*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)*** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Компьютерные Системы и сети (ИУ6)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 12**

**Название лабораторной работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дисциплина:** Языки интернет программирования

Студент гр. ИУ6-31б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И С Марчук**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

## *Задание* Вариант 1.16.

## ЛР 8. Ruby on Rails

Разработать веб-приложение, имеющее HTML-страницу с формой ввода данных и HTML-страницу для представления результатов. Результат

расчёта должен быть представлен в форме таблицы, оформленной с помощью элемента table или отдельными ячейками div и имеющей не менее

двух колонок. Если по условию задания результат может быть представлен только в виде одной строки таблицы, необходимо реализовать вывод

промежуточных результатов расчёта в качестве дополнительных строк.

В этом случае первой колонкой таблицы будет порядковый номер итерации.

Под вводом с клавиатуры в тексте заданий следует понимать ввод в

поле ввода данных формы на HTML-странице.

**Текст задания:**

Дано натуральное число m. Написать программу, определяющую такое

натуральное число n, что двоичная запись числа n получается из двоичной записи числа m изменением порядка цифр на обратный порядок их

следования. Например: 6 = 110 , а 3 = 011. Вывести на печать числа и

их двоичное представление.

**ЛР 12. Сессии. Выполнение авторизации. Интеграционные тесты.**

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 /.

Текст Программы

***routes.rb:***

Rails.application.routes.draw do

root 'start\_page#index'

get 'start\_page/index'

get 'result' => 'start\_page#result', :as => 'result'

get '/login' => 'sessions#registration'

post '/login' => 'sessions#create'

get '/logout' => 'sessions#destroy'

get '/signup' => 'users#new'

get '/users' => 'users#index'

post '/users' => 'users#create'

get '/users/:id' => 'users#show'

get '/users/:id/edit' => 'users#edit'

post '/users/:id/update' => 'users#update'

get '/users/:id/destroy' => 'users#destroy'

end

***application\_controller.rb:***

class ApplicationController < ActionController::Base

#protect\_from\_forgery with: :exception

def current\_user

@current\_user ||= User.find(session[:user\_id]) if session[:user\_id]

end

helper\_method :current\_user

def authorize

redirect\_to '/login' unless current\_user

end

end

***sessions\_controller.rb:***

class SessionsController < ApplicationController

def registration

end

def create

user = (User.where(:mail => params[:mail]))[0]

if user && (user[:password] == (params[:password]))

session[:user\_id] = user.id

redirect\_to '/'

else

redirect\_to '/login'

end

end

def destroy

session[:user\_id] = nil

redirect\_to '/login'

end

end

***start\_page\_controller.rb:***

class StartPageController < ApplicationController

before\_action :authorize

def index ()

end

def result

@m = params[:pm].to\_i

@bin\_m = ' '

@temp\_arr = Array.new()

@ans = 0

@bin\_ans = ' '

i = 0

temp\_m = @m

while temp\_m > 0 do

@ans = @ans \* 2 + (temp\_m % 2)

@bin\_ans = @bin\_ans + (temp\_m % 2).to\_s

@bin\_m = (temp\_m % 2).to\_s + @bin\_m

# новую строку в матрицу

@temp\_arr.push Array.new(3)

# ячейки строки

@temp\_arr[-1][0] = @ans

@temp\_arr[-1][1] = @bin\_ans

@temp\_arr[-1][2] = @bin\_m

temp\_m = temp\_m / 2

i = i + 1

end

end

end

***users\_controller.rb:***

class UsersController < ApplicationController

before\_action :set\_user, only: [:show, :edit, :update, :destroy]

