

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

по лабораторной работе № 1 2

Название: Сессии. Выполнение авторизации. Интеграционные тесты.

Дисциплина: Интернет-Программирование

Студент	<u>ИУ6-32Б</u>	<u>(Подпись, дата)</u>	<u>А.Е.Медведев</u>
	(Группа)		(И.О. Фамилия)
Преподаватель		<u>(Подпись, дата)</u>	<u>(И.О. Фамилия)</u>

Москва, 2020

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 /.

Исходный код

```
#application_controller.rb
class ApplicationController < ActionController::Base
  helper_method :current_user
  before_action :require_auth, except: [:new, :edit, :destroy, :create, :show, :index]
  #before_action :require_auth, only: [:index]

  def current_user
    if session[:user_id]
      @current_user ||= User.find(session[:user_id])
    else
      @current_user = nil
    end
  end

  def require_auth
    current_user
    redirect_to login_path unless @current_user
  end
end
```

```
end  
end
```

```
#com_controller.rb
```

```
class ComController < ApplicationController
```

```
  def index  
    render layout: false  
  end
```

```
  def view
```

```
    @result = History.create_with_output(params[:arr]).output
```

```
    if @result =~ /\d/
```

```
      @result = JSON.parse(@result)
```

```
    end
```

```
    respond_to do |format|
```

```
      format.html
```

```
      format.json do
```

```
        render json:
```

```
          {type: @result.class.to_s, value: @result }
```

```
      end
```

```
    end
```

```
  end
```

```
end
```

```
#session_controller.rb
```

```
class SessionsController < ApplicationController
```

```
  def new
```

```
  end
```

```
  def create
```

```
    user = User.find_by_name(params[:name])
```

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

```
      session[:user_id] = user.id
```

```
      redirect_to root_url, notice: "Logged in!"
```

```
    else
```

```
      flash.now[:alert] = "Name or password is invalid"
```

```
      render "new"
```

```
    end
```

```
  end
```

```
def destroy
  session[:user_id] = nil
  redirect_to root_url, notice: "Logged out!"
end
end
```

#user_controller.rb

```
class UsersController < SessionsController
```

```
  before_action :set_user, only: [: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 = User.new(user_params)
```

```
    respond_to do |format|
```

```
      if @user.save
```

```
        format.html { redirect_to @user, 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
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
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(:name, :password, :password_confirmation)
end
end
```

```
#history.rb /model/
class History < ApplicationRecord
  validates :arr, presence: true, uniqueness: true
```

validates :output, presence: true

```
class << self
  def create_with_output(str)
    if where(arr: str).count().zero?
      create(arr: str, output: out(str))
    end
    @result = where(arr: str).first
  end

  private

  def out(str)
    if str == '' || str.nil?
      'input nil'
    elsif !str.nil? && str.strip.split.map(&:to_i).join(' ') != str.strip
      "don't numbers"
    else
      get_perfect_numbers(convert_input_to_numbers(str))
    end
  end

  def perfect_number?(num)
    s = (1..(num / 2)).select { |a| (num % a).zero? }
    num == s.compact.inject(:+)
  end

  def get_perfect_numbers(arr)
    out = []
    inp = []
    arr = arr.drop_while { |y| !perfect_number?(y) }
    arr.each do |x|
      if !perfect_number?(x)
        out.append(inp) if inp.any?
        inp = Array.new
      else
        inp.append(x)
      end
    end
    out.append(inp) if inp.size.positive?
  end
end
```

```
def convert_input_to_numbers(line)
  line.split.map(&:to_i)
end
end
end
```

```
#user.rb /model/
class User < ApplicationRecord
  has_secure_password

  validates :name, presence: true, uniqueness: true
end
```

```
#index.html.erb
<% if current_user %>
  Logged in as
  <%= link_to "Log Out", logout_path %>
<script>
  number_url= location.protocol + "://" + location.host + "/com/view.json"
  function send_query(){
    var frm = document.forms.number_form;
    var param_str = "?" + "arr=" + frm.arr.value
    if (param_str == "") return false;
    var my_JSON_object = {};
    var http_request = new XMLHttpRequest();
    http_request.open("GET", number_url+param_str, true);
    http_request.onreadystatechange = function () {
      var done = 4, ok = 200;
      if (http_request.readyState == done && http_request.status == ok) {
        my_JSON_object = JSON.parse(http_request.responseText);
        show_result(my_JSON_object);
      }
    };
    http_request.send(null);
    return false;
  }
  function show_result(data){
    var result = document.getElementById("result");
    result.innerHTML = "";
    if (data.type != "Array")
    {
```

```

        result.innerHTML = "<hr/>Result is: " + data.value + "<hr/>";
    }
    else
    {
        var length = 0, ii = 0;
        for (var i = 0; i < data.value.length; ++i)
        {
            result.innerHTML += "<hr>Array: " + data.value[i] + "</hr>";
            if (data.value[i].length > length)
            {
                length = data.value[i].length;
                ii = i;
            }
        }
        result.innerHTML += "<hr>max length:" + length + "</hr>";
        result.innerHTML += "<hr>Array with max length:" + data.value[ii] + "</hr>";
    }
}
</script>
</head>
<body>
<form name="number_form" method="get">
    <label for="numbers">Search perfect number:</label>
    <input id="arr" name="arr" type="text" /> <br/>
    <input name="commit" type="submit" value="Get result" onclick="return
send_query();" />
</form>
<div id="result"></div>

<% else %>
    <%= link_to "Sign Up", signup_path %> or
    <%= link_to "Log In", login_path %>
<% end %>

#new.html.erb
<h1>Sessions#new</h1>
<p>Find me in app/views/sessions/new.html.erb</p>
<p id="alert"><%= alert %></p><h1>Login</h1><%= form_tag sessions_path do |
form| %>
    <div class="field">
        <%= label_tag :name %>
        <%= text_field_tag :name %>

