|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |
| --- |
| **по зачетной работе** |

**Дисциплина:** Языки интернет-программирования

**Тема:** Dotop – сервис для обмена дотфайлами

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-33Б |  |  | Д. И. Мироненко |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Э. Р. Маняшев |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Описание**

Дотфайл - это скрытый файл в операционных системах Linux и MacOS, имя которого начинается с точки. Эти файлы обычно являются файлами конфигурации для различных программ и скрыты по умолчанию для того, чтобы домашний каталог пользователя оставался чистым и чтобы избежать загромождения конфигурационными файлами. Примерами дотфайлов являются .bashrc, .gitconfig и .vimrc.

Мой сервис обмена такими файлами решает несколько проблем. Во-первых, часто бывает трудно написать свой конфигурационный файл для какой-нибудь программы в силу отсутствия знаний и опыта, поэтому на этом сайте пользователь может изучить конфиги других людей и на их примере написать свой. Во-вторых, бывают случаи, когда нет желания писать свой конфиг, а хочется взять уже готовый и наслаждаться всеми возможностями программы, и в этом тоже поможет мой сайт, на котором пользователь на основании оценок пользователей (голосов) может выбрать лучший или наиболее подходящий для него дотфайл.

**Код программы**

**Файл “app/controllers/application\_controller.rb”**

|  |
| --- |
| # frozen\_string\_literal: true  # Controller for application logic  class ApplicationController < ActionController::Base  before\_action :set\_locale  private  def after\_sign\_out\_path\_for(\_resource\_or\_scope)  dotfiles\_path  end  def after\_sign\_in\_path\_for(\_resource\_or\_scope)  dotfiles\_path  end  def default\_url\_options  { locale: I18n.locale }  end  def set\_locale  I18n.locale = extract\_locale || I18n.default\_locale  end  def extract\_locale  parsed\_locale = params[:locale]  return unless I18n.available\_locales.map(&:to\_s).include?(parsed\_locale)  parsed\_locale.to\_sym  end  end |

**Файл “app/controllers/bookmarks\_controller.rb”**

|  |
| --- |
| # frozen\_string\_literal: true  # Controller for bookmarks logic  class BookmarksController < ApplicationController  include Pagy::Backend  before\_action :authenticate\_user!  def index  search\_params = params.permit(:format, :page, :commit, :locale,  q: %i[title\_or\_description\_or\_config\_type\_or\_user\_username\_cont config\_type\_eq s])  @q = Dotfile.ransack(search\_params[:q])  order\_by = get\_order\_(search\_params)  dotfiles = @q.result(distinct: true).order(order\_by => :desc)  @bookmarks\_pagy, @bookmarks = pagy(dotfiles, items: 10)  rescue Pagy::OverflowError  redirect\_to bookmarks\_index\_path(page: 1)  end  private  def get\_order\_(search\_params)  if !search\_params[:q].nil? && !search\_params[:q][:s].nil?  order\_by = search\_params[:q][:s].split[0]  order\_by = (order\_by == 'created\_at' ? :created\_at : :cached\_weighted\_like\_score)  else  order\_by = :cached\_weighted\_like\_score  end  order\_by  end  end |

**Файл “app/controllers/users\_controller.rb”**

|  |
| --- |
| # frozen\_string\_literal: true  # Controller for user profile page  class UsersController < ApplicationController  def show  @user = User.find(params[:id])  end  end |

**Файл “app/controllers/dotfiles\_controller.rb”**

|  |
| --- |
| # frozen\_string\_literal: true  # Controller for dotfiles logic  class DotfilesController < ApplicationController  include Pagy::Backend  before\_action :set\_dotfile, only: %i[show edit update destroy vote bookmark]  before\_action :authenticate\_user!, except: %i[index show]  before\_action :correct\_user, only: %i[edit update destroy]  # GET /dotfiles or /dotfiles.json  def index  search\_params = params.permit(:format, :page, :commit, :locale,  q: %i[title\_or\_description\_or\_config\_type\_or\_user\_username\_cont config\_type\_eq s])  @q = Dotfile.ransack(search\_params[:q])  order\_by = get\_order\_(search\_params)  dotfiles = @q.result(distinct: true).order(order\_by => :desc)  @dotfiles\_pagy, @dotfiles = pagy(dotfiles, items: 2)  rescue Pagy::OverflowError  redirect\_to dotfiles\_path(page: 1)  end  def get\_order\_(search\_params)  if !search\_params[:q].nil? && !search\_params[:q][:s].nil?  order\_by = search\_params[:q][:s].split[0]  order\_by = (order\_by == 'created\_at' ? :created\_at : :cached\_weighted\_like\_score)  else  order\_by = :cached\_weighted\_like\_score  end  order\_by  end  def bookmark  @dotfile.bookmark!(current\_user)  respond\_to do |format|  format.html do  redirect\_to @dotfile  end  format.turbo\_stream do  render turbo\_stream: turbo\_stream.replace(  @dotfile, partial: 'dotfiles/dotfile',  locals: { dotfile: @dotfile, show\_config: params[:show\_config] }  )  end  end  end  def vote  case params[:type]  when 'upvote'  @dotfile.upvote!(current\_user)  when 'downvote'  @dotfile.downvote!(current\_user)  else  return redirect\_to request.url, alert: 'no such vote type'  end  respond\_to do |format|  format.html do  redirect\_to @dotfile  end  format.turbo\_stream do  render turbo\_stream: turbo\_stream.replace(@dotfile, partial: 'dotfiles/dotfile',  locals: { dotfile: @dotfile, show\_config: params[:show\_config] })  end  end  end  # GET /dotfiles/1 or /dotfiles/1.json  def show; end  # GET /dotfiles/new  def new  @dotfile = current\_user.dotfiles.build  end  # GET /dotfiles/1/edit  def edit; end  # POST /dotfiles or /dotfiles.json  def create  @dotfile = current\_user.dotfiles.build(dotfile\_params)  respond\_to do |format|  if @dotfile.save  format.html { redirect\_to dotfile\_url(@dotfile), notice: I18n.t('notices.create\_dotfile') }  format.json { render :show, status: :created, location: @dotfile }  else  format.html { render :new, status: :unprocessable\_entity }  format.json { render json: @dotfile.errors, status: :unprocessable\_entity }  end  end  end  # PATCH/PUT /dotfiles/1 or /dotfiles/1.json  def update  respond\_to do |format|  if @dotfile.update(dotfile\_params)  format.html { redirect\_to dotfile\_url(@dotfile), notice: I18n.t('notices.update\_dotfile') }  format.json { render :show, status: :ok, location: @dotfile }  else  format.html { render :edit, status: :unprocessable\_entity }  format.json { render json: @dotfile.errors, status: :unprocessable\_entity }  end  end  end  # DELETE /dotfiles/1 or /dotfiles/1.json  def destroy  @dotfile.destroy!  respond\_to do |format|  format.html { redirect\_to dotfiles\_url, notice: I18n.t('notices.destroy\_dotfile') }  format.json { head :no\_content }  end  end  def correct\_user  @dotfile = current\_user.dotfiles.find\_by(id: params[:id])  redirect\_to dotfiles\_path, notice: I18n.t('notices.not\_authorized\_to\_edit') if @dotfile.nil?  end  private  # Use callbacks to share common setup or constraints between actions.  def set\_dotfile  @dotfile = Dotfile.find(params[:id])  end  # Only allow a list of trusted parameters through.  def dotfile\_params  params.require(:dotfile).permit(:title, :description, :config\_type, :file, :user\_id)  end  end |

