|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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 **09.03.03 «Прикладная информатика»**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 12 |

**Название:**

Сессии. Выполнение авторизации. Хранение

состояния средствами браузера и средствами

сервера. Интеграционные тесты.

**Дисциплина:** Языки Интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-34Б |  | 16.12.2022 | С.А. Рахманов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д. В. Малахов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы:** Получение практических навыков в создании веб-приложений, использующих аутентификацию. Получить навыки написания интеграционных тестов приложений.

**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 /.

**Тексты измененных файлов**

***seq\_controller.rb***

# frozen\_string\_literal: true

# Top-level class

class SeqController < ApplicationController

before\_action :authenticate\_user!

def input; end

def show

params[:str] ? @str = params[:str] : nil

arr = @str.split&.map(&:to\_i)

if arr && !arr&.empty?

tmp = helpers.find\_seqs(arr)

@found = tmp.map { |x| x.join(' ') }.join(' | ')

@max\_found = helpers.find\_max(tmp)

else

@max\_found = @str

@found = @str

end

end

private

def authenticate\_user!

if user\_signed\_in?

puts "User #{current\_user} is authenticated"

else

puts "User isn't authenticated"

redirect\_to new\_user\_session\_path

end

end

end

***users\_controller.rb***

# frozen\_string\_literal: true

# top-level documentation

class UsersController < ApplicationController

def show

@user = current\_user

end

def show\_all

@users = User.all

end

end

***seq\_helper.rb***

# frozen\_string\_literal: true

# top-level documentation

module SeqHelper

def find\_seqs(arr)

found = []

tmp\_found = []

arr.take(arr.size - 1).each\_index do |i|

tmp\_found << arr[i]

unless arr[i] < arr [i + 1]

found << tmp\_found

tmp\_found = []

end

end

tmp\_found << arr[-1]

found << tmp\_found

found

end

def find\_max(alr\_found)

sizes = alr\_found.map(&:size)

alr\_found[sizes.index(sizes.max)]

end

end

***user.rb***

# frozen\_string\_literal: true

class User < ApplicationRecord

# Include default devise modules. Others available are:

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

devise :database\_authenticatable, :registerable,

:recoverable, :rememberable, :validatable

end

***application.html.erb***

<!DOCTYPE html>

<html>

<head>

<title>Lab12</title>

<meta name="viewport" content="width=device-width,initial-scale=1">

<%= csrf\_meta\_tags %>

<%= csp\_meta\_tag %>

<%= stylesheet\_link\_tag "application", "data-turbo-track": "reload" %>

<%= javascript\_importmap\_tags %>

<!-- CSS only -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">

<!-- JavaScript Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<style>

#user\_nav {

font-size: 20px;

}

}

</style>

</head>

<body class="bg-light">

<div class="col-md-10 mx-auto my-4 p-5 shadow">

<div class="container">

<% flash.each do |type, msg| %>

<div class='alert alert-primary alert-dismissible' role="alert">

<%= msg %>

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

</div>

<% end %>

</div>

<%= yield %>

</div>

<div class="col-md-3 mx-auto my-4 align-items-end" id="user\_nav">

<% if current\_user %>

<%= link\_to "Edit Profile", edit\_user\_registration\_path, 'data-turbo-method': :update %> |

<%= link\_to "Show Profile", user\_show\_path, 'data-turbo-method': :show %> |

<%= link\_to "Log out", destroy\_user\_session\_path, 'data-turbo-method': :delete %>

<% else %>

<%= link\_to "Sign Up", new\_user\_registration\_path %> |

<%= link\_to "Log In", new\_user\_session\_path %> |

<%= link\_to "Show users", "/users/show\_all" %>

<% end %>

</div>

</body>

</html>

***devise/sessions/new.html.erb***

<h2>Log in</h2>

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

<div class="field">

<%= f.label :email %><br />

<%= f.email\_field :email, autofocus: true, autocomplete: "email" %>

</div>

<div class="field">

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

<%= f.password\_field :password, autocomplete: "current-password" %>

</div>

<% if devise\_mapping.rememberable? %>

<div class="field">

<%= f.check\_box :remember\_me %>

<%= f.label :remember\_me %>

</div>

<% end %>

<div class="actions">

<%= f.submit "Log in", data: {turbo: false} %>

</div>

<% end %>

<%= render "devise/shared/links" %>

***devise/registrations/new.html.erb***

<h2>Sign up</h2>

<%= form\_for(resource, as: resource\_name, url: registration\_path(resource\_name)) do |f| %>

<%= render "devise/shared/error\_messages", resource: resource %>

<div class="field">

<%= f.label :email %><br />

<%= f.email\_field :email, autofocus: true, autocomplete: "email" %>

</div>

<div class="field">

<%= f.label :password %>

<% if @minimum\_password\_length %>

<em>(<%= @minimum\_password\_length %> characters minimum)</em>

<% end %><br />

<%= f.password\_field :password, autocomplete: "new-password" %>

</div>

<div class="field">

<%= f.label :password\_confirmation %><br />

<%= f.password\_field :password\_confirmation, autocomplete: "new-password" %>

</div>

<div class="actions">

<%= f.submit "Sign up", data: {turbo: false} %>

</div>

<% end %>

<%= render "devise/shared/links" %>

***seq/input.html.erb***

<h1 class="h1">Finding of growing sequences</h1>

<p>Print given, max sequence and all sequences </p>

<%= form\_tag(show\_path, :method => "get") do %>

<%= label\_tag("Input string:") %>

<%= text\_field\_tag :str, params[:str], id: 'input' %> <br/>

<br/>

<%= submit\_tag("Correct and print", class: 'btn btn-primary') %>

<% end %>

***seq/show.html.erb***

<h1 class="h1">Finding of growing sequences</h1>

<p>Print given, max sequence and all sequences </p>

<p id="str">str: <%= @str %></p>

<p id="max found: ">max found: <%= @max\_found %></p>

<p id="found: ">found: <%= @found %></p>

<%= link\_to "Repeat input", '/' %>

***users/show.html.erb***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Пользователь</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>

