Tai socialinis tinkle tipo aplikacija kurioje žmonės gali ieškoti vienodą hobį arba interesą turinčių asmenų.

Controller:

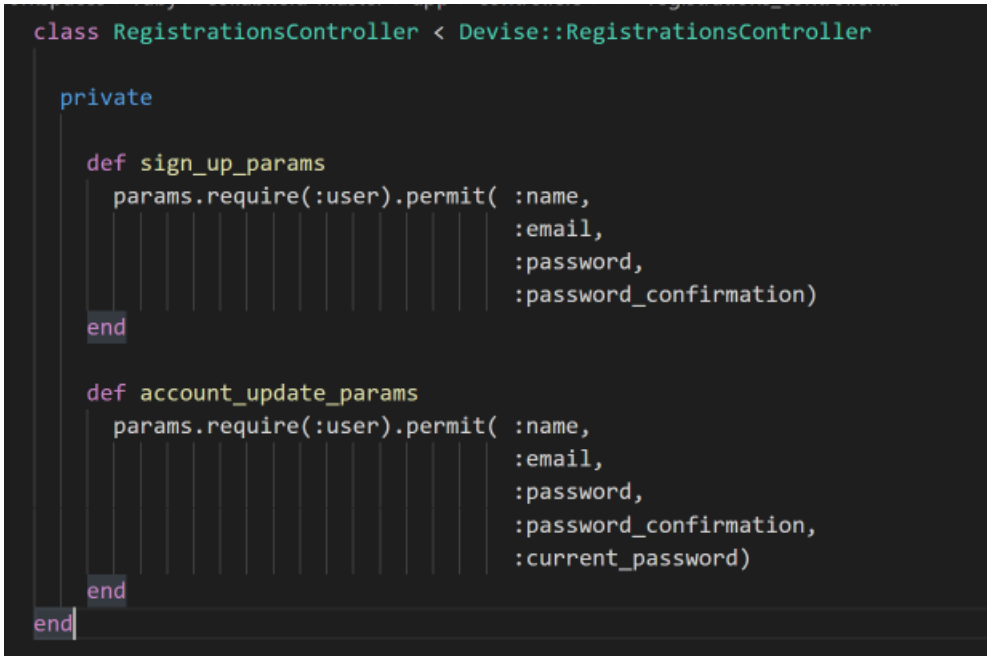
Visi kontroleriai paveldi: Action Controller

```ruby
class PagesController < ApplicationController

  def index
    @hobby_posts = Post.by_branch('hobby').limit(8)
    @study_posts = Post.by_branch('study').limit(8)
    @team_posts = Post.by_branch('team').limit(8)
    @contacts = user_signed_in? ? current_user.all_active_contacts : ''
  end

end
```

Pages controller naudojamas valdyti statiniams ir specialiems puslapiams. Jis veikia kaip root controlleris. Taip kontroleris kuris gauna duomenis is posts lentelės ir gražiną į view dalį.

```ruby
class PostsController < ApplicationController
  before_action :redirect_if_not_signed_in, only: [:new]

  def show
    @post = Post.find(params[:id])
    if user_signed_in?
      @message_has_been_sent = conversation_exist?
    end
  end

  def new
    @branch = params[:branch]
    @categories = Category.where(branch: @branch)
    @post = Post.new
  end

  def create
    @post = Post.new(post_params)
    if @post.save
      redirect_to post_path(@post)
    else
      redirect_to root_path
    end
  end


  def hobby
    posts_for_branch(params[:action])
  end

  def study
    posts_for_branch(params[:action])
  end

  def team
    posts_for_branch(params[:action])
  end

  private

  def conversation_exist?
    Private::Conversation.between_users(current_user.id, @post.user.id).present?
  end

  def post_params
    params.require(:post).permit(:content, :title, :category_id)
                         .merge(user_id: current_user.id)
  end

  def posts_for_branch(branch)
    @categories = Category.where(branch: branch)
    @posts = get_posts.paginate(page: params[:page])
    respond_to do |format|
      format.html
      format.js { render partial: 'posts/posts_pagination_page' }
    end
  end

  def get_posts
    PostsForBranchService.new({
      search: params[:search],
      category: params[:category],
      branch: params[:action]
    }).call
  end

end
```

