**Описание проекта**

Реализовать сайт, на котором можно создавать посты и комментировать их. Для того, чтобы создать пост или оставить комментарий, необходимо зарегистрироваться и выполнить авторизацию.

**Код изменённых файлов:**

**posts\_controller.rb**

class ApplicationController < ActionController::Base

  include SessionHelper

  private

  def require\_login

    return if signed\_in?

    flash[:danger] = 'Требуется логин'

    redirect\_to session\_login\_url

  end

end

**application\_controller.rb**

class PostsController < ApplicationController

  before\_action :set\_post, only: %i[show edit update destroy]

  before\_action :require\_login, only: %i[create new update destroy comment]

  # GET /posts

  # GET /posts.json

  def index

    @posts = Post.all

  end

  # GET /posts/1

  # GET /posts/1.json

  def show

    @comments = Comment.all.find\_all { |i| i.post\_id == @post.id }

  end

  # GET /posts/new

  def new

    @post = Post.new

  end

  # GET /posts/1/edit

  def edit; end

  # POST /posts

  # POST /posts.json

  def create

    @post = Post.new(title: params[:title], text: params[:text], user\_id: current\_user.id)

    current\_user.update\_posts\_count

    respond\_to do |format|

      if @post.save

        format.html { redirect\_to @post, notice: 'Post was successfully created.' }

        format.json { render :show, status: :created, location: @post }

      else

        format.html { render :new }

        format.json { render json: @post.errors, status: :unprocessable\_entity }

      end

    end

  end

  # PATCH/PUT /posts/1

  # PATCH/PUT /posts/1.json

  def update

    respond\_to do |format|

      if @post.update(title: params[:title], text: params[:text])

        format.html { redirect\_to @post, notice: 'Post was successfully updated.' }

        format.json { render :show, status: :ok, location: @post }

      else

        format.html { render :edit }

        format.json { render json: @post.errors, status: :unprocessable\_entity }

      end

    end

  end

  # DELETE /posts/1

  # DELETE /posts/1.json

  def destroy

    Comment.all.find\_all { |i| i.post\_id == @post.id }.each(&:destroy)

    @post.destroy

    respond\_to do |format|

      format.html { redirect\_to posts\_url, notice: 'Post was successfully destroyed.' }

      format.json { head :no\_content }

    end

  end

  def comment

    @post = Post.find(params[:id])

    @text = params[:text]

    if current\_user && @text && @post

      @comment = Comment.new(text: @text, post\_id: @post.id, user\_id: current\_user.id)

      @comment.save

    end

    if @post then redirect\_to @post

    else redirect\_to root\_path

    end

  end

  private

  # Use callbacks to share common setup or constraints between actions.

  def set\_post

    @post = Post.find(params[:id])

  end

  # Only allow a list of trusted parameters through.

  def post\_params

    params.require(:post).permit(:title, :text)

  end

end

**session\_controller.rb**

class SessionController < ApplicationController

  def login; end

  def create

    user = User.find\_by(username: params[:login])

    if user&.authenticate(params[:password])

      sign\_in user

      redirect\_to root\_path

    else

      flash.now[:danger] = 'Неверный логин или пароль'

      redirect\_to session\_login\_path

    end

  end

  def logout

    sign\_out

    redirect\_to session\_login\_path

  end

end

**users\_controller.rb**

class UsersController < ApplicationController

  before\_action :set\_user, only: %i[show edit update destroy]

  # GET /users

  # GET /users.json

  def index

    if !current\_user&.is\_admin

      redirect\_to root\_path

    else

      @users = User.all

    end

  end

  # GET /users/1

  # GET /users/1.json

  def show

    redirect\_to root\_path unless current\_user&.is\_admin

  end

  # GET /users/new

  def new

    @user = User.new

  end

  # GET /users/1/edit

  def edit; end

  # POST /users

  # POST /users.json

  def create

    @user = User.new(user\_params)

    @user.posts\_count = 0

    @user.is\_admin = false

    respond\_to do |format|

      if @user.save

        sign\_in @user

        format.html { redirect\_to root\_path, notice: 'User was successfully created.' }

        format.json { render :show, status: :created, location: @user }

      else

        format.html { render :new }

        format.json { render json: @user.errors, status: :unprocessable\_entity }

      end

    end

  end

  # PATCH/PUT /users/1

  # PATCH/PUT /users/1.json

  def update

    respond\_to do |format|

      if @user.update(user\_params)

        format.html { redirect\_to @user, notice: 'User was successfully updated.' }

        format.json { render :show, status: :ok, location: @user }

      else

        format.html { render :edit }

        format.json { render json: @user.errors, status: :unprocessable\_entity }

      end

    end

  end

  # DELETE /users/1

  # DELETE /users/1.json

  def destroy

    Comment.all.find\_all { |i| i.user\_id == @user.id }.each(&:destroy)

    @user.destroy

    respond\_to do |format|

      format.html { redirect\_to users\_url, notice: 'User was successfully destroyed.' }

      format.json { head :no\_content }

    end

  end

  private

  # Use callbacks to share common setup or constraints between actions.

  def set\_user

    @user = User.find(params[:id])

  end

  # Only allow a list of trusted parameters through.

  def user\_params

    params.require(:user).permit(:username, :password, :password\_conformation)

  end

end

**session\_helper.rb**

module SessionHelper

    def sign\_in(user)

        cookies.signed[:user\_id] = { value: user.id, expires: 20.days }

        self.current\_user = user

      end

      def signed\_in?