<link rel="stylesheet" href="view.css">

</head>

<body>

<p><strong>email</strong> <%= @user.email %></p>

<p><strong>created at</strong> <%= @user.created\_at %></p>

<p><strong>updated at</strong> <%= @user.updated\_at %></p>

<div class="price">

<div class="row align-items-end">

<div class="col">

<%= button\_to "Вернуться назад", root\_path,

:class => "btn btn-primary btn-lg active", :method => :get %>

</div>

</div>

</div>

</body>

***users/show\_all.html.erb***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Пользователи</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>

<link rel="stylesheet" href="view.css">

</head>

<body>

<div class="price">

<div class="row align-items-end">

<div class="col">

<% @users.each do |user| %>

<%= user.email %>

</br>

<% end %>

</div>

</br>

<div class="col">

<%= button\_to "Вернуться назад", root\_path,

:class => "btn btn-primary ", :method => :get %>

</div>

</div>

</div>

</body>

***routes.html.erb***

# frozen\_string\_literal: true

Rails.application.routes.draw do

devise\_for :users

resources :users, only: [:index]

root to: 'seq#input'

get 'seq/show', as: :show

get 'users/show', as: :user\_show

get '/users/sign\_out'

get '/users/show\_all'

# Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html

# Defines the root path route ("/")

# root "articles#index"

end

***20221216104243\_devise\_create\_users.rb***

# frozen\_string\_literal: true

# top-level documentation

class DeviseCreateUsers < ActiveRecord::Migration[7.0]

def change

create\_table :users do |t|

## Database authenticatable

t.string :email, null: false, default: ''

t.string :encrypted\_password, null: false, default: ''

## Recoverable

t.string :reset\_password\_token

t.datetime :reset\_password\_sent\_at

## Rememberable

t.datetime :remember\_created\_at

## Trackable

# t.integer :sign\_in\_count, default: 0, null: false

# t.datetime :current\_sign\_in\_at

# t.datetime :last\_sign\_in\_at

# t.string :current\_sign\_in\_ip

# t.string :last\_sign\_in\_ip

## Confirmable

# t.string :confirmation\_token

# t.datetime :confirmed\_at

# t.datetime :confirmation\_sent\_at

# t.string :unconfirmed\_email # Only if using reconfirmable

## Lockable

# t.integer :failed\_attempts, default: 0, null: false # Only if lock strategy is :failed\_attempts

# t.string :unlock\_token # Only if unlock strategy is :email or :both

# t.datetime :locked\_at

t.timestamps null: false

end

add\_index :users, :email, unique: true

add\_index :users, :reset\_password\_token, unique: true

# add\_index :users, :confirmation\_token, unique: true

# add\_index :users, :unlock\_token, unique: true

end

end

***schema.rb***

# frozen\_string\_literal: true

# This file is auto-generated from the current state of the database. Instead

# of editing this file, please use the migrations feature of Active Record to

# incrementally modify your database, and then regenerate this schema definition.

#

# This file is the source Rails uses to define your schema when running `bin/rails

# db:schema:load`. When creating a new database, `bin/rails db:schema:load` tends to

# be faster and is potentially less error prone than running all of your

# migrations from scratch. Old migrations may fail to apply correctly if those

# migrations use external dependencies or application code.

#

# It's strongly recommended that you check this file into your version control system.

ActiveRecord::Schema[7.0].define(version: 20\_221\_216\_104\_243) 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

end

***spec/models/user\_spec..rb***

# frozen\_string\_literal: true

require 'spec\_helper'

require 'rails\_helper'

RSpec.describe Devise, type: :system do

let(:email\_user) { 'gar@amir.tat' }

let(:password\_user) { 'fweu98254jkbfwbiuygbwoiuybvSYHIUEF' }

describe 'User sign up' do

scenario 'sign up without params' do

visit new\_user\_registration\_path

click\_button 'Sign up'

expect(page).to have\_content("errors prohibited this user from being saved:\nEmail can't be blank\nPassword can't be blank\n")

end

scenario 'sign up without email' do

visit new\_user\_registration\_path

fill\_in 'Password', with: password\_user

fill\_in 'Password confirmation', with: password\_user

click\_button 'Sign up'

expect(page).to have\_content("error prohibited this user from being saved:\nEmail can't be blank\n")

end

scenario 'sign up without password' do

visit new\_user\_registration\_path

fill\_in 'Email', with: email\_user

fill\_in 'Password confirmation', with: password\_user

click\_button 'Sign up'

expect(page).to have\_content("Password confirmation doesn't match Password\n")

end

scenario 'sign up, but password confirmation does not match password' do

visit new\_user\_registration\_path

fill\_in 'Email', with: email\_user

fill\_in 'Password', with: password\_user

fill\_in 'Password confirmation', with: "#{password\_user}1"

click\_button 'Sign up'

expect(page).to have\_content("Password confirmation doesn't match Password\n")

end

scenario 'sign up with correct params' do

visit new\_user\_registration\_path

fill\_in 'Email', with: email\_user

fill\_in 'Password', with: password\_user

fill\_in 'Password confirmation', with: password\_user

click\_button 'Sign up'

