



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(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 зачетной работе

Дисциплина: Языки Интернет-программирования

Название зачетной работы: Создание гог-приложения с использованием всех навыков, полученных в ходе курса

Студент гр. ИУ6-34Б

(Подпись, дата)

К.Ю. Каташинский

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2020

Условие задания:

Основная задача зачётной работы - продемонстрировать полученные знания в создании собственного веб-приложения. **Тему зачётной работы допустимо выбрать самостоятельно, но обязательно согласовать с преподавателем.**

Обязательные требования к программе:

- Для реализации использовать *Ruby on Rails*.
- Необходимо иметь контроллеры, обеспечивающие обработку запросов.
- Необходимо использовать модели для хранения данных в БД.
- Необходимо обеспечить аутентификацию пользователей.
- При реализации клиентской части необходимо применить код на языке *Javascript* и таблицы стилей *CSS*.
- Провести интернационализацию приложения и обеспечить вывод надписей на русском языке (см. пример в лекции 11).

Результат приложить в виде двух файлов:

1. архив, содержащий *RoR*-приложение;
2. *pdf*-отчет, в котором должны присутствовать фрагменты добавленного кода.

Приложение должно содержать **полный набор тестов**, позволяющих проверить все аспекты его функционирования.

Архив приложения можно подготовить командами:

```
git add . # есть риск собрать здесь мусор, не отмеченный в .gitignore!
```

```
git commit -m "initial"
```

```
git archive master | bzip2 > STUDENT_NAME_final_work.tar.bz2
```

Отчет должен содержать:

- ФИО, номер группы;
- чётко сформулированную задачу, выполняемую в работе;
- список файлов, которые были созданы или модифицированы в процессе выполнения работы, а также исходный код этих файлов;
- изображения форм интерфейса пользователя;
- исходный код тестов и описание цели тестирования каждым конкретным тестом (допустимо в виде комментариев в коде программ тестирования);
- отчет *Rubocop* по всем контроллерам.

Тема зачетной работы:

создание сервиса поиска работы, позволяющего клиентам откликаться на вакансию, а организациям – создавать ее.

Код файла application.js:

```
function to_org()
{
    document.getElementById('user').hidden = true;
    document.getElementById('organisation').hidden = false;
}
function to_user()
{
    document.getElementById('user').hidden = false;
    document.getElementById('organisation').hidden = true;
}
function show_job_creation()
{
    document.getElementById('job').hidden = false;
    document.getElementById('li').hidden = true;
}
function cancel_job_creation()
{
    document.getElementById('li').hidden = false;
    document.getElementById('job').hidden = true;
}
```

Код файла base.scss:

```
body {
    font-family: -apple-system, BlinkMacSystemFont, sans-serif;
    text-align: center;
}
.inf-header {
    font-size: 15pt;
    text-decoration: underline;
}
header {
    font-size: 15pt;
    color: red;
    text-align: left;
}
#user-information {
    float: right;
}
a {
    text-decoration: none;
    color: blue;
    cursor: pointer;
}
a:hover {
    color: green;
}
ul {
    width: 500px;
    list-style: none;
    margin: auto;
    padding: 0;
    font-size: 15pt;
}
li a {
    display: block;
    padding: 15px;
    text-decoration: none;
    color: black;
```

Продолжение кода файла base.scss:

```
border: 1px solid #ccc;
background-color: #f0f0f0;
}
li a:hover {
    color: white;
    background-color: green;
}
table {
    width: 45%;
    margin: auto;
    padding-left: 170px;
}
.enter {
    width: 200px;
    height: 15pt;
}
#website-name {
    color:red;
    font-style: italic;
    display:inline-block;
    margin:auto;
}
.account-name {
    color:black;
    text-decoration: underline;
}
```

Код файла organization_interaction.scss:

```
ol {
    text-align: left;
    font-size: 15pt;
}
#table_add_worker {
    margin: 0;
    padding-left: 0px;
    text-align: left;
}
.information_text {
    font-size: 15pt;
    text-align: left;
    font-style: italic;
}
#table_add_worker tr {
    text-align: center;
    font-size: 15pt;
}
td .exit:hover {
    color: red;
}
```