        !current\_user.nil?

      end

      def sign\_out

        cookies.signed[:user\_id] = nil

        self.current\_user = nil

      end

      def current\_user=(user)

        @current\_user = user

      end

      def current\_user

        @current\_user ||= User.find\_by(id: cookies.signed[:user\_id])

      end

end

**comment.rb**

class Comment < ApplicationRecord

    belongs\_to :user

    belongs\_to :post

end

**post.rb**

class Post < ApplicationRecord

    belongs\_to :user

    has\_many :comments

end

**user.rb**

class User < ApplicationRecord

    has\_secure\_password

    validates :username, uniqueness: true

    has\_many :posts

    has\_many :comments

    def update\_posts\_count

        self.posts\_count+=1

        save

    end

end

**login.html.erb**

<style>

  #reg{

      position:relative;

      top:15px;

      left: 40px;

      font-size: 15px;

  }

  .container{

      margin-left: 700px;

  }

  .form-control{

      width: 250px;

  }

  .btn-primary{

      margin-top: 20px;

      width:70px;

      height: 35px;

      font-size:15px;

  }

   .back{

    position:absolute;

    font-size:20px;

  }

</style>

<span class="back"><%=link\_to 'К списку постов',root\_path %></span>

<div class="container">

<%= form\_with url: session\_create\_url, method: :post, local: true do |f| %>

    <%=label\_tag ("Логин")%>

    <%= f.text\_field :login, required: true, label: 'Логин', class: "form-control" %>

    <%=label\_tag ("Пароль")%>

    <%= f.password\_field :password, required: true, label: 'Пароль', class: "form-control" %>

    <%= f.submit 'Войти', class: "btn-primary" %>

<% end %>

<span id="reg"><%= link\_to 'Зарегистрироваться', new\_user\_path %></span>

</div>

**posts/\_form.html.erb**

<style>

  .area{

    width: 500px;

    height: 150px;

  }

  #title{

    width: 500px;

  }

</style>

<%= form\_with(model: post, local: true) do  %>

  <% if post.errors.any? %>

    <div id="error\_explanation">

      <h2><%= pluralize(post.errors.count, "error") %> prohibited this post from being saved:</h2>

      <ul>

        <% post.errors.full\_messages.each do |message| %>

          <li><%= message %></li>

        <% end %>

      </ul>

    </div>

  <% end %>

  <div class="field">

    Тема

    <%= text\_field\_tag :title, nil, class: "form-control tit" %>

  </div>

  <div class="field">

    Текст

    <textarea id="text" name ="text" type="text" class="form-control area"></textarea>

  </div>

  <div class="actions">

    <%= submit\_tag "Отправить",class: "btn-primary" %>

  </div>

<% end %>

**posts/edit.html.erb**

<style>

#delete{

    position:relative;

    left: 500px;

}

#cancel{

    position:relative;

    top: 30px;

    right: 100px;

}

</style>

<h1>Редактирование поста</h1>

<%= render 'form', post: @post %>

<span id="delete"><%= link\_to 'Удалить пост', @post, method: :delete, data: { confirm: 'Вы уверены?' } %></span>

<span id="cancel"><%= link\_to 'Отмена', @post %></span>

**posts/index.html.erb**

<script type="text/javascript">

var time = setInterval(function() {

  var date = new Date();

  var hours = date.getHours();

  var mins = date.getMinutes();

  var sec = date.getSeconds();

  var res="Текущее время: ";

  if (hours < 10) res+= "0" + hours + ":";

  else res+=hours + ":";

  if (mins < 10) res+= "0" + mins + ":";

  else res+=mins + ":";

  if (sec < 10) res+= "0" + sec;

  else res+=sec;

  document.getElementById("time").innerHTML = (res);

}, 20);

</script>

<style>

  #time{

    font-size: 20px;

    position: absolute;

    top: 5px;

    left: 800px;

    font-weight: bold;

  }

  #create{

    font-size: 30px;

    position: absolute;

    left: 1000px;

    top: 130px;

  }

  .table{

    margin-left: 50px;

    width: 750px;

  }

  th{

    text-align: center;

    font-size: 25px;

  }

  td{

    font-size: 25px;

  }

  .exit{

    position:relative;

    left: 1650px;

    font-size: 20px;

  }

  .login{

    position:relative;

    left:1650px;

    font-size: 20px;

  }

  .username{

    position:relative;

    left: 1500px;

    font: bold, serif;

    font-size: 25px;