# GET /users

# GET /users.json

def index

@users = User.all

end

# GET /users/1

# GET /users/1.json

def show

@user = (User.where(:id => params[:id]))[0]

end

# GET /users/new

def new

@user = User.new

end

# GET /users/1/edit

def edit

@user = (User.where(:id => params[:id]))[0]

end

# POST /users

# POST /users.json

def create

@user = User.new({"mail":params[:mail], "password":params[:password]})

respond\_to do |format|

if @user.save

format.html { redirect\_to '/', notice: 'User was successfully created.' }

format.json { render :show, status: :created, location: @user }

else

format.html { render :new }

format.json { render json: @user.errors, status: :unprocessable\_entity }

end

end

end

# PATCH/PUT /users/1

# PATCH/PUT /users/1.json

def update

respond\_to do |format|

if @user.update({mail:params[:mail], password:params[:password]})

format.html { redirect\_to "/users/#{@user[:id]}", notice: 'Пользователь был успешно изменен!.' }

format.json { render :show, status: :ok, location: "/users/#{@user[:id]}"}

else

format.html { render :edit }

format.json { render json: @user.errors, status: :unprocessable\_entity }

end

end

end

# DELETE /users/1

# DELETE /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

end

***sessions/registration.html.erb:***

<h2>Марчук Иван ИУ6-31б Вариант 16</h2>

<h4>Вход в систему(попробуйте логин:admin@yandex.ru пароль:admin)</h4>

<!-- форма входа -->

<%= form\_tag('/login', :method => "post") do %>

<%= label\_tag("Почта:") %>

<%= text\_field\_tag(:mail) %>

<br/>

<%= label\_tag("Пароль:") %>

<%= password\_field\_tag(:password) %>

<br/>

<%= submit\_tag("Вход") %>

<% end %>

<br/>

Нет учетной записи? <%= link\_to 'Регистрация', '/signup' %>

<br/>

<%= link\_to 'Список всех пользователей', '/users' %><% end %>

***start\_page/index.html.erb:***

<H3>Начальная страница</H3>

<H2>Марчук Иван ИУ6-31б Вариант 16</H2>

<!-- форма ввода отправляющая метод get на страничку result -->

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

<%= label\_tag("Введите число m:") %>

<%= number\_field\_tag(:pm) %>

<br/>

<%= submit\_tag("Посчитать") %>

<% end %>

<%= link\_to 'Выйти из системы', "/logout" %>

***start\_page/result.html.erb:***

<H3>Введенное значение m: <%= @m %></H3>

<H3>Решение:</H3>

<table border="1" bordercolor=black>

<tr><td> № </td><td> Ответ </td><td> дв. ответ </td><td> дв. m </td></tr>

<% @temp\_arr.size.times do |i| %>

<tr>

<td><%= i %></td>

<td><%= @temp\_arr[i][0] %></td>

<td><%= @temp\_arr[i][1] %></td>

<td><%= @temp\_arr[i][2] %></td>

</tr>

<% end %>

</table>

<p>m в двоичной форме: <%= @temp\_arr[-1][2] %></p>

<p>Ответ: <%= @temp\_arr[-1][0] %> В двоичной форме: <%= @temp\_arr[-1][1] %></p>

<%= link\_to 'Выйти из системы', "/logout" %>

***users /edit.html.erb:***

<h1>Редактирование пользователя</h1>

<%= form\_tag("/users/#{@user[:id]}/update", :method => "post") do %>

<%= label\_tag("Почта:") %>

<%= text\_field\_tag(:mail, @user[:mail]) %>

<br/>

<%= label\_tag("Пароль:") %>

<%= password\_field\_tag(:password, @user[:password]) %>

<br/>

<%= submit\_tag("Изменить") %>

<% end %>

<%= link\_to 'Перейти к сведениям о пользователе', "/users/#{@user[:id]}" %> |

<%= link\_to 'Назад', '/' %>

***users /index.html.erb:***

<p id="notice"><%= notice %></p>

<h3>Список пользователей:</h3>

<table border = "1">

<thead>

<tr>

<th>Почта</th>

<!--th>Password</th-->

<th> </th>

<th colspan="3"></th>

</tr>

</thead>

<tbody>

<% @users.size.times do |i| %>

<tr>

<td><%= @users[i].mail %></td>

<!--td>< %= user.password %></td-->

<td> </td>

<td>

<%= link\_to 'Показать', "/users/#{@users[i][:id]}" %>

<%= link\_to 'Редактировать', "/users/#{@users[i][:id]}/edit"%>

<%= link\_to 'Удалить', "/users/#{@users[i][:id]}/destroy", method: :delete, data: { confirm: 'Вы действительно хотите удалить пользователя?' } %>

