



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. Интеграционные тесты

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-33Б
(Группа)

(Подпись, дата)

И.А. Нуруллаев
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

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 /.

Решение:

```
WPL_bmstu - application_controller.rb

1 class ApplicationController < ActionController::Base
2   include SessionHelper
3
4   before_action :require_login
5
6   private
7
8   def require_login
9     redirect_to session_login_url unless signed_in?
10  end
11 end
12
```

```
WPL_bmstu - session_controller.rb

1 # Session controller
2 class SessionController < ApplicationController
3   skip_before_action :require_login, only: %i[login create]
4   def login; end
5
6   def create
7     user = User.find_by username: params[:username]
8     if user&.authenticate(params[:password])
9       sign_in user
10      redirect_to root_url
11    else
12      redirect_to session_login_url
13    end
14  end
15
16  def logout
17    cookies.permanent[:remember_token] = nil
18    redirect_to session_login_url
19  end
20 end
21
```

```
WPL_bmstu - session_helper.rb

1 module SessionHelper
2   def sign_in(user)
3     cookies.signed[:user_id] = { value: user.id, expires: 10.days }
4     self.current_user = user
5   end
6
7   def signed_in?
8     !current_user.nil?
9   end
10
11  def sign_out
12    cookies.signed[:user_id] = nil
13    self.current_user = nil
14  end
15
16  def current_user=(user)
17    @current_user = user
18  end
19
20  def current_user
21    @current_user ||= User.find_by(id: cookies.signed[:user_id])
22  end
23 end
24
```

```
1 # Controller for Simons
2 class SimonsController < ApplicationController
3   def initialize
4     super
5     @simons = Enumerator.new do |steps|
6       a1 = 1
7       a2 = 2
8       a3 = 3
9       iteration = 1
10
11       loop do
12         fact = a1 * a2 * a3
13         steps << { iteration: iteration, f: fact, a1: a1, a2: a2, a3: a3, fits: factorial?(fact) }
14         a1 = a2
15         a2 = a3
16         a3 += 1
17         iteration += 1
18       end
19     end
20   end
21
22   def result
23     @max_n = params[:max_n].to_i
24     @result = @simons.take_while { |step| step[:iteration] < @max_n }.select { |step| step[:fits] }
25
26     respond_to do |format|
27       format.html
28       format.json do
29         render json: {
30           type: @result.class.to_s,
31           value: @result
32         }
33       end
34     end
35   end
36
37   def input; end
38
39   private
40
41   def factorial?(number)
42     i = 1
43     n = number.to_f
44     while (n % i).zero?
45       n /= i
46       i += 1
47     end
48
49     n == 1
50   end
51 end
52
```

```

1  # Users controller
2  class UsersController < ApplicationController
3    before_action :set_user, only: %i[show edit update destroy]
4    skip_before_action :require_login, only: %i[new create]
5
6    def index
7      @users = User.all
8    end
9
10   def show; end
11
12   def new
13     @user = User.new
14   end
15
16   def edit; end
17
18   def create
19     @user = User.new(user_params)
20
21     respond_to do |format|
22       if @user.save
23         sign_in @user
24         format.html { redirect_to user_url(@user), notice: 'User was successfully created.' }
25         format.json { render :show, status: :created, location: @user }
26       else
27         format.html { render :new, status: :unprocessable_entity }
28         format.json { render json: @user.errors, status: :unprocessable_entity }
29       end
30     end
31   end
32
33   def update
34     respond_to do |format|
35       if @user.update(user_params)
36         format.html { redirect_to user_url(@user), notice: 'User was successfully updated.' }
37         format.json { render :show, status: :ok, location: @user }
38       else
39         format.html { render :edit, status: :unprocessable_entity }
40         format.json { render json: @user.errors, status: :unprocessable_entity }
41       end
42     end
43   end
44
45   def destroy
46     @user.destroy
47
48     respond_to do |format|
49       format.html { redirect_to users_url, notice: 'User was successfully destroyed.' }
50       format.json { head :no_content }
51     end
52   end
53
54   private
55
56   def set_user
57     @user = User.find(params[:id])
58   end
59
60   def user_params
61     params.require(:user).permit(:username, :password, :password_confirmation)
62   end
63 end
64

