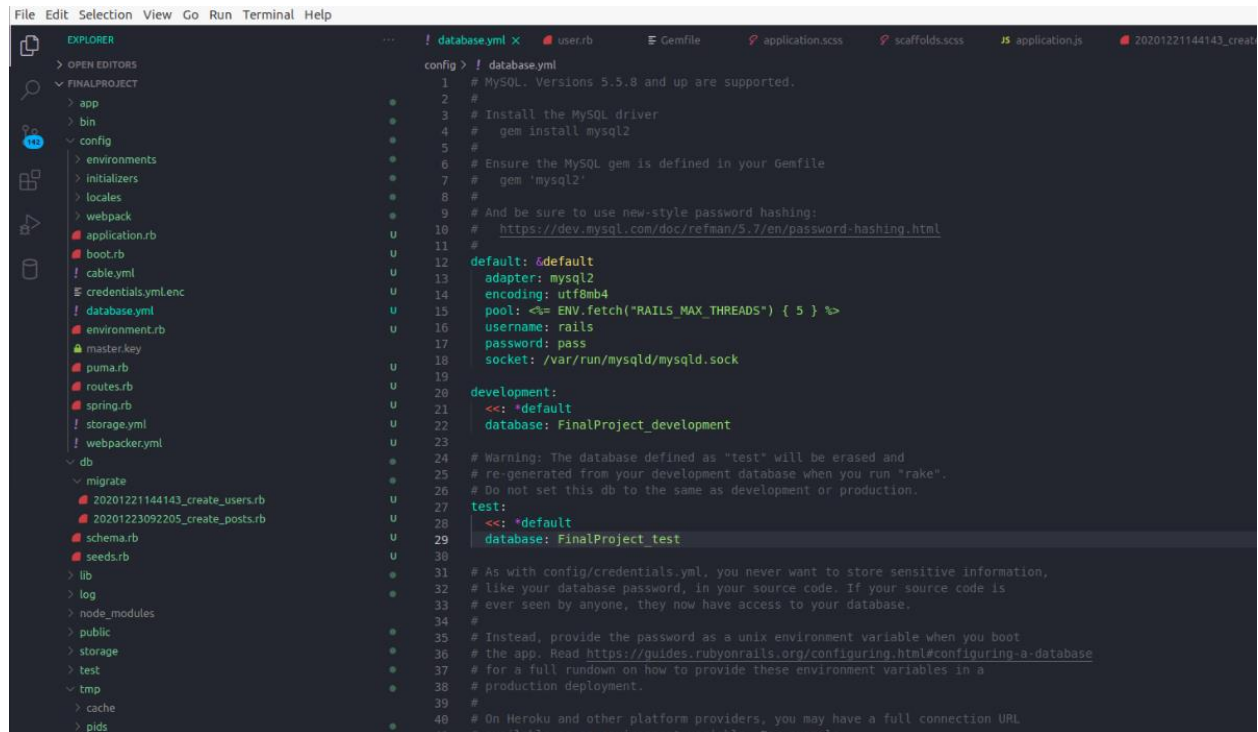
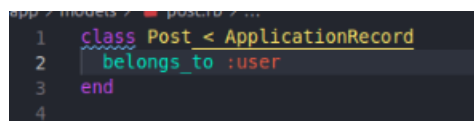


Projekte įgyvendintas paprastas blogas, kur prisijungę vartotojai gali daryti įrašus. Duomenų bazė panaudota MySQL

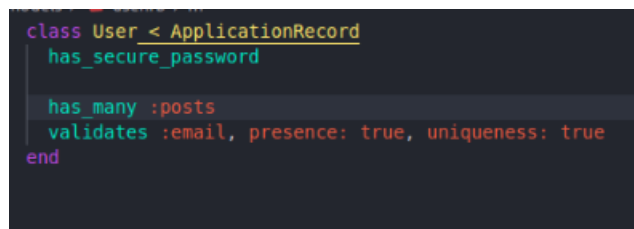


```
1 # MySQL. Versions 5.5.8 and up are supported.
2 #
3 # Install the MySQL driver
4 #   gem install mysql2
5 #
6 # Ensure the MySQL gem is defined in your Gemfile
7 #   gem 'mysql2'
8 #
9 # And be sure to use new-style password hashing:
10 #   https://dev.mysql.com/doc/refman/5.7/en/password-hashing.html
11 #
12 default: <default
13   adapter: mysql2
14   encoding: utf8mb4
15   pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
16   username: rails
17   password: pass
18   socket: /var/run/mysqld/mysqld.sock
19
20 development:
21   <<: *default
22   database: FinalProject_development
23
24 # Warning: The database defined as "test" will be erased and
25 # re-generated from your development database when you run "rake".
26 # Do not set this db to the same as development or production.
27 test:
28   <<: *default
29   database: FinalProject_test
30
31 # As with config/credentials.yml, you never want to store sensitive information,
32 # like your database password, in your source code. If your source code is
33 # ever seen by anyone, they now have access to your database.
34 #
35 # Instead, provide the password as a unix environment variable when you boot
36 # the app. Read https://guides.rubyonrails.org/configuring.html#configuring-a-database
37 # for a full rundown on how to provide these environment variables in a
38 # production deployment.
39 #
40 # On Heroku and other platform providers, you may have a full connection URL
41 # available as an environment variable. For example:
```

Tarp vartotojų ir įrašų panaudotas ActiveRecord Associations.



```
1 class Post < ApplicationRecord
2   belongs_to :user
3 end
4
```



```
class User < ApplicationRecord
  has_secure_password

  has_many :posts

  validates :email, presence: true, uniqueness: true
end
```

Atsidarius tinklalapį, matome vartotojų sąrašą.

## Avialable Users

email@email.com

User Posts

email2@email.com

User Posts

admin@email.com

User Posts

Paspaudus ant User Posts, matome to vartotojo įrašus.

Name:

Post 1

Content:

This is an example post

Name:

Post 2

Content:

This is an example post

Užsiregistravus ir prisijungus, viršuje matome pasirinkimą My account

Logged in as email@email.com

My account

Log Out

## Avialable Users

email@email.com

User Posts

email2@email.com

User Posts

admin@email.com

User Posts

Paspaudus ant jo matome savo įrašų sąrašą, galime pridėti naujus, redaguoti ar ištrinti senus.

## Posts

Post 1 [Edit](#) [Destroy](#)

Post 2 [Edit](#) [Destroy](#)

[Back](#) [New Post](#)

## New Post

Name

Content

[Create Post](#)

[Back](#)

Prisijungus kaip administratoriui („hardcodintas“ prisijungimas *admin@email.com* slaptažodis – *testtest*) ir nuėjus į vartotojo įrašus, mes matome galimybę juos ištrinti.

---

Name:

Post 1

Delete

Content:

This is an example post

Name:

Post 2

Delete

Content:

This is an example post

Name:

Post 3

Delete

Content:

This is also a post