Posts Controller skirtas valdyti visa irašų duomenų keliavimą MVC modelyje.



```ruby
class RegistrationsController < Devise::RegistrationsController

  private

    def sign_up_params
      params.require(:user).permit( :name,
                                    :email,
                                    :password,
                                    :password_confirmation)
    end

    def account_update_params
      params.require(:user).permit( :name,
                                    :email,
                                    :password,
                                    :password_confirmation,
                                    :current_password)
    end
end
```

Registration controller skirtas registruoti vartotojams. Šis kontroleris kuris overridina devise gemo kontroleriu sign_up_params ir account_update_params metodus, kurie ir suteikia autentifikacijos funkcionalumą.

```ruby
class ApplicationController < ActionController::Base
  protect_from_forgery with: :exception
  before_action :opened_conversations_windows
  before_action :all_ordered_conversations
  before_action :set_user_data

  def all_ordered_conversations
    if user_signed_in?
      @all_conversations = OrderConversationsService.new({user: current_user}).call
    end
  end

  def opened_conversations_windows
    if user_signed_in?
      # opened conversations
      session[:private_conversations] ||= []
      session[:group_conversations] ||= []
      @private_conversations_windows = Private::Conversation.includes(:recipient, :messages)
                                       .find(session[:private_conversations])
      @group_conversations_windows = Group::Conversation.find(session[:group_conversations])
    else
      @private_conversations_windows = []
      @group_conversations_windows = []
    end
  end

  def redirect_if_not_signed_in
    redirect_to root_path if !user_signed_in?
  end

  def redirect_if_signed_in
    redirect_to root_path if user_signed_in?
  end

  private

  def set_user_data
    if user_signed_in?
      gon.group_conversations = current_user.group_conversations.ids
      gon.user_id = current_user.id
      cookies[:user_id] = current_user.id if current_user.present?
      cookies[:group_conversations] = current_user.group_conversations.ids
    else
      gon.group_conversations = []
    end
  end

end
```

Application controller skirtas kaip minėjau pernaudoti kodą per visus kontrolerius kurie paveldi šią klasę. Šiame kontroleryje sudėti visi metodai kurie naudojami visų kontrolerių šioje programoje. Pagrindinis tikslas užtikrinti funkcijų kurios veikia visuomet esant klientui svetainėje kaip messaging arba set_user_data, kuris naudojamas visuomet.

Routing:

```ruby
Rails.application.routes.draw do
  devise_for :users, :controllers => {:registrations => "registrations"}

  devise_scope :user do
    get 'login', to: 'devise/sessions#new'
  end

  devise_scope :user do
    get 'signup', to: 'devise/registrations#new'
  end

  root to: 'pages#index'
  get 'messenger', to: 'messengers#index'
  get 'get_private_conversation', to: 'messengers#get_private_conversation'
  get 'get_group_conversation', to: 'messengers#get_group_conversation'
  get 'open_messenger', to: 'messengers#open_messenger'

  resources :posts do
    collection do
      get 'hobby'
      get 'study'
      get 'team'
    end
  end

  namespace :private do
    resources :conversations, only: [:create] do
      member do
        post :close
        post :open
      end
    end
    resources :messages, only: [:index, :create]
  end

  namespace :group do
    resources :conversations do
      member do
        post :close
        post :open
      end
    end
    resources :messages, only: [:index, :create]
  end

  resources :contacts, only: [:create, :update, :destroy]

end
```

Panaudoti partial templates, _navigation.html.erb

Navigation partial layout panaudotas tam kad visiems puslapiams butu surenderinta navigacija application.html.erb faile.

```
<%= render 'layouts/navigation' %>
```

Models:

User skirtas svetainės naudotojams

Bootstrap:

Bootstrapo panaudojimui įrašytas bootstrap-sass gemas, kartu su sass-rails gemu.

Taip pat pakeistas app/assets/stylesheets/application.css į application.scss.

