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. Интеграционные тесты**

Студент гр. \_ИУ6-34**\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лавренов И.С.**

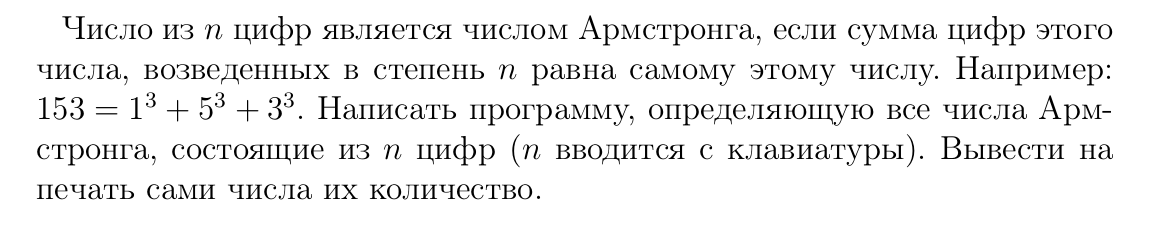
(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2017

Задание.

Созданная миграция 20171223004549\_create\_users.rb:

class CreateUsers < ActiveRecord::Migration[5.1]

def change

create\_table :users do |t|

t.string :username

t.timestamps

end

end

end

Созданная миграция 20171223004715\_add\_password\_digest\_to\_users.rb:

class AddPasswordDigestToUsers < ActiveRecord::Migration[5.1]

def change

add\_column :users, :password\_digest, :string

end

end

Созданная миграция 20171223101245\_add\_remember\_token\_to\_users.rb:

class AddRememberTokenToUsers < ActiveRecord::Migration[5.1]

def change

add\_column :users, :remember\_token, :string

add\_index :users, :remember\_token

end

end

Код модели (user.rb):

class User < ApplicationRecord

has\_secure\_password

validates :username, presence: true, uniqueness: true

before\_create :create\_remember\_token

def User.new\_remember\_token

SecureRandom.urlsafe\_base64

end

def User.encrypt(token)

Digest::SHA1.hexdigest(token.to\_s)

end

private

def create\_remember\_token

self.remember\_token = User.encrypt(User.new\_remember\_token)

end

end

Код представления /users/\_form.html.erb:

<%= form\_with(model: user, local: true) do |form| %>

<% if user.errors.any? %>

<div id="error\_explanation">

<div class='alert alert-danger'>

The form contains <%= pluralize(user.errors.count, "error") %>

<!--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>

</div>

<% end %>

<div class="row">

<div class="span6 offset3">

<%= form\_for(@user) do |f| %>

<%= f.label :username %><br>

<%= f.text\_field :username %><br>

<br>

<%= f.label :password %><br>

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

<br>

<%= f.label :password\_confirmation, "Confirmation" %><br>

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

<%= f.submit "Create my account", class: "btn btn-large btn-primary" %>

<% end %>

</div>

</div>

<% end %>

Код представления /partials/\_nav.html.erb:

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">

<div class="navbar-nav">

<a class="nav-item nav-link active" href="<%= root\_path %>">Home</a>

<% if signed\_in? %>

<a class="nav-item nav-link active" href="<%= session\_destroy\_path %>">Sign out</a>

<% else %>

<a class="nav-item nav-link active" href="<%= new\_user\_path %>">Sign up</a>

<a class="nav-item nav-link active" href="<%= session\_new\_path %>">Sign in</a>

<% end %>

</div>

</div>

</nav>

Код представления /partials/\_flash.html.erb:

<% flash.each do |message\_type, message| %>

<div class="row">

<div class="col">

<div class="alert alert-<%= message\_type %> alert-dismissible fade show" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close" >

<span aria-hidden="true">&times;</span>

</button>

<%= message.html\_safe %>

</div>

</div>

</div>

<% end %>

Код представления /layouts/application.html.erb:

<!DOCTYPE html>

<html>

<head>

<title>Lr8</title>

<%= csrf\_meta\_tags %>

<link rel = "stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700|Material+Icons">