```



```
</div> <div class="field">
  <%= label_tag :password %>
  <%= password_field_tag :password %>
</div> <div class="actions">
  <%= submit_tag "Login" %>
</div>
<% end %>
```

```
#user db
class CreateUsers < ActiveRecord::Migration[6.0]
  def change
    create_table :users do |t|
      t.string :name
      t.string :password_digest

      t.timestamps
    end
  end
end
```

```
#history.rb db
class CreateHistories < ActiveRecord::Migration[6.0]
  def change
    create_table :histories do |t|
      t.string :arr
      t.string :output

      t.timestamps
    end
  end
end
```

```
#routes.rb
Rails.application.routes.draw do
  root 'com#index'
  get 'sessions/new'
  get 'sessions/create'
  get 'sessions/destroy'
  resources :users
  get 'com/index'
  get 'com/view'
  resources :sessions, only: [:new, :create, :destroy]
```

```
get 'input', to: 'users#new', as: 'signup'
get 'login', to: 'sessions#new', as: 'login'
get 'logout', to: 'sessions#destroy', as: 'logout'
# For details on the DSL available within this file, see https://guides.rubyonrails.org/routing.html
end
```

```
#history_test.rb
require 'test_helper'
```

```
class HistoryTest < ActiveSupport::TestCase
  # test "the truth" do
  #   assert true
  # end
  test 'should save data into db' do
    instance = History.new(arr: 10)
    assert_not instance.save
  end
```

```
  test 'should get data from db' do
    assert History.create_with_output('asd').valid?
  end
```

```
  test 'should get 1 data from db' do
    assert History.create_with_output('6 1 2').valid?
  end
```

```
  test 'should unique data' do
    l = History.new()
    l.arr = '6 1 2'
    l.output = '[[6]]'
    l.save!
    r = History.find_by(arr: '6 1 2')
    assert_equal "[[6]]", r.output
  end
```

```
  test 'should 2 data' do
    l = History.new()
    l.arr = "6 1 2"
    l.output = "[[6]]"
    l.save!
```

```
r = History.new()
r.arr = "6 1 2"
r.output = "[[6]]"
r.save
```

```
    assert_equal 1, History.where(arr: '6 1 2').count
end
end
```

```
#session_test.rb
# frozen_string_literal: true
```

```
require 'application_system_test_case'
```

```
class SessionsTest < ApplicationSystemTestCase
  test 'should not login user with invalid credentials' do
    visit login_url
    fill_in 'Name', with: '123'
    fill_in 'Password', with: '123'
    click_on 'Login'
```

```
    assert_text 'Name or password is invalid'
  end
```

```
  test 'register and count' do
    login = '12'
    password = '12'
```

```
    visit signup_url
```

```
    fill_in 'Name', with: login
    fill_in 'Password', with: password
    fill_in 'Password confirmation', with: password
    click_on 'Create User'
```

```
    assert_text 'User was successfully created.'
    visit login_url
```

```
    fill_in 'Name', with: login
    fill_in 'Password', with: password
    click_on 'Login'
```

```
assert_text "Logged in as Log Out\nSearch perfect number:"
```

```
fill_in 'arr', with: '6 28 0 2 6'
```

```
click_on 'Get result'
```

```
assert_text "Logged in as Log Out\nSearch perfect number:\nArray: 6,28\nArray:  
6\nmax length:2\nArray with max length:6,28"  
end
```

```
test 'register login and logout' do
```

```
  login = '12345'
```

```
  password = '12345'
```

```
  visit signup_url
```

```
  fill_in 'Name', with: login
```

```
  fill_in 'Password', with: password
```

```
  fill_in 'Password confirmation', with: password
```

```
  click_on 'Create User'
```

```
  visit login_url
```

```
  fill_in 'Name', with: login
```

```
  fill_in 'Password', with: password
```

```
  click_on 'Login'
```

```
  assert_text "Logged in as Log Out\nSearch perfect number:"
```

```
  click_on 'Log Out'
```

```
  assert_text 'Sign Up or Log In'
```

```
end
```





```
end
```