expect(page).to have\_content("Welcome! You have signed up successfully.\n")

end

end

describe 'User sign in' do

before :each do

visit new\_user\_registration\_path

fill\_in 'Email', with: email\_user

fill\_in 'Password', with: password\_user

fill\_in 'Password confirmation', with: password\_user

click\_button 'Sign up'

click\_link 'Log out'

end

scenario 'sign in without params' do

visit new\_user\_session\_path

click\_button 'Log in'

expect(page).to have\_content("Invalid Email or password.\n")

end

scenario 'sign in with params' do

fill\_in 'Email', with: email\_user

fill\_in 'Password', with: password\_user

click\_button 'Log in'

expect(page).to have\_content("Signed in successfully.\n")

end

scenario 'sign in with params and make calculations' do

fill\_in 'Email', with: email\_user

fill\_in 'Password', with: password\_user

click\_button 'Log in'

fill\_in 'input', with: '1 2 3 4 3 4 5 1 1 1 5 3 4 5 6 10 0 -1 -3'

click\_button 'Correct and print'

expect(page).to have\_current\_path('/seq/show?str=1+2+3+4+3+4+5+1+1+1+5+3+4+5+6+10+0+-1+-3&commit=Correct+and+print')

end

end

describe 'User try to make calculations' do

scenario 'failure make calculations' do

visit '/seq/show?str=1+2+3+4+3+4+5+1+1+1+5+3+4+5+6+10+0+-1+-3&commit=Correct+and+print'

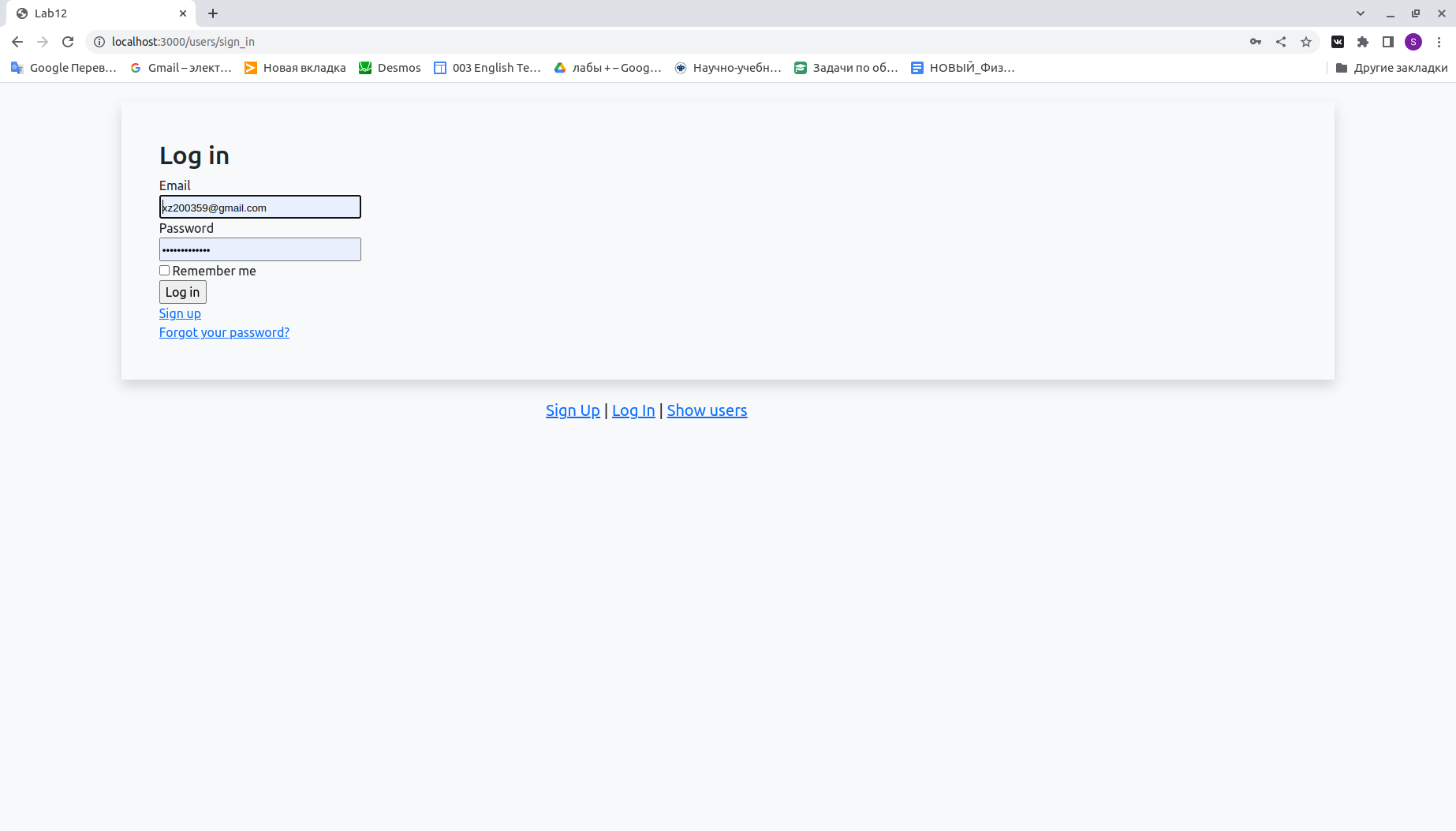
expect(page).to have\_current\_path(new\_user\_session\_path)

