|  |  |
| --- | --- |
| 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 **09.03.01 Информатика и вычислительная техника**

**Отчет**

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

**Дисциплина:** Языки интернет-программирования

**Вариант:** 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-33Б |  |  | Д. И. Мироненко |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | В. Д. Шульман |
|  |  |  | (Подпись, дата) | (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 |

**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  # 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**

A white paper with black lines

Description automatically generated

**Рисунок 1** – регистрация нового пользователя

**A screenshot of a computer

Description automatically generated**

**Рисунок 2** – создание новой сессии

**A screenshot of a calendar

Description automatically generated**

**Рисунок 3** – список зарегистрированных пользователей

**A screenshot of a computer

Description automatically generated**

**Рисунок 4** – ввод данных для вычисления

**A white rectangular object with a black border

Description automatically generated**

**Рисунок 5** – результат вычисления

**A screenshot of a computer

Description automatically generated**

**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.