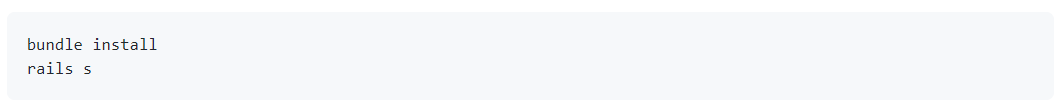
ruby on rails project

Getting project up & running

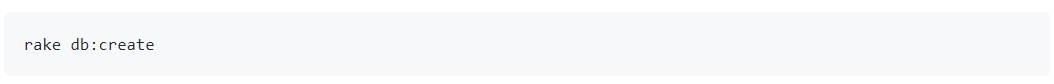
Firstly, make sure that **mysql2** is installed on the machine.

To run the project run following commands in the project directory:

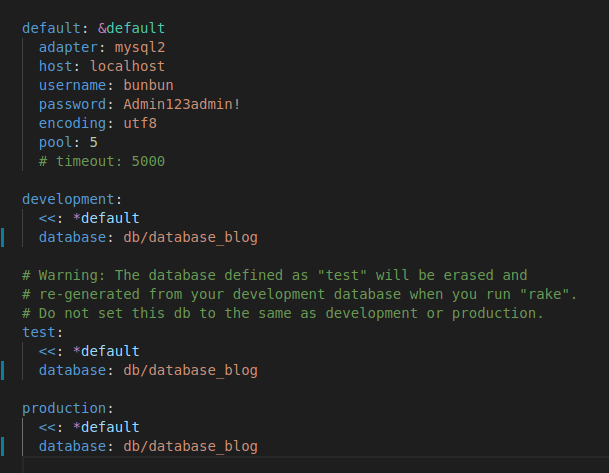


Creating a database

**In case there are issues with the database and the project isn‘t loading,** in `*/config/database.yml*`file you want to change the *database.yml file.*

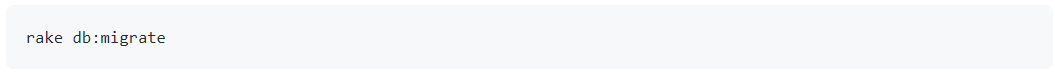


Basically it's recommended to use different gems for each environment, you should also create three databases, each for **development, testing, and production** environment. You can configure them in your `*config/database.yml*` file.



