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

|  |  |
| --- | --- |
|  | ***«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.

Вариант 10

Студент гр.  **ИУ6-33Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Захаров К.А**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Самарев Р.С.**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

**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.

Измененные файлы:

config/routes.rb  
app/models/user.rb  
db/schema.rb  
app/views/lb12  
app/controllers/lb12\_controller.rb  
test/controllers/lb12\_controller\_test.rb  
spec/requests/devises\_spec.rb

**input.html.erb**

**<h1>Lb12#input</h1>**

**<p>Find me in app/views/lb12/input.html.erb</p>**

**<% if flash[:error].present? -%>**

**<p>Error: <%= flash[:error] %></p>**

**<% end -%>**

**<%= form\_tag("/lb12/view", :method => "get") do %>**

**<%= label\_tag("Input border value") %>**

**<%= number\_field\_tag(:input) %>**

**<%= submit\_tag("result") %>**

**<% end %>**

**<%= link\_to ('Log out'), destroy\_user\_session\_path, method: :delete %>**

**view.html.erb**

**<h1>Lb12#view</h1>**

**<p>Find me in app/views/lb12/view.html.erb</p>**

**<table border="2">**

**<% if @result == "Wrong input!" then %>**

**<p><%= @result %></p>**

**<% else %>**

**<table>**

**<tr class="h">**

**<td>#</td>**

**<td>p</td>**

**<td>M</td>**

**</tr>**

**<% @result.map do |el| %>**

**<tr>**

**<td class="h"><%= el[:id] %></td>**

**<td><%= el[:p] %></td>**

**<td><%= el[:M] %></td>**

**</tr>**

**<% end %>**

**</table>**

**<% end %>**

**<%= link\_to ('Return'), lb12\_input\_path %>**

**app/controllers/lb12\_controller.rb**

**require 'prime'**

**class Lb12Controller < ApplicationController**

**before\_action :authenticate\_user!**

**before\_action :parse\_params, only: :view**

**def input; end**

**def view**

**if @input.blank? || @input =~ /-/ || @input == '0'**

**flash[:error] = 'Wrong input!'**

**redirect\_to lb12\_input\_path**

**return**

**end**

**n = @input.to\_i**

**p = (Math.log(n) / Math.log(2)).to\_i**

**p += 1 if Prime.prime?(n)**

**i = 1**

**arr\_hash = []**

**arr = (1..p).to\_a.keep\_if { |num| Prime.prime?(2\*\*num - 1) }.map do |el|**

**arr\_hash.push(id: i, p: el, M: (2\*\*el - 1))**

**i += 1**

**end**

**@result = arr\_hash**

**end**

**test/controllers/lb12\_controller\_test.rb**

**require 'test\_helper'**

**class Lb12ControllerTest < ActionDispatch::IntegrationTest**

**fixtures :users**

**test "should get input" do**

**get lb12\_input\_url, params: {}**

**assert\_response :redirect**

**end**

**end**

**config/routes.rb**

**Rails.application.routes.draw do**

**devise\_for :users**

**root 'lb12#input'**

**get 'lb12/input'**

**get 'lb12/view'**

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

**end**

**db/schema.rb**

**ActiveRecord::Schema.define(version: 2018\_12\_09\_181003) do**

**create\_table "users", force: :cascade do |t|**

**t.string "email", default: "", null: false**

**t.string "encrypted\_password", default: "", null: false**

**t.string "reset\_password\_token"**

**t.datetime "reset\_password\_sent\_at"**

**t.datetime "remember\_created\_at"**

**t.datetime "created\_at", null: false**

**t.datetime "updated\_at", null: false**

**t.index ["email"], name: "index\_users\_on\_email", unique: true**

**t.index ["reset\_password\_token"], name: "index\_users\_on\_reset\_password\_token", unique: true**

**end**

**app/models/user.rb**

**class User < ApplicationRecord**

**# Include default devise modules. Others available are:**

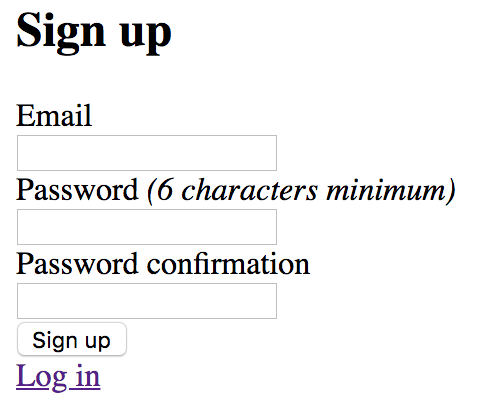
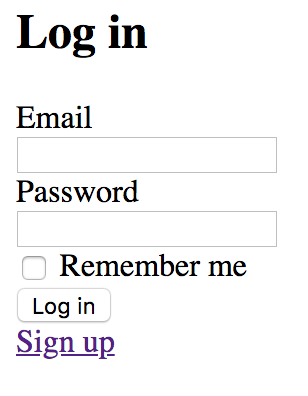
**# :confirmable, :lockable, :timeoutable, :trackable and :omniauthable**

**devise :database\_authenticatable, :registerable,**

**:rememberable, :validatable**

**end**

**Работа программы:**

****

**Автоматические тесты:**

**Run options: --seed 10607**

**# Running:**

**.**

**Finished in 0.438706s, 2.2794 runs/s, 2.2794 assertions/s.**

**1 runs, 1 assertions, 0 failures, 0 errors, 0 skips**

**Проверка rubocop:**

**Inspecting 49 files**

**.................................................**

**49 files inspected, no offenses detected**

**Вывод:** Модифицировали код приложения ЛР 8 таким образом, чтобы приложение требовала регистрацию и аутентификацию пользователя перед вычислениями.