  }

</style>

<div id ="time"></div>

<p id="notice"><%= notice %></p>

<header>

  <% if current\_user%>

    <span class="username"><%= current\_user.username%></span>

    <span class = "exit"><%= link\_to 'Выйти', session\_logout\_path %></span>

  <% else %>

     <span class="login"><%=link\_to 'Войти', session\_login\_path %></span>

  <% end %>

</header>

<h1>Список постов</h1>

<table class ="table">

  <thead>

    <tr>

      <th >Автор</th>

      <th>Заголовок</th>

    </tr>

  </thead>

  <tbody>

    <% @posts.each do |post| %>

        <td><%= post.user.username %></td>

        <td><%= post.title %></td>

        <td><%= link\_to 'Перейти', post %></td>

      </tr>

    <% end %>

  </tbody>

</table>

<br>

<span id="create"><%= link\_to 'Создать пост', new\_post\_path %></span>

**posts/show.html.erb**

<script type="text/javascript">

var time = setInterval(function() {

  var date = new Date();

  var hours = date.getHours();

  var mins = date.getMinutes();

  var sec = date.getSeconds();

  var res="Текущее время: ";

  if (hours < 10) res+= "0" + hours + ":";

  else res+=hours + ":";

  if (mins < 10) res+= "0" + mins + ":";

  else res+=mins + ":";

  if (sec < 10) res+= "0" + sec;

  else res+=sec;

  document.getElementById("time").innerHTML = (res);

}, 20);

</script>

<head>

  <style>

   #time{

    font-size: 20px;

    position: absolute;

    top: 5px;

    left: 800px;

    font-weight: bold;

  }

  #add-com{

    font-weight: bold;

    font-size: 20px;

  }

  #enter{

    font-size: 20px;

    color: black;

    font-weight: bold;

  }

  .form-control{

    height: 100px;

    width: 500px;

    margin-bottom: 5px;

  }

  #com-title{

    font-weight: bold;

    font-size: 30px;

  }

  .comments{

    padding-top: 40px;

    padding-left: 100px;

  }

  .text{

    padding: 10px;

    padding-left: 20px;

    padding-top:20px;

    font-size: 20px;

  }

  .edit{

    float:right;

    position:absolute;

    top: 125px;

    left: 1200px;

    font-size: 20px;

  }

  .author{

    padding: 15px;

    padding-left: 0px;

    font-size: 35px;

  }

  .author-name{

    font-weight: bold;

  }

  .title{

    font-size: 35px;

    padding-top: 10px;

    padding-bottom: 15px;

  }

  .back{

    position:absolute;

    font-size:20px;

  }

  .exit{

    position:relative;

    left: 1650px;

    font-size: 20px;

  }

  .post{

    background-color: Gainsboro;

    margin-top: 50px;

    margin-left: 300px;

    padding-bottom: 20px;

  }

  .post-name{

    font-size: 30px;

  }

  .login{

    position:relative;

    left:1650px;

    font-size: 20px;

  }

  .username{

    position:relative;

    left: 1550px;

    font: bold, serif;

    font-size: 25px;

  }

  </style>

</head>

<header>

  <% if current\_user%>

    <span class="username"><%= current\_user.username%></span>

    <span class = "exit"><%= link\_to 'Выйти', session\_logout\_path %></span>

  <% else %>

     <span class="login"><%=link\_to 'Войти', session\_login\_path %></span>

  <% end %>

  <span class="back"><%=link\_to 'К списку постов',root\_path %></span>

</header>

<div id ="time"></div>

<div class = "post container">

  <p id="notice"><%= notice %></p>

  <div class = "author">Автор:<span class="author-name"><%= @post.user.username %></div></span>

  <div class='title'>Тема: <%= @post.title%></div>

  <div class="text"><%= @post.text %></div>

  <% if current\_user&.username == @post.user.username || current\_user&.is\_admin%>

     <div class="edit"><%= link\_to 'Удалить пост', @post, method: :delete, data: { confirm: 'Вы уверены?' } %></div>

  <% end %>

</div>

<div class ="comments">

    <span id="com-title">Комментарии</span>

       <% @comments.each do|comment|%>

         <div class = "com">

            <strong><%= comment.user.username%></strong>

            <p><%= comment.text %></p>

          </div>

        <% end %>

    <% if current\_user%>

      <%= form\_with url: comment\_path(@post), method: :post, local: true do %>

        <span id="add-com"><%= label\_tag("Введите комментарий") %></span>

        <textarea id="text" name ="text" type="text" class="form-control"></textarea>

        <%= submit\_tag("Отправить",class: "btn-primary")%>

      <% end %>

    <% else %>

      <span id="enter"><%=link\_to 'Войдите, чтобы написать комментарий', session\_login\_path %></span>