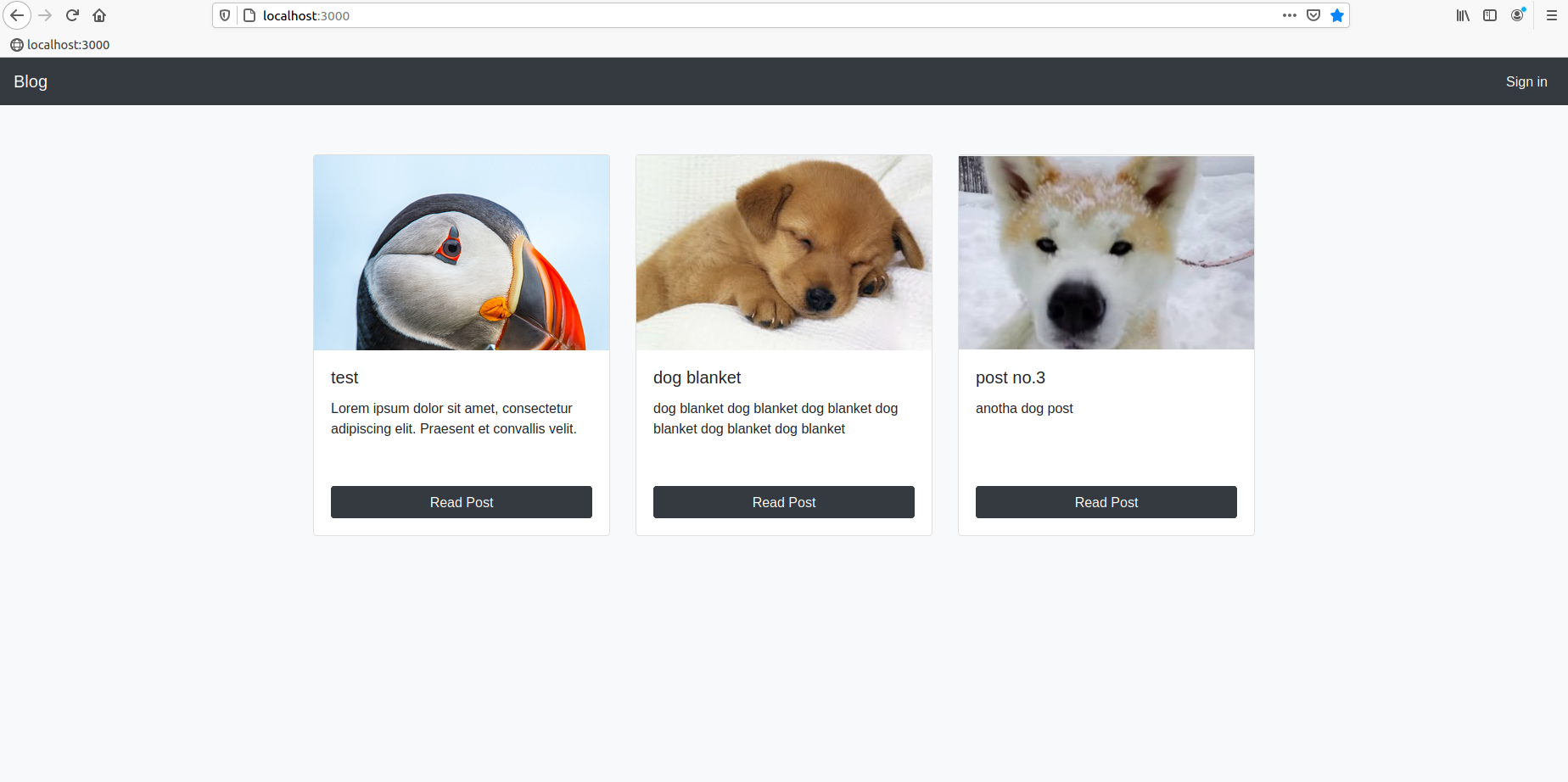
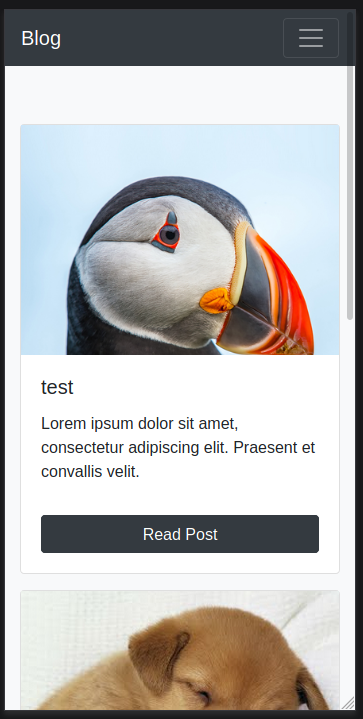
Migrating

Migrations setup the tables in the database. After that, you want to run the following command to get it up and running:

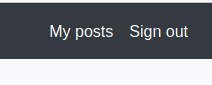


This should set our project up and it should be up and running.

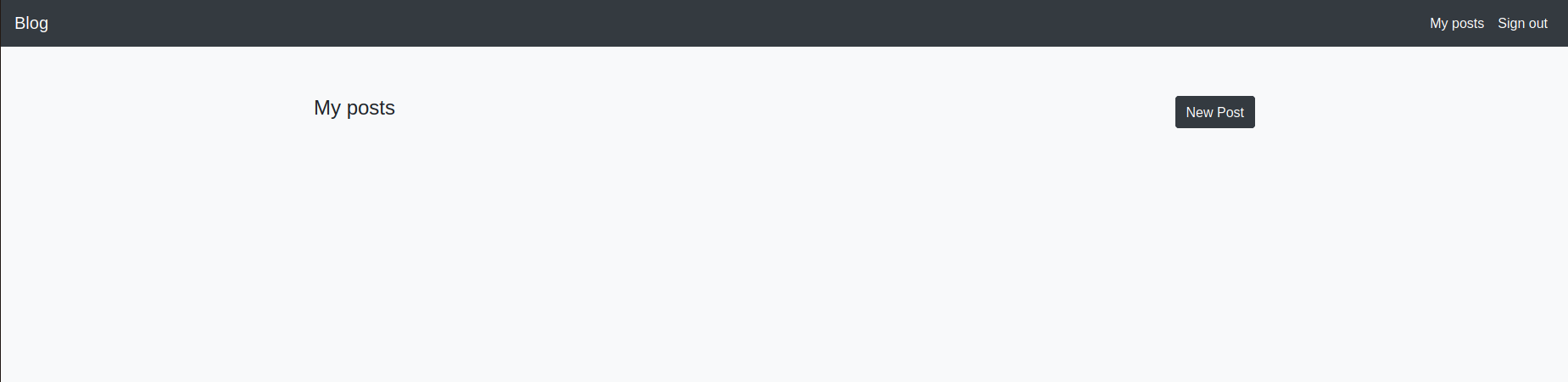
To test it, we can run ***rails s*** and head to [***http://localhost:3000/***](http://localhost:3000/) to see our project. For front-end styling we used bootstrap.