```

WPL_bmstu - input.html.erb

```
1 <h1>Гипотеза Симона</h1>
2 <p>Существует гипотеза Симона о факториале.<br/>
3 Она гласит, что существует 4 факториала, которые представимы в виде произведения трех последовательных чисел.<br/>
4 Например:  $4! = 2 \cdot 3 \cdot 4$ .<br/>
5 Определить эту четверку, подсчитать действительное количество таких факториалов, если это возможно и тем самым доказать неправильность гипотезы.<br/>
6 Вывести на печать результаты каждой итерации и финальных расчетов.
7 </p>
8 <%= form_tag(simons_result_path, :method => "get") do %>
9   <%= label_tag "Максимальное количество итераций n:" %>
10   <%= number_field_tag 'max_n', '20', min: 1 %>
11   <%= submit_tag "Найти числа" %>
12 <% end %>
13 <%= link_to 'Просмотр пользователей', url_for(controller: :users, action: :index), class: 'btn btn-link btn-sm' %>
14 <%= link_to 'Выход', url_for(controller: :session, action: :logout), class: 'btn btn-link btn-sm' %>
```

WPL_bmstu - result.html.erb

```
1 <h1>Результат</h1>
2 <p>Существует гипотеза Симона о факториале.<br/>
3 Она гласит, что существует 4 факториала, которые представимы в виде произведения трех последовательных чисел.<br/>
4 Например:  $4! = 2 \cdot 3 \cdot 4$ .<br/>
5 Определить эту четверку, подсчитать действительное количество таких факториалов, если это возможно и тем самым доказать неправильность гипотезы.<br/>
6 Вывести на печать результаты каждой итерации и финальных расчетов.
7 </p>
8 <% if @result.empty? %>
9   <%= "Что-то пошло не так, попробуйте еще раз или обратитесь в поддержку" %>
10 <% else %>
11   <div class="result">
12     <h2>Найденные факториалы соответствующие гипотезе Симона</h2>
13     <p>Число итераций: <%= @max_n %></p>
14     <p id="result">Найдено чисел: <%= @result.size %></p>
15     <table>
16       <tr>
17         <th>Факториал</th>
18         <th>Последовательность</th>
19       </tr>
20       <% @result.each do |step| %>
21         <tr>
22           <td><%= step[:f] %></td>
23           <td><%= step[:a1] %>, <%= step[:a2] %>, <%= step[:a3] %></td>
24         </tr>
25       <% end %>
26     </table>
27   </div>
28 <% end %>
29 <%= link_to "Повторить попытку", :simons_input %>
```

WPL_bmstu - login.html.erb

```
1 <%= form_tag url_for(controller: :session, action: :create), method: 'post' do %>
2   <%= label_tag 'username' %>
3   <%= text_field_tag :username %>
4   <%= label_tag 'password' %>
5   <%= password_field_tag 'password' %>
6   <%= submit_tag 'Логин' %>
7 <% end %>
8 <%= link_to 'Регистрация', url_for(controller: :users, action: :new), class: 'btn btn-link btn-sm' %>
```

WPL_bmstu - _user.html.erb

```
1 <div id="<%= dom_id user %>">
2   <p>
3     <strong>Пользователь:</strong>
4     <%= user.username %>
5   </p>
6
7   <p>
8     <strong>Пароль (зашифрованный):</strong>
9     <%= user.password_digest %>
10  </p>
11 </div>
12
```

WPL_bmstu - edit.html.erb

```
1 <h1>Изменить пользователя</h1>
2
3 <%= render "form", user: @user %>
4
5 <br>
6
7 <div>
8   <%= link_to "Показать пользователя", @user %> |
9   <%= link_to "Обратно к пользователям", users_path %>
10 </div>
11
```

WPL_bmstu - index.html.erb

```
1 <p style="color: green"><%= notice %></p>
2
3 <h1>Пользователи</h1>
4
5 <div id="users">
6   <% @users.each do |user| %>
7     <%= render user %>
8     <p>
9       <%= link_to "Показать пользователя", user %>
10    </p>
11    <% end %>
12  </div>
13
14 <%= link_to "Новый пользователь", new_user_path %>
15 <%= link_to 'обратно к расчетам', url_for(controller: :simons, action: :input), class: 'btn btn-link btn-sm' %>
```