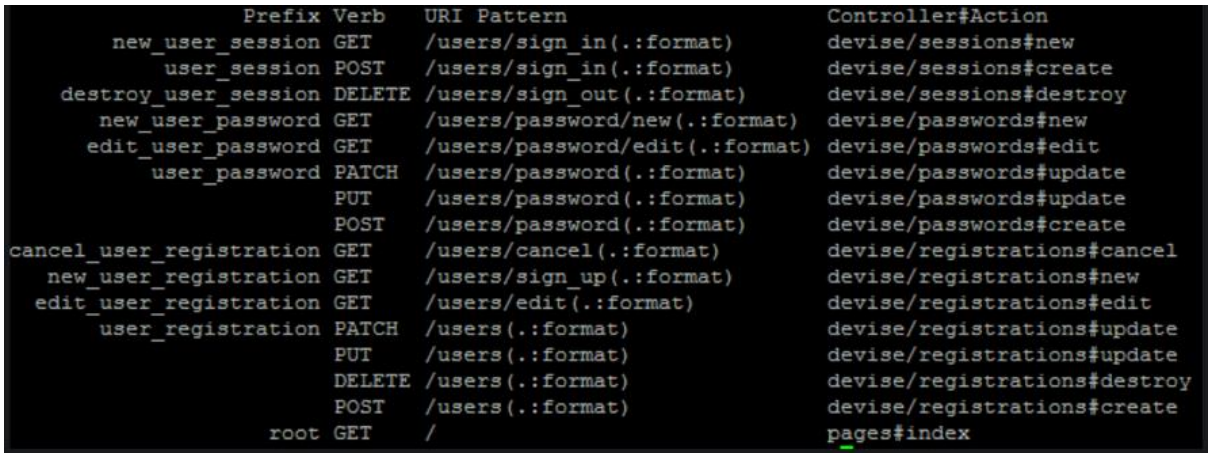
Pridėtas jquery-rails gemas.

app/assets/javascripts/application.js

Helpers:

Panaudotas tam kad paimti logiką iš view dalių. PagesController, turi pages_helper.rb faila, kuris bus prieinamas visiems failams viduje views/pages aplanke.

Routes:



```
                 Prefix Verb   URI Pattern                    Controller#Action
       new_user_session GET    /users/sign_in(.:format)       devise/sessions#new
           user_session POST   /users/sign_in(.:format)       devise/sessions#create
   destroy_user_session DELETE /users/sign_out(.:format)      devise/sessions#destroy
      new_user_password GET    /users/password/new(.:format)  devise/passwords#new
     edit_user_password GET    /users/password/edit(.:format) devise/passwords#edit
         user_password PATCH  /users/password(.:format)      devise/passwords#update
                       PUT    /users/password(.:format)      devise/passwords#update
                       POST   /users/password(.:format)      devise/passwords#create
cancel_user_registration GET    /users/cancel(.:format)       devise/registrations#cancel
   new_user_registration GET    /users/sign_up(.:format)      devise/registrations#new
  edit_user_registration GET    /users/edit(.:format)         devise/registrations#edit
      user_registration PATCH  /users(.:format)              devise/registrations#update
                       PUT    /users(.:format)              devise/registrations#update
                       DELETE /users(.:format)              devise/registrations#destroy
                       POST   /users(.:format)              devise/registrations#create
                 root GET    /                             pages#index
```