**Файл “app/helpers/application\_helper.rb”**

|  |
| --- |
| module ApplicationHelper  include Pagy::Frontend  end |

**Файл “app/helpers/dotfiles\_helper.rb”**

|  |
| --- |
| module DotfilesHelper  def upvote\_label(message, user)  label\_text = if user.voted\_up\_on? message, vote\_scope: "like"  "UNvote"  else  "UPvote"  end  tag.span do  "#{message.cached\_scoped\_like\_votes\_up} #{label\_text}"  end  end  def downvote\_label(message, user)  label\_text = if user.voted\_down\_on? message, vote\_scope: "like"  "UNvote"  else  "DOWNvote"  end  tag.span do  "#{message.cached\_scoped\_like\_votes\_down} #{label\_text}"  end  end  def upvote\_label\_styles(dotfile, user)  if user&.voted\_up\_on? dotfile, vote\_scope: "like"  "background-color: #e9ecef;"  end  end  def downvote\_label\_styles(dotfile, user)  if user&.voted\_down\_on? dotfile, vote\_scope: "like"  "background-color: #e9ecef;"  end  end  def bookmark\_label\_styles(message, user)  if user.voted\_up\_on? message, vote\_scope: "bookmark"  "background-color: #e9ecef;"  end  end  def bookmark\_label\_styles(dotfile, user)  if user&.voted\_up\_on? dotfile, vote\_scope: "bookmark"  "background-color: #e9ecef;"  end  end  end |

**Файл “app/assets/stylesheets/home.scss”**

|  |
| --- |
| .content {  position: absolute;  left: 50%;  top: 50%;  -webkit-transform: translate(-50%, -50%);  transform: translate(-50%, -50%);  }  .header {  width: 400px;  margin-left: 100px;  }  .title {  font-size: 100px;  font-weight: 700;  }  .subtitle {  font-size: 24px;  font-weight: 500;  color: grey;  margin-top: -40px;  }  .actions {  width: 100%;  display: flex;  justify-content: center;  margin-top: 35px;  }  .log-in-btn {  margin-left: 10px;  } |

**Файл “app/models/dotfile.rb”**

|  |
| --- |
| class Dotfile < ApplicationRecord  has\_one\_attached :file  belongs\_to :user  validates :title, presence: true, format: { with: /\A[a-zA-Z0-9\_.\s]+\Z/ }  validates :description, presence: true, format: { with: /\A[a-zA-Z0-9\_.\s]+\Z/ }  validates :config\_type, presence: true, format: { with: /\A[a-zA-Z\_.\s]+\Z/ }  validates :file, presence: true  validate :allowed\_file\_extension    acts\_as\_votable  def bookmark!(user)  if user.voted\_up\_on? self, vote\_scope: "bookmark"  unvote\_by user, vote\_scope: "bookmark"  else  upvote\_by user, vote\_scope: "bookmark"  end  end  def upvote!(user)  if user.voted\_up\_on? self, vote\_scope: "like"  unvote\_by user, vote\_scope: "like"  else  upvote\_by user, vote\_scope: "like"  end  end  def downvote!(user)  if user.voted\_down\_on? self, vote\_scope: "like"  unvote\_by user, vote\_scope: "like"  else  downvote\_by user, vote\_scope: "like"  end  end  def allowed\_file\_extension  allowed\_extensions = %w[fish lua conf]  if file.present? && !allowed\_extensions.include?(File.extname(file.filename.to\_s).delete("."))  errors.add(:file, I18n.t('errors.invalid\_extension'))  end  end  def self.ransackable\_attributes(auth\_object = nil)  ["title", "description", "config\_type", "username"]  end  def self.ransackable\_associations(auth\_object = nil)  ["user"]  end  end |