WPL_bmstu - _form.html.erb

```
1 <%= form_with(model: user) do |form| %>
2   <% if user.errors.any? %>
3     <div style="color: red">
4       <h2><%= pluralize(user.errors.count, "error") %> prohibited this user from being saved:</h2>
5
6       <ul>
7         <% user.errors.each do |error| %>
8           <li><%= error.full_message %></li>
9         <% end %>
10      </ul>
11    </div>
12  <% end %>
13
14  <div>
15    <%= form.label :username, style: "display: block" %>
16    <%= form.text_field :username %>
17  </div>
18
19  <div>
20    <%= form.label :password, style: "display: block" %>
21    <%= form.password_field :password %>
22  </div>
23
24  <div>
25    <%= form.label :password_confirmation, style: "display: block" %>
26    <%= form.password_field :password_confirmation %>
27  </div>
28
29  <div>
30    <%= form.submit %>
31  </div>
32 <% end %>
33
```

WPL_bmstu - new.html.erb

```
1 <h1>Новый пользователь</h1>
2
3 <%= render "form", user: @user %>
4
5 <br>
6
7 <div>
8   <%= link_to "Обратно", users_path %>
9 </div>
10
```

WPL_bmstu - show.html.erb

```
1 <p style="color: green"><%= notice %></p>
2
3 <%= render @user %>
4
5 <div>
6   <%= link_to "Изменить пользователя", edit_user_path(@user) %> |
7   <%= link_to "Обратно к пользователям", users_path %>
8
9   <%= button_to "Удалить пользователя", @user, method: :delete %>
10 </div>
11
```



```
1 require 'test_helper'
2
3 class UsersControllerTest < ActionDispatch::IntegrationTest
4   setup do
5     @password = Faker::Lorem.word
6     @user = User.create(username: Faker::Lorem.word, password: @password, last_login_at: DateTime.now)
7     post users_url, params: { user: { login: @user.username, password: @password, last_login: @user.last_login_at } }
8   end
9
10  test 'should get index' do
11    get users_url
12    assert_response :success
13  end
14
15  test 'should get new' do
16    get new_user_url
17    assert_response :success
18  end
19
20  test 'should create user' do
21    assert_difference('User.count') do
22      post users_url,
23        params: { user: { last_login_at: DateTime.now, password: Faker::Lorem.word,
24                          username: Faker::Lorem.word } }
25    end
26
27    assert_redirected_to user_url(User.last)
28  end
29
30  test 'should show user' do
31    get user_url(@user)
32    assert_response :success
33  end
34
35  test 'should get edit' do
36    get edit_user_url(@user)
37    assert_response :success
38  end
39
40  test 'should update user' do
41    patch user_url(@user),
42      params: { user: { last_login_at: @user.last_login_at, password: @password + Faker::Lorem.word,
43                        username: @user.username + Faker::Lorem.word } }
44    assert_redirected_to user_url(@user)
45  end
46
47  test 'should destroy user' do
48    assert_difference('User.count', -1) do
49      delete user_url(@user)
50    end
51
52    assert_redirected_to users_url
53  end
54 end
55
```

```

WPL_bmstu - session_flow_test.rb

1  require 'test_helper'
2
3  class SessionFlowTest < ActionDispatch::IntegrationTest
4    test 'unauthorized user will be redirected to login page' do
5      get simons_input_path
6      assert_redirected_to controller: :session, action: :login
7    end
8
9    test 'user with incorrect credentials will be redirected to login page' do
10     post session_create_url, params: { username: Faker::Lorem.word, password: Faker::Lorem.word }
11     assert_redirected_to controller: :session, action: :login
12   end
13
14   test 'user with correct credentials will see the root' do
15     password = Faker::Lorem.word
16
17     user = User.create(username: Faker::Lorem.word, password_digest: password, last_login_at: DateTime.now)
18
19     post users_url, params: { user: { login: user.username, password: password, last_login: user.last_login_at } }
20
21     get simons_input_path
22     assert_response :success
23   end
24
25   test 'user can logout' do
26     password = Faker::Lorem.word
27
28     user = User.create(username: Faker::Lorem.word, password_digest: password, last_login_at: DateTime.now)
29
30     post users_url,
31         params: { user: { username: user.username, password_digest: password, last_login: user.last_login_at } }
32
33     get session_logout_url
34
35     get simons_input_path
36
37     assert_redirected_to controller: :session, action: :login
38   end
39 end
40

```

Тестирование

```

PS C:\Users\might\Desktop\BMSTU\sem_3\WPL_bmstu\Lab12> rails test
Running 11 tests in a single process (parallelization threshold is 50)
Run options: --seed 34975

# Running:

.....

Finished in 5.161389s, 2.1312 runs/s, 3.0999 assertions/s.
11 runs, 16 assertions, 0 failures, 0 errors, 0 skips

```

Результат:

Идет перенаправление на вход

← → ↺ 🏠 📄 localhost:3000/session/login

Username

Password

Логин

[Регистрация](#)

Регистрация пользователя

← → ↺ 🏠 📄 localhost:3000/users/new

Новый пользователь

Username

Password

Password confirmation

Create User

[Обратно](#)

← → ↺ 🏠 📄 localhost:3000/users/1

User was successfully created.

