



**«Московский государственный технический университет  
имени Н.Э. Баумана»**

**(национальный исследовательский университет)**

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

## **О т ч е т**

**по Зачетной работе**

**Дисциплина :Языки Интернет-программирования**

Студент гр. ИУ6 346

\_\_\_\_\_  
(Подпись, дата)

**Погосян В.В**

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия)

**Основная задача:** создание веб-приложения для размещения статей.

Поставленные задачи:

- 1.Добавить форму аутентификации пользователей.
- 2.Реализовать возможность редактирование данных пользователя.
- 3.Функционал для сброса пароль пользователя.
- 4.Функционал для смены языка страницы en/ru .
- 5.Добавить форму создания статей.
- 6.Реализовать возможность добавления изображений в статью.
- 7.Добавить форму редактирования статей, пользователем-создателем.
- 8.Реализовать возможность удаления статей, пользователем-создателем.
- 9.Реализовать возможность добавления комментариев.

### Код файла application.js

```
require("@rails/ujs").start()
require("turboLinks").start()
require("@rails/activestorage").start()
require("channels")
```

### Код файла application.scss

```
@import "bootstrap";
```

### Код файла main.sass

```
.main
  padding-top: 1em

.field_with_errors
  label
    color: #d01612
  input, textarea
    border-color: #d01612

.tag_cloud
  margin-top: 2em

.submenu
  ul.subnav
    list-style: none
    padding-left: 0

ul.nested_nav
```

```
padding-left: 1em
```

### **Код файла pagination.sass**

```
.pagination ul
  padding: 0

.page_delimeter
  background: #ececec
  text-align: center
  padding: .5em
```

### **Код файла post.sass**

```
article.post
  margin-bottom: 1em
  padding: 1em 0
  border-bottom: 1px solid lightblue
  h1, h2
    margin-bottom: 0
  section.post-summary
    padding: .5em 0
  section.post-body
    padding-top: 1em
  .img-show
    display: block
    margin-bottom: 2em
  .thumb
    float: left
    margin-right: 1em

.editor
  min-height: 165px
```

### **Код файла application\_controller.rb**

```
class ApplicationController < ActionController::Base
  protect_from_forgery with: :exception
  add_flash_types :success, :danger, :info, :warning
  before_action :set_locale

  private
```

```

def set_locale
  I18n.locale = params[:locale] || I18n.default_locale
end

def default_url_options(options = {})
  { locale: I18n.locale }.merge options
end

end

```

### **Код файла posts\_controller.rb**

```

class PostsController < ApplicationController

  before_action :authenticate_user!, except: [:index, :show]

  before_action :set_post, only: [ :show, :edit, :update, :destroy]

  def index
    @posts = Post.all
  end

  def myindex
    @posts = Post.all
    if current_user != nil
      @posts = Post.where('user_id== ?', current_user.id).order(:id)
    end
  end

  def show
  end

  def new
    @post = Post.new
  end

  def create
    @post = Post.new(post_params)
    @post.user_id = current_user.id
    if @post.save
      redirect_to @post, success: 'Post successfully created'
    else
      render :new, danger: 'Post not created'
    end
  end
end

```

```

    end
  end

  def edit
  end

  def update
    if @post.user_id == current_user.id && @post.update(post_params)
      redirect_to @post, success: 'Post successfully updated'
    else
      render :edit, danger: 'Post not updated'
    end
  end

  def destroy
    @post.destroy
    redirect_to posts_path, success: 'Post successfully deleted'
  end

  private

  def set_post
    @post = Post.find(params[:id])
  end

  def post_params
    params.require(:post).permit(:user_id, :title, :summary, :body, :image)
  end
end

```

### **Код файла post.rb**

```

class Post < ApplicationRecord
  mount_uploader :image, ImageUploader
  validates :title, :summary, :body, presence: true #проверка на заполненные поля
  belongs_to :user

end

```

### **Код файла user.rb**

```

class User < ApplicationRecord

```

```

# Include default devise modules. Others available are:
# :confirmable, :lockable, :timeoutable, :trackable and :omniauthable
devise :database_authenticatable, :registerable,
       :recoverable, :rememberable, :validatable, :confirmable
has_many :posts
end

