*Федеральное государственное бюджетное образовательное учреждение* *высшего профессионального образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

**Отчет**

**по Лабораторной работе №12**

**Дисциплина: Языки интернет программирования**

**Название лабораторной работы: Сессии. Выполнение авторизации. Хранение состояния средствами браузера и сервера. Интеграционные тесты.**

**Вариант 24**

Студент гр. ИУ6-32 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Скударнов И. В.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Самарёв Р. С.**

(Подпись, дата) (И.О. Фамилия)

Москва, 2016

**Задание:**

Модифицировать код лабораторной работы 8. Релизовать в приложении возможность регистрации и авторизации, запретить доступ до вычислений без авторизации.

**Код:**

**users\_controller.rb**

class UsersController < ApplicationController

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

def index

@users = User.all

respond\_to do |format|

format.html

format.xml { render xml: @users.map { |user| { user: {

id: user.id,

login: user.login,

password: user.password\_filtered,

created\_at: user.created\_at,

updated\_at: user.updated\_at

} } } }

end

end

def show

set\_user

end

def new

@user = User.new

end

def edit

set\_user

end

include SessionHelper

def create

@user = User.new(user\_params)

if @user.save\_new\_user

sign\_in @user

redirect\_to root\_path

else

redirect\_to signin\_path, notice: 'user' + user\_params[:login] + 'exists'

end

end

def update

set\_user

if @user.update\_exist\_user(user\_params)

redirect\_to user\_path, notice: 'user' + @user.login + 'updated'

else

redirect\_to user\_edit\_path, notice: 'user' + user\_params[:login] + 'exists'

end

end

def destroy

set\_user

@user.destroy

redirect\_to root\_path

end

private

def set\_user

@user = User.find session[:current\_user\_id]

end

def user\_params

params.require(:user).permit(:login, :password)

end

end

**user.rb– модель**

class User < ApplicationRecord

def password\_filtered

password.size.times.to\_a.map { '\*' }.join

end

def has\_password?(submitted\_password)

password == submitted\_password

end

def self.authenticate(login, submitted\_password)

user = find\_by\_login(login)

return nil if user.nil?

return user if user.has\_password?(submitted\_password)

end

def save\_new\_user

save unless User.find\_by\_login login

end

def update\_exist\_user(params)

update(params) if User.find\_by\_login(params[:login]).nil? || User.find\_by\_login(params[:login]) == self

end

def password

super

end

def save

super

end

def update(params)

super(params)

end

end

**session\_controller.rb**

class SessionController < ApplicationController

def new

end

def create

user = User.authenticate params[:session][:login], params[:session][:password]

if user.nil?

redirect\_to signin\_path, notice: 'invalid\_login\_password'

else

sign\_in user

redirect\_to root\_path

end

end

def destroy

sign\_out

redirect\_to root\_path

end

end

**new.html.erb – представление контроллера Session**

<h1>Session#new</h1>

<p>Find me in app/views/session/new.html.erb</p>

<div class='container'>

<div class='headline'><%= 'signin' %></div>

<div>

<%= form\_for(:session, :url => session\_path) do |f| %>

<div class='field'>

<%= f.label 'login', class: 'user\_input\_label', for: 'session\_login' %>

<%= f.text\_field :login %>

</div>

<div class='field'>

<%= f.label 'password', class: 'user\_input\_label', for: 'session\_password' %>

<%= f.password\_field :password %>

</div>

<div class='field'>

<%= f.submit 'signin' %>

</div>

<% end %>

</div>

<div id='notice'><%= notice %></div>

<div>

<span id='new\_user\_span'><%= 'new\_user?' %> &nbsp;</span>

<%= link\_to 'signup\_now', signup\_path %>

</div>

</div>

**session\_helper.rb**

module SessionHelper

def sign\_in(user)