</td>

</tr>

<% end %>

</tbody>

</table>

<br>

<%= link\_to 'Новый пользователь', '/signup' %>

***users /new.html.erb:***

<h3>Добавленипе пользователя</h3>

<%= form\_tag('/users', :method => "post") do %>

<%= label\_tag("Почта:") %>

<%= text\_field\_tag(:mail) %>

<br/>

<%= label\_tag("Пароль:") %>

<%= password\_field\_tag(:password) %>

<br/>

<%= submit\_tag("Добавить") %>

<% end %>

<%= link\_to 'Назад', '/' %>

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

<p id="notice"><%= notice %></p>

<p>

<strong>Mail:</strong>

<%= @user.mail %>

</p>

<p>

<strong>Password:</strong>

<%= @user.password %>

</p>

<%= link\_to 'Редактировать', "/users/#{@user[:id]}/edit" %> |

<%= link\_to 'Назад', '/' %>

**Файл миграции 20200107144113\_create\_users.rb:**

class CreateUsers < ActiveRecord::Migration[6.0]

def change

create\_table :users do |t|

t.string :mail

t.string :password

t.timestamps

end

end

end

**my\_result\_spec.rb**

require 'rails\_helper'

RSpec.describe MyResult, type: :model do

describe 'add record to database' do

it 'should add record to database and check it' do

field = MyResult.new(pm: 1500, result: '{"result":[[0], ["0"], ["0"]]}')

MyResult.find\_by(pm: 1500).destroy unless field.valid?

field.save

expect(MyResult.where(pm: 1500).exists?).to be\_truthy

end

end

context 'if program trying to add not valid options' do

it 'should say that not valid input is not valid' do

exs = MyResult.find\_by(pm: 144)

MyResult.new(pm: 144, result: '{"result":[[0], ["0"], ["0"]]}').save if exs.nil?

expect(MyResult.new(pm: 144).valid?).to be\_falsy

MyResult.find\_by(pm: 144).destroy

unless exs.nil?

MyResult.new(pm: 144, result: exs.result).save

end

end

end

end

**start\_page\_controller\_test.rb**

require 'test\_helper'

class StartPageControllerTest < ActionDispatch::IntegrationTest

# test "the truth" do

# assert true

# end

test 'test get root page' do

get root\_path

assert\_response :success

end

test 'test get view page' do

get '/result'

assert\_response :success

end

# удалить если показывать

#test "empty get" do

# get '/result'

# assert\_equal assigns[:bin\_m], ' '

# assert\_equal assigns[:ans], 0

# assert\_equal assigns[:bin\_ans], ' '

#end

#test "get with param m = 144" do

# get '/result', params: {mp: 144}

# assert\_equal " 10010000", assigns[:bin\_m]

# assert\_equal 9, assigns[:ans]

# assert\_equal " 00001001", assigns[:bin\_ans]

#end

End

**start\_page\_controller\_spec.rb**

require 'rails\_helper'

RSpec.describe StartPageController, type: :controller do

context 'app behaviour if user not register' do

it 'should redirect to signin page if user do not auth' do

get :index

expect(response).to redirect\_to('/login')

expect(response).to have\_http\_status(302)

end

end

end

**users\_routing\_spec.rb**

require "rails\_helper"