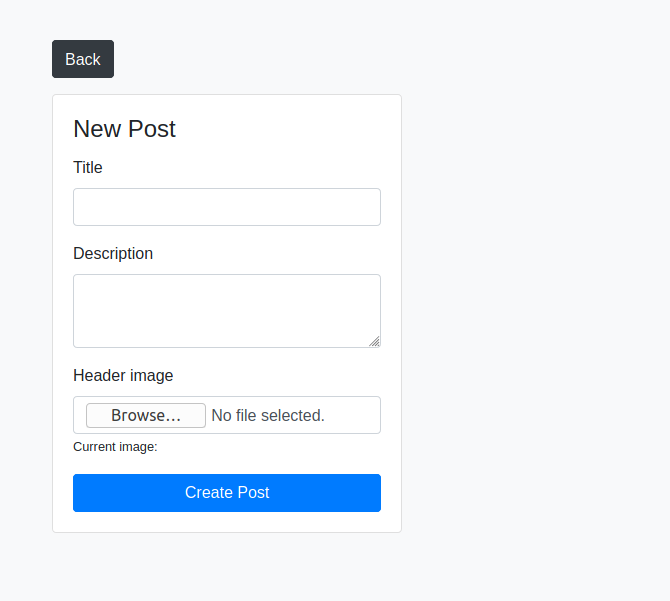
User doesn‘t need to be logged in to read the posts, but will need to create / log into an account if he/she wants to create a post.



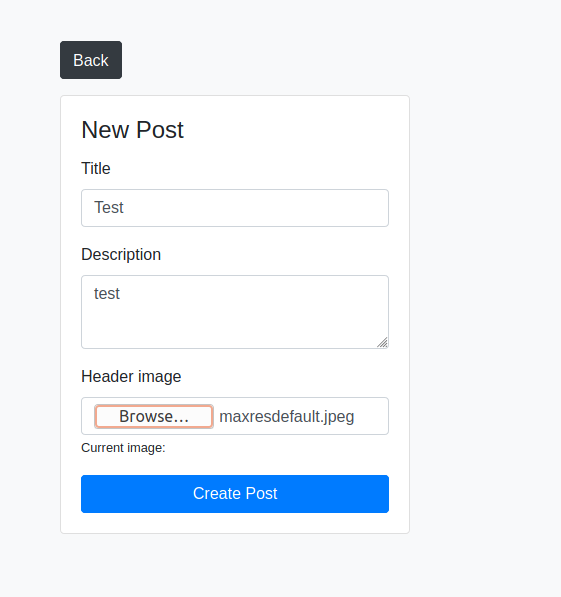
Once logged in, user will see two buttons at the top right corner. One is to log out of the account and another one is to see all the posts made by a user.