    <% end %>

</div>

**users/\_form.html.erb**

<script type="text/javascript">

var time = setInterval(function() {

  var date = new Date();

  var hours = date.getHours();

  var mins = date.getMinutes();

  var sec = date.getSeconds();

  var res="Текущее время: ";

  if (hours < 10) res+= "0" + hours + ":";

  else res+=hours + ":";

  if (mins < 10) res+= "0" + mins + ":";

  else res+=mins + ":";

  if (sec < 10) res+= "0" + sec;

  else res+=sec;

  document.getElementById("time").innerHTML = (res);

}, 20);

</script>

<style>

  #time{

    font-size: 20px;

    position: absolute;

    top: 5px;

    left: 800px;

    font-weight: bold;

  }

  .form-control{

    width: 250px;

  }

  .btn-primary{

    margin-top: 20px;

    width:150px;

    height: 35px;

    font-size:15px;

  }

  .container{

    padding-top: 20px;

    margin-left: 760px;

  }

</style>

<div id ="time"></div>

<div class = "container">

<%= form\_with(model: user, local: true) do |form| %>

  <% if user.errors.any? %>

    <div id="error\_explanation">

      <h2><%= pluralize(user.errors.count, "error") %> prohibited this user from being saved:</h2>

      <ul>

        <% user.errors.full\_messages.each do |message| %>

          <li><%= message %></li>

        <% end %>

      </ul>

    </div>

  <% end %>

  <div class="field">

    <%= form.label :username %>

    <%= form.text\_field :username, class: "form-control"  %>

  </div>

  <div class="field">

    <%= form.label :password %>

    <%= form.password\_field :password, required: true, class: "form-control" %>

  </div>

  <div class="field">

    <%= form.label :password\_confirmation %>

    <%= form.password\_field :password\_confirmation, required: true, class: "form-control" %>

  </div>

  <div class="actions">

    <%= form.submit "Создать", class: "btn-primary" %>

  </div>

<% end %>

</div>

**session\_flows\_test.rb – проверка возможности перехода между страницами с авторизацией и без, возможности создания пользователя, постов и комментариев, их добавление в базу данных.**

require 'test\_helper'

class SessionFlowsTest < ActionDispatch::IntegrationTest

  test 'unauthorized user will be redirected to login page' do

    get new\_post\_path

    assert\_redirected\_to controller: :session, action: :login

  end

  test 'user with incorrect credentials will be redirected to login page' do

    post session\_create\_url, params: { login: 'kek', password: 'kek' }

    assert\_redirected\_to controller: :session, action: :login

  end

 test 'user with correct credentials will see the root' do

    password = 'kek'

    user = User.create(username: 'lol', password: password, password\_confirmation: password, is\_admin: false, posts\_count: 0)

    post session\_create\_url, params: { login: user.username, password: password }

    assert\_redirected\_to controller: :posts, action: :index

  end

  test 'user can logout' do

    password = 'bbb'

    user = User.create(username: 'aaa',password: password, password\_confirmation: password)

    post session\_create\_url, params: { login: user.username, password: password }

    get session\_logout\_url

    assert\_redirected\_to controller: :session, action: :login

 end

   test 'not admin can not watch user list' do

     username = 'vvv'

     password = 'vvv'

     user = User.create(username: username, password: password, password\_confirmation: password)

     post session\_create\_url, params: { login: user.username, password: password }

     get users\_url

     assert\_redirected\_to controller: :posts, action: :index

   end

   test 'admin can  watch user list' do

    username = 'vvv'

    password = 'vvv'

    user = User.create(username: username, password: password, password\_confirmation: password, is\_admin: true)

    post session\_create\_url, params: { login: user.username, password: password }

    get users\_url

    assert request.url ==  "http://www.example.com/users"

  end

  test 'user can comemnting' do

    username = 'vvv'

    password ='vvv'

    user = User.create(username: username, password: password, password\_confirmation: password)

    post session\_create\_url, params: { login: user.username, password: password }

    post = Post.create(title: 'text', text: 'text', user\_id: user.id)

    get "/posts/#{post.id}"

    assert request.url ==  "http://www.example.com/posts/#{post.id}"

    post "/posts/#{post.id}/comment"

    assert\_redirected\_to post

  end

  test 'not logged user can not commenting' do

    username = 'vvv'

    password ='vvv'

    user = User.create(username: username, password: password, password\_confirmation: password)

    post = Post.create(title: 'text', text: 'text', user\_id: user.id)

    get "/posts/#{post.id}"

    assert request.url ==  "http://www.example.com/posts/#{post.id}"

