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**

О Т Ч Е Т

по лабораторной работе № 1 2

Название: Сессии. Выполнение авторизации. Интеграционные тесты.

Дисциплина: Языки Интернет-программирования

Студент	ИУ6-35Б		19.12.2023	В. И. Мамыкин
	(Группа)		(Подпись, дата)	(И.О. Фамилия)
Преподаватель				Е. Ю. Гаврилова
			(Подпись, дата)	(И.О. Фамилия)

Москва, 2023

Цель: 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 /.

Запросы на добавление, изменение и вывод данных (пример с тестовыми данными):

```
INSERT INTO users (email, password_digest, remember_token, created_at, updated_at)
VALUES ('test@email.com', '$2a$12$fPPfThxkvqZwkVqpNbc3u.ehzJv6v3NanqPkzQyPK5TiCx3vr/joO', 'b95a184d03b82f167ef75d6668096eb1e6dd4d2a', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP);
UPDATE users
SET password_digest='123456789'
WHERE id = 1;
SELECT *
FROM users;
```

	id	email	created_at	updated_at	password_digest	remember_token
1	11	or4cular@yandex.ru	2023-12-18 11:03:57.351782	2023-12-18 17:50:27.376157	$2a$12$HkDyqTzM1o1kKd.ST0ycBeYENv47yajjXXeMMXE.R4JXJlt6GUFyq	89970edc932f1961f207399575ddbf3ac4a5600d
2	12	or4cular1@yandex.ru	2023-12-18 17:52:04.763225	2023-12-19 07:58:01.497866	$2a$12$AUti4NUbTtYcrOHHdAaLne3X.FeTxSsqTQYHJf4B0FddxHmjtAw3u	186baf41ad868669ba8f37bb716654055a57d77d

Рисунок 1 – вывод таблицы БД

Демонстрация таблицы без пароля:

```
CREATE VIEW SHOW_USERS AS
SELECT id, email, created_at, updated_at
FROM users;
SELECT *
```

FROM SHOW_USERS

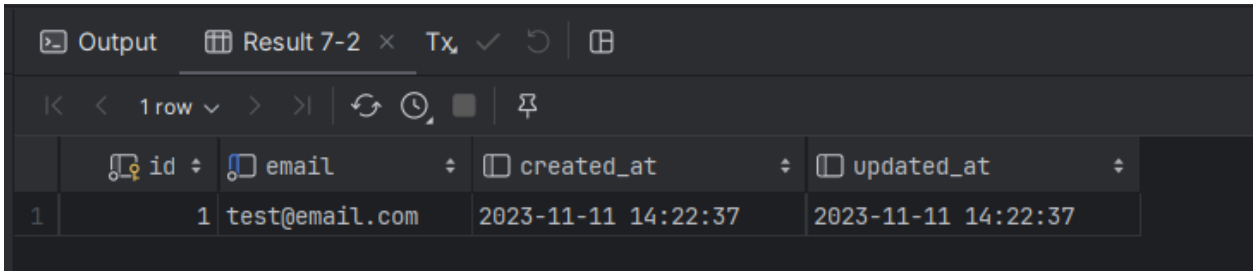

Рисунок 2 – вывод запроса

stylesheets/application.scss

```
/*
 *= require_tree .
 *= require_self
 */

@import "bootstrap";

table {
  border: 3px solid black;
  border-collapse: collapse;
  width: 20%;
}

td {
  border: 1px solid #333;
  text-align: center;
}
```

stylesheets/custom.css.scss

```
@import "bootstrap";

html, body {
  height: 100%;
}

html {
  position: relative;
}

footer {
  position: fixed;
  width: 100%;
  bottom: 0;
}

input, textarea, select, .uneditable-input {
  border: 1px solid #bbb;
  width: 100%;
  margin-bottom: 15px;
}
```

```
input {
  height: auto !important;
}
```

controllers/application_controller.rb

```
class ApplicationController < ActionController::Base
  protect_from_forgery with: :exception
  include SessionsHelper
end
```

controllers/sessions_controller.rb

```
class SessionsController < ApplicationController
  def new
  end

  def create
    msg_text = ''
    msg_status = :success

    email = params[:session][:email]
    password = params[:session][:password]

    puts password

    respond_to do |format|
      user = User.find_by(email: email.downcase)
```

```
      if !user
        msg_text = 'Пользователя не существует'
        msg_status = :danger
      elsif !user.authenticate(password)
        msg_text = 'Неверный пароль'
        msg_status = :danger
      end

      if msg_status == :success
        sign_in user
        msg_text = 'Вы успешно вошли'
        flash[msg_status] = msg_text
        format.html { redirect_to input_path }
        format.json { render :show, status: :created, location: input_path }
      else
        flash.now[msg_status] = msg_text
        format.html { render :new, status: :unprocessable_entity }
        format.json { render json: @user.errors, status: :unprocessable_entity }
      end
    end
  end

  def destroy
    sign_out
    redirect_to root_url
  end
end
```