**Файл “app/models/user.rb”**

|  |
| --- |
| class Dotfile < ApplicationRecord  has\_one\_attached :file  belongs\_to :user  validates :title, presence: true, format: { with: /\A[a-zA-Z0-9\_.\s]+\Z/ }  validates :description, presence: true, format: { with: /\A[a-zA-Z0-9\_.\s]+\Z/ }  validates :config\_type, presence: true, format: { with: /\A[a-zA-Z\_.\s]+\Z/ }  validates :file, presence: true  validate :allowed\_file\_extension    acts\_as\_votable  def bookmark!(user)  if user.voted\_up\_on? self, vote\_scope: "bookmark"  unvote\_by user, vote\_scope: "bookmark"  else  upvote\_by user, vote\_scope: "bookmark"  end  end  def upvote!(user)  if user.voted\_up\_on? self, vote\_scope: "like"  unvote\_by user, vote\_scope: "like"  else  upvote\_by user, vote\_scope: "like"  end  end  def downvote!(user)  if user.voted\_down\_on? self, vote\_scope: "like"  unvote\_by user, vote\_scope: "like"  else  downvote\_by user, vote\_scope: "like"  end  end  def allowed\_file\_extension  allowed\_extensions = %w[fish lua conf]  if file.present? && !allowed\_extensions.include?(File.extname(file.filename.to\_s).delete("."))  errors.add(:file, I18n.t('errors.invalid\_extension'))  end  end  def self.ransackable\_attributes(auth\_object = nil)  ["title", "description", "config\_type", "username"]  end  def self.ransackable\_associations(auth\_object = nil)  ["user"]  end  end |

**Файл “config/routes.rb”**

|  |
| --- |
| Rails.application.routes.draw do  scope "(:locale)", locale: /#{I18n.available\_locales.join("|")}/ do  devise\_for :users, controllers: { registrations: 'users/registrations' }  devise\_scope :user do  get '/users/sign\_out' => 'devise/sessions#destroy'  end  resources :dotfiles  resources :dotfiles do  member do  patch :vote  patch :bookmark  end  end  root 'home#index'  get 'bookmarks/index'  resources :users, only: [:show]  end  end |

**Файл “config/locales/en.yml”**

|  |
| --- |
| en:  global:  title: "Dotop"  copyright: "© 2023 Daniil Mironenko"  activerecord:  models:  dotfile: "Dotfile"  attributes:  dotfile:  title: "Title"  description: "Description"  config\_type: "Config Type"  file: "File"  helpers:  submit:  create: "Create"  update: "Apply"  header:  explore: "Explore"  saved: "Saved"  create: "Create"  log\_in: "Log In"  profile:  title: "Profile"  show: "Profile"  edit: "Edit"  log\_out: "Log Out"  home:  index:  subtitle: "Share your awesome dotfiles"  forms:  searching:  search: "Search"  search\_placeholder: "Title, description, author, type"  config\_type: "Config Type"  order\_by: "Order"  votes: "Votes"  recent: "Recent"  search\_button: "Search"  creating\_dotfile:  title: "Creating"  editing\_dotfile:  title: "Editing"  delete: "Delete"  creating\_user:  title: "Creating"  editing\_user:  title: "Editing"  change\_email: "Change Email"  new\_email: "New Email"  change\_password: "Change Password"  new\_password: "New Password"  confirm\_new\_password: "Confirm New Password"  save\_changes: "Save Changes"  current\_password: "Current Password"  apply: "Apply"  delete: "Delete"  sign\_up:  title: "Sign Up"  username: "Username"  email: "Email"  password: "Password"  confirm\_password: "Confirm Password"  submit: "Sign Up"  log\_in: "Log In"  log\_in:  title: "Log In"  email: "Email"  password: "Password"  remember\_me: "Remember"  submit: "Log In"  sign\_up: "Sign Up"  user:  username: "Username"  email: "Email"  edit: "Edit"  errors:  invalid\_extension: "has invalid extension"  notices:  create\_dotfile: "Dotfile was successfully created."  update\_dotfile: "Dotfile was successfully updated."  destroy\_dotfile: "Dotfile was successfully destroyed."  not\_authorized\_to\_edit: "Not authorized to edit this." |

**Файл “config/locales/ru.yml”**

|  |
| --- |
| ru:  global:  title: "Dotop"  copyright: "© 2023 Даниил Мироненко"  activerecord:  models:  dotfile: "Конфиг"  attributes:  dotfile:  title: "Название"  description: "Описание"  config\_type: "Тип Конфига"  file: "Файл"  helpers:  submit:  create: "Создать"  update: "Применить"  header:  explore: "Поиск"  saved: "Сохраненные"  create: "Создать"  log\_in: "Войти"  profile:  title: "Профиль"  show: "Профиль"  edit: "Редактировать"  log\_out: "Выйти"  home:  index:  subtitle: "Делитесь ваши потрясающими конфигами"  forms:  searching:  search: "Поиск"  search\_placeholder: "Название, описание, автор, тип"  config\_type: "Тип Конфига"  order\_by: "Сортировать"  votes: "Голоса"  recent: "Недавние"  search\_button: "Искать"  creating\_dotfile:  title: "Создание"  editing\_dotfile:  title: "Редактирование"  delete: "Удалить"  creating\_user:  title: "Создание"  editing\_user:  title: "Редактирование"  change\_email: "Изменить почту"  new\_email: "Новая почта"  change\_password: "Изменить пароль"  new\_password: "Новый пароль"  confirm\_new\_password: "Повторите новый пароль"  save\_changes: "Сохранить изменения"  current\_password: "Текущий пароль"  apply: "Применить"  delete: "Удалить"  sign\_up:  title: "Регистрация"  username: "Имя пользователя"  email: "Email"  password: "Пароль"  confirm\_password: "Повторите пароль"  submit: "Зарегистрироваться"  log\_in: "Вход"  log\_in:  title: "Вход"  email: "Email"  password: "Пароль"  remember\_me: "Запомнить"  submit: "Войти"  sign\_up: "Зарегистрироваться"  user:  username: "Имя пользователя"  email: "Почта"  edit: "Изменить"  errors:  invalid\_extension: "имеет недопустимое расширение"  notices:  create\_dotfile: "Конфиг был успешно создан."  update\_dotfile: "Конфиг был успешно изменен."  destroy\_dotfile: "Конфиг был успешно удален."  not\_authorized\_to\_edit: "Вы не авторизованы для редактирования этого." |