session[:current\_user\_id] = user.id

end

def sign\_out

session[:current\_user\_id] = nil

end

end

**application\_controller.rb**

class ApplicationController < ActionController::Base

protect\_from\_forgery with: :exception

include SessionHelper

before\_action :authenticate, except: [:new, :create, :signup, :create\_user]

def current\_user

session[:current\_user\_id] && User.find\_by\_id(session[:current\_user\_id])

end

def authenticate

puts "Current user is #{current\_user}, id is #{session[:current\_user\_id]}"

unless current\_user

redirect\_to signin\_path

end

end

before\_action :set\_locale

def set\_locale

I18n.locale = browser\_locales.first || I18n.default\_locale

end

def browser\_locales

request.env['HTTP\_ACCEPT\_LANGUAGE'].split(/;|,/).reject do |x|

x.start\_with?('q')

end & I18n.available\_locales.map(&:to\_s)

end

end

**routes.rb**

Rails.application.routes.draw do

root 'palind#input'

get 'session/new'

get 'palind/input'

get 'palind/view', to: 'palind#view'

get 'signup', to: 'users#new'

post 'create\_user', to: 'users#create'

get 'user', to: 'users#show'

post 'user', to: 'users#update'

get 'user\_edit', to: 'users#edit'

get 'users', to: 'users#index'

get 'destroy\_user', to: 'users#destroy'

get 'signin', to: 'session#new'

get 'signout', to: 'session#destroy'

post 'session', to: 'session#create'

# For details on the DSL available within this file, see http://guides.rubyonrails.org/routing.html

end

**user\_flows\_test.rb**

require 'test\_helper'

class UserFlowsTest < ActionDispatch::IntegrationTest

test 'registration and calculation'do

get signup\_path

assert\_response :success

post create\_user\_path, params: { user:{ login: 'login', password: 'password' } }

assert\_redirected\_to controller: 'palind', action: 'input'

follow\_redirect!

get '/palind/view', params: { n1: 50 }

answer = 3

result = assigns[:result].first

assert\_in\_delta answer, result

end

test 'registration and walk through pages'do

get signup\_path

assert\_response :success

post create\_user\_path, params: { user:{ login: 'login', password: 'password' } }

assert\_redirected\_to controller: 'palind', action: 'input'

follow\_redirect!

get '/palind/view', params: { n1: 50 }

assert\_response :success

get user\_path

assert\_response :success

get users\_path

assert\_response :success

get user\_edit\_path

assert\_response :success

post user\_path, params: { user:{ login: 'login', password: 'password' } }

assert\_redirected\_to controller: 'users', action: 'show'