controllers/sequences_controller.rb

```
# frozen_string_literal: true
class SequencesController < ApplicationController
  def input
    unless signed_in?
      redirect_to signin_path
    end
  end

  def view
    unless signed_in?
      redirect_to signin_path
    end

    longest_subsequence = []  # Самая длинная подпоследовательность
    current_subsequence = []  # Текущая подпоследовательность
    all_subsequences = [] # Все подпоследовательности
    # unless params[:v2].nil?
    sequence = params[:n]&.split(' ')&.map(&:to_i)
    sequence&.each do |number|
      if (Math.sqrt(number) % 1).zero?
        # Если число является полным квадратом
        current_subsequence << number
      else
        # Если число не является полным квадратом
        longest_subsequence = current_subsequence.clone if
current_subsequence.length > longest_subsequence.length
        cur = current_subsequence.clone
        all_subsequences << cur.join(' ')
```

```
      all_subsequences.pop if all_subsequences[all_subsequences.size - 1] == ''
      current_subsequence = []
    end
  end
  cur = current_subsequence.clone
  all_subsequences << cur.join(' ') if cur.length.positive?
  longest_subsequence = current_subsequence.clone if current_subsequence.length > longest_subsequence.length
  all_subsequences.pop if all_subsequences[all_subsequences.size - 1] == ''
  subsequence_count = longest_subsequence.length
  @result = [all_subsequences, longest_subsequence.join(' '), subsequence_count.to_s, sequence&.join(' ')]
  @table = '<table>' # Начало таблицы
  if @result[2] != '0'
    @table +=
      "<tr><td>Введенная последовательность:</td></tr><tr><td>#{@result[3]}</td></tr><tr><td>Подпо следовательности:</td></tr>"

    @result[0].each do |res|
      @table += "<tr><td>#{res}</td></tr>"
    end

    @table +=
      "<tr><td>Самая длинная подпоследовательность:</td></tr><tr><td>#{@result[1]}</td></tr><tr><td>Ее длина:</td></tr><tr><td>#{@result[2]}</td></tr>"
  else
    @table +=
```

```
        "<tr><td>Введенная
последовательность:</td></tr><tr><td>#{@result[3]}</td></tr><tr><td>Подпо
лседовательностей квадратов натуральных чисел нет</td></tr>"
    end
    @table += '</table>' # Конец таблицы
    @tmp = @result.clone
    @result = [@tmp, @table]
  end
end
```

controllers/users_controller.rb

```
class UsersController < ApplicationController
  before_action :set_user, only: %i[ show edit update destroy ]

  # GET /users or /users.json
  def index
    @users = User.all
  end

  # GET /users/1 or /users/1.json
  def show
  end

  # GET /users/new
  def new
    @user = User.new
  end

  # GET /users/1/edit
  def edit
```

```
  end

  # POST http://127.0.0.1:3000/signup
  def create
    msg_text = ''
    msg_status = :success

    email = params[:user][:email]
    password = params[:user][:password]
    password_confirmation = params[:user][:password_confirmation]
    puts password

    @user = User.new(user_params)

    respond_to do |format|

      if @user
        if User.find_by_email(email)
          msg_text = 'Пользователь уже зарегестрирован!'
          msg_status = :danger
        elsif password != password_confirmation
          msg_text = 'Пароль для подтверждения введен неверно'
          msg_status = :danger
        elsif       !email.match?('[a-z0-9]+[_a-z0-9\.-]*[a-z0-9]+@[a-z0-9-]+(\.[a-z0-9-
]+)*(\.[a-z]{2,4})')
          msg_text = 'Введите почту корректно'
          msg_status = :danger
        end

        if msg_status == :success and @user.save
```

```
        sign_in @user
        msg_text = 'Спасибо за регистрацию'
        flash[msg_status] = msg_text

        format.html { redirect_to input_path }
        format.json { render :show, status: :created, location: input_path }
      else
        flash.now[msg_status] = msg_text
        format.html { render :new, status: :unprocessable_entity }
        format.json { render json: @user.errors, status: :unprocessable_entity }
      end
    end
  end
end

# PATCH/PUT /users/1 or /users/1.json
def update
  respond_to do |format|
    if @user.update(user_params)
      format.html { redirect_to user_url(@user), notice: "User was successfully updated." }
      format.json { render :show, status: :ok, location: @user }
    else
      format.html { render :edit, status: :unprocessable_entity }
      format.json { render json: @user.errors, status: :unprocessable_entity }
    end
  end
end

# DELETE /users/1 or /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

  # Only allow a list of trusted parameters through.
  def user_params
    params.require(:user).permit(:email, :password)
  end
end
```

helpers/sessions_helper.rb

```
module SessionsHelper
  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 sign_out
    current_user.update_attribute(:remember_token,
                                  User.encrypt(User.new_remember_token))
    cookies.delete(:remember_token)
    self.current_user = nil
  end

  def current_user=(user)
    @current_user = user
  end

  # Пользователь является вошедшим если в сессии существует текущий пользователь, т.е., если current_user не является nil
  def signed_in?
    !current_user.nil?
  end

  # Поиск текущего пользователя с помощью remember_token.
  def current_user
    remember_token = User.encrypt(cookies[:remember_token])
    @current_user ||= User.find_by(remember_token: remember_token)
  end
end
```

javascript/controllers/application.js

```
// Configure your import map in config/importmap.rb. Read more: https://github.com/rails/importmap-rails
import "@hotwired/turbo-rails"
import "controllers"

import "jquery"
import "jquery_ujs"
import "popper"
import "bootstrap"
import "src/main"
```

javascript/src/main.js

```
$(document).on('click', '.btn-close', function () {
   $('.alert').fadeOut();
});
```