**Файл “config/locales/devise.en.yml”**

|  |
| --- |
| en:  devise:  confirmations:  confirmed: "Your email address has been successfully confirmed."  send\_instructions: "You will receive an email with instructions for how to confirm your email address in a few minutes."  send\_paranoid\_instructions: "If your email address exists in our database, you will receive an email with instructions for how to confirm your email address in a few minutes."  failure:  already\_authenticated: "You are already signed in."  inactive: "Your account is not activated yet."  invalid: "Invalid %{authentication\_keys} or password."  locked: "Your account is locked."  last\_attempt: "You have one more attempt before your account is locked."  not\_found\_in\_database: "Invalid %{authentication\_keys} or password."  timeout: "Your session expired. Please sign in again to continue."  unauthenticated: "You need to sign in or sign up before continuing."  unconfirmed: "You have to confirm your email address before continuing."  mailer:  confirmation\_instructions:  subject: "Confirmation instructions"  reset\_password\_instructions:  subject: "Reset password instructions"  unlock\_instructions:  subject: "Unlock instructions"  email\_changed:  subject: "Email Changed"  password\_change:  subject: "Password Changed"  omniauth\_callbacks:  failure: "Could not authenticate you from %{kind} because \"%{reason}\"."  success: "Successfully authenticated from %{kind} account."  passwords:  no\_token: "You can't access this page without coming from a password reset email. If you do come from a password reset email, please make sure you used the full URL provided."  send\_instructions: "You will receive an email with instructions on how to reset your password in a few minutes."  send\_paranoid\_instructions: "If your email address exists in our database, you will receive a password recovery link at your email address in a few minutes."  updated: "Your password has been changed successfully. You are now signed in."  updated\_not\_active: "Your password has been changed successfully."  registrations:  destroyed: "Bye! Your account has been successfully cancelled. We hope to see you again soon."  signed\_up: "Welcome! You have signed up successfully."  signed\_up\_but\_inactive: "You have signed up successfully. However, we could not sign you in because your account is not yet activated."  signed\_up\_but\_locked: "You have signed up successfully. However, we could not sign you in because your account is locked."  signed\_up\_but\_unconfirmed: "A message with a confirmation link has been sent to your email address. Please follow the link to activate your account."  update\_needs\_confirmation: "You updated your account successfully, but we need to verify your new email address. Please check your email and follow the confirmation link to confirm your new email address."  updated: "Your account has been updated successfully."  updated\_but\_not\_signed\_in: "Your account has been updated successfully, but since your password was changed, you need to sign in again."  sessions:  signed\_in: "Signed in successfully."  signed\_out: "Signed out successfully."  already\_signed\_out: "Signed out successfully."  unlocks:  send\_instructions: "You will receive an email with instructions for how to unlock your account in a few minutes."  send\_paranoid\_instructions: "If your account exists, you will receive an email with instructions for how to unlock it in a few minutes."  unlocked: "Your account has been unlocked successfully. Please sign in to continue."  errors:  messages:  already\_confirmed: "was already confirmed, please try signing in"  confirmation\_period\_expired: "needs to be confirmed within %{period}, please request a new one"  expired: "has expired, please request a new one"  not\_found: "not found"  not\_locked: "was not locked"  not\_saved:  one: "1 error prohibited this %{resource} from being saved:"  other: "%{count} errors prohibited this %{resource} from being saved:" |