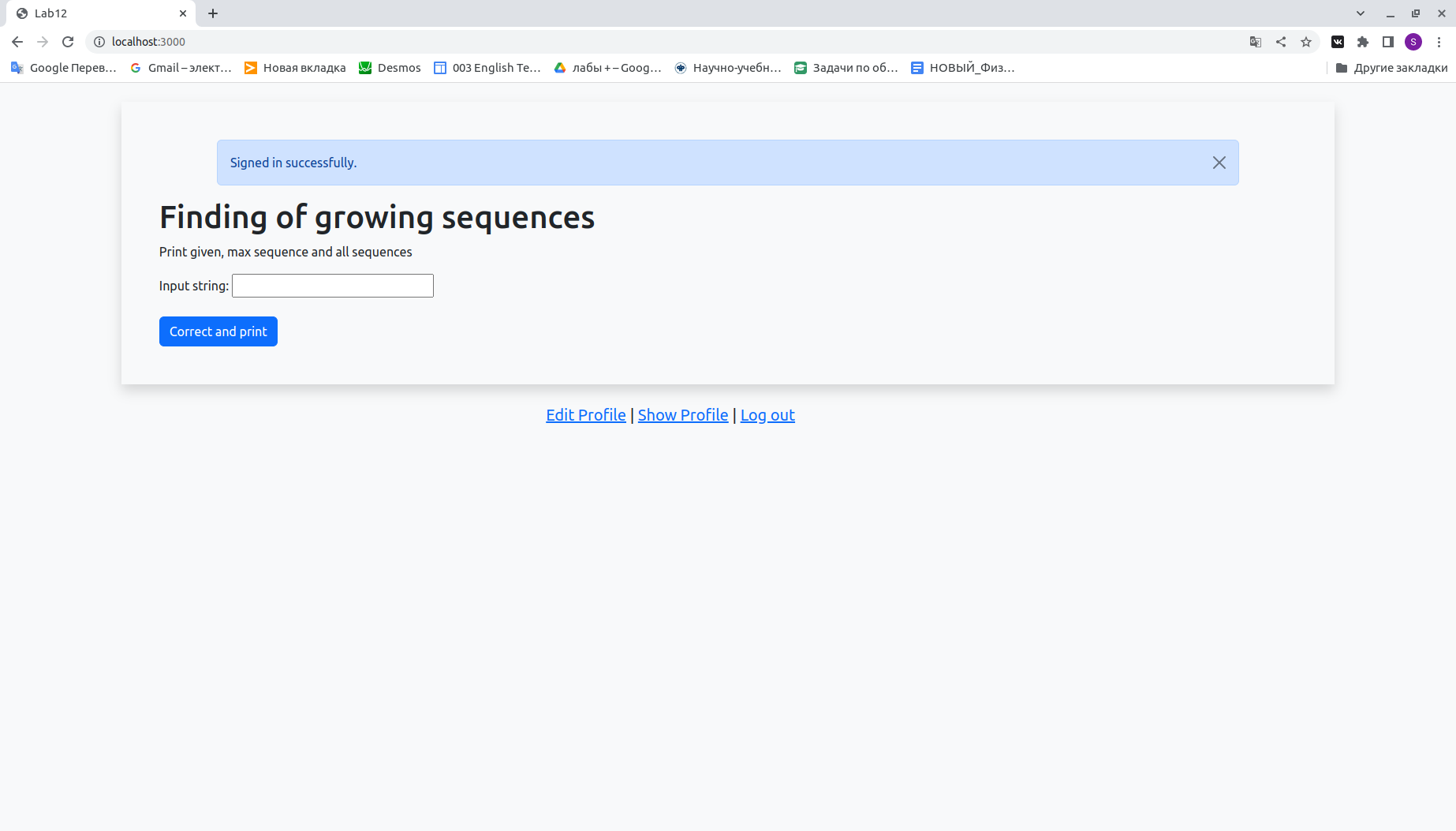
end

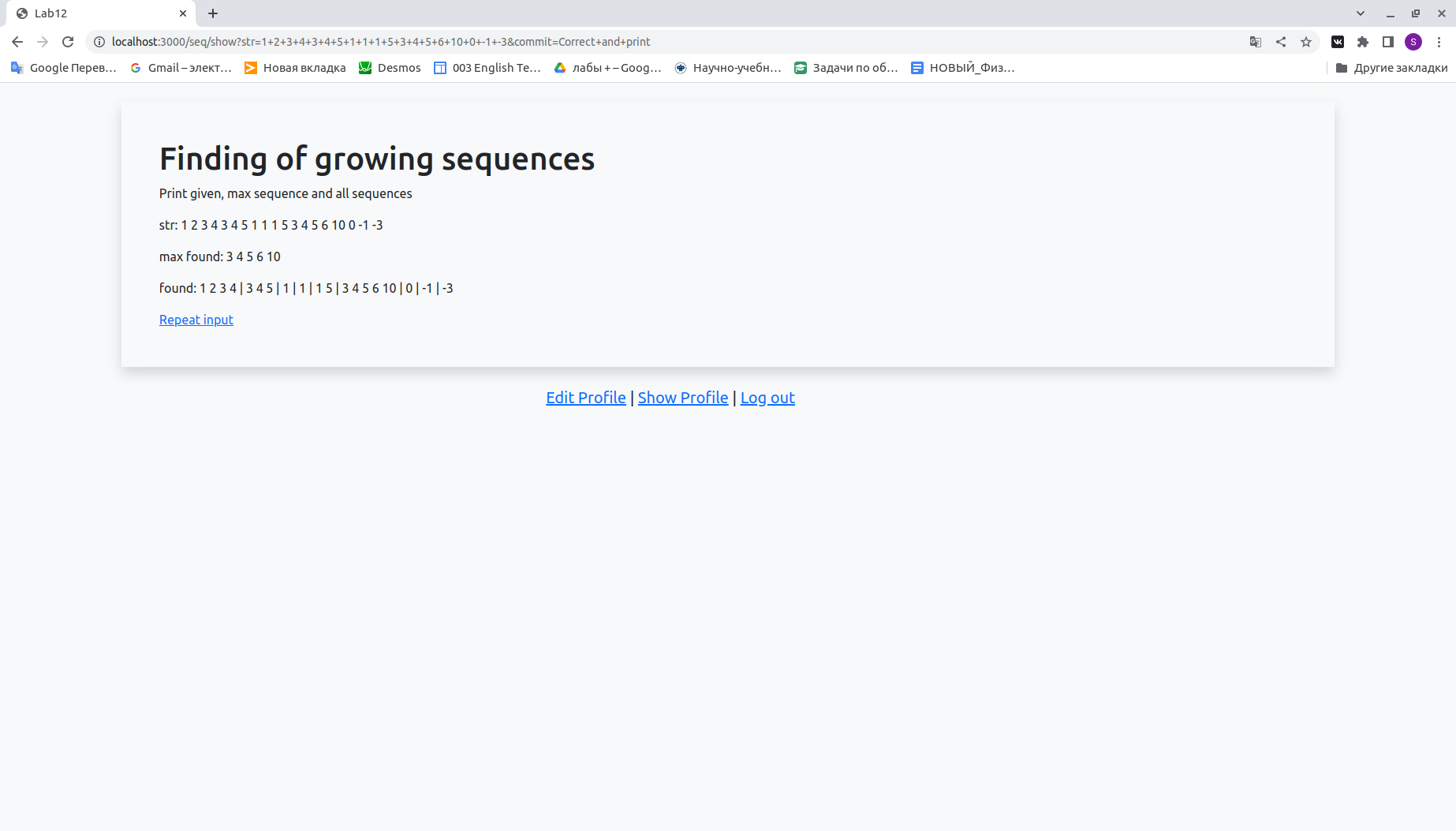
end

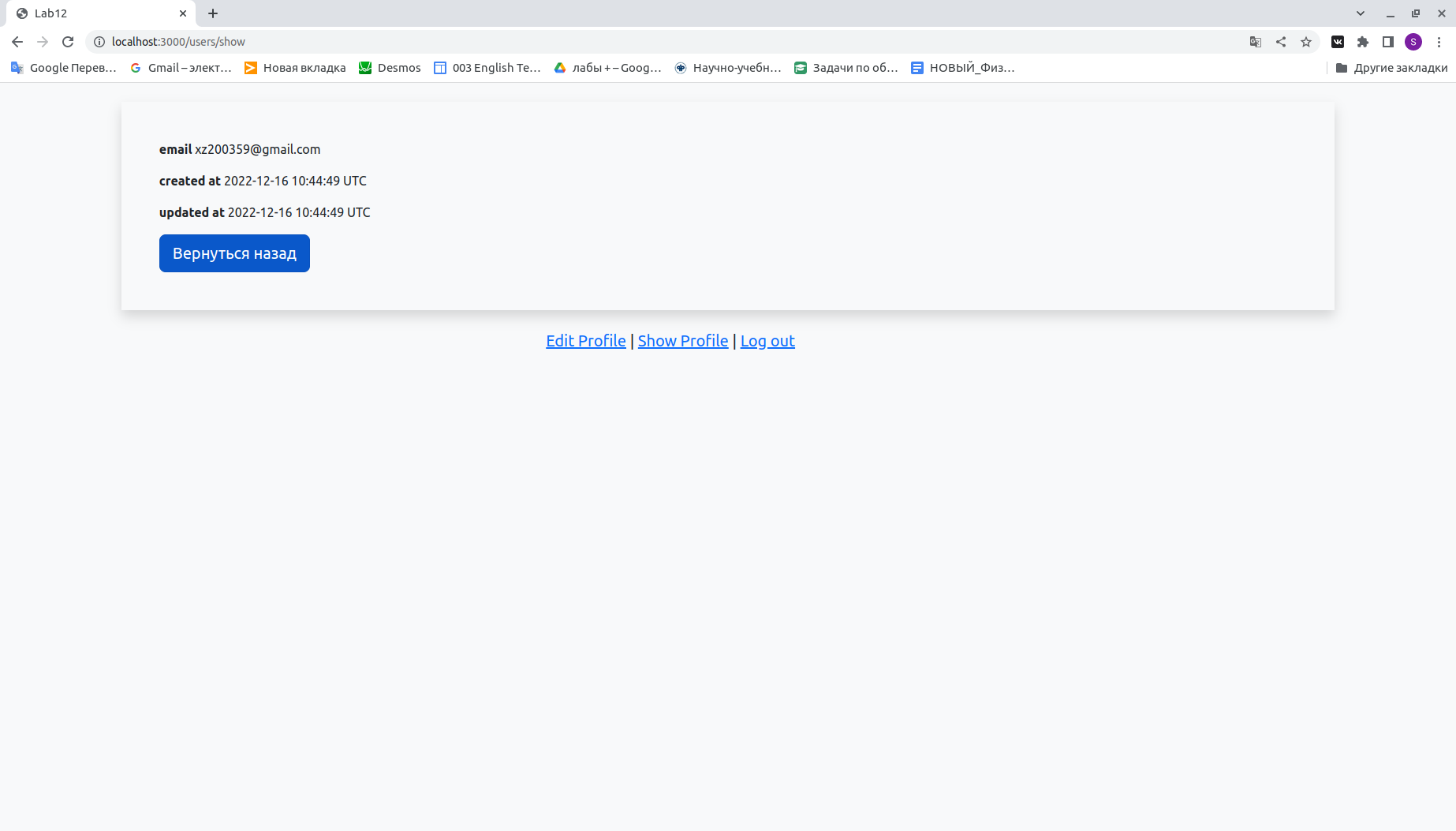
