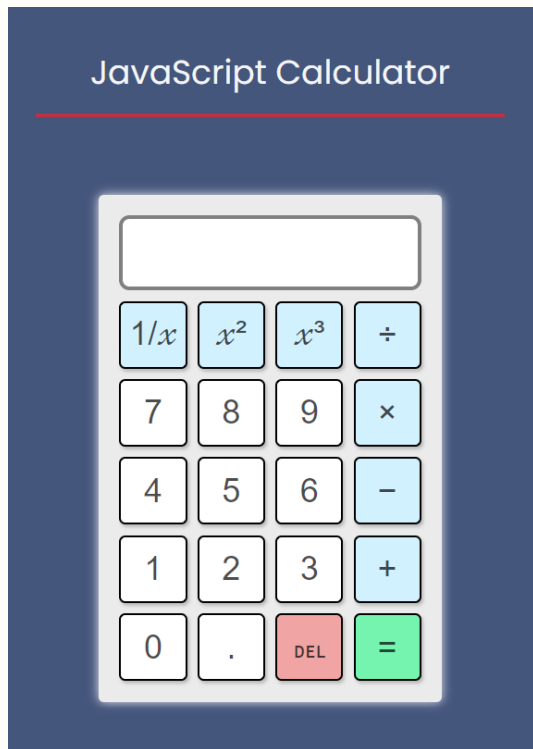


Figure 26: Calculator

Number to roman converter:

<https://codepen.io/bojan-vujic/full/RwQNQWW>

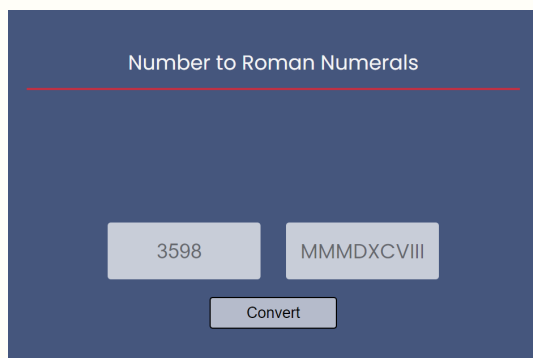


Figure 27: Number to roman numeral

JavaScript Tic-Tac-Toe. My daughter and I enjoyed playing it:

<https://codepen.io/bojan-vujic/full/xxYwJQV>



Figure 28: Tic-Tac-Toe

Hangman

My daughter likes to play it a lot. It took me slightly more than 500 lines of code to write it in a multiplayer mode. This is pure javascript, however, using django it would be a lot easier to organize things.

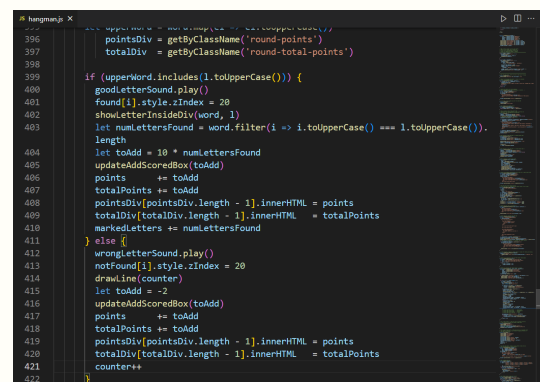


Figure 29: Hangman