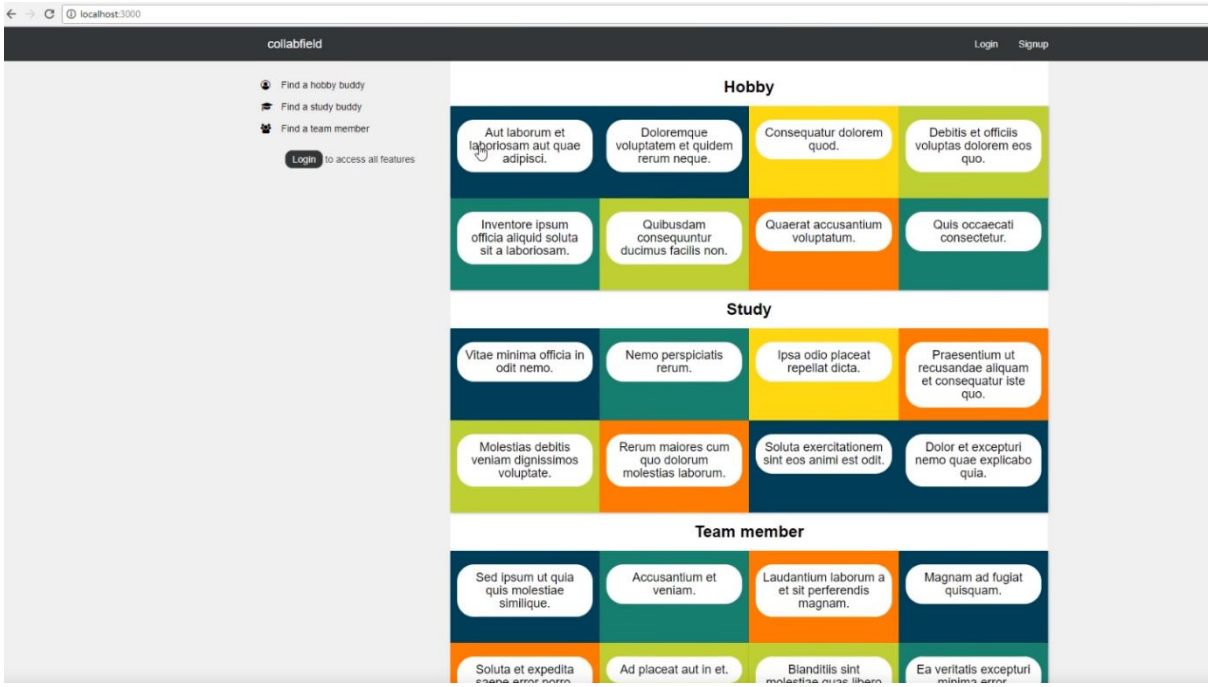
Autentifikacija:

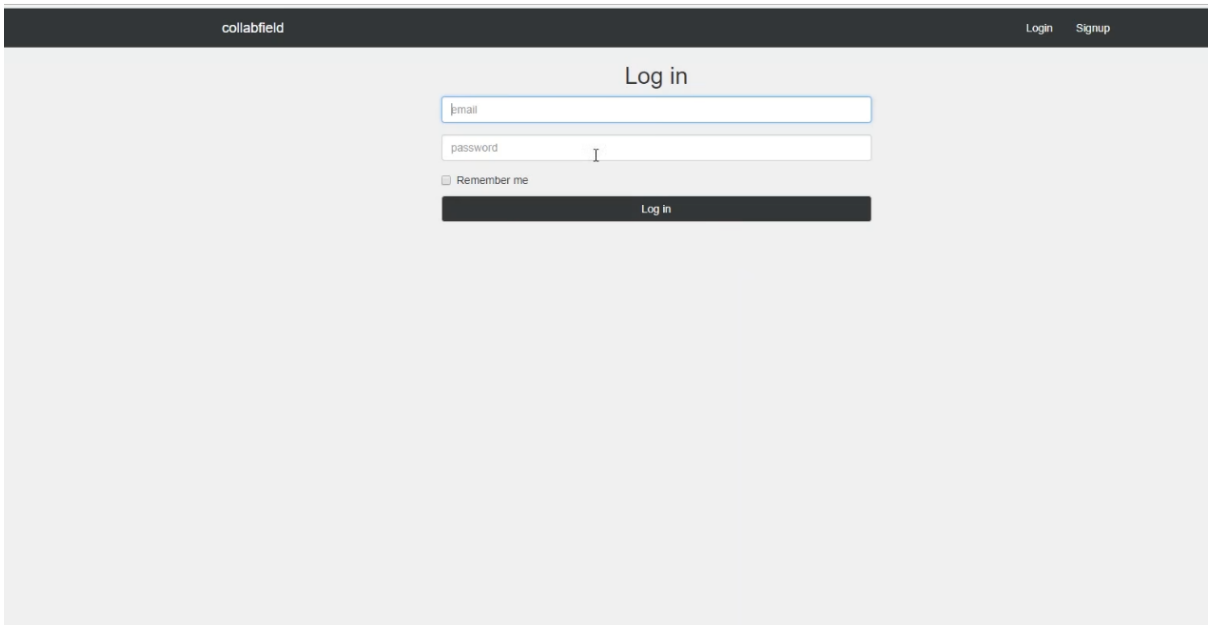
Autentifikacijai panaudotas devise gemas.