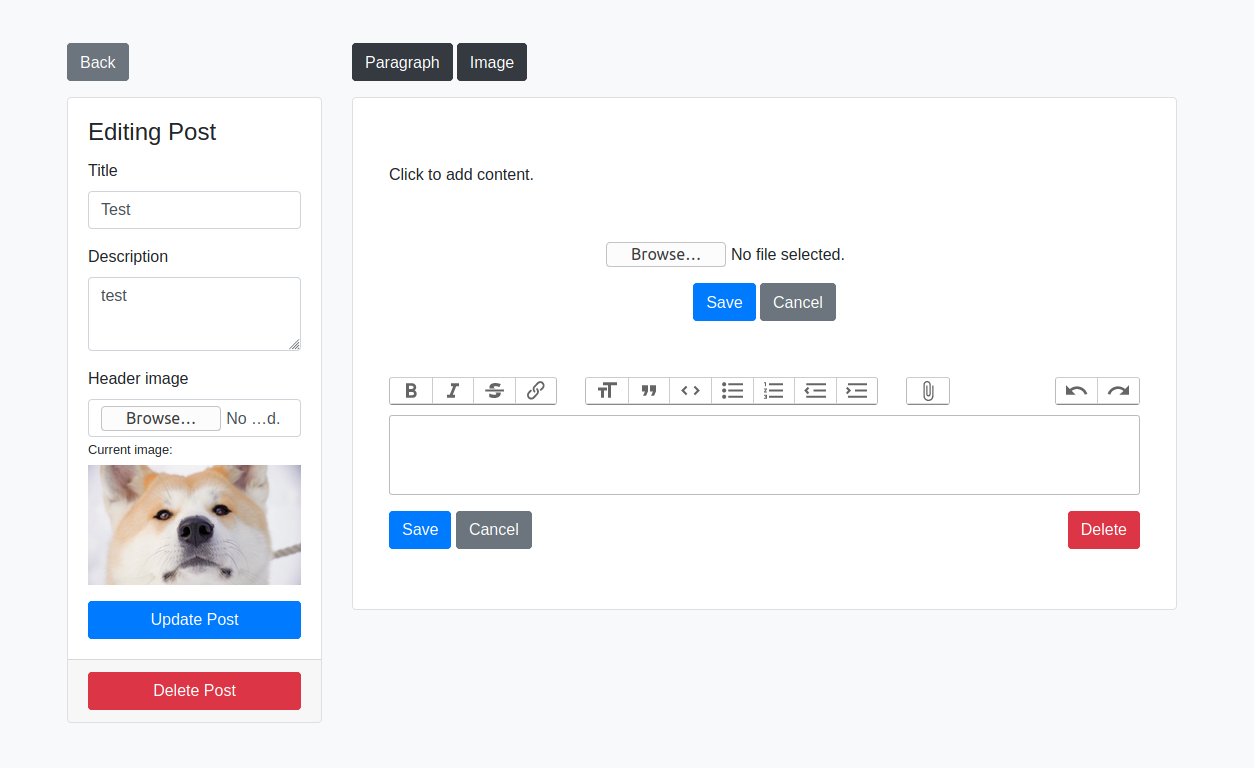
In this case we do not have any posts made with this user. So to create one we just click on „*New Post*“.



This will bring the user into `*/posts/new`* and a user can write down initial information about the post. User can give post a title, description and add a header image which will be displayed as a main image.