RSpec.describe UsersController, type: :routing do

describe "routing" do

it "routes to #index" do

expect(:get => "/users").to route\_to("users#index")

end

it "routes to #new" do

expect(:get => "/signup").to route\_to("users#new")

end

it "routes to #show" do

expect(:get => "/users/1").to route\_to("users#show", :id => "1")

end

it "routes to #edit" do

expect(:get => "/users/1/edit").to route\_to("users#edit", :id => "1")

end

it "routes to #create" do

expect(:post => "/users").to route\_to("users#create")

end

it "routes to #update via POST" do

expect(:post => "/users/1/update").to route\_to("users#update", :id => "1")

end

it "routes to #destroy" do

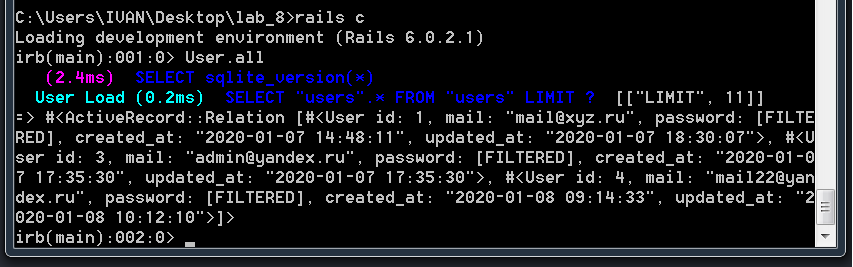
expect(:get => "/users/1/destroy").to route\_to("users#destroy", :id => "1")

