



**«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 лабораторной работе № 12**

**Дисциплина: Языки Интернет-программирования.**

Студент гр. ИУ6-33Б

\_\_\_\_\_  
(Подпись, дата)

**Дасов Т.Д.**

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия)

Москва, 2019

## ***Задание.***

Модифицировать код приложения ЛР 8 таким образом, чтобы вычисление было невозможно без регистрации пользователя и аутентификации при помощи логина/пароля.

- Сгенерировать при помощи генератора scaffold ресурс для регистрации пользователей.
- Создать БД и выполнить миграцию соответствующим запросом rake.
- Проверить возможность добавления, редактирования информации и получения списка пользователей.
- Удалить отображение поля пароля при просмотре списка пользователей.
- Добавить контроллер сессий.
- Реализовать форму для ввода логина/пароля при обращении по адресу /. Добавить ссылку на регистрацию нового пользователя. При успешном вводе логина/пароля должно осуществляться перенаправление на страницу ввода параметров для вычисления.
- Реализовать при помощи контроллера сессий во всех действиях контроллера проверку о того, прошел ли пользователь аутентификацию или нет (с выдачей соответствующей отладочной информации).
- Вставить фильтры для запроса аутентификации.
- Подготовить интеграционный тест, позволяющий проверить регистрацию нового пользователя, вход под его именем и выполнение вычислений.
- Подготовить интеграционный тест для проверки невозможности выполнения вычислений без ввода логина/пароля.
- Проверить маршруты приложения с помощью rake routes и убрать лишние. Обеспечить доступ при обращении по адресу /.

## ***Маршруты routes.rb:***

```
Rails.application.routes.draw do
  resources :users

  root 'main#index', :as => 'root'

  get 'admin/auth' => 'admin#auth', :as => 'authorize'
  get 'admin/handling', :as => 'handle'
  get 'admin/res', :as => 'res'

  get 'main/result' => 'main#result', :as => 'result'
  get 'main/index' => 'main#index', :as => 'index'

  get 'session/signin' => 'session#auth', :as => 'signin'
  get 'session/auth' => 'session#authorize_user', :as => 'auth'
  get 'session/signout' => 'session#sign_out', :as => 'signout'
end
```

## ***Контроллер (application\_controller.rb):***

```
# frozen_string_literal: true

# ApplicationController class
class ApplicationController < ActionController::Base
  before_action :authorize, except: %i[auth new create authorize_user]

  def current_user
    @current_user ||= session[:current_user_id] &&
      User.find_by(id: session[:current_user_id])
  end
end
```

```

def authorize
  if current_user
    puts "Пользователь: #{current_user.login}"
  else
    puts 'Пользователь не аутентифицирован!'
    redirect_to signin_path
  end
end
end
end

```

### ***Контроллер сессий (session\_controller.rb)***

```

# frozen_string_literal: true

# SessionController class
class SessionController < ApplicationController
  def auth; end

  def authorize_user
    usr = User.authenticate(params[:login], params[:pass])
    if usr.nil?
      redirect_to signin_url
    else
      session[:current_user_id] = usr.id
      redirect_to root_url
    end
  end

  def sign_out
    session[:current_user_id] = nil
    redirect_to signin_url
  end
end

```

### ***Контроллер пользователя (user\_controller.rb)***

```

# frozen_string_literal: true

# UserController class
class UsersController < ApplicationController
  before_action :set_user, only: %i[show edit update destroy]

  # GET /users
  # GET /users.json
  def index
    @users = User.all
  end

  # GET /users/1
  # GET /users/1.json
  def show; 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_params
  @user = User.create_user(params)
  respond_to do |format|
    if !@user.nil? && @user.save
      format.html { redirect_to @user, notice: 'User was successfully created.' }
    else
      format.html { render :new }
    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
  @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

def user_params
  params.require(:user).permit(:login, :pass_d, :pass)
end
end