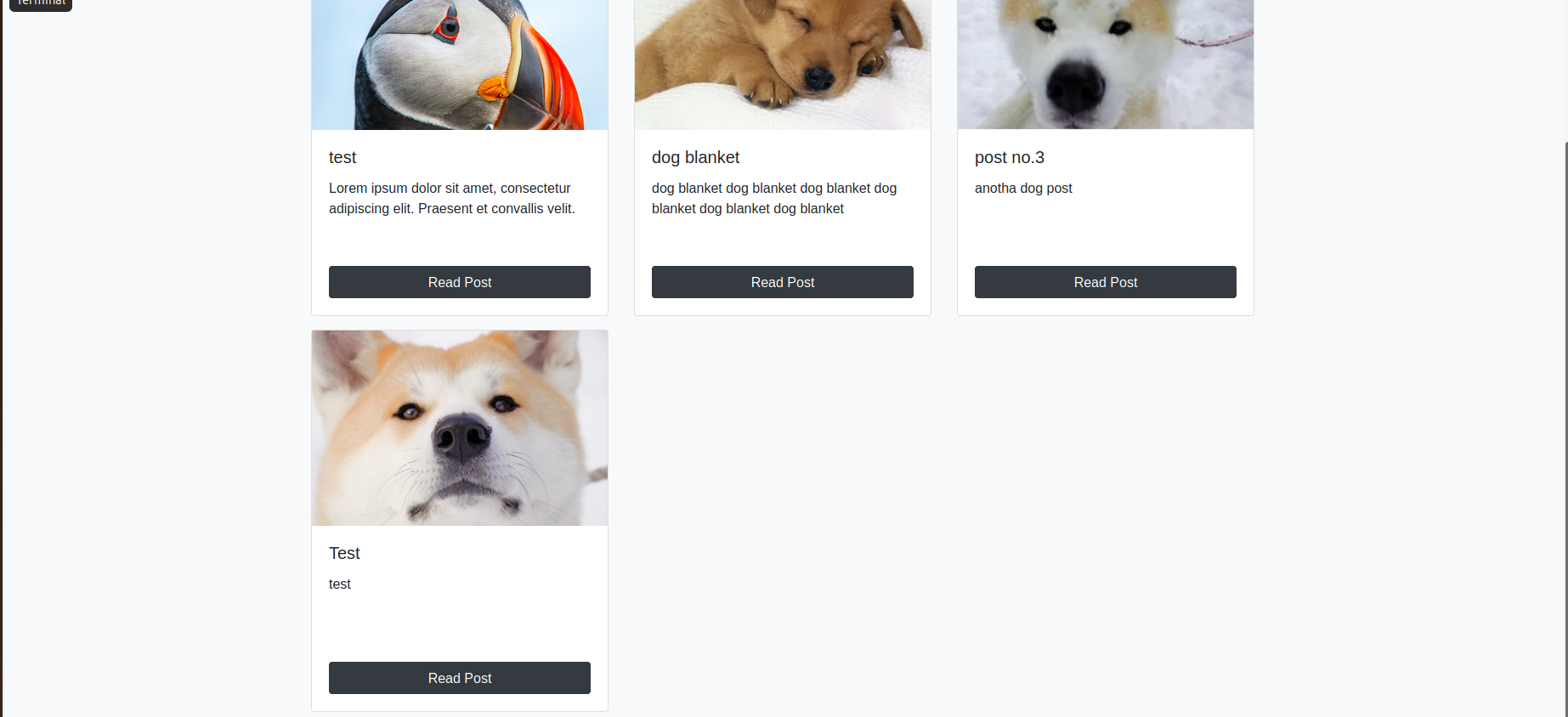


After writing some information about the post, once we click on „*Create Post“* it will create a post in a database and it will bring user into post edit screen. And it will now be shown in the home screen.

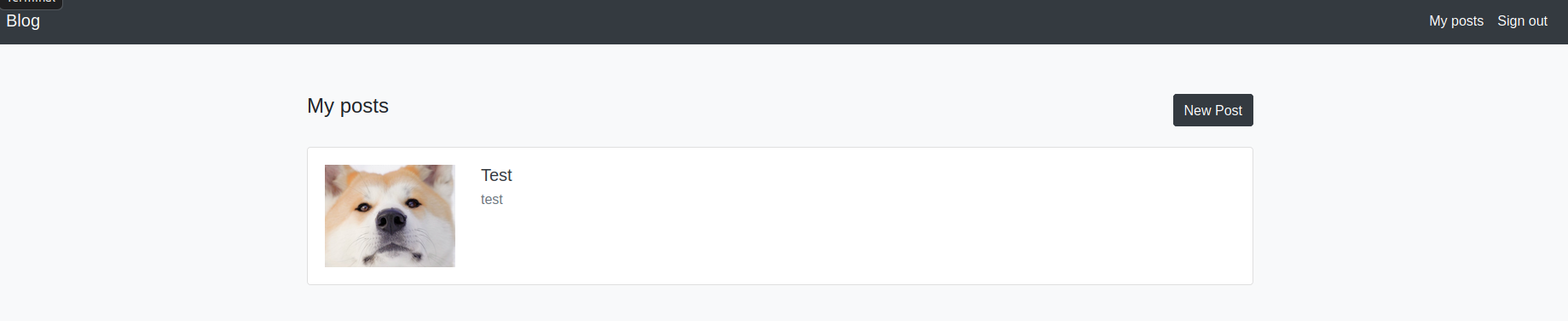
**

If we click on the „Update Post“ button we will save the post as it is.