**Файл “config/locales/devise.views.ru.yml”**

|  |
| --- |
| ru:  activerecord:  attributes:  user:  confirmation\_sent\_at: Дата отправки подтверждения  confirmation\_token: Токен подтверждения  confirmed\_at: Дата подтверждения  created\_at: Дата создания  current\_password: Текущий пароль  current\_sign\_in\_at: Дата текущего входа  current\_sign\_in\_ip: IP текущего входа  email: Email  encrypted\_password: Зашифрованный пароль  failed\_attempts: Неудачные попытки  last\_sign\_in\_at: Дата последнего входа  last\_sign\_in\_ip: IP последнего входа  locked\_at: Дата блокировки  password: Пароль  password\_confirmation: Подтверждение пароля  remember\_created\_at: Дата создания запоминания  remember\_me: Запомнить меня  reset\_password\_sent\_at: Дата отправки сброса пароля  reset\_password\_token: Токен сброса пароля  sign\_in\_count: Количество входов  unconfirmed\_email: Неподтвержденный email  unlock\_token: Токен разблокировки  updated\_at: Дата обновления  models:  user:  few: Пользователя  many: Пользователей  one: Пользователь  other: Пользователи  devise:  confirmations:  confirmed: Ваш адрес эл. почты успешно подтвержден.  new:  resend\_confirmation\_instructions: Выслать повторно письмо с активацией  send\_instructions: В течение нескольких минут вы получите письмо с инструкциями по подтверждению адреса эл. почты.  send\_paranoid\_instructions: Если ваш адрес эл. почты есть в системе, то в течение нескольких минут вы получите письмо с инструкциями по его подтверждению.  failure:  already\_authenticated: Вы уже вошли в систему.  inactive: Ваша учетная запись еще не активирована.  invalid: Неправильный %{authentication\_keys} или пароль.  last\_attempt: У вас осталась последняя попытка ввода пароля до блокировки учетной записи.  locked: Ваша учетная запись заблокирована.  not\_found\_in\_database: Неправильный %{authentication\_keys} или пароль.  timeout: Ваш сеанс закончился. Пожалуйста, войдите в систему снова.  unauthenticated: Вам необходимо войти в систему или зарегистрироваться.  unconfirmed: Вы должны подтвердить вашу учетную запись.  mailer:  confirmation\_instructions:  action: Активировать  greeting: Здравствуйте, %{recipient}!  instruction: 'Вы можете активировать свою учетную запись, нажав ссылку снизу:'  subject: Инструкции по подтверждению учетной записи  email\_changed:  greeting: Здравствуйте, %{recipient}!  message: Мы пытаемся связаться с вами, что бы сообщить, что ваш email был изменен на %{email}.  message\_unconfirmed:  subject: Email изменен  password\_change:  greeting: Приветствуем, %{recipient}!  message: Мы пытаемся связаться с вами, что бы сообщить, что ваш пароль был изменен.  subject: Пароль изменен  reset\_password\_instructions:  action: Изменить пароль  greeting: Здравствуйте, %{recipient}!  instruction: 'Вы (или кто-то еще) запросили изменение пароля. Для изменения пароля нажмите ссылку ниже:'  instruction\_2: Если вы не запрашивали изменение пароля - проигнорируйте это сообщение  instruction\_3: Ваш пароль не изменится пока вы не нажмете на ссылку и не введете новый.  subject: Инструкции по восстановлению пароля  unlock\_instructions:  action: Разблокировать учетную запись  greeting: Здравствуйте, %{recipient}!  instruction: 'Нажмите ссылку для активации учетной записи:'  message: Ваша учетная запись была заблокирована в связи с превышением лимита неудачных попыток входа.  subject: Инструкции по разблокировке учетной записи  omniauth\_callbacks:  failure: Вы не можете войти в систему с учетной записью из %{kind}, т.к. "%{reason}".  success: Вход в систему выполнен с учетной записью из %{kind}.  passwords:  edit:  change\_my\_password: Изменить мой пароль  change\_your\_password: Изменить пароль  confirm\_new\_password: Повторите новый пароль  new\_password: Новый пароль  new:  forgot\_your\_password: Забыли пароль?  send\_me\_reset\_password\_instructions: Выслать новый пароль  no\_token: Доступ к этой странице возможен только по ссылке из письма о восстановлении пароля. Если вы пришли по такой ссылке, пожалуйста, убедитесь, что Вы скопировали всю ссылку целиком.  send\_instructions: В течение нескольких минут вы получите письмо с инструкциями по восстановлению пароля.  send\_paranoid\_instructions: Если ваш адрес эл. почты есть в системе, то в течение нескольких минут вы получите письмо с инструкциями по восстановлению пароля.  updated: Ваш пароль изменен. Теперь вы вошли в систему.  updated\_not\_active: Ваш пароль изменен.  registrations:  destroyed: До свидания! Ваша учетная запись удалена. Надеемся снова увидеть вас.  edit:  are\_you\_sure: Вы уверены?  cancel\_my\_account: Удалить учетную запись  currently\_waiting\_confirmation\_for\_email: 'Ожидается подтверждение адреса E-mail: %{email}'  leave\_blank\_if\_you\_don\_t\_want\_to\_change\_it: оставьте поле пустым, если не хотите его менять  title: Редактировать %{resource}  unhappy: Разочарованы?  update: Обновить  we\_need\_your\_current\_password\_to\_confirm\_your\_changes: введите текущий пароль для подтверждения изменений  new:  sign\_up: Регистрация  signed\_up: Добро пожаловать! Вы успешно зарегистрировались.  signed\_up\_but\_inactive: Вы успешно зарегистрированы. Однако, вы не можете войти в систему, потому что ваша учетная запись не активирована.  signed\_up\_but\_locked: Вы успешно зарегистрированы. Однако, вы не можете войти в систему, потому что ваша учетная запись заблокирована.  signed\_up\_but\_unconfirmed: Письмо со ссылкой для подтверждения было отправлено на ваш адрес эл. почты. Пожалуйста, перейдите по ссылке, чтобы подтвердить учетную запись.  update\_needs\_confirmation: Вы успешно обновили данные учетной записи, но необходимо подтвердить новый адрес эл. почты. Пожалуйста, проверьте почтовый ящик и перейдите по ссылке, чтобы закончить процедуру проверки нового адреса эл. почты.  updated: Ваша учетная запись изменена.  updated\_but\_not\_signed\_in: Ваш аккаунт успешно обновлен, но так как пароль был изменен, необходимо выполнить вход еще раз  sessions:  already\_signed\_out: Выход из системы уже выполнен.  new:  sign\_in: Войти  signed\_in: Вход в систему выполнен.  signed\_out: Выход из системы выполнен.  shared:  links:  back: Назад  didn\_t\_receive\_confirmation\_instructions: Не получили подтверждение?  didn\_t\_receive\_unlock\_instructions: Не получили код разблокировки?  forgot\_your\_password: Забыли пароль?  sign\_in: Войти  sign\_in\_with\_provider: 'Войти с помощью: %{provider}'  sign\_up: Регистрация  minimum\_password\_length:  few: "(минимум %{count} знака)"  many: "(минимум %{count} знаков)"  one: "(минимум %{count} знак)"  other: "(минимум %{count} знаков)"  unlocks:  new:  resend\_unlock\_instructions: Выслать подтверждение заново  send\_instructions: В течение нескольких минут вы получите письмо с инструкциями по разблокировке учетной записи.  send\_paranoid\_instructions: Если ваша учетная запись существует, то в течение нескольких минут вы получите письмо с инструкциями по ее разблокировке.  unlocked: Ваша учетная запись разблокирована. Теперь вы можете войти в систему.  errors:  messages:  already\_confirmed: уже подтверждён. Пожалуйста, попробуйте войти в систему  confirmation\_period\_expired: должен быть подтвержден в течение %{period}, пожалуйста, повторите запрос на подтверждение  expired: устарела. Пожалуйста, запросите новую  not\_found: не найден  not\_locked: не заблокирован  not\_saved:  few: "Сохранение не удалось из-за %{count} ошибок:"  many: "%{resource}: сохранение не удалось из-за %{count} ошибок:"  one: "%{resource}: сохранение не удалось из-за %{count} ошибки:"  other: "%{resource}: сохранение не удалось из-за %{count} ошибки:" |