Код файла user_interaction.scss:

```
#user-information a {
    border: 1px solid #333;
    display: inline-block;
    padding: 2px 10px;
    color: black;
}
```

Продолжение кода файла user_interaction.scss:

```
#user-information a:hover {
  color: white;
  background-color: rgb(81, 206, 255);
}
.success {
  border: 1px solid #333;
  display: inline-block;
  padding: 5px 40px;
  color: white;
  background-color: green;
}
.success:hover {
  background-color: yellowgreen
}
.change {
  font-size: small;
  font-style: italic;
  cursor: pointer;
}
```

Код файла application_controller.rb:

```
class ApplicationController < ActionController::Base
  protect_from_forgery with: :exception
  before_action :check_auth, :get_today_date
  private def check_auth
    if @account = Account.find_by_session_id(session[:session_id])
      user_id = @account.user_id
      organisation_id = @account.organisation_id
      @user = User.find_by_id(user_id)
      @organisation = Organisation.find_by_id(organisation_id)
    end
  end
  private def get_today_date
    @today_date = Date.strptime(Time.now.strftime("%m/%d/%Y"), "%m/%d/%Y")
  end
end
```

Код файла base_controller.rb:

```
require 'digest/sha1'
class BaseController < ApplicationController
  def index
    @organisations = Organisation.all
    @jobs = Job.all.select{|job| ((@today_date -
Date.strptime(job.created_at.strftime('%m.%d.%Y'), '%m.%d.%Y')).to_i * 24) < 720 }
  end
end
```

Код файла organization_interaction_controller.rb:

```
class OrganisationInteractionController < ApplicationController
  def organisation
    id = params[:id].to_i
    @current_organisation = Organisation.find_by_id(id)
    unless @current_organisation
      redirect_to "/"
      return
    end
  end
end
```

Продолжение кода файла organization_interaction_controller.rb:

```

    @jobs_count = @current_organisation.jobs.all.select{|job| ((@today_date -
Date.strptime(job.created_at.strftime('%m.%d.%Y'), '%m.%d.%Y')).to_i * 24) < 720
}.length
    @has_access = false
    if @organisation
      if @organisation.id == @current_organisation.id
        @has_access = true
      end
    end
  end
end
def job
  id = params[:id].to_i
  @job = Job.find_by_id(id)
  unless @job
    redirect_to "/"
    return
  end
  @created_at = @job.created_at.strftime('%m.%d.%Y')
  if ((@today_date - Date.strptime(@created_at, '%m.%d.%Y')).to_i * 24) >= 720
    redirect_to "/"
    return
  end
  organisation_id = @job.organisation_id
  @current_organisation = Organisation.find_by_id(organisation_id)
  @need_to_do = @job.need_to_do.split(";")
  @has_access = false
  @has_applied = false
  if @user
    @has_applied = true if Middle.find_by_job_id_and_user_id(id, @user.id)
  elsif @organisation
    if @organisation.id == @job.organisation_id
      middles = @job.middles
      @users = middles.map {|element| User.find_by_id(element.user_id) }
      @has_access = true
    end
  end
end
def create_job
  unless @organisation
    redirect_to "/"
    return
  end
  name = params[:name]
  need_to_do = params[:need_to_do]
  description = params[:description]
  job = @organisation.jobs.create(name: name, need_to_do: need_to_do,
description: description)
  jobs_count = @organisation.jobs.all.select{|job| ((@today_date -
Date.strptime(job.created_at.strftime('%m.%d.%Y'), '%m.%d.%Y')).to_i * 24) < 720
}.length
  new_description = description.split.select.with_index {|element, index| index
< 8 }.join(" ")
  respond_to do |format|
    format.json do
      render json:
        { id: job.id, description: new_description, jobs_count: jobs_count,
is_valid: job.valid? }
    end
  end
end
```

Продолжение кода файла organization_interaction_controller.rb:

```
        end
      end
    end
  def add_worker
    if (!@account) || (@organisation)
      redirect_to "/"
      return
    end
    job_id = params[:job_id].to_i
    summary = params[:summary]
    @user.middles.create(summary: summary, job_id: job_id)
    redirect_to "/job/#{job_id}"
  end
end
```