```

### ***Сервисный класс для нахождения дружественных чисел (friend\_numbers.rb):***

```

# frozen_string_literal: true

# Friend Number interactor class
class FriendNumbers
  include Interactor

  before :find_in_db

  private

```

```

# rubocop:disable Metrics/AbcSize
def find_in_db
  if Calculation.find_res(context.input).nil?
    context.res = calc(context.input)
    Calculation.insert_res(context.input, context.res)
  else
    context.res = Calculation.find_res(context.input)
  end
end
# rubocop:enable Metrics/AbcSize

def calc(_range)
  arr = []
  par = params[:n].to_i
  par.times do |i|
    par.times do |j|
      arr << [i, j] if compare_nums(i, j) && !arr.include?([j, i]) && i != j
    end
  end
  arr
end

def find_dev(x_val)
  res = []
  divider = x_val / 2
  while divider.positive?
    res.push(divider) if (x_val % divider).zero?
    divider -= 1
  end
  res
end
end

```

***Сервисный класс для преобразования данных БД в удобный для последующего использования формат (convert\_object.rb):***

```

# frozen_string_literal: true

# Converts object from database to array
class ConvertObject
  include Interactor

  def call
    context.res = context.obj.map { |el| [el.input, el.json_response] }
  end
end

```

***Модель User (user.rb)***

```

# frozen_string_literal: true

require 'digest'

# User model
class User < ApplicationRecord
  validates :login, uniqueness: true

  class << self
    def authenticate(login, pass)
      usr = User.find_by(login: login)
    end
  end
end

```

```

    return nil if usr.nil?
    return usr if usr.check_passw?(pass)
end

def create_user(params)
  data = params[:user]
  hash = Digest::SHA256.hexdigest(data[:pass])
  hash_d = Digest::SHA256.hexdigest(data[:pass_d])
  return User.new unless hash == hash_d

  create(login: data[:login], password_digest: hash)
end

def check_passw?(password)
  password_digest == Digest::SHA256.hexdigest(password)
end
end

```

### ***Представление для аутентификации пользователя пользователя (auth.html.erb):***

```

<%= form_tag auth_url, method: 'get' do %>
<%= label_tag 'Логин' %>
<%= text_field_tag :login %></br>
<%= label_tag 'Пароль' %>
<%= password_field_tag :pass %></br>
<%= submit_tag 'Войти' %>
<% end %>
<%= link_to 'Еще не с нами? Зарегистрируйтесь!', new_user_path %>

```

### ***Форма для регистрации нового пользователя (\_form.html.erb):***

```

<div class="input-form">
  <h1>Новый пользователь</h1>
  <%= 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 'Логин' %>
      <%= form.text_field :login, class: 'input-field' %>
    </div>

    <div class="field">
      <%= form.label 'Пароль' %>
      <%= form.password_field :pass, class: 'input-field' %>
    </div>

    <div class="field">

```

```

    <%= form.label 'Пароль еще раз' %>
    <%= form.password_field :pass_d, class: 'input-field' %>
  </div>

  <div class="actions" style="margin-top: 10px;">
    <%= form.submit 'Зарегистрироваться', class: 'button-def' %>
  </div>
<% end %>
</div>

```

### ***Представление для редактирования пользователя (show.html.erb)***

```

<p>
  <strong>Логин:</strong>
  <%= @user.login %>
</p>

<p>
  <strong>Пароль:</strong>
  <%= @user.password_digest %>
</p>

<%= link_to 'Редактировать', edit_user_path(@user) %> |
<%= link_to 'Назад', users_path %>

```

### ***Отображение списка пользователей (index.html.erb):***

```

<p id="notice"><%= notice %></p>

<h1>Пользователи</h1>

<table>
  <thead>
    <tr>
      <th>Логин</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <% @users.each do |user| %>
      <tr>
        <td><%= user.login %></td>
        <td><%= link_to 'Показать', user %></td>
        <td><%= link_to 'Редактировать', edit_user_path(user) %></td>
        <td><%= link_to 'Удалить', user, method: :delete, data: { confirm: 'Вы уверены?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>

<br>

<%= link_to 'Новый пользователь', new_user_path %>