models/user.rb

```
class User < ApplicationRecord
  before_save { self.email = email.downcase }
  before_create :create_remember_token

  VALID_EMAIL_REGEX = /\A[\w+\-.]+@[a-z\d\-.]+\.[a-z]+\z/i
  has_secure_password
  validates :email, presence: true,
          format: { with: VALID_EMAIL_REGEX },
          uniqueness: { case_sensitive: false }
```

```
  validates :password, length: { minimum: 4 }

  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
```

integration/authentication_pages_test.rb

```
require "test_helper"
#
https://www.softcover.io/read/28fdb94f/ruby_on_rails_tutorial_3rd_edition/sign_up

class AuthenticationPagesTest < ActionDispatch::IntegrationTest
  def add_record(email, password)
    record = User.new(:email => email, :password => password)
    record.save
    record
  end
```

```
  ################################### Sign up
#######################################
  # Проверяем доступность страницы регистрации
  test "test registration page access" do
    get signup_url
    assert_response :success
  end

  # Проверяем, что нельзя зарегестрировать того же пользователя
  test 'attempt to register with existing user details' do
    # Создаем пользователя
    add_record('test@test.com', '123456')

    get signup_url
    assert_response :success

    post users_url, params: { "authenticity_token" => "token", "user" => { "email"
=> "test@test.com", "password" => "123456", "password_confirmation" =>
"123456" } }

    assert_response 422
  end

  # Проверяем, что пользователя можно зарегестрировать
  test 'successfully user registration' do
    get signup_url
    assert_response :success

    # Смотрим, что такой пользователь только 1
```

```
    assert_difference 'User.count', 1 do
      post users_url, params: { "authenticity_token" => "token", "user" => { "email"
=> "test@test.com", "password" => "123456", "password_confirmation" =>
"123456" } }
      follow_redirect!
    end

    assert_template 'input'
    assert_response 200
  end

  ##################################### Sign in
########################################
  # Проверяем доступность страницы входа
  test "test login page access" do
    get signin_url
    assert_response :success
  end

  test 'successfully user login' do
    add_record('test@test.com', '123456')

    assert_difference 'User.count', 0 do
      post sessions_url, params: { "authenticity_token" => "token", "session" => {
"email" => "test@test.com", "password" => "123456" } }
      follow_redirect!
    end

    assert_template 'input'
    assert_response 200
```

```
  end

  test 'login of a non-existent user' do
    post sessions_url, params: { "authenticity_token" => "token", "session" => {
"email" => "test@test.com", "password" => "123456" } }

    assert_template 'sessions/new'
    assert_response 422
  end

  test 'login without password' do
    post sessions_url, params: { "authenticity_token" => "token", "session" => {
"email" => "test@test.com", "password" => "" } }

    assert_template 'sessions/new'
    assert_response 422
  end

  ################################### Sign out
#######################################
  test "test logout success" do
    # Добавляем тестового юзера в БД
    add_record('test@test.com', '123456')

    assert_difference 'User.count', 0 do
      # login
      post sessions_url, params: { "authenticity_token" => "token", "session" => {
"email" => "test@test.com", "password" => "123456" } }
      follow_redirect!
    end
```

```
    assert_difference 'User.count', 0 do
      # Logout
      delete signout_url
      follow_redirect! # перенаправлены в input, там проверочка, что не
залогинились и иедм в login
      follow_redirect! # из input в login
    end

    assert_template 'sessions/new'
    assert_response 200
  end

  ##################################### Вычисления не возможны без входа
########################################
  test "Calculations are impossible without sign in" do
    # view
    get view_url, params: { n: '10' }

    # Если не вошли, значит редиректимся в signin
    assert_response 302

    # input
    get input_url
    assert_response 302
  end
end
```

layouts/_footer.html.erb

```
<footer style="background-color: #1A237E; color: white;">
```

```
  <div class="text-center p-3 text-light">
    made by Mamykin V. I.
  </div>
</footer>
```

layouts/_header.html.erb

```
<header>
  <nav class="navbar navbar-expand-lg navbar-dark" style="background-color: #1A237E; color: white;">
    <div class="container-fluid">
      <a class="navbar-brand" href="#" style="border: 1px solid white; border-radius: 10px;">LW12</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div id="navbarNav" style="align-content: center; margin-left: auto; margin-right: auto;">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <% if signed_in? %>
            <li class="nav-item">
              <%= link_to 'Ввод', input_path, class: "nav-link  #{ request.path == input_path || request.path == '/' ? 'active' : ''  }" %>
            </li>
            <li class="nav-item">
              <%= link_to 'Вывод', view_path, class: "nav-link  #{ request.path == view_path ? 'active' : ''  }" %>
            </li>
            <li class="nav-item">
```

```
          <%= link_to 'Выход', signout_path, method: "delete", class: "nav-link  #{ request.path == signout_path ? 'active' : "  }" %>
          </li>
        <% else %>
          <li class="nav-item">
          <%= link_to 'Вход', signin_path, class: "nav-link  #{ request.path == signin_path ? 'active' : "  }" %>
        <% end %>
       </ul>
     </div>
   </div>
  </nav>
</header>
```

layouts/_shim.html.erb

```
<!--[if lt IE 9]>
<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

layouts/application.html.erb

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <title>sequences</title>
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <link rel="icon" type="image/png" href="/assets/logo.png">
  <%= csrf_meta_tags %>
```

```
  <%= csp_meta_tag %>

  <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
  <%= javascript_importmap_tags %>
  <%= render 'layouts/shim' %>
</head>

<body>
<%= render 'layouts/header' %>
<div class="container">
  <% flash.each do |key, value| %>
    <div class="alert alert-<%= key %> alert-dismissible fade show" role="alert"><%= value %>
      <button id="flash-close" type="button" class="btn-close" data-dismiss="alert">
      </button>
    </div>
  <% end %>
  <%= yield %>
</div>

<%= render 'layouts/footer' %>
</body>
</html>
```

sessions/new.html.erb

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


<div class="row">
```

```
  <div class="col-4"></div>
  <div class="col-4">
    <div class="span6 offset3">
      <h1>Вход</h1>
      <%= form_for(:session, url: sessions_path) do |f| %>