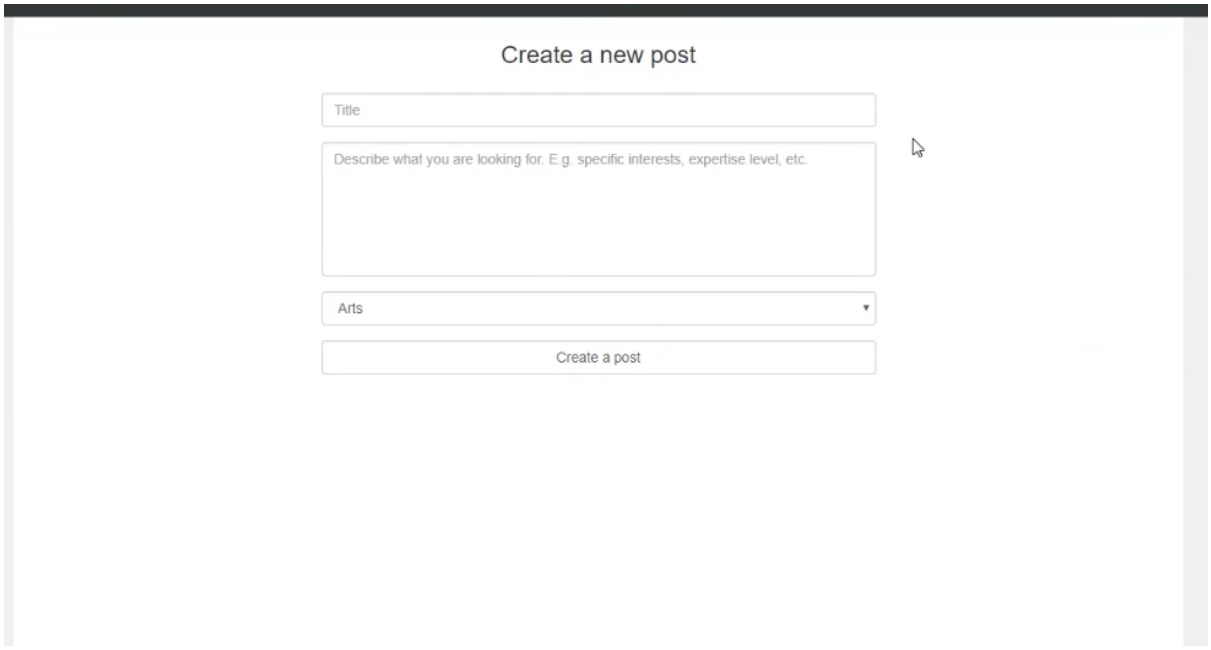
Duomenų bazė: Postegreqsl

-=-=-=-=-=-==-=-=-=-=-=-

Veikiančio aplikacijos nuotraukos:


localhost:3000
collabfield
Login
Signup
Find a hobby buddy
Find a study buddy
Find a team member
Login to access all features
Hobby
Aut laborum et laboriosam aut quae adipisci.
Doloremque voluptatem et quidem rerum neque.
Consequatur dolorem quod.
Debitis et officiis voluptas dolorem eos quo.
Inventore ipsum officia aliquid soluta sit a laboriosam.
Quibusdam consequuntur ducimus facilis non.
Quaerat accusantium voluptatum.
Quis occaecati consectetur.
Study
Vitae minima officia in odit nemo.
Nemo perspiciatis rerum.
Ipsa odio placeat repellat dicta.
Praesentium ut recusandae aliquam et consequatur iste quo.
Molestias debitis veniam dignissimos voluptate.
Rerum maiores cum quo dolorum molestias laborum.
Soluta exercitationem sint eos animi est odit.
Dolor et excepturi nemo quae explicabo quia.
Team member
Sed ipsum ut quia quis molestiae similique.
Accusantium et veniam.
Laudantium laborum a et sit perferendis magnam.
Magnam ad fugiat quisquam.
Soluta et expedita
Ad placeat aut in et.
Blanditiis sint
Ea veritatis excepturi


collabfield
Login
Signup
Log in
email
password
Remember me
Log in


Create a new post
Title
Describe what you are looking for. E.g. specific interests, expertise level, etc.
Arts
Create a post