```

### ***Интеграционный тест приложения (создание пользователя и попытка входа):***

```
require 'test_helper'

class SessionControllerTest < ActionDispatch::IntegrationTest
  test 'cant go anywhere without login' do
    get signout_url
    get index_url
    assert_response :redirect
    assert_redirected_to signin_url
  end

  test 'create fake user and make some stuff' do
    unless usr = User.find_by(login: 'test').nil?
      User.find_by(login: 'test').destroy
    end
    User.create(login: 'test', password_digest: '1234').save
    get signout_url
    get auth_url, params: { login: 'test', pass: '1234' }
    get result_url, params: { n: '300' }
    User.find_by(login: 'test').destroy if usr
  end
end
```

### ***RSpec тест (поведение приложения если пользователь не аутентифицирован):***

```
# frozen_string_literal: true

require 'rails_helper'

RSpec.describe MainController, type: :controller do
  context 'app behaviour if user not register' do
    it 'should redirect to signin page if user do not auth' do
      get :index
      expect(response).to redirect_to(signin_url)
      expect(response).to have_http_status(302)
    end
  end
end
```



## Пример работы приложения:

Дружественные числа!

Зарегистрироваться

Логин

Пароль

Войти

[Еще не с нами? Зарегистрируйтесь!](#)

Рис. 1 (Страница авторизации)

Дружественные числа!

Зарегистрироваться

Новый  
пользователь

Логин

Пароль

Пароль еще раз

Зарегистрироваться

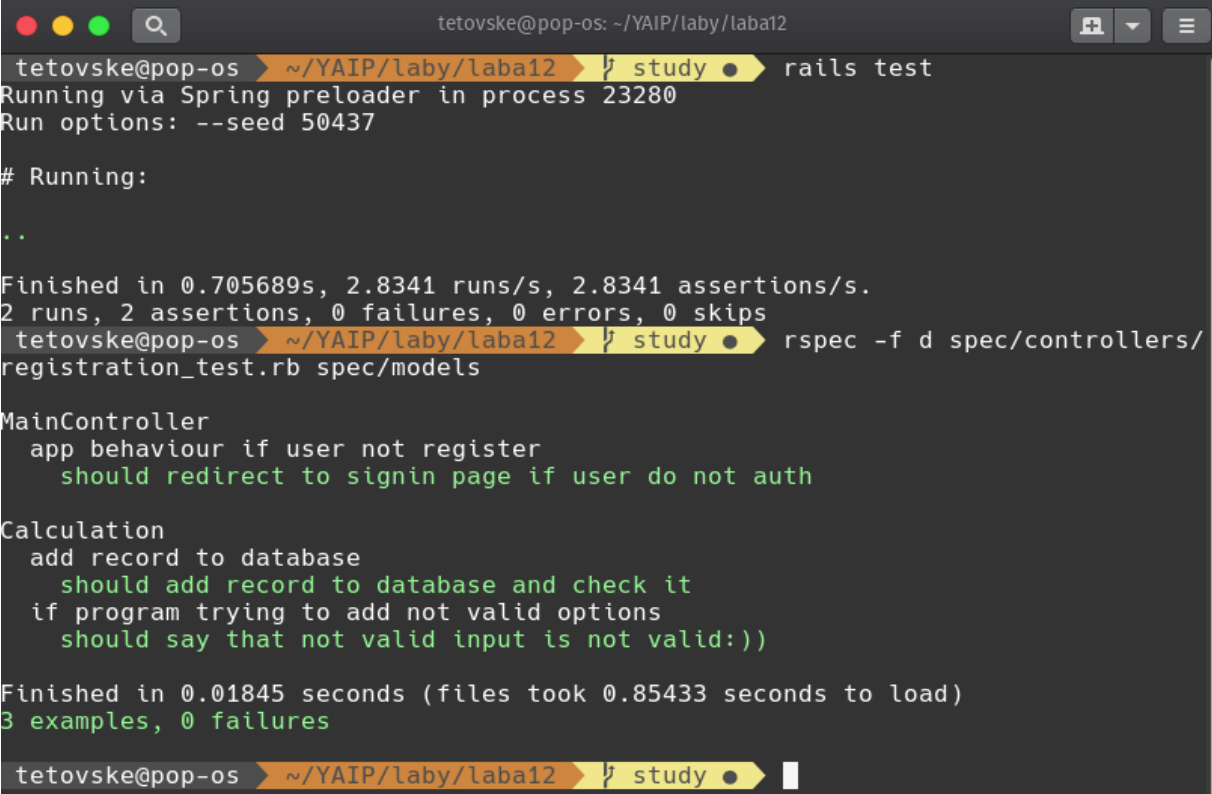
Рис. 2 (Страница регистрации нового пользователя)