```

## Код файла image\_uploader.rb

```

class ImageUploader < CarrierWave::Uploader::Base
  # Include RMagick or MiniMagick support:
  # include CarrierWave::RMagick
  include CarrierWave::MiniMagick

  # Choose what kind of storage to use for this uploader:
  storage :file
  # storage :fog

  # Override the directory where uploaded files will be stored.
  # This is a sensible default for uploaders that are meant to be mounted:
  def store_dir
    "uploads/#{model.class.to_s.underscore}/#{mounted_as}/#{model.id}"
  end

  # Provide a default URL as a default if there hasn't been a file uploaded:
  # def default_url(*args)
  #   # For Rails 3.1+ asset pipeline compatibility:
  #   # ActionController::Base.helpers.asset_path("fallback/" + [version_name,
"default.png"].compact.join('_'))
  #
  #   "/images/fallback/" + [version_name, "default.png"].compact.join('_')
  # end

  # Process files as they are uploaded:
  # process scale: [200, 300]
  #
  # def scale(width, height)
  #   # do something
  # end

  version :large do
    process resize_to_fit: [900, 900]
  end
end

```

```

version :thumb do
  process resize_to_fit: [200, 200]
end

# Add a white list of extensions which are allowed to be uploaded.
# For images you might use something like this:
def extension_whitelist
  %w(jpg jpeg gif png)
end

# Override the filename of the uploaded files:
# Avoid using model.id or version_name here, see uploader/store.rb for details.
# def filename
#   "something.jpg" if original_filename
# end
end

```

### Код файла `confirmations/new.html.erb`

```

<h2>Resend confirmation instructions</h2>

<%= form_for(resource, as: resource_name, url: confirmation_path(resource_name), html:
{ method: :post }) do |f| %>
  <%= render "devise/shared/error_messages", resource: resource %>

  <div class="field">
    <%= f.label :email %><br />
    <%= f.email_field :email, autofocus: true, autocomplete: "email", value:
(resource.pending_reconfirmation? ? resource.unconfirmed_email : resource.email) %>
  </div>

  <div class="actions">
    <%= f.submit "Resend confirmation instructions" %>
  </div>
<% end %>

<%= render "devise/shared/links" %>

```

### Код файла `passwords/edit.html.erb`

```

<h2><%= t('.chng') %></h2>

```

```

<%= form_for(resource, as: resource_name, url: password_path(resource_name), html:
{ method: :put }) do |f| %>
  <%= render "devise/shared/error_messages", resource: resource %>
  <%= f.hidden_field :reset_password_token %>

  <div class="form-group">
    <%= f.label :password, t('.np') %><br />
    <% if @minimum_password_length %>
      <em>(<%= @minimum_password_length %> <%= t('.min') %> </em><br />
    <% end %>
    <%= f.password_field :password, autofocus: true, autocomplete: "new-password", class:
'form-control' %>
  </div>

  <div class="form-group">
    <%= f.label :password_confirmation, t('.cnp') %><br />
    <%= f.password_field :password_confirmation, autocomplete: "new-password", class: 'form-
control' %>
  </div>

  <div class="actions">
    <%= f.submit t('.chng'), class: 'btn btn-success' %>
  </div>
<% end %>

<%= render "devise/shared/links" %>

```

## Код файла passwords/new.html.erb

```

<h2><%= t('.for') %></h2>

<%= form_for(resource, as: resource_name, url: password_path(resource_name), html:
{ method: :post }) do |f| %>
  <%= render "devise/shared/error_messages", resource: resource %>

  <div class="form-group">
    <%= f.label :email %><br />
    <%= f.email_field :email, autofocus: true, autocomplete: "email", class: 'form-control' %>
  </div>

  <div class="actions">
    <%= f.submit t('.send'), class: 'btn btn-success' %>
  </div>
<% end %>

```



```
<%= render "devise/shared/links" %>
```

## Код файла registrations/edit.html.erb

```
<h2><%= t('.ed') %> <%= resource_name.to_s.humanize %></h2>
```

```
<%= form_for(resource, as: resource_name, url: registration_path(resource_name), html:
{ method: :put }) do |f| %>
  <%= render "devise/shared/error_messages", resource: resource %>
```

```
<div class="form-group">
  <%= f.label :email %><br />
  <%= f.email_field :email, autofocus: true, autocomplete: "email", class: 'form-control' %>
</div>
```

```
<% if devise_mapping.confirmable? && resource.pending_reconfirmation? %>
  <div>Currently waiting confirmation for: <%= resource.unconfirmed_email %></div>
<% end %>
```

```
<div class="form-group">
  <%= f.label :password %> <i></i><br />
  <%= f.password_field :password, autocomplete: "new-password", class: 'form-control' %>
  <% if @minimum_password_length %>
    <br />
    <em><%= @minimum_password_length %> <%= t('.min')%></em>
  <% end %>
</div>
```

```
<div class="form-group">
  <%= f.label :password_confirmation %><br />
  <%= f.password_field :password_confirmation, autocomplete: "new-password", class: 'form-
control' %>
</div>
```

```
<div class="form-group">
  <%= f.label :current_password %> <i></i><br />
  <%= f.password_field :current_password, autocomplete: "current-password", class: 'form-
control' %>
</div>
```

```

<div class="actions">
  <%= f.submit t('.update'), class: 'btn btn-success' %>
  <%= link_to t('.back'), :back, class: 'btn btn-outline-secondary' %>
</div>
<% end %>