Пользователь: tenessinum

Пароль (зашифрованный): \$2a\$12\$SVU9ZhpVg60OytSAjtasRed1oe3xI.50aRK6CNWXyVAKPZovrn2IS

[Изменить пользователя](#) | [Обратно к пользователям](#)

[Удалить пользователя](#)

Теперь есть доступ к расчетам

← → ↺ 🏠 📄 localhost:3000/simons/input

Гипотеза Симона

Существует гипотеза Симона о факториале.

Она гласит, что существует 4 факториала, которые представимы в виде произведения трех последовательных чисел.

Например: $4! = 2 \cdot 3 \cdot 4$.

Определить эту четверку, подсчитать действительное количество таких факториалов, если это возможно и тем самым доказать неправильность гипотезы.

Вывести на печать результаты каждой итерации и финальных расчетов.

Максимальное количество итераций n:

[Просмотр пользователей](#) [Выход](#)

Результат

Существует гипотеза Симона о факториале.

Она гласит, что существует 4 факториала, которые представимы в виде произведения трех последовательных чисел.

Например: $4! = 2 \cdot 3 \cdot 4$.

Определить эту четверку, подсчитать действительное количество таких факториалов, если это возможно и тем самым доказать неправильность гипотезы.

Вывести на печать результаты каждой итерации и финальных расчетов.

Найденные факториалы соответствующие гипотезе Симона

Число итераций: 20

Найдено чисел: 4

Факториал	Последовательность
6	1, 2, 3
24	2, 3, 4
120	4, 5, 6
720	8, 9, 10

[Повторить попытку](#)

Выйдем и создадим еще пользователя

Пользователи

Пользователь: tenessinum

Пароль (зашифрованный): \$2a\$12\$SVU9ZhpVg60OytSAjtasRed1oe3xI.50aRK6CNWXyVAKPZovrn2IS

[Показать пользователя](#)

Пользователь: igorek

Пароль (зашифрованный): \$2a\$12\$EBwOJ2djxhRYLRGH1EISUuOEwqhjvthXm49f2Y38w/bGjhSf8FdEK

[Показать пользователя](#)

[Новый пользователь обратно к расчетам](#)

Просмотр пользователя

Пользователь: tenessinum

Пароль (зашифрованный): \$2a\$12\$SVU9ZhpVg60OytSAjtasRed1oe3xI.50aRK6CNWXyVAKPZovrn2IS

[Изменить пользователя](#) | [Обратно к пользователям](#)

[Удалить пользователя](#)

Удалим пользователя

localhost:3000/users

User was successfully destroyed.

Пользователи

Пользователь: igorek

Пароль (зашифрованный): \$2a\$12\$EBwOJ2djhRYLRGH1EISUuOEwqhjvthXm49f2Y38w/bGjhSf8FdEK

[Показать пользователя](#)

[Новый пользователь обратно к расчетам](#)

Обновим имя и пароль пользователя

localhost:3000/users/2/edit

Изменить пользователя

Username

igorek

Password

Password confirmation

Update User

[Показать пользователя](#) | [Обратно к пользователям](#)

localhost:3000/users/2

User was successfully updated.

Пользователь: igorek_v2

Пароль (зашифрованный): \$2a\$12\$QygBQK.CGsjI5/m4xeEaNuZMYTdrRP.FfopHitP.zjTZzuVoTvvrVG

[Изменить пользователя](#) | [Обратно к пользователям](#)

Удалить пользователя

Добавим еще пользователей и посмотрим что хранится в БД

id	username	password_digest	created_at	updated_at
1	igorek_v2	\$2a\$12\$QygBQK.CGs...	2022-11-25 03:33:45....	2022-11-25 03:35:45....
2	tenessinum	\$2a\$12\$pzWO4lnSvl...	2022-11-25 03:37:49....	2022-11-25 03:37:49....
3	i_love_lab_12	\$2a\$12\$Izsy00ZJvI5l...	2022-11-25 03:38:20....	2022-11-25 03:38:20....

Итоговый код данной лабораторной работы доступен по ссылке:

https://github.com/tenessinum/WPL_bmstu/tree/main/Lab12