end

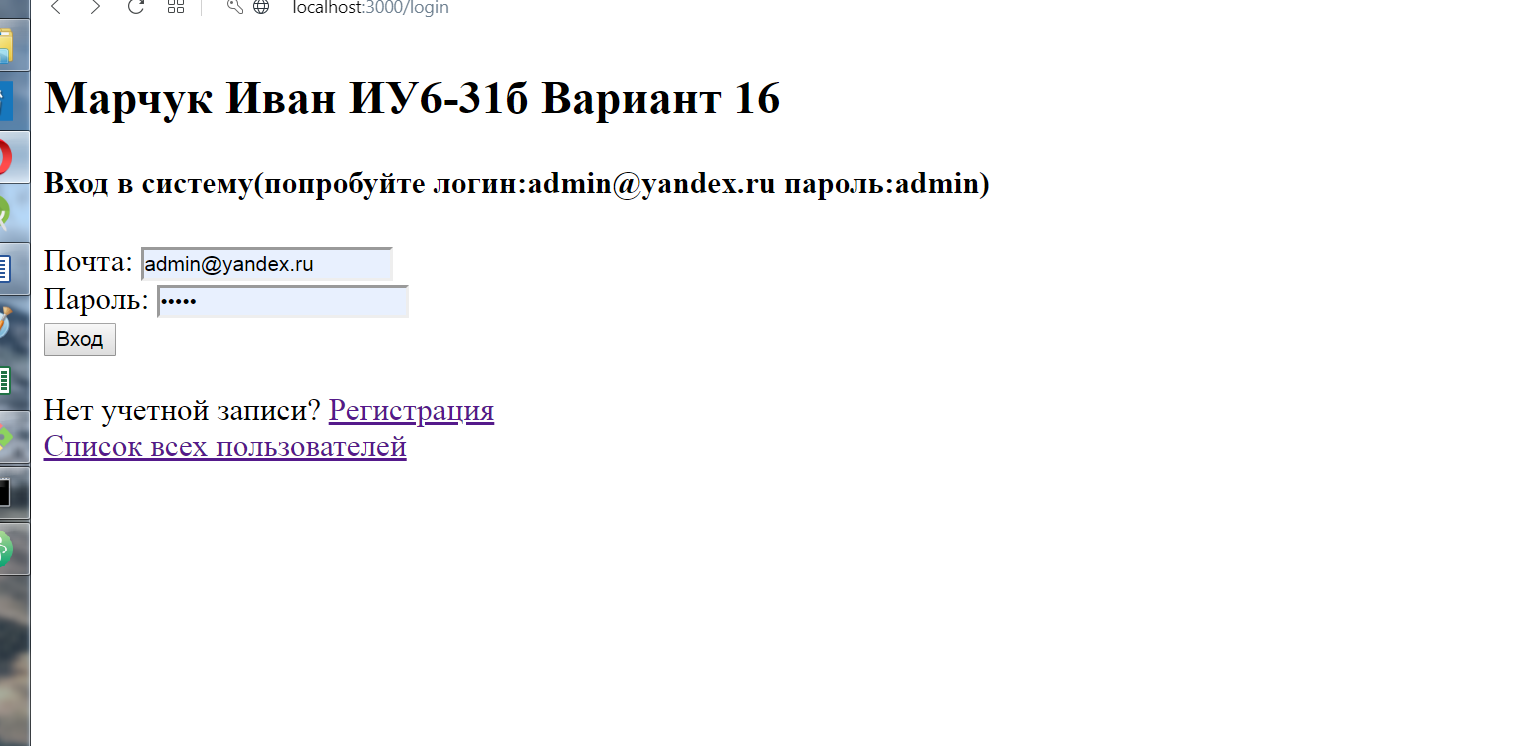
end

end

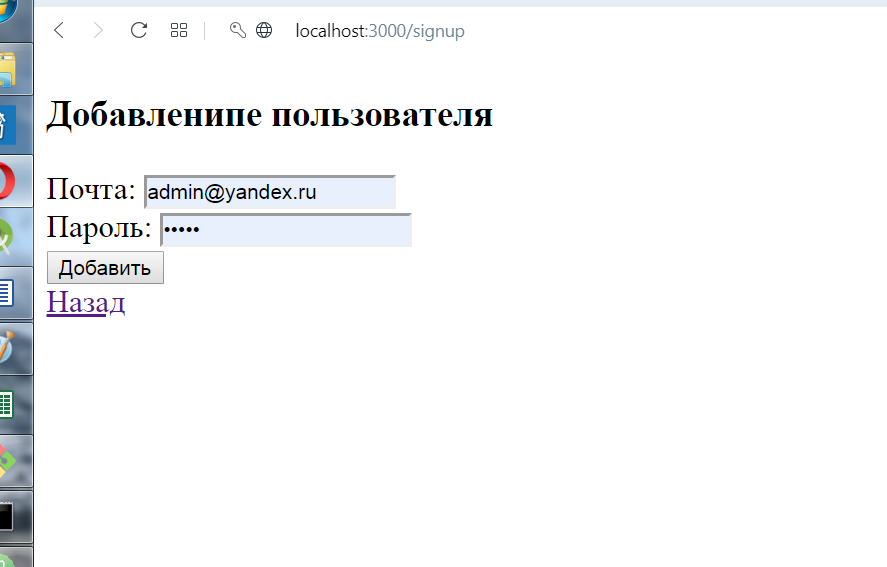
Скриншоты