<%= stylesheet\_link\_tag 'application', media: 'all', 'data-turbolinks-track': 'reload' %>

<%= javascript\_include\_tag 'application', 'data-turbolinks-track': 'reload' %>

</head>

<body>

<%= render 'partials/nav' %>

<div class = 'container' id = 'main-container'>

<%= yield %>

</div>

</body>

</html>

Код представления /session/new.html.erb:

<%= render 'partials/flash' %>

<%= provide(:title, "Sign in") %>

<h1>Sign in</h1>

<div class ='row'>

<div class = 'span6 offset3'>

<!--%= form\_tag session\_index\_path do%-->

<%= form\_tag url\_for(controller: :session, action: :create), method: 'post' do %>

<%= label\_tag :username %>

<%= text\_field\_tag :username %>

<%= label\_tag :password %>

<%= password\_field\_tag :password%>

<%= submit\_tag 'Sign in', class: "btn btn-large btn-primary"%>

<% end %>

<p> New user? <%= link\_to "Sign up now!", users\_new\_path %></p>

</div>

</div>

Код контроллера application.rb:

class ApplicationController < ActionController::Base

protect\_from\_forgery with: :exception

before\_action :require\_login

include SessionHelper

private

def require\_login

unless signed\_in?

flash[:danger] = 'You should sign in'

redirect\_to session\_new\_path

end

end

end

Код контроллера users\_controller.rb:

class UsersController < ApplicationController

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

skip\_before\_action :require\_login, only: [:new, :create]

# 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

sign\_in @user

format.html { redirect\_to @user }#, notice: 'User was successfully created.' }

format.json { render :show, status: :created, location: @user }

flash[:success] = 'Welcome'

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

# 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

# Never trust parameters from the scary internet, only allow the white list through.

def user\_params

params.require(:user).permit(:username, :password, :password\_confirmation)

end

end

Код контроллера session\_controller.rb:

class SessionController < ApplicationController

skip\_before\_action :require\_login, only: [:new, :create]

def new # sign up

end

def create # log in

user = User.find\_by\_username(params[:username])

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

#session[:user\_id] = user.id

sign\_in user

redirect\_to root\_path

#redirect\_to user

else

flash[:danger] = 'Invalid username or password'

redirect\_to session\_new\_path

#render 'new'

end

end

def destroy # log out

sign\_out

redirect\_to session\_new\_path

end

end

Код хелпера /app/helpers/session\_helper.rb:

module SessionHelper

def sign\_in(user)

remember\_token = User.new\_remember\_token

cookies.permanent[:remember\_token] = remember\_token

user.update\_attribute(:remember\_token, User.encrypt(remember\_token))

self.current\_user = user

end

def signed\_in?

!current\_user.nil?

end

def current\_user=(user) #?self.current\_user = user?

@current\_user = user

end

def current\_user

remember\_token = User.encrypt(cookies[:remember\_token])

@current\_user ||= User.find\_by\_remember\_token(remember\_token) # если @current\_user не определен

end

def sign\_out

current\_user.update\_attribute(:remember\_token, User.encrypt(User.new\_remember\_token))

cookies.delete(:remember\_token)

self.current\_user = nil

end

end

Код интеграционных тестов /integration/user\_integration\_test.rb:

require 'test\_helper'

class UserIntegrationTest < ActionDispatch::IntegrationTest

test 'require login' do

get lr8\_output\_path, params: { word: 1 }

assert\_redirected\_to session\_new\_path

end

test 'get root after error sign in' do

post session\_create\_path, params: { username: 'user', password: 'password' }

get session\_new\_path

assert\_response :success

end

test 'get root after success sign in' do

post users\_path, params: { user: { username: 'user', password: 'password' } }

get root\_path

assert\_response :success

end

test 'create users' do

old\_count = User.count

post users\_path, params: { user: { username: 'user', password: 'password' } }

new\_count = User.count

assert new\_count > old\_count

end

end

Маршруты routes.rb:

Rails.application.routes.draw do

get 'session/new'

post 'session/create'

get 'session/destroy'

resources :users

get 'users/new'

root to: 'lr8#input'