```

## Код файла registrations/new.html.erb

```

<h2><%= t('.sin') %></h2>

<%= form_for(resource, as: resource_name, url: registration_path(resource_name)) do |f| %>
  <%= render "devise/shared/error_messages", resource: resource %>

  <div class="form-group">
    <%= f.label :email %><br />
    <%= f.email_field :email, autofocus: true, autocomplete: "email", class: 'form-control' %>
  </div>

  <div class="form-group">
    <%= f.label :password %>
    <% if @minimum_password_length %>
    <em>(<%= @minimum_password_length %> characters minimum)</em>
    <% end %><br />
    <%= f.password_field :password, autocomplete: "new-password", class: 'form-control' %>
  </div>

  <div class="form-group">
    <%= f.label :password_confirmation %><br />
    <%= f.password_field :password_confirmation, autocomplete: "new-password", class: 'form-control' %>
  </div>

  <div class="actions">
    <%= f.submit t('.sinn'), class: 'btn btn-primary' %>
  </div>
<% end %>

```

## Код файла sessions/new.html.erb

```

<h1><%= t('.sin') %></h1>

<%= form_for(resource, as: resource_name, url: session_path(resource_name)) do |f| %>

```

```

<div class="form-group">
  <%= f.label :email %><br />
  <%= f.email_field :email, autofocus: true, autocomplete: "email", class: 'form-control' %>
</div>

<div class="form-group">
  <%= f.label :password %><br />
  <%= f.password_field :password, autocomplete: "current-password", class: 'form-control' %>
</div>

<% if devise_mapping.rememberable? %>
  <div class="form-group">
    <%= f.check_box :remember_me %>
    <%= f.label :remember_me %>
  </div>
<% end %>

<div class="actions">
  <%= f.submit t('.sinn'), class: 'btn btn-success' %>
  <%= link_to t('.for'), new_password_path(resource_name), class: 'btn btn-outline-secondary' %>
</div>
<% end %>

```

## Код файла application.html.erb

```

<!DOCTYPE html>
<html>
  <head>
    <title>Lab13</title>
    <%= csrf_meta_tags %>
    <%= csp_meta_tag %>

    <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track': 'reload' %>
    <%= javascript_include_tag 'application', 'data-turbolinks-track': 'reload' %>
  </head>

  <body>

    <div class="container">
      <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;">
        <div class="collapse navbar-collapse" id="navbarText">

```

```

<table>
  <tr>
    <th> <%= link_to t('.allp'), posts_path, class: 'btn btn-outline-primary' %> </th>
    <% if user_signed_in? %>
      <th> <%= link_to t('.newp'), new_post_path, class: 'btn btn-outline-primary' %>
</th>
    <% end %>
  </tr>
</table>
</div>

<div class="navbar-nav pull-xs-right">
  <table>
    <tr>
      <% if user_signed_in? %>
        <th> <a> <%= button_to t('.edit'), edit_user_registration_path, :method => "get",
class: "btn btn-outline-warning" %> </a> </th>
        <th> <a> <%= button_to t('.sout'), destroy_user_session_path, method: :delete, class:
"btn btn-outline-warning" %> </a> </th>
        <% else %>
          <th> <a> <%= link_to t('.sin'), new_user_session_path, class: "btn btn-outline-info"
%> </a> </th>
          <th> <a> <%= link_to t('.regg'), new_user_registration_path, class: "btn btn-outline-
info" %> </a> </th>
          <% end %>
          <th> <%= link_to "Pyc", { locale: :ru }, class: "btn btn-outline-secondary" %> </th>
          <th> <%= link_to "En", { locale: :en }, class: "btn btn-outline-secondary" %> </th>
        </tr>
      </table>
    </div>

</nav>
</div>

<div class="container main">
  <div class="row">
    <div class="col-md-12">
      <% flash.each do |name, message| %>
        <div class="alert <%= bootstrap_class_for(name) %>">
          <button class="close" data-dismiss="alert">&times</button>
          <%= message %>
        </div>
      <% end %>
    </div>
  </div>
</div>

```

```

    <%= yield %>
  </div>

</body>
</html>

```

## Код файла \_disqus.html.erb

```

<div id="disqus_thread"></div>
<script>
  var disqus_config = function () {
    <%= # <1%# this.page.url = '<%= request.base.url + request.original_fullpath %2>'; %>
    this.page.url = 'http://localhost.com<%= request.original_fullpath %>';
    this.page.identifier = '<%= @post.id %>';
  };
  (function() { // DON'T EDIT BELOW THIS LINE
    var d = document, s = d.createElement('script');
    s.src = '//blogdev.disqus.com/embed.js';
    s.setAttribute('data-timestamp', +new Date());
    (d.head || d.body).appendChild(s);
  })();
</script>
<noscript>Please enable JavaScript to view the <a href="https://disqus.com/?
ref_noscript">comments powered by Disqus.</a></noscript>