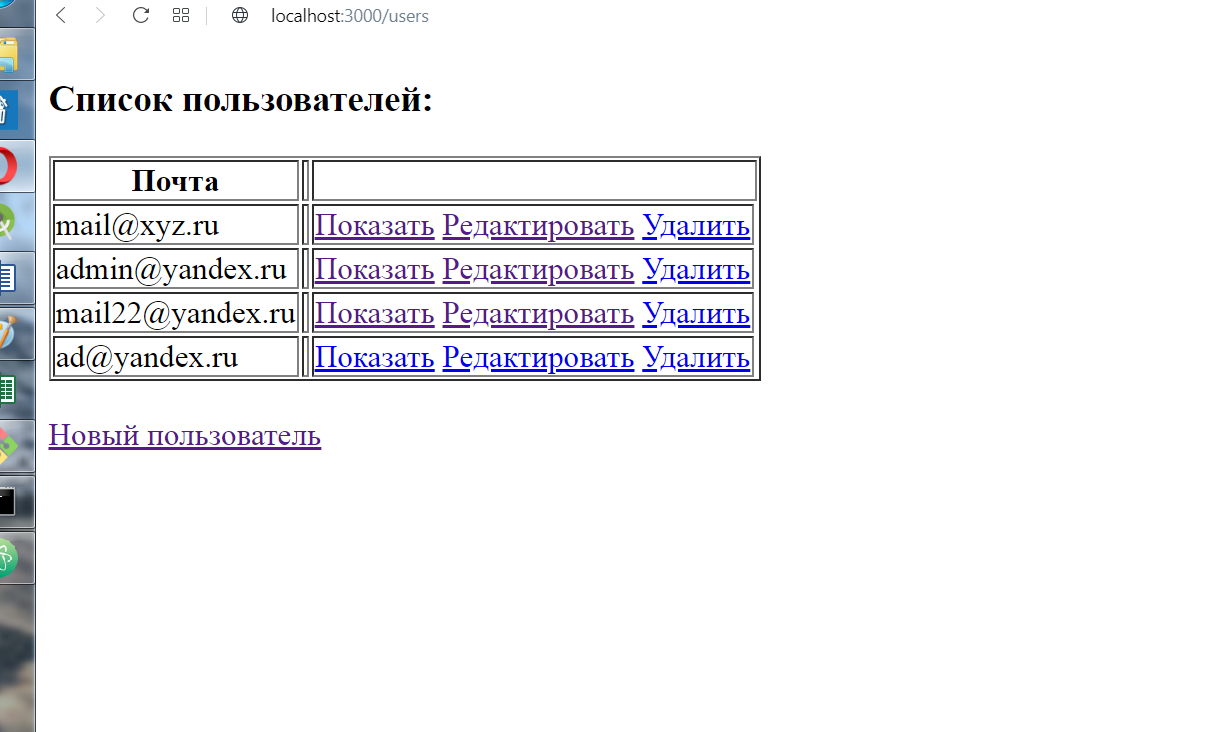
БД приложения в XML формате



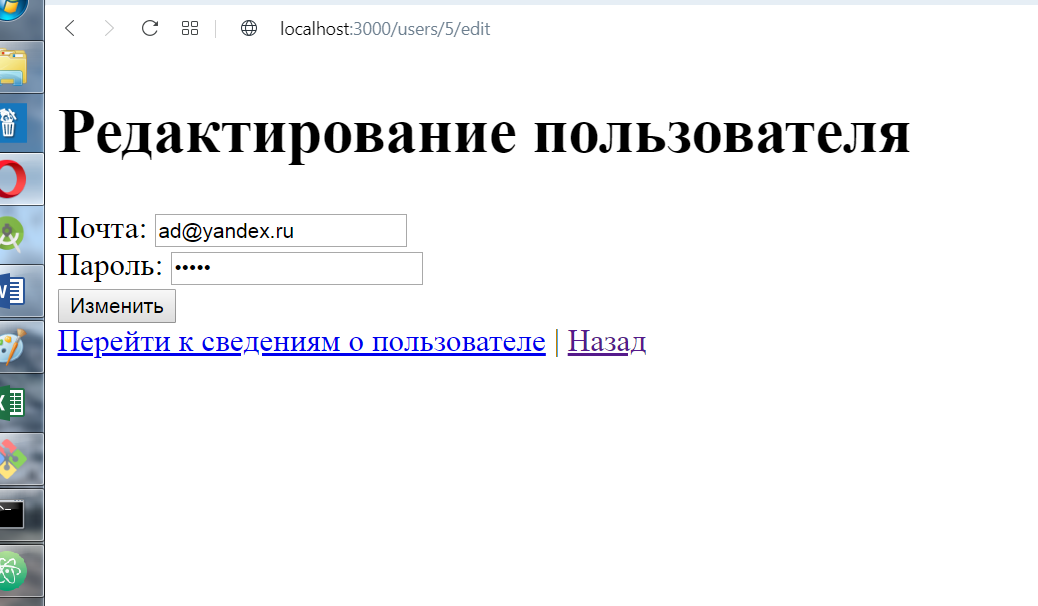