Код файла user_interaction_controller.rb:

```
class UserInteractionController < ApplicationController
  def login
    if @account
      redirect_to "/"
    end
    @error = (params[:error] == nil) ? '' : params[:error]
  end
  def logout
    if @account
      @account.destroy
    end
    redirect_to "/"
  end
  def auth
    if @account
      redirect_to "/"
    end
    email = params[:email]
    password = Digest::SHA1.hexdigest params[:password]
    error = ""
    if (user = User.find_by_email(email)) || (organisation =
Organisation.find_by_email(email))
      if user
        unless user.password == password
          error = "Введен неверный пароль"
        else
          user.accounts.create(session_id: session[:session_id])
        end
      else
        unless organisation.password == password
          error = "Введен неверный пароль"
        else
          organisation.accounts.create(session_id: session[:session_id])
        end
      end
    else
      error = 'Пользователь/компания не найдены'
    end
    if error == ""
      redirect_to "/"
      return
    else
      redirect_to :action => "login", :error => error
    end
  end
end
```

Продолжение кода файла user_interaction_controller.rb:

```
        return
      end
    end
  end
  def register
    if @account
      redirect_to "/"
    end
    @error1 = (params[:error1] == nil) ? '' : params[:error1]
    @error2 = (params[:error2] == nil) ? '' : params[:error2]
  end
  def create_new_user
    if @account
      redirect_to "/"
      return
    end
    name = params[:name]
    email = params[:email]
    surname = params[:surname]
    phone = params[:phone]
    password = params[:password]
    another_password = params[:another_password]
    if password != another_password
      redirect_to :action => "register", :error1 => 'Введенные пароли не
совпадают'
      return
    end
    error=''
    if user = User.find_by_email(email)
      error = 'Пользователь с такой почтой уже существует'
    end
    if user = User.find_by_phone(phone)
      error = 'Пользователь с таким номером телефона уже существует'
    end
    unless error == ''
      redirect_to :action => "register", :error1 => error
      return
    end
    password = Digest::SHA1.hexdigest password
    @user = User.new(name: name, surname: surname, email: email, phone: phone,
password: password)
    unless @user.valid?
      redirect_to :action => "register", :error1 => 'Проверьте правильность
введенных данных'
      return
    end
    @user.save
    @user.accounts.create(session_id: session[:session_id])
    redirect_to "/"
  end
  def create_new_organisation
    name = params[:name]
    description = params[:description]
    email = params[:email]
    website_url = params[:website_url]
    phone = params[:phone]
    password = params[:password]
    another_password = params[:another_password]
    if password != another_password
```


Продолжение кода файла user_interaction_controller.rb:

```
        redirect_to :action => "register", :error2 => 'Введенные пароли не
совпадают'
        return
      end
      error = ''
      if organisation = Organisation.find_by_name(name)
        error = 'Организация с таким именем уже существует'
      end
      if organisation = Organisation.find_by_email(email)
        error = 'Организация с такой почтой уже существует'
      end
      if organisation = Organisation.find_by_website_url(website_url)
        error = 'Организация с таким сайтом уже существует'
      end
      if organisation = Organisation.find_by_phone(phone)
        error = 'Организация с таким номером телефона уже существует'
      end
      unless error == ''
        redirect_to :action => "register", :error2 => error
        return
      end
      password = Digest::SHA1.hexdigest password
      @organisation = Organisation.create(name: name, description: description,
website_url: website_url, password: password, phone: phone, email: email)
      unless @organisation.valid?
        redirect_to :action => "register", :error2 => 'Проверьте правильность
введенных данных',
        return
      end
      @organisation.save
      @organisation.accounts.create(session_id: session[:session_id])
      redirect_to "/"
    end
  end
end
```

Код файла account.rb:

```
class Account < ApplicationRecord
  # belongs_to :user
  # belongs_to :organisation
  validates :session_id, presence: true, uniqueness: true
end
```