        <%= f.label :email %>
        <%= f.text_field :email, class: "form-control", required: true %>

        <%= f.label 'Пароль' %>
        <%= f.password_field :password, minlength: 4, class: "form-control", required:
true %>

        <%= f.submit "Войти", class: "btn btn-large btn-primary mt-4" %>
      <% end %>

      <p>Нет аккаунта? <%= link_to "Регистрация", signup_path %></p>
    </div>
  </div>
  <div class="col-4"></div>
</div>
```

sequences/input.html.erb

```
<div class="row">
  <div class="span6 offset3">
    <div class="row">
      <div class="col-4">
```

```
    </div>
    <div class="col-4">
     <h4 style="align-content: center; margin-left: auto; margin-right: auto;">Sequences#input</h4>
     <form action="<%= view_path %>" method="get" accept-charset="UTF-8">
      <div class="form-group">
       <label for="n"> Последовательность: </label>
       <input type="text" id="n" name="n" pattern="^([0-9]+[\s]{0,1})+" value="1 2 3 4 9 16 3 2 4 10" required/>
      </div>

      <button type="submit" class="btn btn-large btn-primary mt-4">Ввести</button>
     </form>
    </div>
    <div class="col-4">
    </div>
   </div>
  </div>
</div>
```

sequences/view.html.erb

```
<div style="align-content: center; margin-left: auto; margin-right: auto;">
```

```
    <div class="row">
      <div class="col-3">
      </div>
      <div class="col-6" style="margin-left: 180px; margin-right: auto;">
        <h4>Sequences#view</h4>
        <%= @table.html_safe %>
        <br/>
        <%= link_to "Рассчитать заново", input_path %>
      </div>
      <div class="col-3">
      </div>
    </div>
</div>
```

users/new.html.erb

```
<% provide(:title, 'Sign up') %>



<div class="row">
  <div class="col-4"></div>
  <div class="col-4">
    <div class="span6 offset3">
      <h1>Регистрация</h1>
```

```
    <%= form_for(@user) do |f| %>

      <%= f.label :email %>
      <%= f.text_field :email, class: "form-control", required: true %>

      <%= f.label 'Пароль' %>
      <%= f.password_field :password, class: "form-control", required: true %>

      <%= f.label 'Подтверждение пароля', 'Confirmation' %>
      <%= f.password_field :password_confirmation, minlength: 4, class: "form-control", suggested: "new-password", required: true %>

      <%= f.submit "Зарегестрироваться", class: "btn btn-large btn-primary mt-4" %>
      <p>Уже есть аккаунт? <%= link_to 'Вход', signin_path%></p>
    <% end %>
   </div>
  </div>
  <div class="col-4"></div>
</div>
```

environments/development.rb

```
require "active_support/core_ext/integer/time"
```

```
Rails.application.configure do
  # Settings specified here will take precedence over those in config/application.rb.

  # In the development environment your application's code is reloaded any time
  # it changes. This slows down response time but is perfect for development
  # since you don't have to restart the web server when you make code changes.
  config.cache_classes = false

  # Do not eager load code on boot.
  config.eager_load = false

  # Show full error reports.
  config.consider_all_requests_local = true

  # Enable server timing
  config.server_timing = true

  # Enable/disable caching. By default caching is disabled.
  # Run rails dev:cache to toggle caching.
  if Rails.root.join("tmp/caching-dev.txt").exist?
    config.action_controller.perform_caching = true
    config.action_controller.enable_fragment_cache_logging = true

    config.cache_store = :memory_store
    config.public_file_server.headers = {
      "Cache-Control" => "public, max-age=#{2.days.to_i}"
    }
  else
    config.action_controller.perform_caching = false
```

```
  config.cache_store = :null_store
end

# Store uploaded files on the local file system (see config/storage.yml for options).
config.active_storage.service = :local

# Don't care if the mailer can't send.
config.action_mailer.raise_delivery_errors = false

config.action_mailer.perform_caching = false

# Print deprecation notices to the Rails logger.
config.active_support.deprecation = :log

# Raise exceptions for disallowed deprecations.
config.active_support.disallowed_deprecation = :raise

# Tell Active Support which deprecation messages to disallow.
config.active_support.disallowed_deprecation_warnings = []

# Raise an error on page load if there are pending migrations.
config.active_record.migration_error = :page_load

# Highlight code that triggered database queries in logs.
config.active_record.verbose_query_logs = true

# Suppress logger output for asset requests.
config.assets.quiet = true

# Raises error for missing translations.
```

```
  # config.i18n.raise_on_missing_translations = true

  # Annotate rendered view with file names.
  # config.action_view.annotate_rendered_view_with_filenames = true

  # Uncomment if you wish to allow Action Cable access from any origin.
  # config.action_cable.disable_request_forgery_protection = true

  # config.action_mailer.default_url_options = { host: 'localhost', port: 3000 }
  # config.action_view.preload_links_header = false
  # config.action_view.automatically_disable_submit_tag = false
end
```

initializers/assets.rb

```
# Be sure to restart your server when you modify this file.