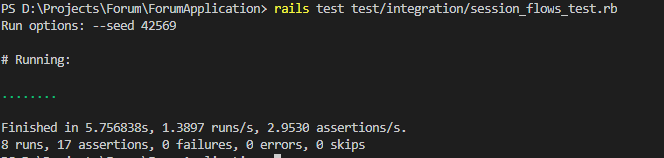
    post "/posts/#{post.id}/comment"

    assert\_redirected\_to controller: :session, action: :login

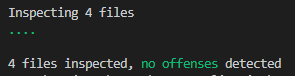
  end

end

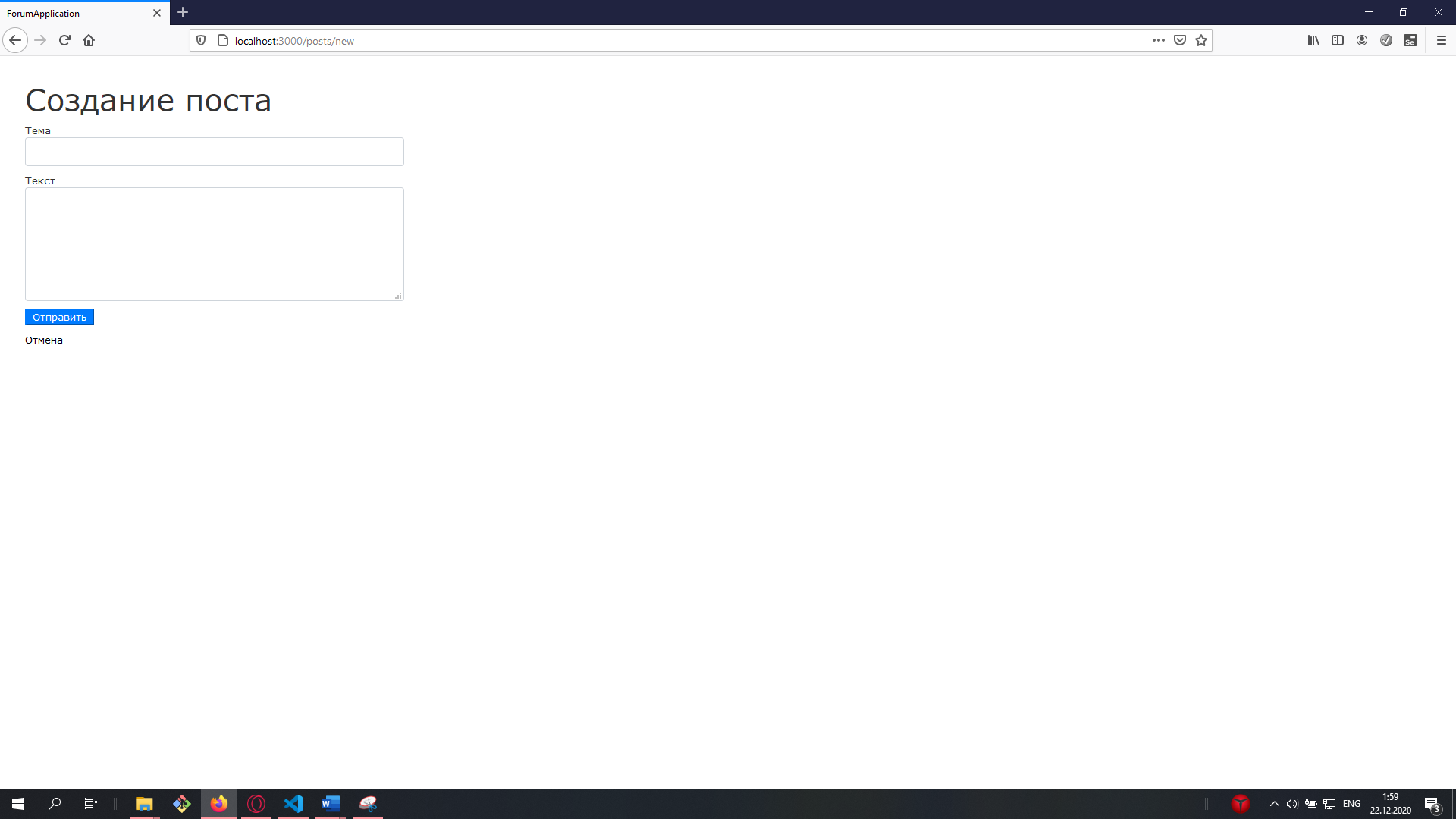
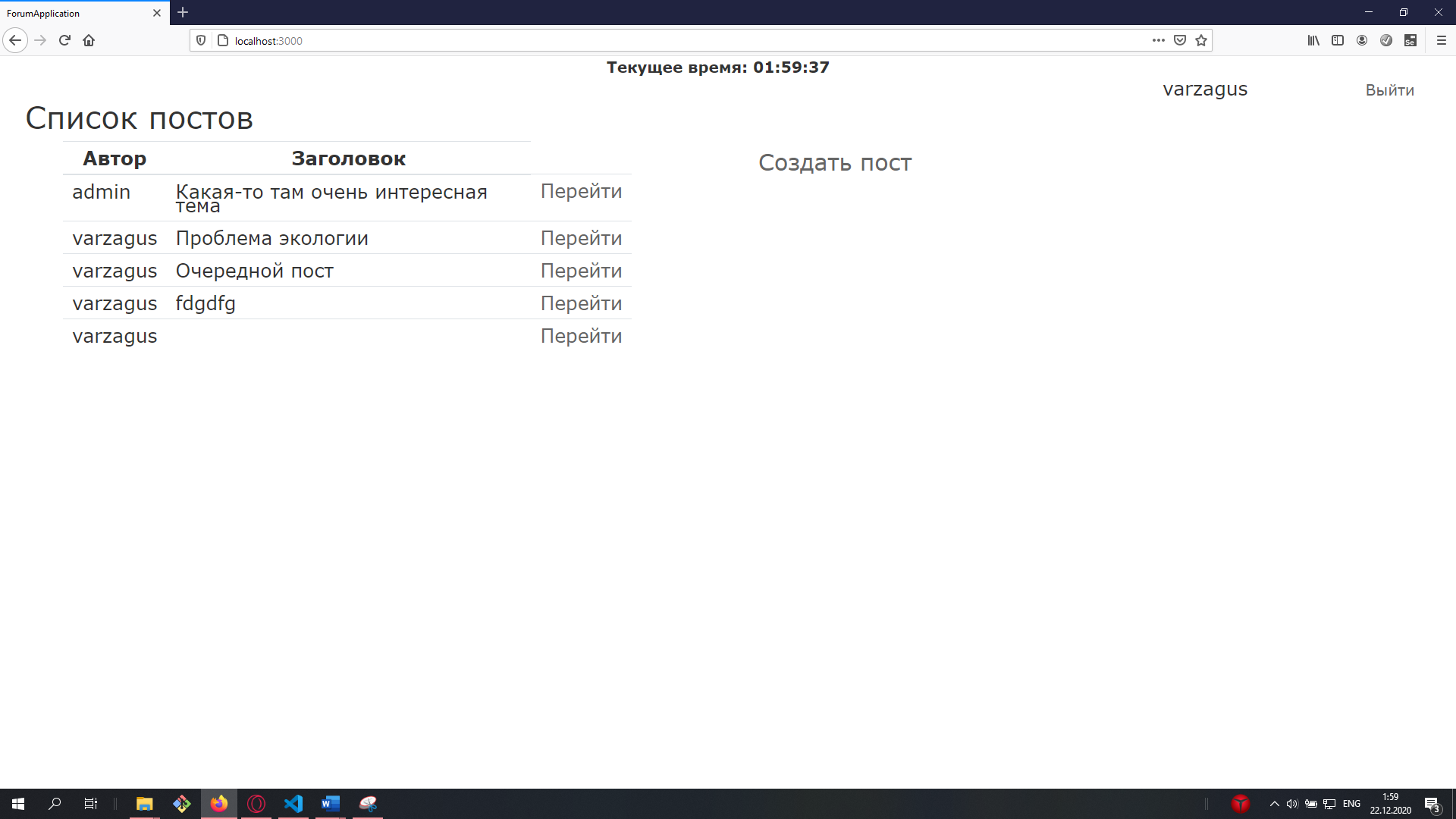
**Работа тестов:**