Страница авторизации



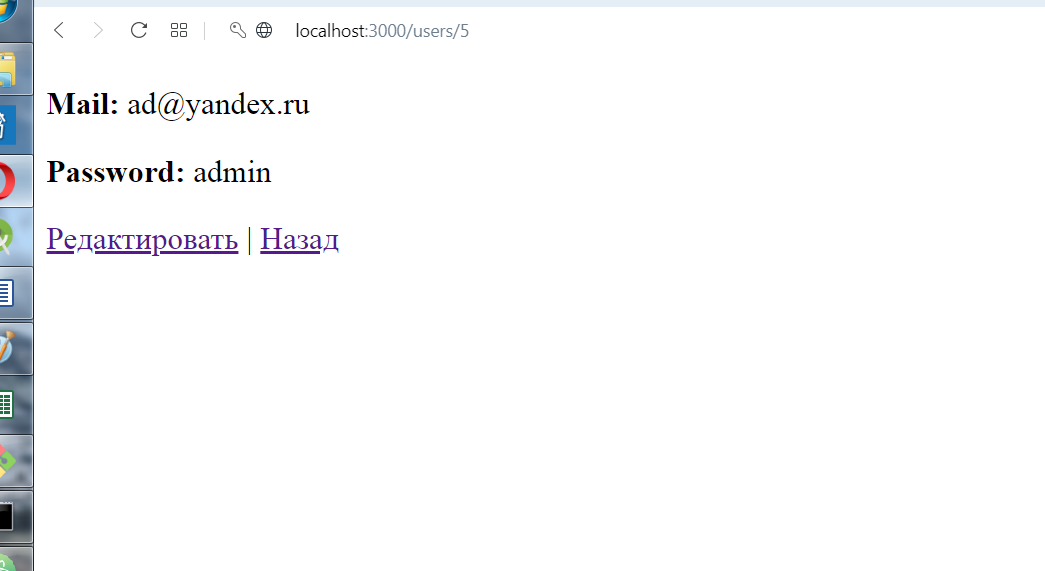
Страница регистрации нового пользователя



