



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 09.03.01 Информатика и вычислительная техника

ОТЧЕТ

по лабораторной работе № 12

Дисциплина: Языки интернет-программирования

Вариант: 15

Студент

ИУ6-33Б

(Группа)

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В. Д. Шульман

(И.О. Фамилия)

Москва, 2023

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 /.

Код программы

Файл “app/controllers/application_controller.rb”

```
# frozen_string_literal: true

class ApplicationController < ActionController::Base
  include SessionHelper

  before_action :require_login

  private
```

```

def require_login
  unless signed_in?
    flash[:danger] = 'Требуется логин'
    redirect_to session_login_url
  end
end
end
end

```

Файл “app/controllers/calcs_contoller.rb”

```

# frozen_string_literal: true

class CalcsController < ApplicationController
  before_action :set_calc, only: %i[show edit update destroy]

  # GET /calcs or /calcs.json
  def index
    @calcs = Calc.all
  end

  # GET /calcs/1 or /calcs/1.json
  def show; end

  # GET /calcs/new
  def new
    @calc = Calc.new
  end

  # GET /calcs/1/edit
  def edit; end

  # POST /calcs or /calcs.json
  def create
    array = params[:value].split
    array.length.times do |i|
      value = Float(array[i])
      value = value.to_i if (value % 1).zero?
      array[i] = value
    end

    @calc = Calc.save_sequence(array, current_user.id)
    @calc = Calc.find_by(input_sequence: array.join(' '))
    @calc = Calc.save_sequence(array) if @calc.nil?
    redirect_to calc_url(@calc)
  rescue
    redirect_to new_calc_path
  end

  # PATCH/PUT /calcs/1 or /calcs/1.json
  def update
    respond_to do |format|
      if @calc.update(calc_params)
        format.html { redirect_to calc_url(@calc), notice: 'Calc was successfully updated.' }
        format.json { render :show, status: :ok, location: @calc }
      else
        format.html { render :edit, status: :unprocessable_entity }
        format.json { render json: @calc.errors, status: :unprocessable_entity }
      end
    end
  end

  # DELETE /calcs/1 or /calcs/1.json
  def destroy
    @calc.destroy

    respond_to do |format|
      format.html { redirect_to calcs_url, notice: 'Calc was successfully destroyed.' }
      format.json { head :no_content }
    end
  end

  private

  # Use callbacks to share common setup or constraints between actions.
  def set_calc
    @calc = Calc.find(params[:id])
  end
end

```

```
# Only allow a list of trusted parameters through.
def calc_params
  params.require(:calc).permit(:input_sequence)
end
end
```

Файл “app/controllers/session_controller.rb”

```
# frozen_string_literal: true

class SessionController < ApplicationController
  skip_before_action :require_login, only: %i[login create]

  def login; end

  def create
    user = User.find_by(username: params[:login])

    if user&.authenticate(params[:password])
      sign_in user
      redirect_to root_path
    else
      flash.now[:danger] = 'Неверный логин или пароль'
      redirect_to session_login_path
    end
  end

  def logout
    sign_out
    redirect_to session_login_path
  end
end
```

Файл “app/controllers/users_controller.rb”

```
# frozen_string_literal: true

class UsersController < ApplicationController
  before_action :set_user, only: %i[show edit update destroy]
  skip_before_action :require_login, only: %i[new create]

  # 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 /users or /users.json
  def create
    @user = User.new(user_params)

    respond_to do |format|
      if @user.save
        sign_in @user
        format.html { redirect_to user_url(@user), notice: 'User was successfully created.' }
        format.json { render :show, status: :created, location: @user }
      else
        format.html { render :new, status: :unprocessable_entity }
        format.json { render json: @user.errors, status: :unprocessable_entity }
      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(:username, :password, :password_confirmation)
end
end

```

Файл “app/views/session/login.html.erb”

```

<%= bootstrap_form_with url: session_create_url, method: :post, local: true, class: "container vh-100 d-flex
flex-column justify-content-center" do |f| %>
  <%= f.text_field :login, required: true %><br>
  <%= f.password_field :password, required: true %><br>
  <br>
  <%= f.submit 'Войти' %>
  <%= link_to 'Зарегистрироваться', new_user_path %>
<% end %>

```

Файл “app/views/calcs/index.html.erb”

```

<p style="color: green"><%= notice %></p>

<h1>Запросы</h1>

<div id="calcs" class="container">
  <% @calcs.each do |calc| %>
    <div class="mt-4">
      <%= render calc %>
      <%= link_to "Show this calc", calc, class: "btn btn-secondary mt-2" %>
    </div>
  <% end %>
</div>

<%= link_to "New calc", new_calc_path, class: "btn btn-primary mt-4" %>

```

Файл “app/views/calcs/_calc.html.erb”

```

<div id="<%= dom_id calc %>">
  <div class="card">
    <div class="card-body">
      <p><strong>Input sequence:</strong> <%= calc.input_sequence %></p>
      <% sequences = calc.output_sequences.split('\n') %>
      <% (sequences.length - 1).times.each do |i| %>
        <p><strong>Sequence <%= i + 1 %></strong>: <%= sequences[i] %></p>
      <% end %>
      <p><strong>Longest Sequence</strong>: <%= sequences[sequences.length - 1] %></p>
    </div>
  </div>
</div>