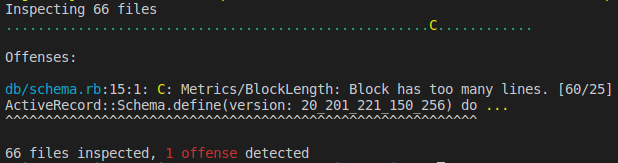
This is how our home page looks like:



And since a user made a post, we can now go into `/*posts*/`and see all the user posts that have been made.



rubocop

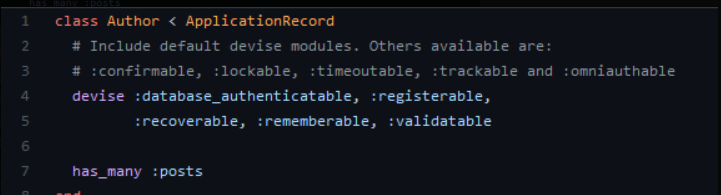


When running the only offence we get is for schema. Schema is too big. But schema was generated so we did not refactor.

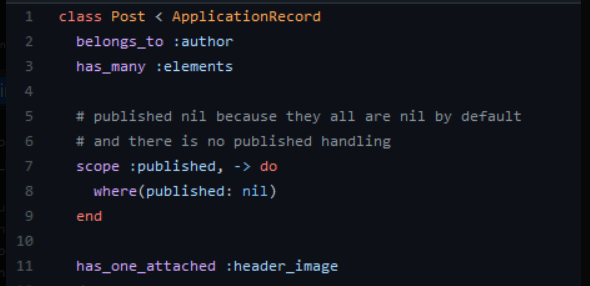
Posts & authors

Logged in user can have many posts assigned to them. Logged in user is an Author.

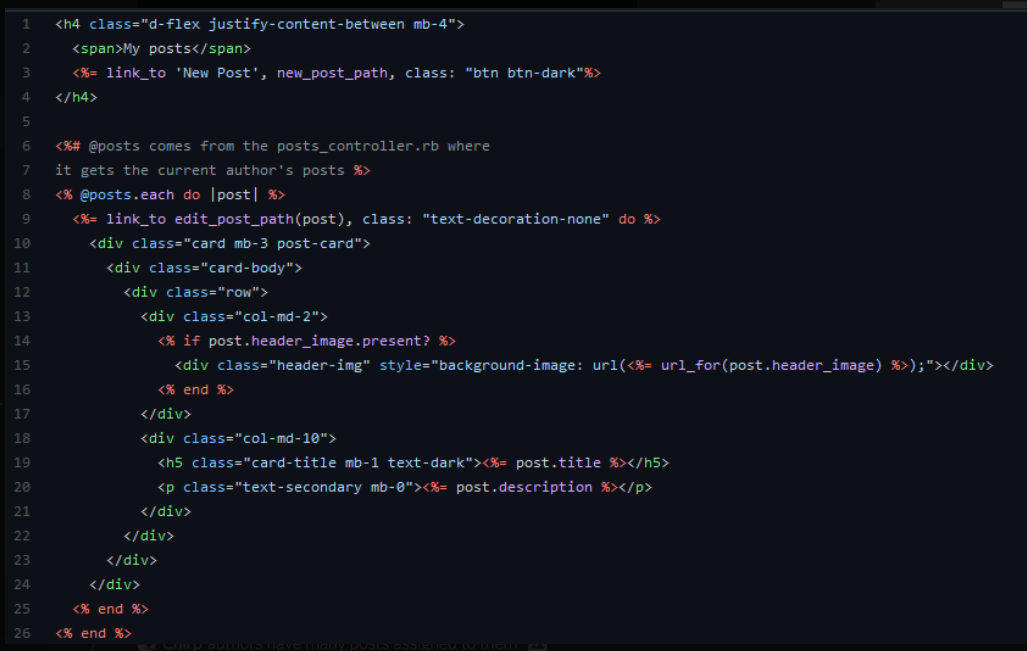
**Author validation is handled by devise gem.**



Each user can see their own posts because, posts are linked to the authors (users) they belong to:



This is a html for the my posts page of logged in user.



Posts are taken by the current signed in author