```

## Код файла \_form.html.erb

```

<% if @post.errors.any? %>
  <div class=" alert alert-danger">
    <h5> <%= t('common.errors') %> </h5>
    <ul>
      <% @post.errors.full_messages.each do |message| %>
        <li> <%= message %> </li>
      <% end %>
    </ul>
  </div>
<% end %>

<%= form_for @post, html: { multipart: true } do |f| %>
  <div class="form-group">
    <%= f.label :title %>
    <%= f.text_field :title, class: 'form-control' %>

```

```

</div>

<div class="form-group">
  <%= f.label :summary %>
  <%= f.text_area :summary, class: 'form-control' %>
</div>

<div class="form-group">
  <%= f.label :body %>
  <%= f.text_area :body, class: 'form-control editor' %>
</div>

<div class="form-group">
  <%= f.file_field :image, class: "btn btn-success"%>
</div>

<div class="form-group">
  <%= f.submit t('.submit', action: @post.new_record? ? t('common.create') :
t('common.edit')), class: 'btn btn-primary'%>
  <%= link_to t('common.back'), :back, class: 'btn btn-outline-secondary' %>
</div>

<% end %>

```

## Код файла \_post.html.erb

```

<article class="post">

  <section class="post-head">
    <h2><%= link_to post.title, post_path(post) %></h2>
    <small>
      <table>
        <tr>
          <th> <time><%= post.created_at.to_date.strftime("%d %B, %Y") %></time> </th>
          <% if user_signed_in? && post.user_id == current_user.id %>
            <th> <%= link_to t('.edit'), edit_post_path(post), :class => "btn btn-warning" %> </th>
            <th> <%= button_to t('.delete'), post_path(post), method: :delete, :class => 'btn btn-
danger' %> </th>
          <% end %>
        </tr>
      </table>
    </small>
  </section>

```

```

<section class="post-summary clearfix">
  <table>
    <tr>
      <th> <%= image_tag post.image.url(:thumb), class: 'img-show collapse' if post.image?
%> </th>
      <th class='font-weight-normal'> <%= post.summary %> </th>
    </tr>
  </table>
</section>

</article>

```

### Код файла posts/edit.html.erb

```

<h1><%= t('.title') %></h1>
<%= render 'form' %>

```

### Код файла index.html.erb

```

<h1><%= t('.title') %></h1>
<%= render @posts %>

```

### Код файла posts/new.html.erb

```

<h1><%= t('.title') %></h1>
<%= render 'form' %>

```

### Код файла show.html.erb

```

<article class="post">

  <section class="post-head">
    <h1><%= @post.title %></h1>
    <small>
      <table>
        <tr>
          <th> <time><%= @post.created_at.to_date.strftime("%d %B, %Y") %></time> </th>
          <% if user_signed_in? && @post.user_id == current_user.id %>
            <th> <%= link_to t('.edit'), edit_post_path(@post), :class => "btn btn-warning" %>
          </th>
          <th> <%= button_to t('.delete'), post_path(@post), method: :delete, :class => 'btn btn-

```

```

danger' %> </th>
  <% end %>
</tr>
</table>
</small>
</section>

<section class="post-body">
  <%= image_tag @post.image.url(:large), class: 'img-show collapse rounded mx-auto d-
block' if @post.image? %>
  <%= @post.body.html_safe %>
</section>

</article>

<%= render 'disqus' %>

```

## Код файла **schema.rb**

ActiveRecord::Schema.define(version: 2019\_12\_19\_211544) do

```

  create_table "posts", force: :cascade do |t|
    t.string "title"
    t.text "summary"
    t.text "body"
    t.datetime "created_at", precision: 6, null: false
    t.datetime "updated_at", precision: 6, null: false
    t.string "image"
    t.integer "user_id"
    t.index ["user_id"], name: "index_posts_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.string "confirmation_token"
    t.datetime "confirmed_at"
    t.datetime "confirmation_sent_at"
    t.string "unconfirmed_email"
    t.datetime "created_at", precision: 6, null: false
    t.datetime "updated_at", precision: 6, null: false
  end

```



```
t.index ["confirmation_token"], name: "index_users_on_confirmation_token", unique: true
t.index ["email"], name: "index_users_on_email", unique: true
t.index ["reset_password_token"], name: "index_users_on_reset_password_token", unique:
true
end

end
```