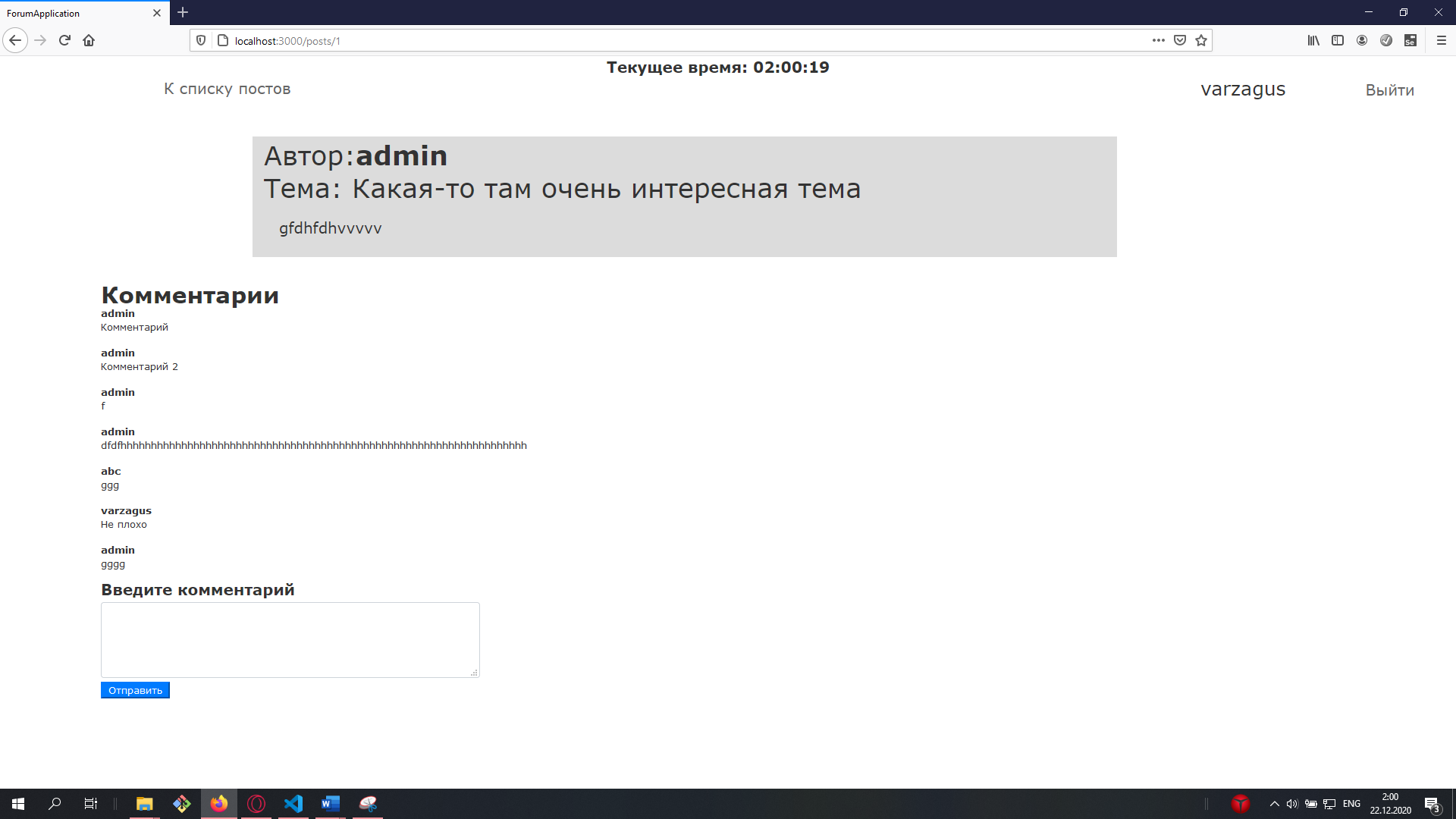


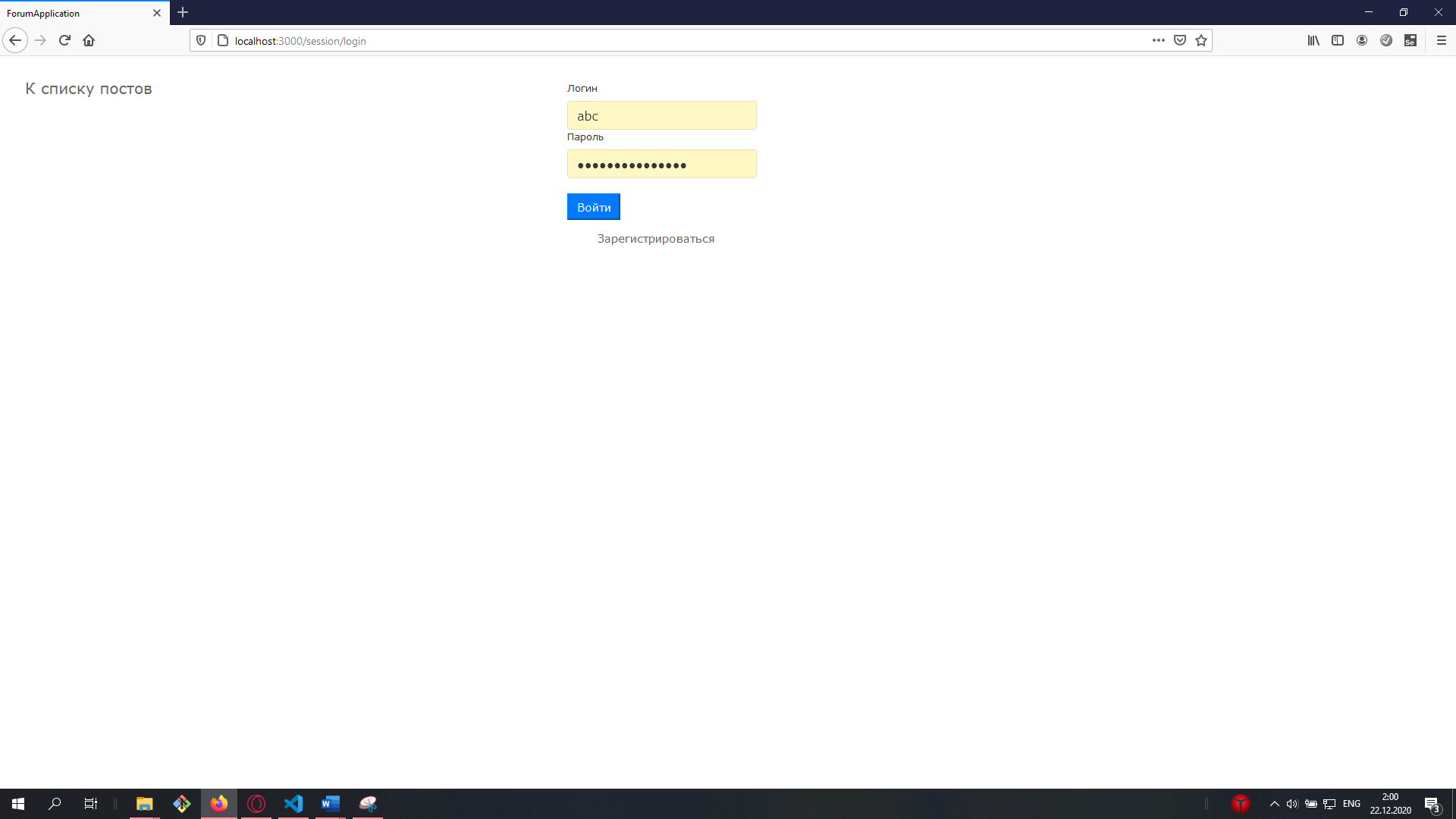
**Проверка контроллеров с помощью rubocop:**