end

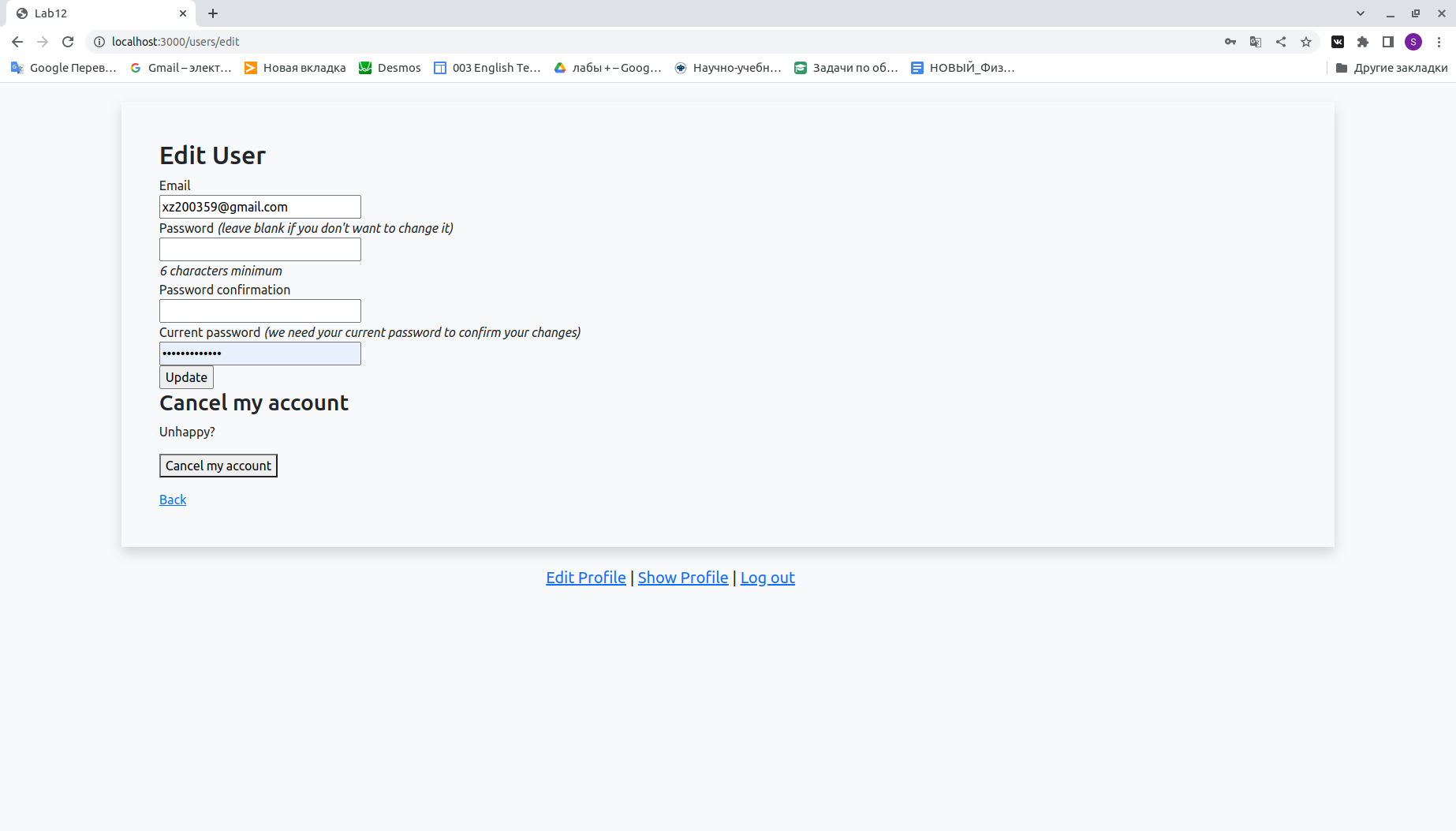
**Результаты выполнения:**

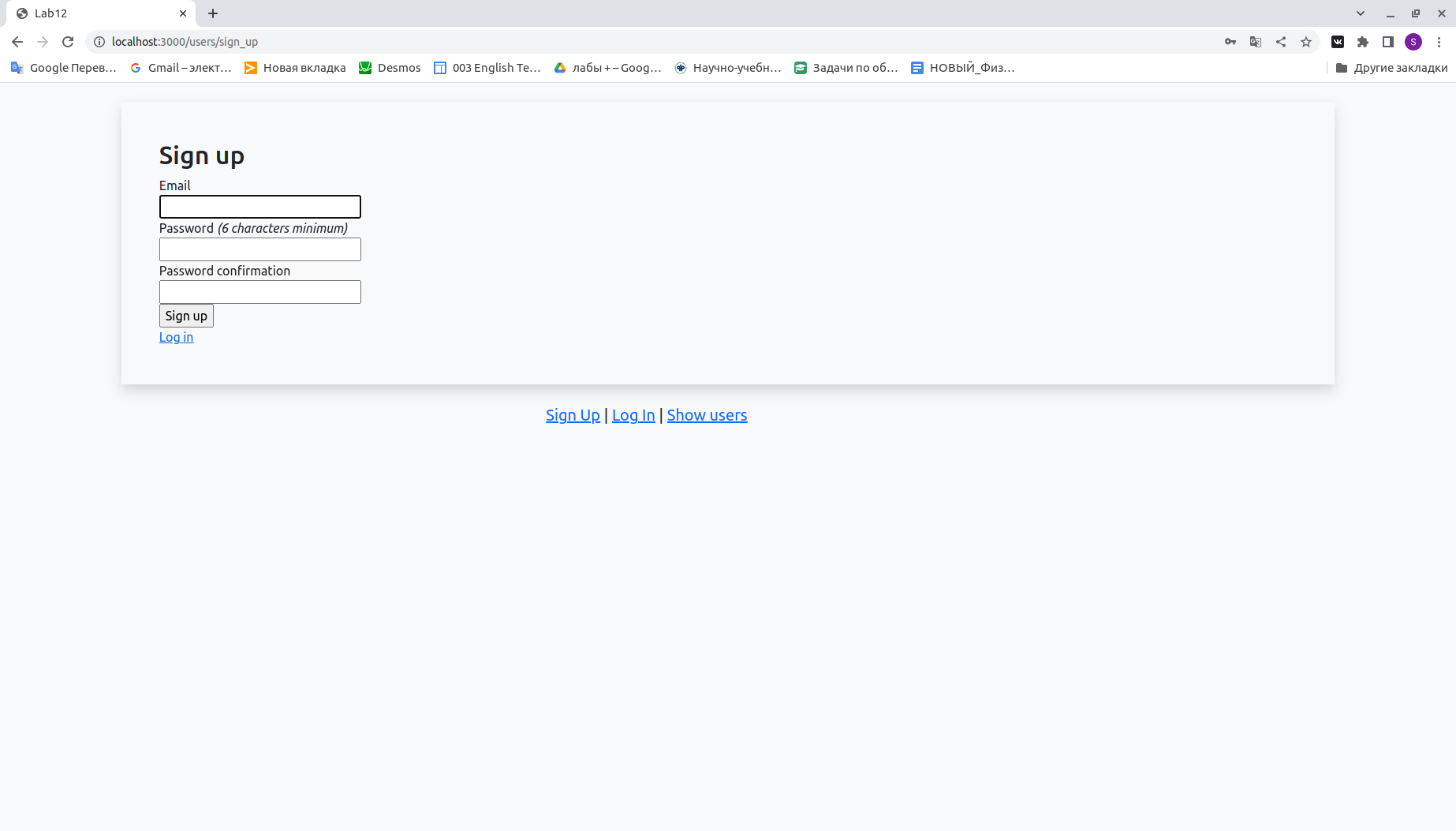