### Код файла activerecord.en.yml

```
en:
  activerecord:
    models:
      post: 'Post'
    attributes:
      post:
        title: 'Title'
        summary: 'Summary'
        body: 'Body'
      user:
        email: 'Email'
        password: 'Password'
        password_confirmation: 'Password confirmation'
        current_password: 'Current password'
        remember_me: 'remember me'
    errors:
      messages:
        blank: "can't be blank"
```

### Код файла activerecord.ru.yml

```
ru:
  activerecord:
    models:
      post: 'Статья'
    attributes:
      post:
        title: 'Название'
        summary: 'Краткая статья'
        body: 'Содержание'
      user:
        email: 'Почта'
        password: 'Пароль'
        password_confirmation: 'Подтвердите пароль'
```

```
current_password: 'Текущий пароль'
remember_me: 'запомнить меня'
errors:
  messages:
    blank: 'не может быть пустым'
```

## Код файла devise.en.yml

# Additional translations at <https://github.com/plataformatec/devise/wiki/I18n>

```
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:"

## Код файла devise.ru.yml

# Русский перевод для <https://github.com/plataformatec/devise/tree/v4.7.1>

# Другие переводы на <https://github.com/plataformatec/devise/wiki/I18n>

ru:

devise:

confirmations:

confirmed: "Ваша учётная запись успешно подтверждена."

send\_instructions: "В течение нескольких минут вы получите письмо с инструкциями по подтверждению вашей учётной записи."

send\_paranoid\_instructions: "Если ваш адрес email есть в нашей базе данных, то в течение нескольких минут вы получите письмо с инструкциями по подтверждению вашей учётной записи."

failure:

already\_authenticated: "Вы уже вошли в систему."

inactive: "Ваша учётная запись ещё не активирована."

invalid: "Неверный %{authentication\_keys} или пароль."

locked: "Ваша учётная запись заблокирована."

last\_attempt: "У Вас осталась еще одна попытка ввести пароль до блокировки."

not\_found\_in\_database: "Неверный %{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 учётную запись."

mailer:

confirmation\_instructions:

subject: "Инструкции по подтверждению учётной записи."

reset\_password\_instructions:

subject: "Инструкции по восстановлению пароля."

unlock\_instructions:

subject: "Инструкции по разблокировке учётной записи."

email\_changed:

subject: "Email адрес был изменён."

password\_change:

subject: "Пароль был изменён."

omniauth\_callbacks:

failure: "Вы не можете войти в систему с учётной записью из %{kind}, так как \"%{reason}\"."

success: "Вход в систему выполнен с учётной записью из %{\kind}."

passwords:

no\_token: "Эта страница доступна только при переходе с email для сброса пароля. Если Вы перешли по ссылке из письма, пожалуйста, убедитесь, что вы использовали полный URL."

send\_instructions: "В течение нескольких минут вы получите письмо с инструкциями по восстановлению вашего пароля."

send\_paranoid\_instructions: "Если ваш адрес email есть в нашей базе данных, то в течение нескольких минут Вы получите письмо с инструкциями по восстановлению вашего пароля."

updated: "Ваш пароль успешно изменён. Теперь вы вошли в систему."

updated\_not\_active: "Пароль успешно изменён."

registrations:

destroyed: "До свидания! Ваша учётная запись успешно удалена. Надеемся снова увидеть вас."

signed\_up: "Добро пожаловать! Вы успешно зарегистрировались."

signed\_up\_but\_inactive: "Вы успешно зарегистрировались. Тем не менее, вы не можете войти, потому что ваша учетная запись ещё не активирована."

signed\_up\_but\_locked: "Вы успешно зарегистрировались. Тем не менее, вы не можете войти, потому что ваша учетная запись заблокирована."

signed\_up\_but\_unconfirmed: "В течение нескольких минут вы получите письмо с инструкциями по подтверждению вашей учётной записи."

update\_needs\_confirmation: "Ваш аккаунт успешно обновлен, необходимо подтвердить Ваш новый адрес email. Пожалуйста, проверьте свою электронную почту и нажмите на ссылку \"Подтвердить\", чтобы завершить обновления email."

updated: "Ваша учётная запись успешно изменена."

updated\_but\_not\_signed\_in: "Ваш аккаунт успешно изменен, но так как ваш пароль изменился, нужно войти снова."

sessions:

signed\_in: "Вход в систему выполнен."

signed\_out: "Выход из системы выполнен."

already\_signed\_out: "Выход из системы выполнен."

unlocks:

send\_instructions: "В течение нескольких минут вы получите письмо с инструкциями по разблокировке вашей учётной записи."

send\_paranoid\_instructions: "Если ваша учётная запись существует, то в течение нескольких минут вы получите письмо с инструкциями по её разблокировке."

unlocked: "Ваша учётная запись разблокирована. Теперь вы вошли в систему."

errors:

messages:

already\_confirmed: "уже подтверждена. Пожалуйста, попробуйте войти в систему"

confirmation\_period\_expired: "должен быть подтвержден в течении %{\period}, пожалуйста, запросите подтверждение ещё раз"

expired: "устарела. Пожалуйста, запросите новую"

not\_found: "не найдена"