# Version of your assets, change this if you want to expire all your assets.
Rails.application.config.assets.version = "1.0"
Rails.application.config.assets.precompile += %w( jquery.min.js jquery_ujs.js bootstrap.min.js popper.js )
```

routes.rb

```
Rails.application.routes.draw do
  resources :users
  resources :sessions, only: [:new, :create, :destroy]
```

```
  match '/input', to: 'sequences#input', via: 'get'
  match '/view', to: 'sequences#view', via: 'get'

  match '/signup', to: 'users#new', via: 'get'
  match '/signin', to: 'sessions#new', via: 'get'
  match '/signout', to: 'sessions#destroy', via: 'delete'

  root 'sequences#input'
end
```

importmap.rb

```
# Pin npm packages by running ./bin/importmap

pin "application", preload: true
pin "@hotwired/turbo-rails", to: "turbo.min.js", preload: true
pin "@hotwired/stimulus", to: "stimulus.min.js", preload: true
pin "@hotwired/stimulus-loading", to: "stimulus-loading.js", preload: true
pin_all_from "app/javascript/controllers", under: "controllers"

pin_all_from 'app/javascript/src', under: 'src'
pin "jquery", to: "jquery.min.js", preload: true
pin "jquery_ujs", to: "jquery_ujs.js", preload: true
pin "popper", to: "popper.js", preload: true
pin "bootstrap", to: "bootstrap.min.js", preload: true
```

Gemfile

```
# frozen_string_literal: true
```

```
source 'https://rubygems.org'
git_source(:github) { |repo| "https://github.com/#{repo}.git" }

gem "jquery-rails"
gem "bootstrap"
gem "sassc-rails"

gem 'rails-controller-testing'

# Bundle edge Rails instead: gem "rails", github: "rails/rails", branch: "main"
gem 'rails', '7.1.2'

# The original asset pipeline for Rails [https://github.com/rails/sprockets-rails]
gem 'sprockets-rails'

# Use sqlite3 as the database for Active Record
gem 'sqlite3'

# Use the Puma web server [https://github.com/puma/puma]
gem 'puma'

# Use JavaScript with ESM import maps [https://github.com/rails/importmap-rails]
gem 'importmap-rails'

# Hotwire's SPA-like page accelerator [https://turbo.hotwired.dev]
gem 'turbo-rails'

# Hotwire's modest JavaScript framework [https://stimulus.hotwired.dev]
gem 'stimulus-rails'
```

```
# Build JSON APIs with ease [https://github.com/rails/jbuilder]
gem 'jbuilder'

# Use Active Model has_secure_password [https://guides.rubyonrails.org/active_model_basics.html#securepassword]
gem "bcrypt"

# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem 'tzinfo-data', platforms: %i[ mingw mswin x64_mingw jruby ]

# Reduces boot times through caching; required in config/boot.rb
gem 'bootsnap', require: false

group :development, :test do
  # See https://guides.rubyonrails.org/debugging_rails_applications.html#debugging-with-the-debug-gem
  gem 'debug', platforms: %i[ mri mingw x64_mingw ]
  gem 'rspec-rails'
end

group :development do
  # Use console on exceptions pages [https://github.com/rails/web-console]
  gem 'web-console'
end

group :test do
  # Use system testing [https://guides.rubyonrails.org/testing.html#system-testing]
  gem 'capybara'
```

```
  gem 'selenium-webdriver'