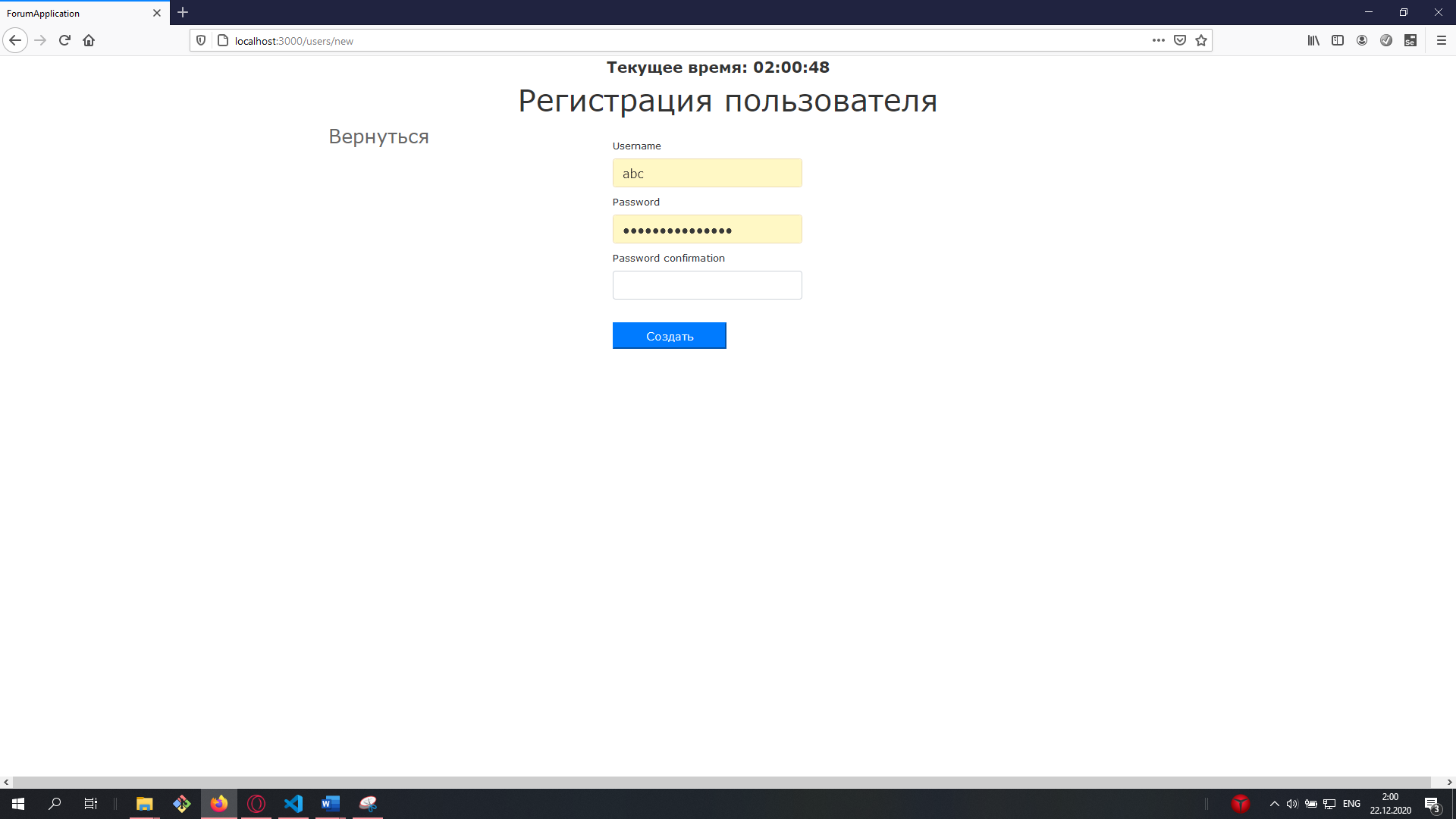


**Внешний вид страниц:**









**Вывод:** были получены практические навыки разработки приложений с использованием Ruby on Rails и их откладки.