**Файл “db/schema.rb”**

|  |
| --- |
| ActiveRecord::Schema[7.1].define(version: 2023\_12\_08\_161150) do  create\_table "active\_storage\_attachments", force: :cascade do |t|  t.string "name", null: false  t.string "record\_type", null: false  t.bigint "record\_id", null: false  t.bigint "blob\_id", null: false  t.datetime "created\_at", null: false  t.index ["blob\_id"], name: "index\_active\_storage\_attachments\_on\_blob\_id"  t.index ["record\_type", "record\_id", "name", "blob\_id"], name: "index\_active\_storage\_attachments\_uniqueness", unique: true  end  create\_table "active\_storage\_blobs", force: :cascade do |t|  t.string "key", null: false  t.string "filename", null: false  t.string "content\_type"  t.text "metadata"  t.string "service\_name", null: false  t.bigint "byte\_size", null: false  t.string "checksum"  t.datetime "created\_at", null: false  t.index ["key"], name: "index\_active\_storage\_blobs\_on\_key", unique: true  end  create\_table "active\_storage\_variant\_records", force: :cascade do |t|  t.bigint "blob\_id", null: false  t.string "variation\_digest", null: false  t.index ["blob\_id", "variation\_digest"], name: "index\_active\_storage\_variant\_records\_uniqueness", unique: true  end  create\_table "dotfiles", force: :cascade do |t|  t.string "title"  t.text "description"  t.string "config\_type"  t.datetime "created\_at", null: false  t.datetime "updated\_at", null: false  t.integer "user\_id"  t.integer "cached\_scoped\_like\_votes\_total", default: 0  t.integer "cached\_scoped\_like\_votes\_score", default: 0  t.integer "cached\_scoped\_like\_votes\_up", default: 0  t.integer "cached\_scoped\_like\_votes\_down", default: 0  t.integer "cached\_weighted\_like\_score", default: 0  t.integer "cached\_weighted\_like\_total", default: 0  t.float "cached\_weighted\_like\_average", default: 0.0  t.integer "cached\_scoped\_bookmark\_votes\_total", default: 0  t.integer "cached\_scoped\_bookmark\_votes\_score", default: 0  t.integer "cached\_scoped\_bookmark\_votes\_up", default: 0  t.integer "cached\_scoped\_bookmark\_votes\_down", default: 0  t.integer "cached\_weighted\_bookmark\_score", default: 0  t.integer "cached\_weighted\_bookmark\_total", default: 0  t.float "cached\_weighted\_bookmark\_average", default: 0.0  t.index ["user\_id"], name: "index\_dotfiles\_on\_user\_id"  end  create\_table "users", force: :cascade do |t|  t.string "email", default: "", null: false  t.string "encrypted\_password", default: "", null: false  t.string "reset\_password\_token"  t.datetime "reset\_password\_sent\_at"  t.datetime "remember\_created\_at"  t.datetime "created\_at", null: false  t.datetime "updated\_at", null: false  t.string "username"  t.index ["email"], name: "index\_users\_on\_email", unique: true  t.index ["reset\_password\_token"], name: "index\_users\_on\_reset\_password\_token", unique: true  t.index ["username"], name: "index\_users\_on\_username", unique: true  end  create\_table "votes", force: :cascade do |t|  t.string "votable\_type"  t.integer "votable\_id"  t.string "voter\_type"  t.integer "voter\_id"  t.boolean "vote\_flag"  t.string "vote\_scope"  t.integer "vote\_weight"  t.datetime "created\_at", null: false  t.datetime "updated\_at", null: false  t.index ["votable\_id", "votable\_type", "vote\_scope"], name: "index\_votes\_on\_votable\_id\_and\_votable\_type\_and\_vote\_scope"  t.index ["votable\_type", "votable\_id"], name: "index\_votes\_on\_votable"  t.index ["voter\_id", "voter\_type", "vote\_scope"], name: "index\_votes\_on\_voter\_id\_and\_voter\_type\_and\_vote\_scope"  t.index ["voter\_type", "voter\_id"], name: "index\_votes\_on\_voter"  end  add\_foreign\_key "active\_storage\_attachments", "active\_storage\_blobs", column: "blob\_id"  add\_foreign\_key "active\_storage\_variant\_records", "active\_storage\_blobs", column: "blob\_id"  end |

**Файл “spec/shared\_contexts.rb”**

|  |
| --- |
| RSpec.shared\_context "api request authentication helper methods" do  def sign\_in(user)  login\_as(user, scope: :user)  end  def sign\_out  logout(:user)  end  end |