end
gem "bcrypt"
```

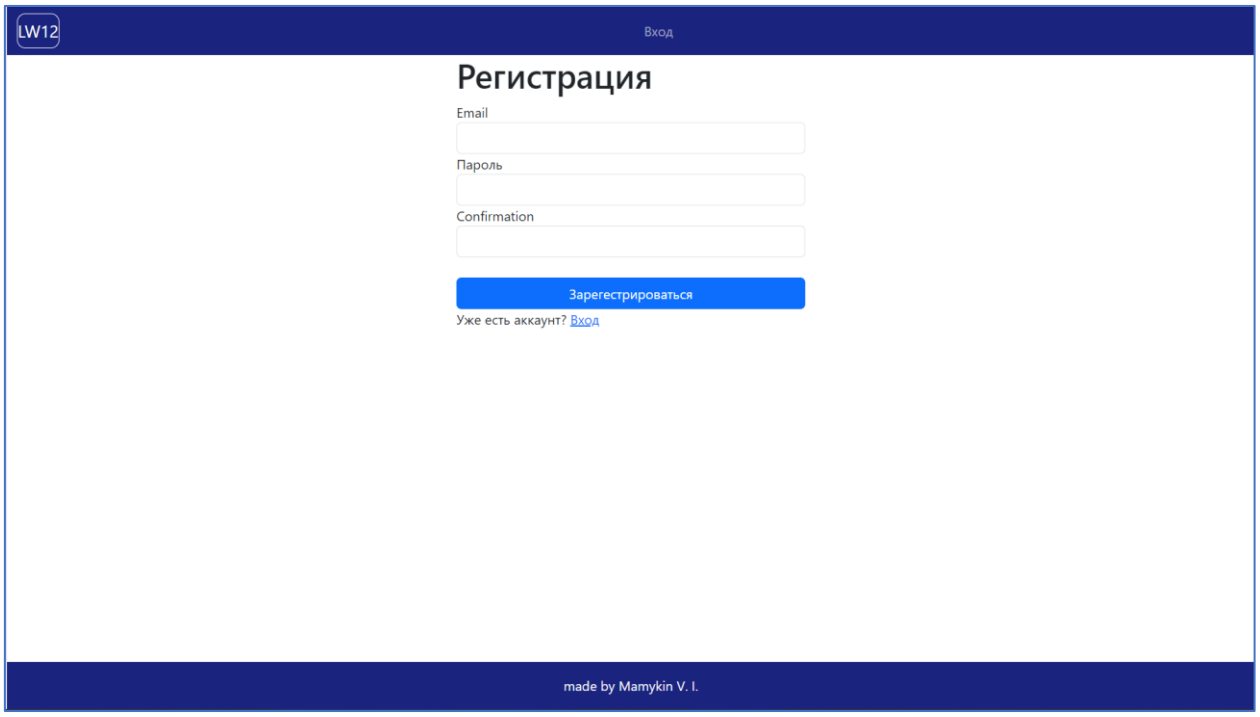

Рисунок 3 – форма регистрации

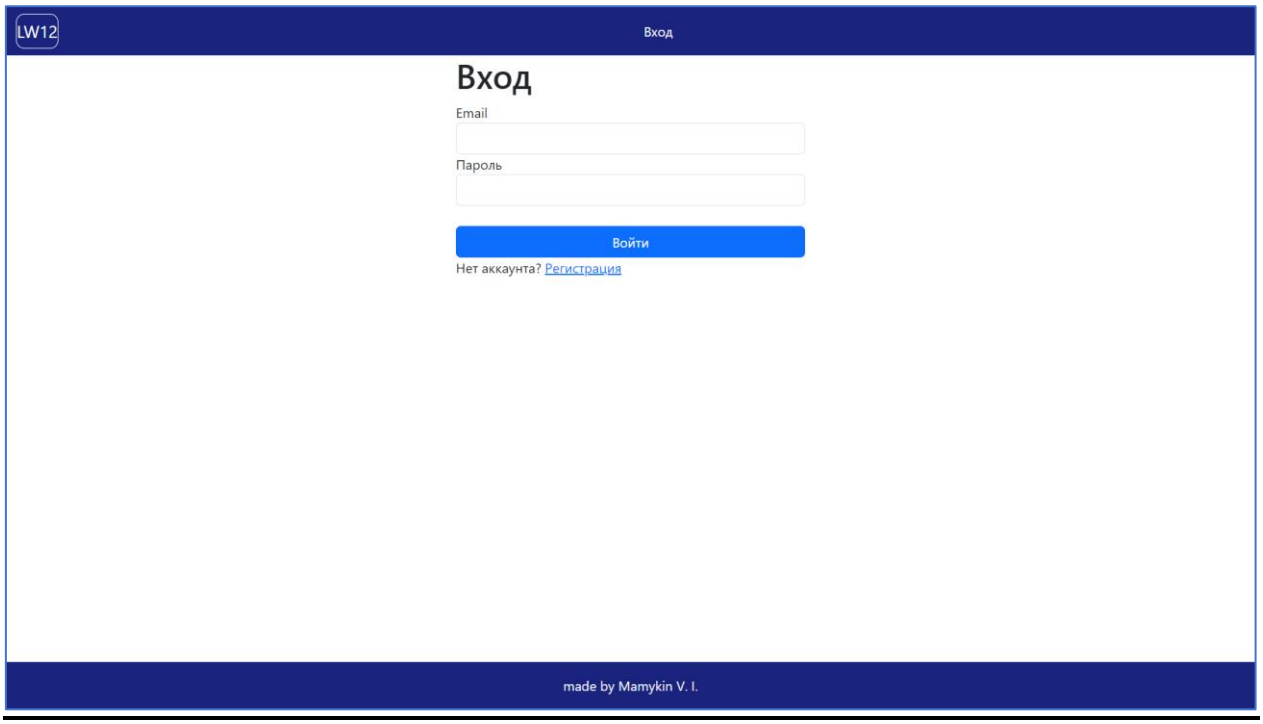

Рисунок 4 – форма входа

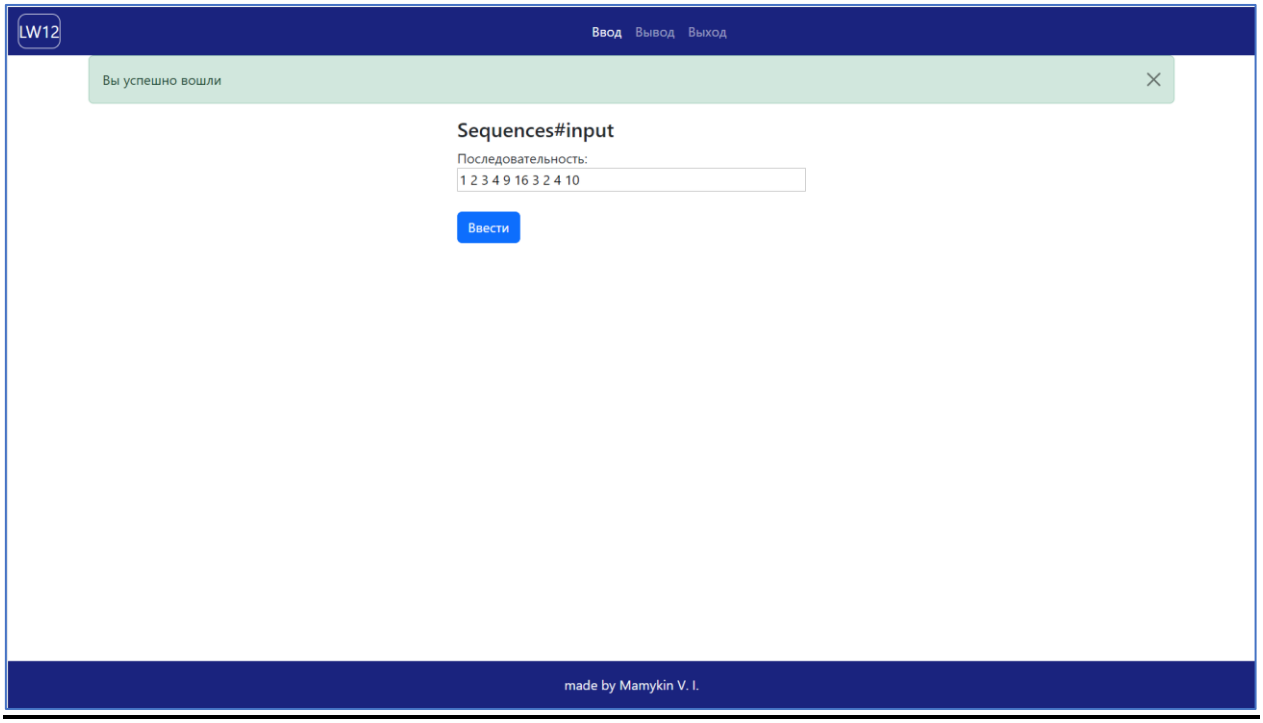

Рисунок 5 – успешный вход

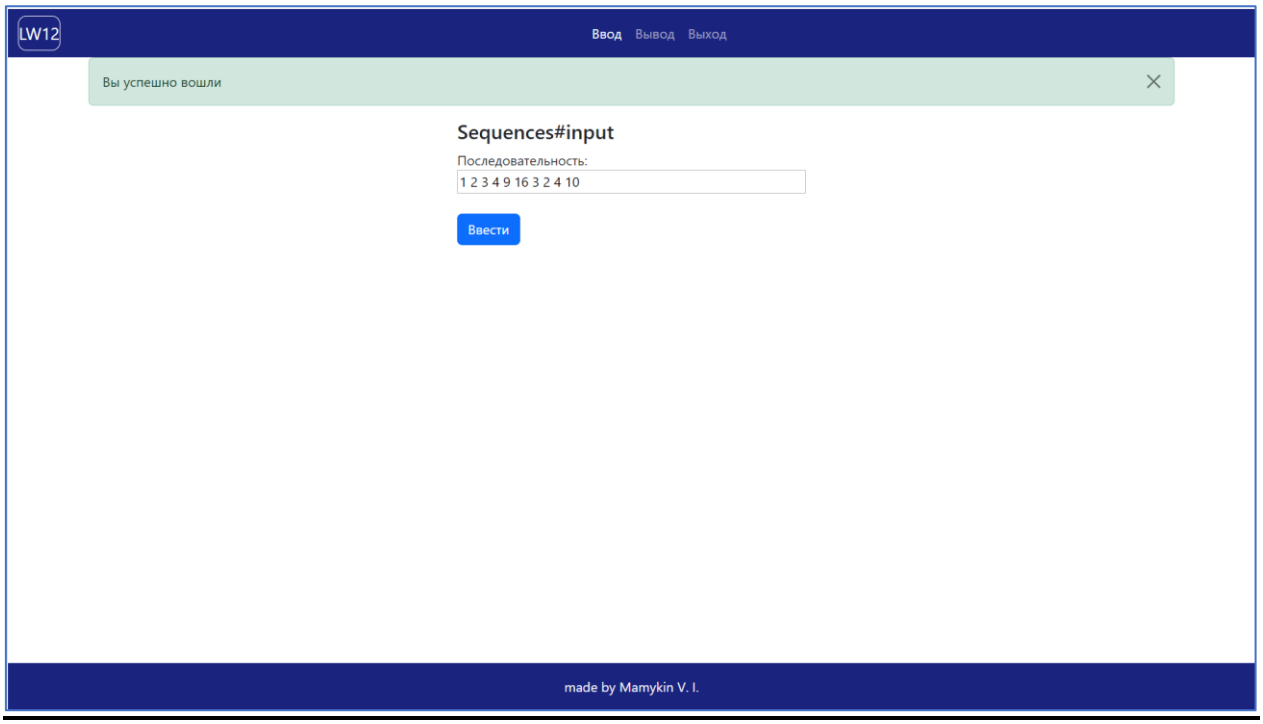

Рисунок 6 – ввод значения

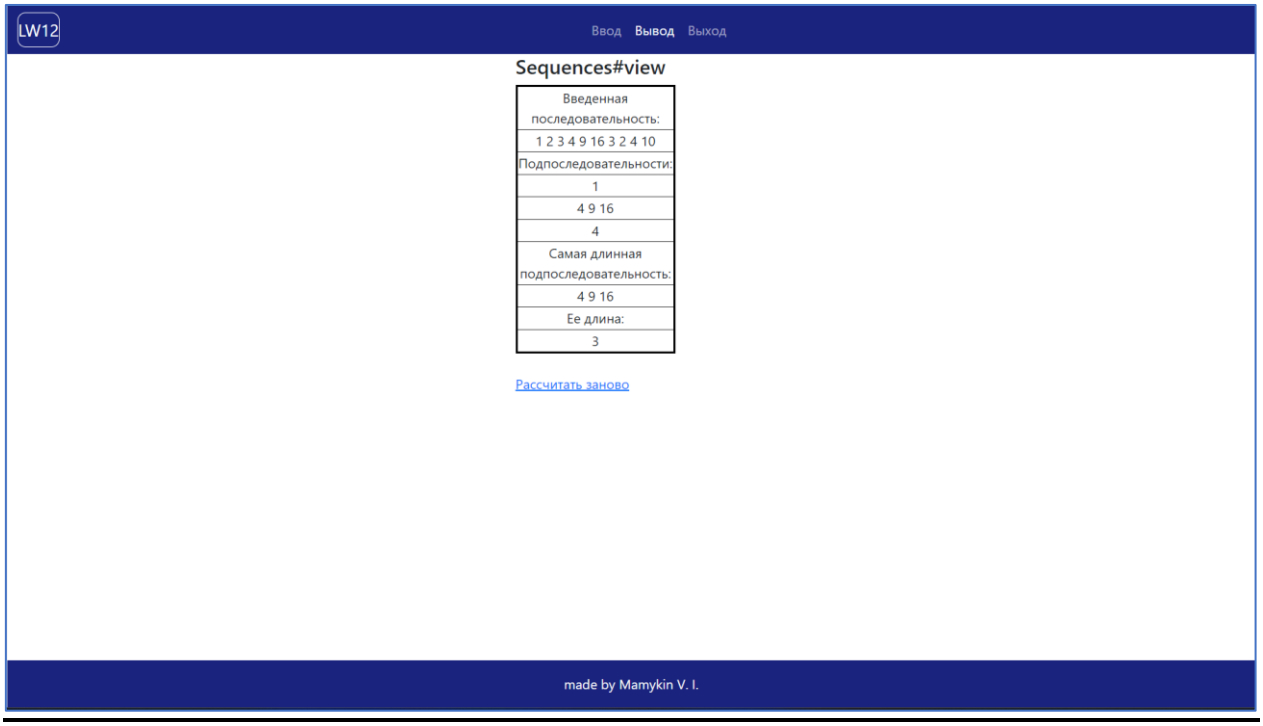

Рисунок 7 – вывод

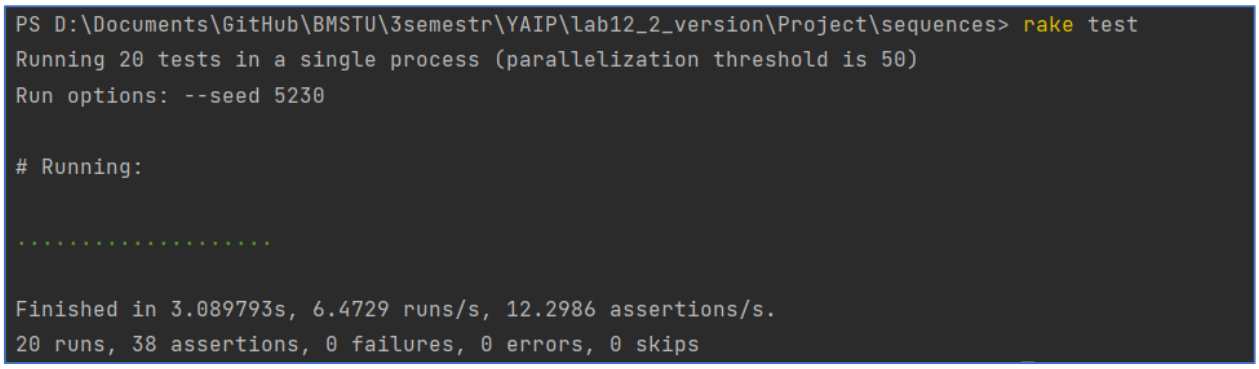

Рисунок 8 – тесты

Распечатка БД:

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL); id integer PRIMARY KEY PRIMARY KEY AUTOINCREMENT

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL); id integer NOT NULL NOT NULL

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL); input integer

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL); result json

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at"

datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL); created_at
    datetime NOT NULL NOT NULL

    results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL); updated_at
    datetime NOT NULL NOT NULL

    results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
        1 0

    results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
        10 1

    results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
        "{\"11\":13,\"17\":19}" 2

    results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
        2023-11-08 16:33:04.789709 3

    results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
        2023-11-08 16:33:04.789709 4

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
2 0

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
11 1

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
"{\"11\":13,\"17\":19}" 2

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
2023-11-08 16:33:07.187956 3

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
2023-11-08 16:33:07.187956 4

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
3 0

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
15 1

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
"{\"17\":19}" 2

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
2023-11-08 16:33:09.666411 3

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
2023-11-08 16:33:09.666411 4

Вывод: было создано веб-приложение, использующее аутентификацию. Изучены способы написания интеграционных тестов таких приложений.