```
not_locked: "не заблокирована"
not_saved:
  one: "%{resource}: сохранение не удалось из-за %{count} ошибки"
  few: "%{resource}: сохранение не удалось из-за %{count} ошибок"
  many: "%{resource}: сохранение не удалось из-за %{count} ошибок"
  other: "%{resource}: сохранение не удалось из-за %{count} ошибки"
```

## Код файла en.yml

```
en:
  devise:
    passwords:
      new:
        for: 'Forgot you password?'
        send: 'Send me reset password instructions'
      edit:
        chng: 'Change your password'
        np: 'New password'
        cnp: 'Confirm new password'
        min: 'characters minimum)'
    sessions:
      new:
        sin: 'Log in'
        sinn: 'Log in'
        for: 'Forgot your password?'
    registrations:
      new:
        sin: 'Sign up'
        sinn: 'Sign up'
      edit:
        ed: 'Edit'
        back: 'Back'
        update: 'Update'
        min: 'characters minimum'
    common:
      errors: 'You have errors'
      back: 'Back'
      create: 'Create'
      edit: 'Edit'
  will_paginate:
    previous_label: '&larr;'
    next_label: '&rarr;'
  layouts:
    application:
      allp: 'All posts'
```

```
newp: 'New post'
sout: 'Log out'
sin: 'Log in'
regg: 'Sign up'
edit: 'Edit user data'
myp: 'My posts'
posts:
  show:
    edit: 'Edit post'
    delete: 'Delete post'
  index:
    title: 'Posts'
  new:
    title: 'New post'
  edit:
    title: 'Edit post'
  form:
    submit:
      '%{action} post'
  post:
    edit: 'Edit post'
    delete: 'Delete post'
```

## Код файла ru.yml

```
ru:
  devise:
    passwords:
      new:
        for: 'Забыли пароль?'
        send: 'Отправьте мне инструкции по сбросу пароля'
      edit:
        chng: 'Поменяйте ваш пароль'
        np: 'Новый пароль'
        snp: 'Подтвердите новый пароль'
        min: 'символов минимум)'
    sessions:
      new:
        sin: 'Вход'
        sinn: 'Войти'
        for: 'Забыли пароль?'
    registrations:
      new:
        sin: 'Регистрация'
```

```
sinn: 'Зарегистрироваться'
edit:
  ed: 'Изменить данные'
  back: 'Назад'
  update: 'Обновить данные'
  min: 'символов минимум'
common:
  errors: 'При заполнении формы возникли ошибки'
  back: 'Назад'
  create: 'Создать'
  edit: 'Изменить'
will_paginate:
  previous_label: '&larr; Туда'
  next_label: 'Сюда &rarr;'
layouts:
  application:
    allp: 'Все статьи'
    newp: 'Новая статья'
    sout: 'Выход'
    sin: 'Вход'
    regg: 'Регистрация'
    edit: 'Редактировать данные пользователя'
    мур: 'Мои статьи'
posts:
  post:
    edit: 'Редактировать статью'
    delete: 'Удалить статью'
  show:
    edit: 'Редактировать статью'
    delete: 'Удалить статью'
  index:
    title: 'Статьи'
  new:
    title: 'Новая статья'
  edit:
    title: 'Редактирование статьи'
  form:
    submit:
      '%{action} статью'
```

## Код файла routes.rb

Rails.application.routes.draw do



```

devise_for :users
scope "(:locale)", locale: /#{I18n.available_locales.join("|")}/ do
  root 'posts#index'
  resources :posts
end

end

```

## Код файла Gemfile

```

source 'https://rubygems.org'
git_source(:github) { |repo| "https://github.com/#{repo}.git" }

ruby '2.6.3'

# Use CoffeeScript for .coffee assets and views
gem 'coffee-rails', '~> 4.2'
# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '~> 6.0.1'
# Use sqlite3 as the database for Active Record
gem 'sqlite3', '~> 1.4'
# Use Puma as the app server
gem 'puma', '~> 4.1'
# Use SCSS for stylesheets
gem 'sass-rails', '>= 6'
# Transpile app-like JavaScript. Read more: https://github.com/rails/webpacker
gem 'webpacker', '~> 4.0'
# Turbolinks makes navigating your web application faster. Read more:
https://github.com/turbolinks/turbolinks
gem 'turbolinks', '~> 5'
# Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
gem 'jbuilder', '~> 2.7'
# Use Redis adapter to run Action Cable in production
# gem 'redis', '~> 4.0'
# Use Active Model has_secure_password
# gem 'bcrypt', '~> 3.1.7'