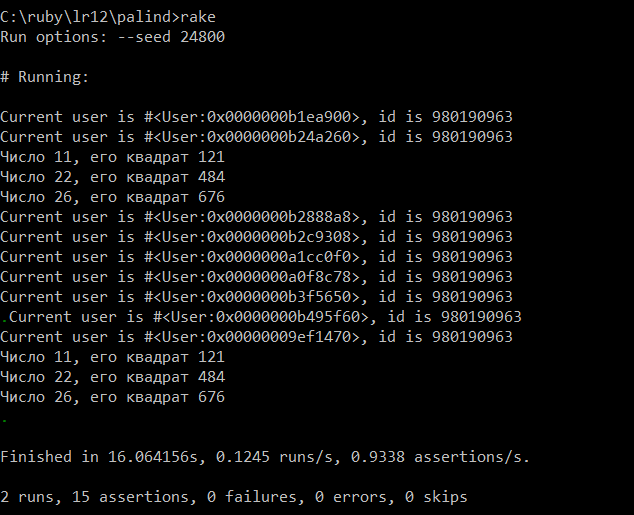
get destroy\_user\_path

assert\_redirected\_to root\_path

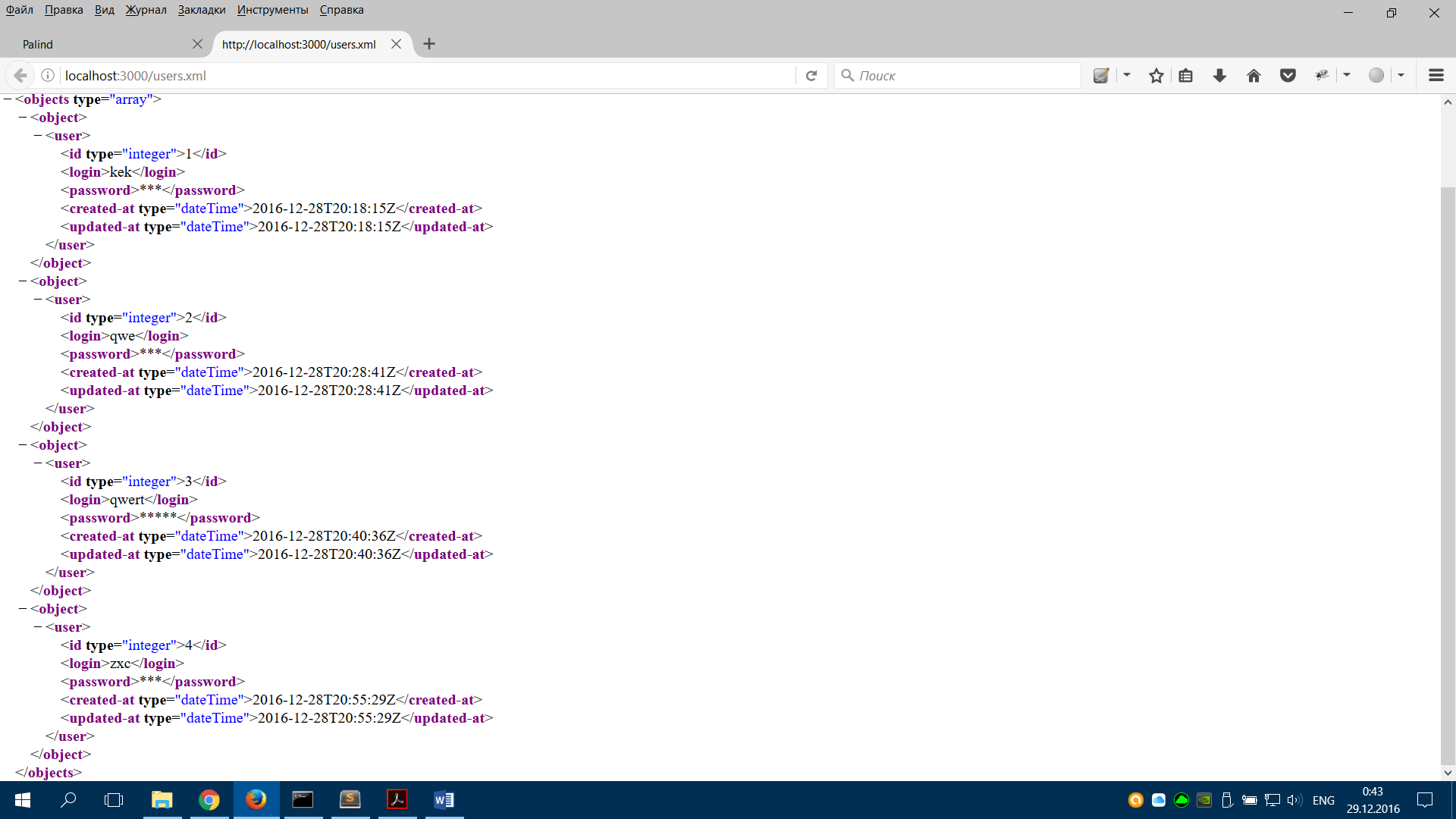
end

end

**Результаты тестирования:**

****

**XML-распечатка базы данных:**

****

**Вывод:** научился работать с сессиями, делать регистрацию и авторизацию пользователей.