**Файл “spec/factories/dotfile.rb”**

|  |
| --- |
| FactoryBot.define do  factory :dotfile do  title { Faker::Lorem.sentence(word\_count: 3) }  description { Faker::Lorem.paragraph(sentence\_count: 3) }  config\_type { "fish" }  user { FactoryBot.create(:user) }  file { Rack::Test::UploadedFile.new('spec/fixtures/config.fish', 'text/fish') }  end  end |

**Файл “spec/factories/user.rb”**

|  |
| --- |
| FactoryBot.define do  password = Faker::Internet.password  factory :user do  username { Faker::Internet.username(specifier: 6..10) }  email { Faker::Internet.email }  password { password }  password\_confirmation { password }  end  end |

**Файл “spec/unit/dotfile\_spec.rb”**

|  |
| --- |
| require 'rails\_helper'  RSpec.describe Dotfile, type: :model do  before(:each) do  @user = User.create!(  username: Faker::Internet.username(specifier: 6..10),  email: Faker::Internet.email,  password: "password",  encrypted\_password: "password"  )  end  subject do  described\_class.new(  title: Faker::Lorem.sentence(word\_count: 3),  description: Faker::Lorem.paragraph(sentence\_count: 3),  config\_type: "fish",  user\_id: @user.id,  file: fixture\_file\_upload(Rails.root.join('spec', 'fixtures', 'config.fish'), 'text/plain')  )  end    it "is valid with valid attributes" do  expect(subject).to be\_valid  end  it "is not valid when no title is entered" do  subject.title = nil  expect(subject).not\_to be\_valid  end  it "is not valid when no description is entered" do  subject.description = nil  expect(subject).not\_to be\_valid  end  it "is not valid when no config\_type is provided" do  subject.config\_type = nil  expect(subject).not\_to be\_valid  end  it "is not valid when no user\_id is given" do  subject.user\_id = nil  expect(subject).not\_to be\_valid  end  it "should belong to a user" do  expect(subject.user).to eq(@user)  end  it "is not valid when title has invalid symbols" do  subject.title = "title&"  expect(subject).not\_to be\_valid  end  it "is not valid when description has invalid symbols" do  subject.description = "description\*"  expect(subject).not\_to be\_valid  end  end |

**Файл “spec/unit/user\_spec.rb”**

|  |
| --- |
| require 'rails\_helper'  RSpec.describe User, type: :model do  subject do  described\_class.new(  username: Faker::Internet.username(specifier: 6..10),  email: Faker::Internet.email,  password: "password",  encrypted\_password: "password"  )  end  it "is valid with valid attributes" do  expect(subject).to be\_valid  end  it "is not valid when email is not entered" do  subject.email = nil  expect(subject).not\_to be\_valid  end  it "is not valid when username is not entered" do  subject.username = nil  expect(subject).not\_to be\_valid  end  it "is valid when encrypted password is not entered" do  subject.encrypted\_password = nil  expect(subject).to be\_valid  end  it "is not valid when username has spaces" do  subject.username = "user name"  expect(subject).not\_to be\_valid  end  it "is not valid when username has invalid symbols" do  subject.username = "username^"  expect(subject).not\_to be\_valid  end  it "is not valid when email has invalid format" do  subject.email = "email"  expect(subject).not\_to be\_valid  end  it "is not valid when email has invalid symbols" do  subject.email = "ma&l@gmail.com"  expect(subject).not\_to be\_valid  end  end |

**Файл “spec/integration/dotfile\_spec.rb”**

|  |
| --- |
| require 'rails\_helper'  require 'shared\_contexts'  RSpec.describe "Dotfile Tests", type: :feature do  include\_context "api request authentication helper methods"  let(:user) { FactoryBot.create(:user) }  let(:dotfile) { FactoryBot.create(:dotfile, user: user) }  before do  sign\_in user  end  scenario "create new dotfile" do  visit new\_dotfile\_path    expect(page).to have\_content("Creating")  title = Faker::Lorem.sentence(word\_count: 3)  description = Faker::Lorem.paragraph(sentence\_count: 3)  find("#title").set(title)  find("#description").set(description)  find("#config\_type").select("fish")  find("#file").attach\_file("spec/fixtures/config.fish")  find("#create\_button").click    expect(page).to have\_content("Dotfile was successfully created.")  expect(page).to have\_content(title)  expect(page).to have\_content(description)  expect(page).to have\_content("fish")  end  scenario "edit title" do  visit edit\_dotfile\_path(id: dotfile.id)  expect(page).to have\_content("Editing")  title = Faker::Lorem.sentence(word\_count: 3)  find("#title").set(title)  click\_on "Apply"  expect(page).to have\_content("Dotfile was successfully updated.")  expect(page).to have\_content(title)  end  scenario "edit description" do  visit edit\_dotfile\_path(id: dotfile.id)  expect(page).to have\_content("Editing")  description = Faker::Lorem.paragraph(sentence\_count: 3)  find("#description").set(description)  click\_on "Apply"  expect(page).to have\_content("Dotfile was successfully updated.")  expect(page).to have\_content(description)  end  scenario "edit config type" do  visit edit\_dotfile\_path(id: dotfile.id)  expect(page).to have\_content("Editing")  config\_type = "tmux"  find("#config\_type").select(config\_type)  click\_on "Apply"  expect(page).to have\_content("Dotfile was successfully updated.")  expect(page).to have\_content(config\_type)  end  end |