# Use Active Storage variant
# gem 'image_processing', '~> 1.2'

gem 'bootstrap', '~> 4.0.0.alpha4'
source 'https://rails-assets.org' do
  gem 'rails-assets-tether', '>= 1.1.0'
end

```

```
gem 'jquery-rails'
gem 'carrierwave', '~> 2.0'
gem "mini_magick"
gem 'trix'
gem 'will_paginate', '~> 3.1.0'
gem 'ancestry'
gem 'devise'
```

```
# Reduces boot times through caching; required in config/boot.rb
gem 'bootsnap', '>= 1.4.2', require: false
```

```
group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug', platforms: [:mri, :mingw, :x64_mingw]
end
```

```
group :development do
  # Access an interactive console on exception pages or by calling 'console' anywhere in the
  code.
  gem 'web-console', '>= 3.3.0'
  gem "letter_opener"
end
```

```
group :test do
  # Adds support for Capybara system testing and selenium driver
  gem 'capybara', '>= 2.15'
  gem 'selenium-webdriver'
  # Easy installation and use of web drivers to run system tests with browsers
  gem 'webdrivers'
end
```

```
# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64_mingw, :jruby]
```

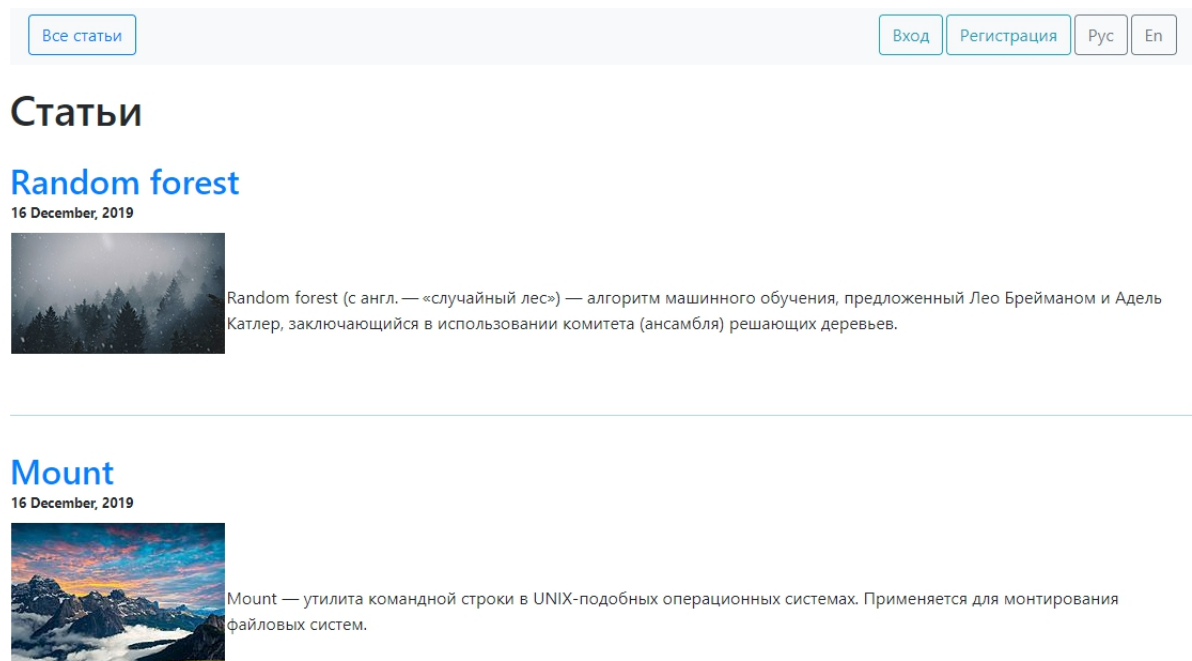


Рис.1 — главная страница

Все статьи

Вход Регистрация Рус En

## Вход

Почта

vilen@mail.r

Пароль

.....

☐ запомнить меня

Войти Забыли пароль?

Рис.2 — вход на сайт.

[Все статьи](#)[Вход](#)[Регистрация](#)[Рус](#)[En](#)

## Регистрация

Почта

Пароль (*6 characters minimum*)

Подтвердите пароль

[Зарегистрироваться](#)

Рис.3 — регистрация на сайте.

[Все статьи](#)

[Вход](#)[Регистрация](#)[Рус](#)[En](#)

## Забыли пароль?

Почта

Отправьте мне инструкции по сбросу пароля

Рис.4 — форма для сброса пароля.

Все статьи

Новая статья

Редактировать данные пользователя

Выход

Рус

En

## Изменить данные User

Почта

vilen@mail.r

Пароль

6 символов минимум

Подтвердите пароль

Текущий пароль

.....

Обновить данные

Назад

Рис.5 — форма для редактирования данных пользователя.

Все статьи

Новая статья

Редактировать данные пользователя

Выход

Рус

En

## Новая статья

Название

Краткая статья

Содержание

Выберите файл

Файл не выбран