```

Файл “app/views/calcs/_form.html.erb”

```
<main class="container vh-100 d-flex flex-column justify-content-center align-items-center">
  <div class="card">
    <div class="card-body">
      <%= bootstrap_form_with url: calcs_url, method: :post, local: true, remote: false do |f| %>
        <strong>Задание:</strong>
        <p>Написать программу, которая вводит цепочку целых чисел (количество чисел не менее 10) и определяет наиболее длинную
монотонно возрастающую их последовательность. Вывести на печать введенную цепочку, все найденные последовательности и наиболее
длинную из них. При программировании использовать функцию.</p>
        <%= f.text_field :value, required: true, label: 'Массив: ' %>
        <br>
        <%= f.submit "Вычислить", class: 'btn btn-primary' %>
      <% end %>
    </div>
  </div>
</main>
```

Файл “app/views/users/index.html.erb”

```
<p style="color: green"><%= notice %></p>

<h1>Пользователи</h1>

<div id="users" class="container">
  <% @users.each do |user| %>
    <div class="mt-4">
      <%= render user %>
      <%= link_to "Show this user", user, class: "btn btn-secondary mt-2" %>
    </div>
  <% end %>
</div>

<%= link_to "New user", new_user_path, class: "btn btn-primary mt-4" %>
```

Файл “app/views/users/_user.html.erb”

```
<div id="<%= dom_id user %>">
  <div class="card">
    <div class="card-body">
      <p><strong>Username:</strong> <%= user.username %></p>
    </div>
  </div>
</div>
```

Файл “app/views/users/_form.html.erb”

```
<%= bootstrap_form_with model: user, local: true, class: "container vh-100 d-flex flex-column justify-content-center"
do |form| %>
  <% if user.errors.any? %>
    <div id="error_explanation">
      <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>
  <% end %>

  <div class="field">
    <%= form.text_field :username %>
  </div>

  <div class="field">
    <%= form.password_field :password, required: true %>
  </div>

  <div class="field">
    <%= form.password_field :password_confirmation, required: true %>
  </div>
```

```
<div class="actions">
  <%= form.submit %>
</div>
<% end %>
```

Файл “app/models/calc.rb”

```
# frozen_string_literal: true

class Calc < ApplicationRecord
  belongs_to :user

  validates :input_sequence, presence: true, uniqueness: true
  validates :output_sequences, presence: true

  class << self
    def save_sequence(array, user_id)
      sequences = []
      longest_sequence = []

      start = 0
      max_len = 0
      curr_len = 1
      array.length.times do |i|
        next if i.zero?

        if array[i - 1].to_i ≤ array[i].to_i
          curr_len += 1
        else
          sequences << array[start, curr_len] if curr_len > 1

          if curr_len > max_len
            longest_sequence = sequences[-1]
            max_len = curr_len
          end

          start = i
          curr_len = 1
        end
      end

      sequences << array[start, curr_len] if curr_len > 1

      if curr_len > max_len
        longest_sequence = sequences[-1]
        max_len = curr_len
      end

      sequences << longest_sequence

      sequences.length.times do |i|
        sequences[i] = sequences[i].join(' ')
      end

      create input_sequence: array.join(' '), output_sequences: sequences.join('\n'), user_id: user_id
    end
  end
end
```

Файл “app/models/user.rb”

```
# frozen_string_literal: true

class User < ApplicationRecord
  has_many :calcs


  has_secure_password

  end
```


Работа программы

Username

Password



Password confirmation



Create User

Рисунок 1 – регистрация нового пользователя

Login

Password

Войти

[Зарегистрироваться](#)

Рисунок 2 – создание новой сессии

Пользователи

Username: user_1

Show this user

Username: user_2

Show this user

New user

Рисунок 3 – список зарегистрированных пользователей

Задание:

Написать программу, которая вводит цепочку целых чисел (количество чисел не менее 10) и определяет наиболее длинную монотонно возрастающую их последовательность. Вывести на печать введенную цепочку, все найденные последовательности и наиболее длинную из них. При программировании использовать функцию.

Массив:

6 8 0 5 9 26 8 6 2 9

Вычислить

Рисунок 4 – ввод данных для вычисления

Input sequence: 6 8 0 5 9 26 8 6 2 9

Sequence 1: 6 8

Sequence 2: 0 5 9 26

Sequence 3: 2 9

Longest Sequence: 0 5 9 26

Back to calcs

Рисунок 5 – результат вычисления

Запросы

Input sequence: 7 9 3 7 9 9 7 3 7 8

Sequence 1: 7 9

Sequence 2: 3 7 9 9

Sequence 3: 3 7 8

Longest Sequence: 3 7 9 9

Show this calc

Input sequence: 6 8 0 5 9 26 8 6 2 9

Sequence 1: 6 8

Sequence 2: 0 5 9 26

Sequence 3: 2 9

Longest Sequence: 0 5 9 26

Show this calc

New calc

Рисунок 6 – история запросов

Вывод

Изучены механизмы сессий и авторизаций; полученные знания отработаны при создании простого web-сайта, не позволяющего производить вычисления без регистрации пользователя и аутентификации при помощи логина/пароля.