**Файл “spec/integration/user\_spec.rb”**

|  |
| --- |
| require 'rails\_helper'  require 'shared\_contexts'  RSpec.describe "User Tests", type: :feature do  include\_context "api request authentication helper methods"  let(:user) { FactoryBot.create(:user) }  scenario "valid sign up" do  visit new\_user\_registration\_path  fill\_in "Username", with: "#{Faker::Internet.username(specifier: 6..10)}"  fill\_in "Email", with: "#{Faker::Internet.email}"  password = Faker::Internet.password  fill\_in "Password", with: "#{password}"  fill\_in "Confirm Password", with: "#{password}"  click\_on "Sign Up"  expect(page).to have\_content("Welcome! You have signed up successfully.")  end  scenario "valid log in" do  visit new\_user\_session\_path  fill\_in "Email", with: user.email  fill\_in "Password", with: user.password  click\_on "Log In"  expect(page).to have\_content("Signed in successfully.")  end  scenario "change user email" do  sign\_in user  visit edit\_user\_registration\_path  fill\_in "New Email", with: "#{Faker::Internet.email}"  fill\_in "Current Password", with: "#{user.password}"  click\_on "Apply"  visit dotfiles\_path  expect(page).to have\_content("Your account has been updated successfully.")  end  scenario "change user password" do  sign\_in user  visit edit\_user\_registration\_path  new\_password = Faker::Internet.password  fill\_in "New Password", with: "#{new\_password}"  fill\_in "Confirm New Password", with: "#{new\_password}"  fill\_in "Current Password", with: "#{user.password}"  click\_on "Apply"  visit dotfiles\_path  expect(page).to have\_content("Your account has been updated successfully.")  end  end |

**Файл “Gemfile”**

|  |
| --- |
| source "https://rubygems.org"  ruby "3.2.2"  gem 'sassc-rails'  # Bundle edge Rails instead: gem "rails", github: "rails/rails", branch: "main"  gem "rails", "~> 7.1.1"  gem 'rails-i18n'  # The original asset pipeline for Rails [https://github.com/rails/sprockets-rails]  gem "sprockets-rails"  # Use sqlite3 as the database for Active Record  gem "sqlite3", "~> 1.4"  # Use the Puma web server [https://github.com/puma/puma]  gem "puma", ">= 5.0"  # Use JavaScript with ESM import maps [https://github.com/rails/importmap-rails]  gem "importmap-rails"  # Hotwire's SPA-like page accelerator [https://turbo.hotwired.dev]  gem "turbo-rails"  # Hotwire's modest JavaScript framework [https://stimulus.hotwired.dev]  gem "stimulus-rails"  # Build JSON APIs with ease [https://github.com/rils/jbuilder]  gem "jbuilder"  # Use Redis adapter to run Action Cable in production  # gem "redis", ">= 4.0.1"  # Use Kredis to get higher-level data types in Redis [https://github.com/rails/kredis]  # gem "kredis"  # Use Active Model has\_secure\_password [https://guides.rubyonrails.org/active\_model\_basics.html#securepassword]  # gem "bcrypt", "~> 3.1.7"  # Windows does not include zoneinfo files, so bundle the tzinfo-data gem  gem "tzinfo-data", platforms: %i[ windows jruby ]  # Reduces boot times through caching; required in config/boot.rb  gem "bootsnap", require: false  # Use Active Storage variants [https://guides.rubyonrails.org/active\_storage\_overview.html#transforming-images]  # gem "image\_processing", "~> 1.2"  gem 'activestorage', '~> 7.1', '>= 7.1.1'  gem 'devise', '~> 4.9', '>= 4.9.3'  gem 'devise-i18n', '~> 1.12.0', ">= 1.12.0"  group :development, :test do  # See https://guides.rubyonrails.org/debugging\_rails\_applications.html#debugging-with-the-debug-gem  gem "debug", platforms: %i[ mri windows ]  end  group :development do  # Use console on exceptions pages [https://github.com/rails/web-console]  gem "web-console"  # Add speed badges [https://github.com/MiniProfiler/rack-mini-profiler]  # gem "rack-mini-profiler"  # Speed up commands on slow machines / big apps [https://github.com/rails/spring]  # gem "spring"  end  group :test do  # Use system testing [https://guides.rubyonrails.org/testing.html#system-testing]  gem "capybara"  gem "selenium-webdriver"  gem "rspec-rails"  gem "factory\_bot\_rails"  gem "faker"  end  gem "ransack", "~> 4.1"  gem "pagy", "~> 6.1"  gem "acts\_as\_votable", "~> 0.14.0" |

**Работа программы**

A screenshot of a computer

Description automatically generated

**Рисунок 1** – домашняя страница

A screenshot of a computer

Description automatically generated

**Рисунок 2** – страница просмотра конфигов

A screenshot of a computer

Description automatically generated

**Рисунок 3** – создание конфига

A screenshot of a computer

Description automatically generated

**Рисунок 4** – редактирование конфига

A screenshot of a login form

Description automatically generated

**Рисунок 5** – регистрация аккаунта

A screenshot of a login screen

Description automatically generated

**Рисунок 6** – вход в аккаунт

A screenshot of a computer

Description automatically generated

**Рисунок 7** – просмотр профиля

A screenshot of a computer

Description automatically generated

**Рисунок 8** – редактирование профиля

A screenshot of a computer

Description automatically generated

**Рисунок 9** – страница сохраненных конфигов

A screenshot of a computer

Description automatically generated

**Рисунок 10** – смена локали сайта

A screenshot of a computer

Description automatically generated

**Рисунок 11** – проверка исходного кода контроллеров rubocop’ом

A screenshot of a computer

Description automatically generated

**Рисунок 12** – выполнение тестов

A screenshot of a computer

Description automatically generated

**Рисунок 13** – просмотр конфига

**Вывод**

Разработан и реализован сайт, удовлетворяющий условиям задания, а также написаны тесты (юнит-тесты и интеграционные тесты) для проверки функционала и корректной работы приложения; проведена проверка исходного кода с использованием линтера “rubocop” для обеспечения соблюдения стандартов кода.