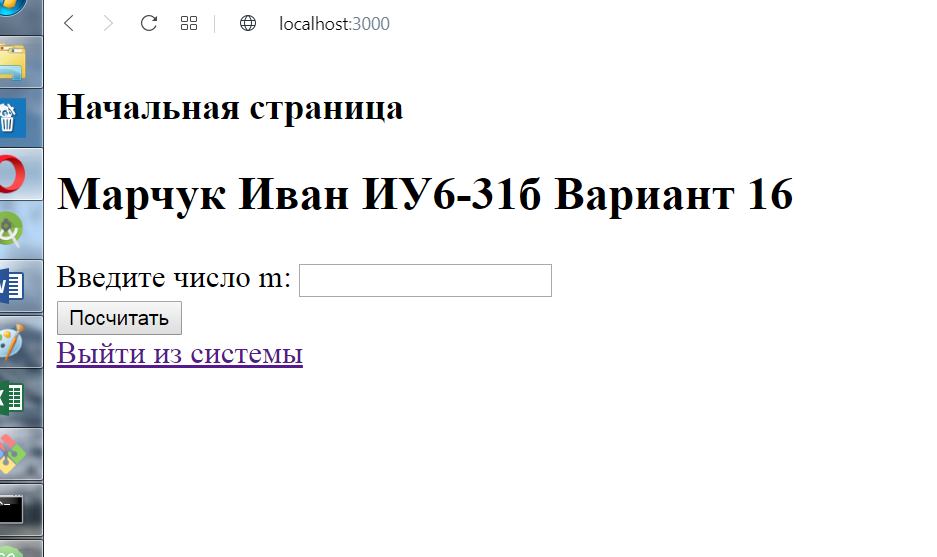
Страница списка пользователей



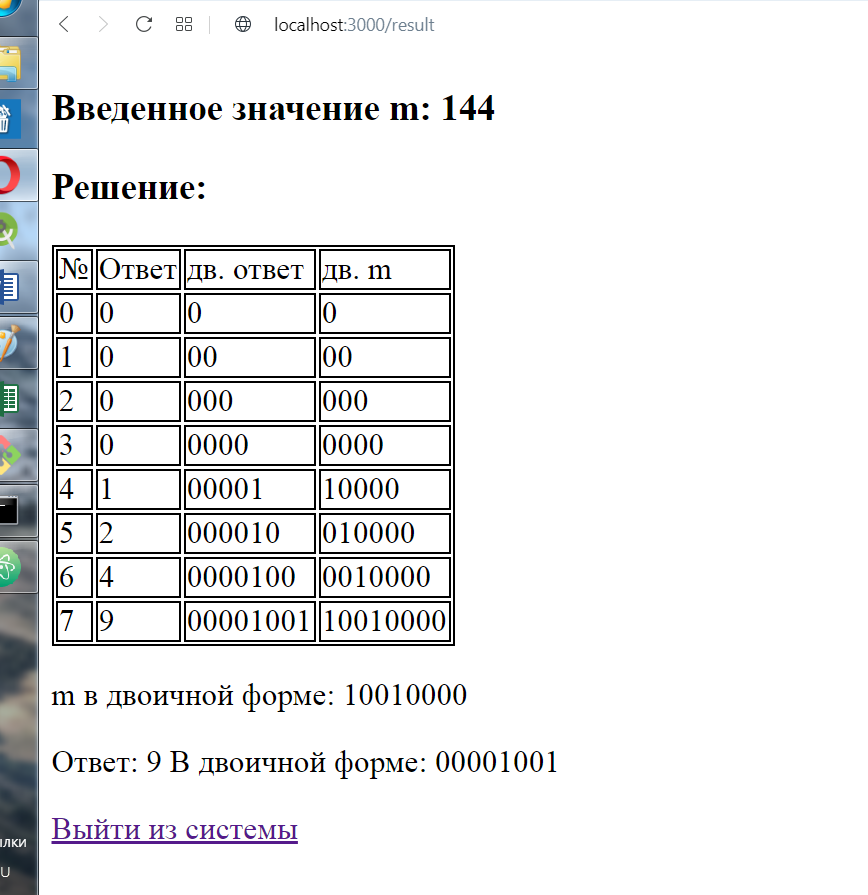
Страница редактирования пользователя



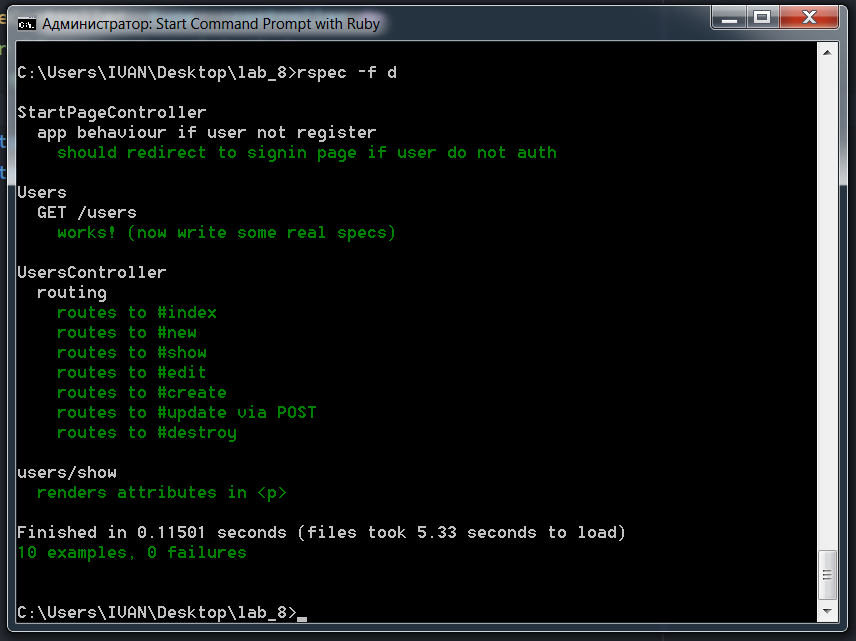
Страница просмотра пользователя



Страница ввода данных



Страница вывода результата



Результаты теста rspec

**Вывод:**

Для решения поставленной задачи созданное ранее web-приложение было модифицировано таким образом, чтобы доступ к страницам расчетов можно было получить только авторизовавшись под одним из пользователей хранящимся в БД.