Создать статью

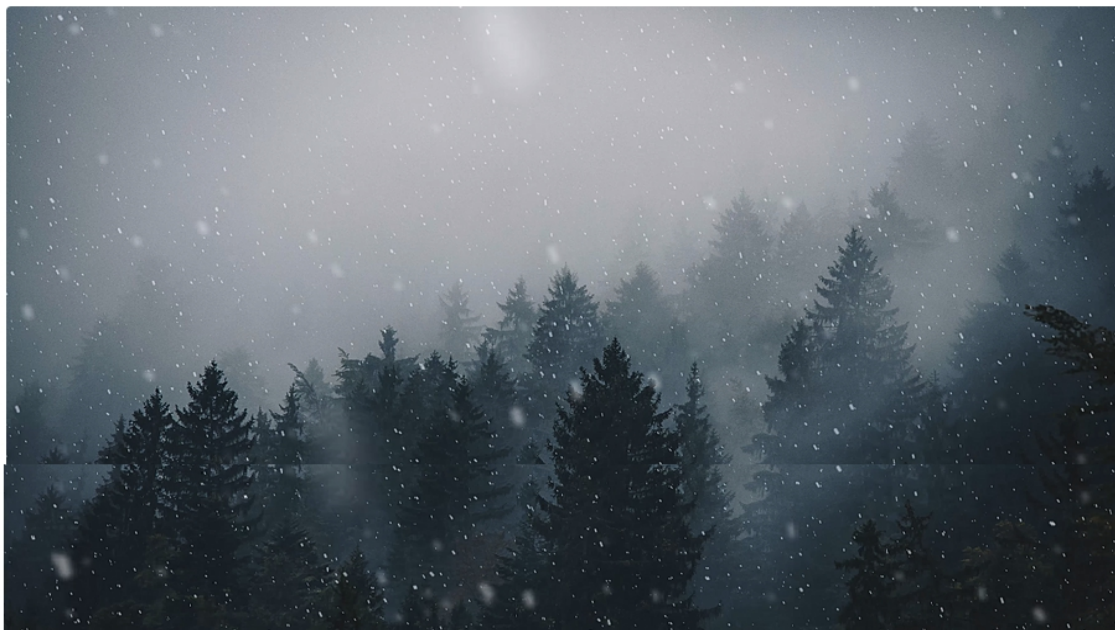
Назад

Рис.7 — форма для создания статьи.

[Все статьи](#)[Новая статья](#)[Редактировать данные пользователя](#)[Выход](#)[Рус](#)[En](#)

# Random forest

16 December, 2019



Random forest (с англ. — «случайный лес») — алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана, и метод случайных подпространств, предложенный Tin Kam Ho. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

1 Comment   Blogdev

1 Vilenchik ▾

♥ Recommend

🐦 Tweet

📌 Share

Sort by Best ▾



Join the discussion...



Vilenchik · 3 days ago



1 ^ | ▾ · Edit · Reply · Share ▸

ALSO ON BLOGDEV

Рис.8 — просмотр сатчи.



Все статьи

Новая статья

Редактировать данные пользователя

Выход

Рус

En

## Редактирование статьи

Название

Random forest

Краткая статья

Random forest (с англ. — «случайный лес») — алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев.

Содержание

Random forest (с англ. — «случайный лес») — алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана, и метод случайных подпространств, предложенный Tin Kam Ho. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

Выберите файл

Файл не выбран

Изменить статью

Назад

Рис.9 — форма редактирования сатьи.

```
D:\Desktop>rubocop application_controller.rb
Inspecting 1 file
C

Offenses:

application_controller.rb:1:1: C: Layout/EndOfLine: Carriage return character missing.
class ApplicationController < ActionController::Base ...
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
application_controller.rb:1:1: C: Style/Documentation: Missing top-level class documentation comment.
class ApplicationController < ActionController::Base
^^^^^
application_controller.rb:1:1: C: Style/FrozenStringLiteralComment: Missing magic comment # frozen_string_literal: true.

class ApplicationController < ActionController::Base
^
application_controller.rb:15:1: C: Layout/EmptyLinesAroundClassBody: Extra empty line detected at class body end.

1 file inspected, 4 offenses detected
```

Рис.10 — отчет rubocop для контроллера application\_controller.rb