## Пользователи

Логин

tet [Показать](#)[Редактировать](#)[Удалить](#)test [Показать](#)[Редактировать](#)[Удалить](#)[Новый пользователь](#)

Рис. 3 (Отображение всех зарегистрированных пользователей)

**Тесты приложения:**

```
tetovske@pop-os: ~/YAIP/laby/laba12
tetovske@pop-os > ~/YAIP/laby/laba12 > study ● rails test
Running via Spring preloader in process 23280
Run options: --seed 50437

# Running:

..

Finished in 0.705689s, 2.8341 runs/s, 2.8341 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
tetovske@pop-os > ~/YAIP/laby/laba12 > study ● rspec -f d spec/controllers/
registration_test.rb spec/models

MainController
  app behaviour if user not register
    should redirect to signin page if user do not auth

Calculation
  add record to database
    should add record to database and check it
  if program trying to add not valid options
    should say that not valid input is not valid:))

Finished in 0.01845 seconds (files took 0.85433 seconds to load)
3 examples, 0 failures

tetovske@pop-os > ~/YAIP/laby/laba12 > study ●
```

Рис. 4 (Пройденные тесты)

```

tetovske@pop-os > ~/YAIP/laby/laba12 > study ● rails c
Running via Spring preloader in process 24425
Loading development environment (Rails 6.0.1)
[1] pry(main)> User.all
(0.1ms) SELECT sqlite_version(*)
User Load (0.1ms) SELECT "users".* FROM "users"
=> [#<User:0x0000563551627cd0
  id: 19,
  login: "tet",
  password_digest: [FILTERED],
  created_at: Sun, 15 Dec 2019 21:38:37 UTC +00:00,
  updated_at: Sun, 15 Dec 2019 21:38:37 UTC +00:00>,
 #<User:0x0000563551769210
  id: 20,
  login: "test",
  password_digest: [FILTERED],
  created_at: Sun, 15 Dec 2019 22:00:10 UTC +00:00,
  updated_at: Sun, 15 Dec 2019 22:00:10 UTC +00:00>]
[2] pry(main)>

```

Рис. 5 (Просмотр содержимого БД через команду «rails c»)

```

tetovske@pop-os > ~/YAIP/laby/laba12 > study ● rubocop app/interactors app/
models app/controllers/admin_controller.rb app/controllers/application_controlle
r.rb app/controllers/session_controller.rb
Inspecting 8 files
.....

8 files inspected, no offenses detected
tetovske@pop-os > ~/YAIP/laby/laba12 > study ●

```

Рис. 6 (Отчет Rubocop)

**Вывод:** в ходе выполнения лабораторной работы был модифицирован код ЛР8 таким образом, что попытка выполнить вычисления без регистрации пользователя невозможна. Для реализации бизнес-логики был использован гем “Interactor”. Приложение было протестировано и проверено на соответствие стилю программой Rubocop.