



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

**О Т Ч Е Т**

по лабораторной работе № 1 2

Название: Сессии. Выполнение авторизации. Интегрированные тесты.

Дисциплина: Языки интернет программирования

Студент

ИУ6-53Б

(Группа)

(Подпись, дата)

Т.Р. Сапарбаев

(И.О. Фамилия)

Преподаватель

Р.С. Самарев

(Подпись, дата)

(И.О. Фамилия)

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 /.

## Маршруты (routes.rb):

```
Rails.application.routes.draw do
  root 'home#input'

  resources :sessions, only: [:new, :create, :destroy]
  resources :users

  get 'home/input', :as => 'input'
  get 'home/output', :as => 'output'

  get "signup", to: "users#new", as: "signup"

  get "login", to: "sessions#new", as: "login"
  get "auth", to: "sessions#create", as: "auth"
  get "logout", to: "sessions#destroy", as: "logout"
end
```

## Главный контроллер (application\_controller.rb):

```
# frozen_string_literal: true

# Application Controller Class
class ApplicationController < ActionController::Base
  before_action :authenticate_user!
  helper_method :current_user

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

  def authenticate_user!
    redirect_to login_path, notice: 'Please login' unless current_user
  end

  def deny_access_to_authorized
    redirect_to input_url if current_user
  end
end
```

## Контроллер сессий (sessions\_controller.rb):

```
# frozen_string_literal: true

# Sessions Controller Class
class SessionsController < ApplicationController
  skip_before_action :authenticate_user!
  def new; end
end
```

```

def create
  user = User.find_by_login(params[:login])
  if user&.authenticate(params[:password])
    session[:user_id] = user.id
    redirect_to input_url
  else
    flash[:alert] = 'invalid data'
    redirect_to login_url
  end
end

def destroy
  session[:user_id] = nil
  redirect_to login_url
end
end

```

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

```

# frozen_string_literal: true

# Users Controller Class
class UsersController < ApplicationController
  skip_before_action :authenticate_user!, only: %i[new create]

  def new; end

  def create
    user = User.new(user_params)
    if user.save
      session[:user_id] = user.id
      redirect_to input_url
    else
      flash[:alert] = 'invalid data'
      redirect_to signup_url
    end
  end

  private

  def user_params
    params.require(:user).permit(:login, :password, :password_confirmation)
  end
end

```

## Модель User (user.rb):

```
# frozen_string_literal: true

class User < ApplicationRecord
  has_secure_password
  validates :login, uniqueness: true
  validates :password_digest,
            presence: true,
            confirmation: true
end
```

## Представление для аутентификации пользователя пользователя(new.html.erb):

```
<p><%= alert %></p>

<div class="form-io">
  <b>Вход в систему</b>
  <%= form_tag auth_path, :method => 'get' do %>
    <div class="field">
      <%= text_field_tag :login, '', placeholder: "Логин", class: 'textarea' %>
    </div>
    <div class="field">
      <%= password_field_tag :password, '', placeholder: "Пароль", class: 'text-
textarea' %>
    </div>
    <div class="actions">
      <%= submit_tag 'Войти', class: 'button' %>
    </div>
  <% end %>
</p>
  Или,
  <%= link_to "зарегистрируйтесь", signup_path %>
</p>
</div>
```

## Форма для регистрации нового пользователя (new.html.erb):

```
<p><%= alert %></p>
<%= form_with(model: User, local: true) do |form| %>
  <div class="form-io">
    <b>Регистрация</b>
    <%= form.text_field :login, placeholder: "Логин", class: 'textarea' %>
    <%= form.password_field :password, placeholder: "Пароль", class: 'text-
textarea' %>
    <%= form.password_field :password_confirmation, place-
holder: "Повторите пароль", class: 'textarea' %>
    <%= form.submit 'Создать', :class => 'button' %>
  </p>
```

```

        У меня уже есть аккаунт,
        <%= link_to 'войти', login_url %>
      </p>
    </div>
  <% end %>

```

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

```

require 'test_helper'

class SessionsControllerTest < ActionDispatch::IntegrationTest
  test 'cant go anywhere without login' do
    get logout_url
    get input_url
    assert_response :redirect
    assert_redirected_to login_url
  end

  test 'create fake user and make some stuff' do
    unless usr = User.find_by(login: 'User').nil?
      User.find_by(login: 'User').destroy
    end
    User.create(login: 'User', password: '1111', password_confirmation: '1111').save
    get auth_url, params: { login: 'User', password: '1111' }
    get output_url, params: { txt: '100' }
    assert_response :success
    User.find_by(login: 'User').destroy if usr
  end
end

```

## Пример работы приложения:

Рисунок 1 – Страница авторизации

**Регистрация**

Логин

Пароль

Повторите пароль

**Создать**

У меня уже есть аккаунт, [войти](#)

Рисунок 2 – Страница регистрации нового пользователя

**Лабораторная работа №12**

Введите число

**Найти**

**Выход**

Рисунок 3 – Главная страница приложения

## Тесты приложения:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR12$ rails test
Running via Spring preloader in process 29723
Run options: --seed 52998

# Running:

..

Finished in 1.944804s, 1.0284 runs/s, 1.5426 assertions/s.
2 runs, 3 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 4 – Тесты приложения

## Отчеты Rubocop:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR12/app/controllers$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.1.
warning: please see https://github.com/whitequark/parser#compatibility-with-ruby-mri.
Inspecting 4 files
.....

4 files inspected, no offenses detected

saparbi@saparbi-TM1703:~/Desktop/YAIP/LR12/app/models$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.1.
warning: please see https://github.com/whitequark/parser#compatibility-with-ruby-mri.
Inspecting 2 files
..

2 files inspected, no offenses detected
```

Рисунок 5 – Отчёт Rubocop

**Вывод:** в ходе выполнения лабораторной работы был модифицирован код ЛР8 таким образом, что попытка выполнить вычисления без регистрации пользователя невозможна. Приложение было протестировано и проверено на соответствие стилю программой Rubocop.