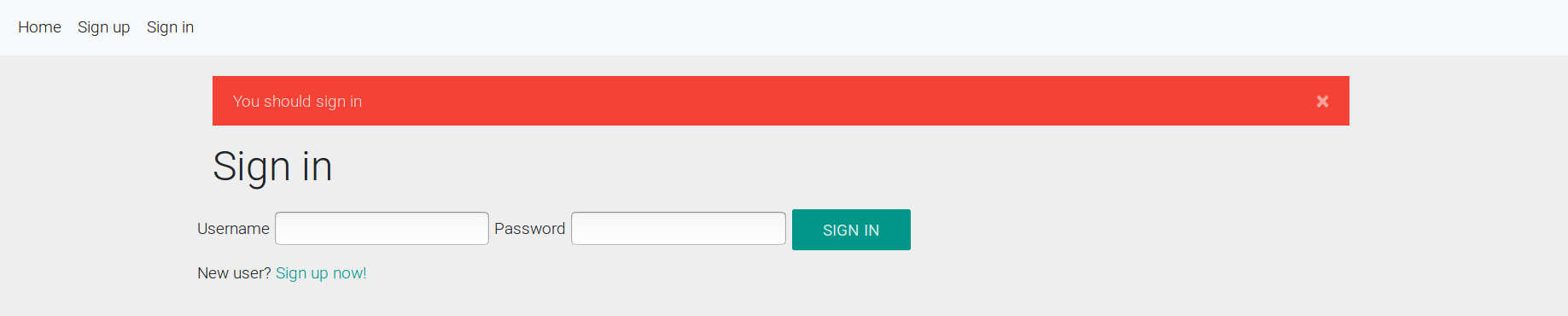
get 'lr8/input'

get 'lr8/output'

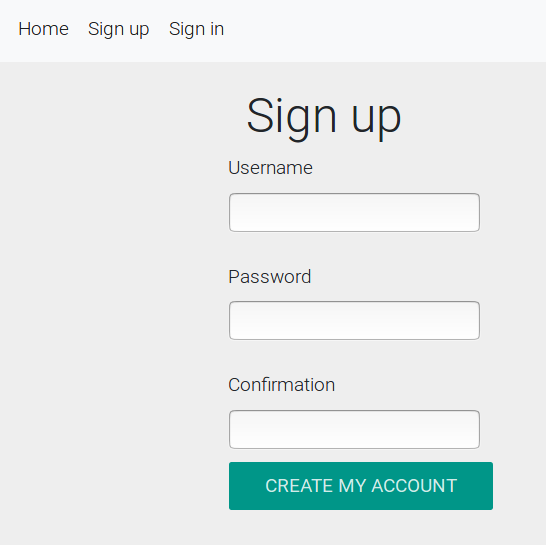
# For details on the DSL available within this file, see http://guides.rubyonrails.org/routing.html

end

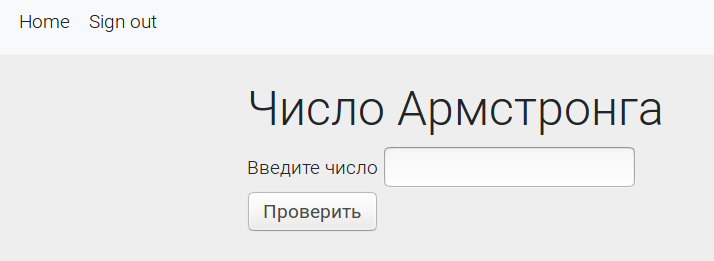
Результаты:



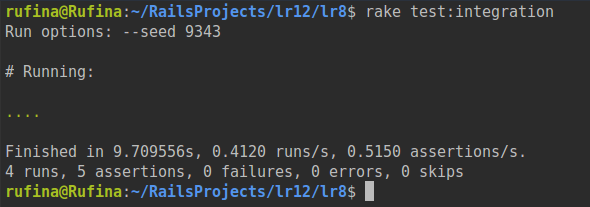
Страница входа



Страница регистрации



Перенаправление на главную страницу после входа



Результат тестирования