Фото



  127.0.0.1:3000

[Sign Up](#) or [Log In](#)



  127.0.0.1:3000/input

New User

Name

Password

Password confirmation

Create User



  127.0.0.1:3000/login

Sessions#new

Find me in app/views/sessions/new.html.erb

Login

Name

Password

Login

← → ↻ 🏠 🔒 127.0.0.1:3000

Logged in as [Log Out](#)

Search perfect number:

Get result

← → ↻ 🏠 🔒 127.0.0.1:3000

Logged in as [Log Out](#)

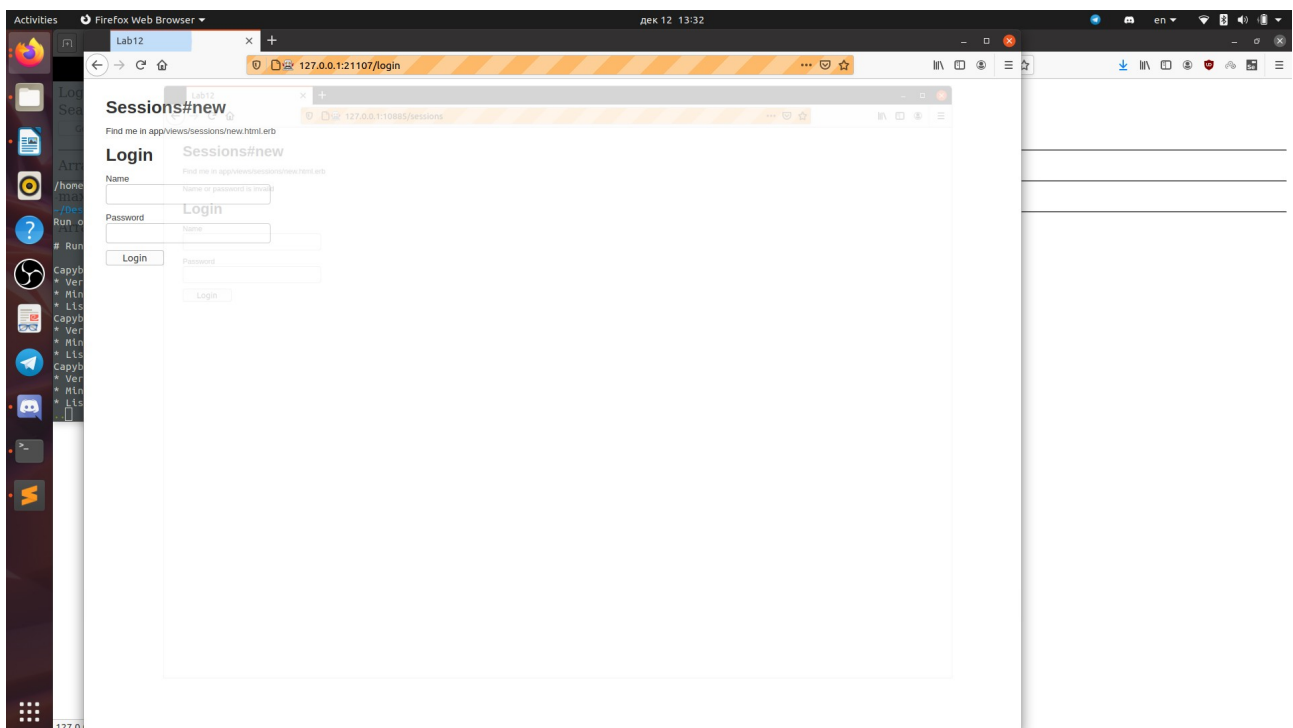
Search perfect number:

Get result

Array: 6

max length:1

Array with max length:6



```
~/Desktop/mgtu/mgtu_3_sem/IP/lab12/lab12(master*) » rake
Run options: --seed 46027

# Running:

.....

Finished in 2.723490s, 4.0389 runs/s, 4.4061 assertions/s.
11 runs, 12 assertions, 0 failures, 0 errors, 0 skips

~/Desktop/mgtu/mgtu_3_sem/IP/lab12/lab12(master*) »
```

```
~/Desktop/mgtu/mgtu_3_sem/IP/lab12/lab12(master*) » rake test:system
Run options: --seed 42760
Array with max length:6

# Running:

Capybara starting Puma...
* Version 4.3.7 , codename: Mysterious Traveller
* Min threads: 0, max threads: 4
* Listening on tcp://127.0.0.1:24121
Capybara starting Puma...
* Version 4.3.7 , codename: Mysterious Traveller
* Min threads: 0, max threads: 4
* Listening on tcp://127.0.0.1:21107
Capybara starting Puma...
* Version 4.3.7 , codename: Mysterious Traveller
* Min threads: 0, max threads: 4
* Listening on tcp://127.0.0.1:10885
...
```

Вывод

В ходе лабораторной работы была разработана система авторизации.