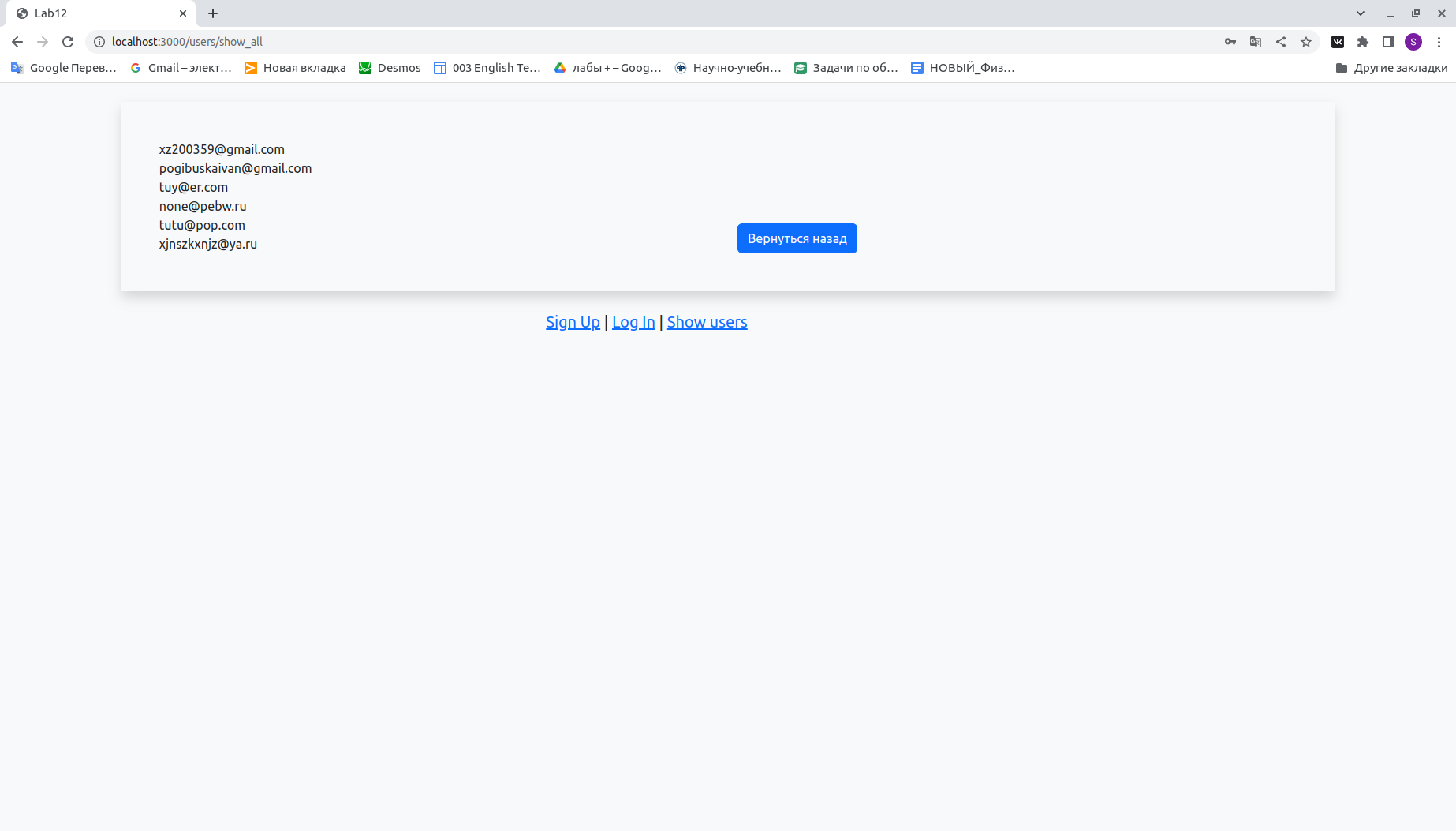




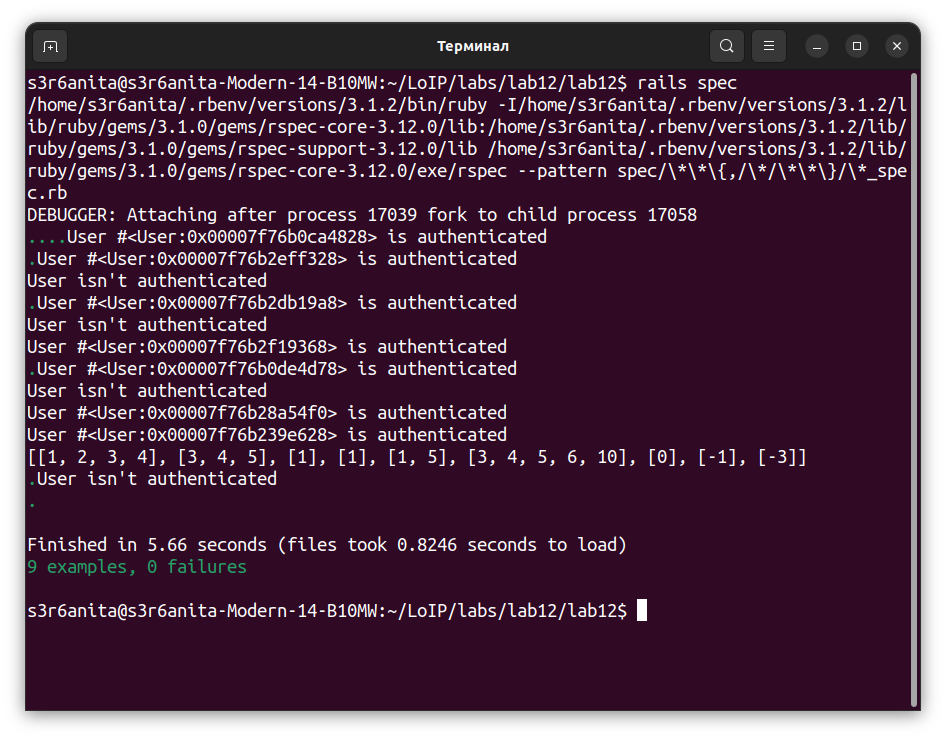




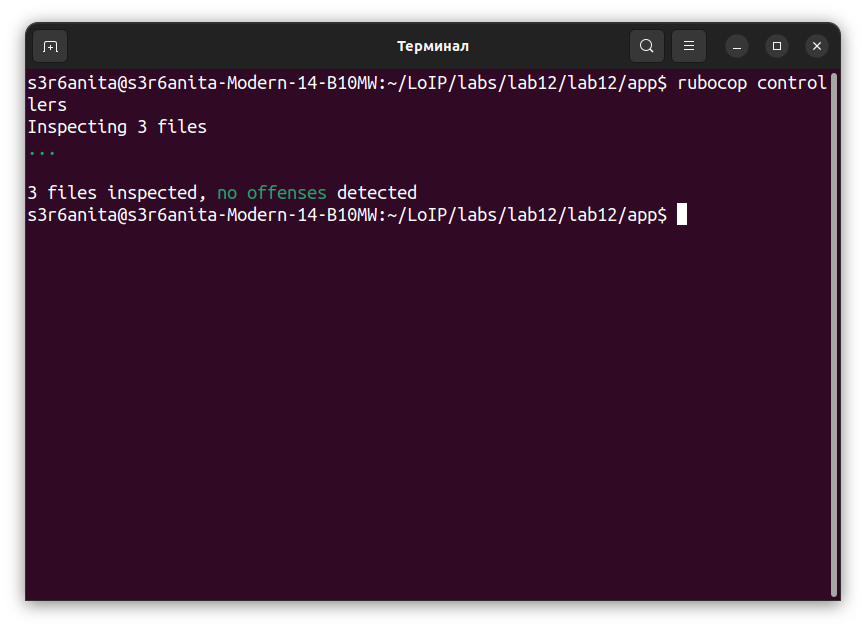




**Тестирование:**



**Результаты проверки анализатором rubocop:**

****

**Вывод:** в результате выполнения лабораторной работы были получены практические навыки в создании веб-приложений, использующих аутентификацию, также были получены навыки написания интеграционных тестов приложений.