Код файла job.rb:

```
class Job < ApplicationRecord
  has_many :middles
  belongs_to :organisation
  validates :name, presence: true, length: { maximum: 50 }, uniqueness: true
  validates :need_to_do, presence: true, length: { maximum: 100 }, uniqueness: true
  validates :description, presence: true, uniqueness: true
end
```

Код файла middle.rb:

```
class Middle < ApplicationRecord
  belongs_to :user
  belongs_to :job
  validates :summary, presence: true, uniqueness: true
end
```

Код файла organisation.rb:

```
class Organisation < ApplicationRecord
  has_many :accounts
  has_many :jobs
  before_save { self.email = email.downcase }
  validates :name, presence: true, length: { maximum: 50 }, uniqueness: true
  validates :description, presence: true, uniqueness: true
  validates :website_url, presence: true, length: { maximum: 50 }, uniqueness: true
  validates :password, presence: true, length: { maximum: 50 }
  VALID_PHONE_REGEX = /\A[+-]?\d+(\.\d+)?\Z/
  validates :phone, presence: true, length: { maximum: 11 }, format: { with:
VALID_PHONE_REGEX }, uniqueness: true
  VALID_EMAIL_REGEX = /\A[\w+\-\.]+\@[a-z\d\-\.]+\.[a-z]+\z/i
  validates :email, presence: true, length: { maximum: 100 }, format: { with:
VALID_EMAIL_REGEX }, uniqueness: true
end
```

Код файла user.rb:

```
class User < ApplicationRecord
  has_many :accounts
  has_many :middles
  before_save { self.email = email.downcase }
  validates :name, presence: true, length: { maximum: 50 }, uniqueness: true
  validates :surname, presence: true, length: { maximum: 50 }, uniqueness: true
  VALID_PHONE_REGEX = /\A[+-]?\d+(\.\d+)?\Z/
  validates :phone, presence: true, length: { maximum: 11 }, format: { with:
VALID_PHONE_REGEX }, uniqueness: true
  validates :password, presence: true, length: { maximum: 50 }
  VALID_EMAIL_REGEX = /\A[\w+\-\.]+\@[a-z\d\-\.]+\.[a-z]+\z/i
  validates :email, presence: true, length: { maximum: 100 }, format: { with:
VALID_EMAIL_REGEX }, uniqueness: true
end
```

Код файла index.html.erb:

```
<h1>Добро пожаловать на сайт поиска работы!</h1>
<p style="font-size: 15pt;"><span style="font-style: italic;">find-job.ru </span>-
новый сайт поиска работы, где каждый может найти работу по душе</p>
<p style="font-size: 15pt;">У нас уже есть организаций - <%= @organisations.length %>
и доступных вакансий - <%= @jobs.length %>.</p>
<br><br>
<h2>Список организаций</h2>
<p style="font-size: 15pt;">Выберите конкретную организацию для просмотра доступных в
ней вакансий.</p>
<ul class="menu">
  <% @organisations.each do |organisation| %>
    <li><a href="/organisation/<%= organisation.id %>"><%= organisation.name
%></a></li>&nbsp;
  <% end %>
</ul>
```

Код файла application.html.erb:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Зачетная работа</title>
    <%= csrf_meta_tags %>
    <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track':
'reload' %>
    <%= javascript_include_tag 'application', 'data-turbolinks-track': 'reload' %>
  </head>
  <body>
    <header>
      <!--<a href="/"></a>-->
      <span id="user-information">
        <% if @account %>
          <% if @user %>
            <span class="account-name">Пользователь <%= @user.name %></span>
          <% elsif @organisation %>
            <span class="account-name">Организация <%= @organisation.name %></span>
          <% end %>
          &nbsp;
          <a href="/logout">Выйти</a>
        <% else %>
          <a href="/login">Войти</a>
        <% end %>
      </span>
      <a href="/"><h2 id="website-name">find-job.ru</h2></a>&nbsp;&nbsp;&nbsp;
      <% if @organisation %>
        <a href="/organisation/<%= @organisation.id %>">Ваша организация</a>
      <% else %>
        <!--<a href="/list">Список организаций</a>-->
      <% end %>
      <hr>
    </header>
    <%= yield %>
  </body>
</html>
```

Код файла job.html.erb:

```
<h1><%= @job.name %></h1>
<div style="text-align: left;">
  <p style="font-size: 15pt;">Вернуться к организации <a href="/organisation/<%=
@current_organisation.id %>"><%= @current_organisation.name %></a></p>
  <p style="font-size: 15pt;">Дата создания вакансии - <%= @created_at %></p>
  <p class="inf-header">Описание вакансии</p>
  <p style="font-size: 15pt;"><%= @job.description %></p>
  <p class="inf-header">Обязательные требования</a></p>
</div>
<ol id="need_to_do">
  <% @need_to_do.each do |item|>
    <li><%= item %></li>
  <% end %>
</ol>
<% if @account %>
  <% if @user %>
    <% unless @has_applied %>
      <%= form_tag("/add_worker", :method => "post") do %>
        <table id="table_add_worker">
```

Продолжение кода файла job.html.erb:

```
<caption style="text-align: left;"><h2>Расскажите нам о
себе</h2></caption>
<tr>
  <td colspan="2"><%= text_area_tag :summary, nil, style:
"width:500px; height:150px; margin-right: 500px;" %></td>
</tr>
<tr>
  <td colspan="2"><%= hidden_field_tag :job_id, @job.id
%></td>
</tr>
<tr>
  <td><%= submit_tag "Отправить", class: "success", style:
"margin-right: 850px;" %></td>
  <td><span style="color: red;"><%= @error %></span></td>
</tr>
</table>
<% end %>
<% else %>
  <p class="information_text">Вы уже подали заявку на эту вакансию.</p>
<% end %>
<% elsif @has_access %>
  <div style="text-align: left;">
    <p class="inf-header">Список откликнувшихся пользователей</a></p>
  </div>
  <% if @users.length > 0 %>
    <p style="text-align: left; font-size: 15pt;">Общее количество: <%=
@users.length %></p>
    <table id="table_add_worker" border="1" style="border-collapse:
collapse;">
      <tr>
        <td>Фамилия</td>
        <td>Имя</td>
        <td>Телефон</td>
        <td>Резюме</td>
      </tr>
      <% @users.each do |user| %>
        <tr>
          <td><%= user.surname %></td>
          <td><%= user.name %></td>
          <td><%= user.phone %></td>
          <td><a onclick="document.getElementById('summary_<%= user.id
%>').hidden = false;">Подробнее</a></td>
        </tr>
        <tr id="summary_<%= user.id %>" hidden>
          <td colspan="3"><%=
user.middles.find_by_job_id(@job.id).summary %></td>
          <td><a class="exit"
onclick="document.getElementById('summary_<%= user.id
%>').hidden =
true;">Закрыть</a></td>
        </tr>
      <% end %>
    </table>
  <% else %>
    <p class="information_text">Пока никто не откликнулся на данную
вакансию.</p>
  <% end %>
<% end %>
<% elsif %>
```

Продолжение кода файла job.html.erb:

```
<p style="font-size: 15pt; text-align: left;"><a href="/login">Войдите</a>, чтобы  
откликнуться на данную вакансию.</p>  
<% end %>
```

Код файла organisation.html.erb:

```
<script>  
  handleAjaxSuccess = function(event) {  
    [data, status, xhr] = event.detail;  
    show_job(data);  
  }  
  document.addEventListener('DOMContentLoaded', function() {  
    document.querySelector("#add").addEventListener(  
      'ajax:success', handleAjaxSuccess)  
  })  
  function show_job(data)  
  {  
    if (!data.is_valid)  
    {  
      document.getElementById("error").innerText = "Проверьте правильность  
введенных данных";  
      return;  
    }  
    document.getElementById("error").innerText = "";  
    let parent = document.getElementById("inside");  
    let last_li = document.getElementById("li");  
    let name = document.getElementById("name").value;  
    let a = document.createElement("a");  
    let li_description = document.createElement("li");  
    let li_last = document.createElement("li");  
    a.href = "/job/" + data.id;  
    a.innerText = name;  
    li_description.innerText = data.description + "...";  
    li_description.style.cssText = "font-size: 14px;";  
    li_last.innerText = '\u00a0';  
    let li = document.createElement("li");  
    li.appendChild(a);  
    parent.insertBefore(li, last_li);  
    parent.insertBefore(li_description, last_li);  
    parent.insertBefore(li_last, last_li);  
    document.getElementById("name").value = "";  
    document.getElementById("need_to_do").value = "";  
    document.getElementById("description").value = "";  
    document.getElementById("jobs_count").innerText = 'Общее количество: ' +  
data.jobs_count;  
  }  
</script>  
<h1><%= @current_organisation.name %></h1>  
<div style="text-align: left;">  
  <p class="inf-header">Описание организации</p>  
  <p style="font-size: 15pt;"><%= @current_organisation.description %></p>  
  <p style="font-size: 15pt;">Сайт организации - <a href="<%=  
@current_organisation.website_url%>"><%= @current_organisation.website_url%></a></p>  
  <p style="font-size: 15pt;"><span style="font-style: italic;">Остались  
вопросы?</span> Позвоните нам - <%= @current_organisation.phone %></p>  
  <p class="inf-header">Доступные вакансии</p>  
  <% if @jobs_count > 0 %>  
    <p style="text-align: left; font-size: 15pt; id="jobs_count" >Общее  
количество: <%= @jobs_count %></p>
```

Продолжение кода файла organisation.html.erb:

```
<% end %>
</div>
<div style="float: left;">
  <ul class="menu" style="margin: 0;">
    <ul id="inside">
      <% if @current_organisation.jobs.length > 0 %>
        <% @current_organisation.jobs.each do |job| %>
          <% if (@today_date -
Date.strptime(job.created_at.strftime('%m.%d.%Y'), '%m.%d.%Y')).to_i * 24) < 720 %>
            <li><a href="/job/<%= job.id %>" class="array"><%= job.name
%></a></li>
            <li style="font-size: 14px;"><%=
job.description.split.select.with_index {|element, index| index < 8 }.join(" ")
%>...</li>
            <li>&nbsp;</li>
          <% end %>
        <% end %>
      <% else %>
        <li><a>Доступных вакансий нет!</a></li>
      <% end %>
      <% if @organisation && @has_access %>
        <li id="li" onclick="show_job_creation()"><a> -- Добавить вакансию --
</a></li>
      <% end %>
    </ul>
  </ul>
  <% if @organisation %>
    <%= form_with(url: "/create_job.json", method: "post", id: "add") do %>
      <table id="job" hidden style="padding-left: 0px;">
        <caption><h2>Добавление вакансии</h2></caption>
        <tr>
          <td colspan="2"><a style="font-size: small; font-style: italic;"
onclick="cancel_job_creation()">Отменить создание</a></td>
        </tr>
        <tr>
          <td>Введите название</td>
          <td><%= text_field_tag :name, nil, class: "enter" %></td>
        </tr>
        <tr>
          <td>Введите обязательные требования (через ";")</td>
          <td><%= text_area_tag :need_to_do, nil, class: "enter" %></td>
        </tr>
        <tr>
          <td>Введите описание</td>
          <td><%= text_area_tag :description, nil, class: "enter" %></td>
        </tr>
        <tr>
          <td><%= submit_tag "Добавить", class: "success" %></td>
          <td><span id="error" style="color: red;"><%= @error
%></span></td>
        </tr>
      </table>
    <% end %>
  <% end %>
  <br>
</div>
```

Код файла login.html.erb:

```
<%= form_tag("/auth", :method => "post") do %>
  <table>
    <caption><h1>Вход</h1></caption>
    <tr>
      <td width="40%">Введите почту</td>
      <td><%= text_field_tag :email, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите пароль</td>
      <td><%= password_field_tag :password, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td><%= submit_tag "Войти", class: "success" %></td>
      <td><span style="color: red;"><%= @error %></span></td>
    </tr>
    <tr style="font-style: italic; font-size: medium;">
      <td><span style="font-size: small;">Еще не зарегистрированы?</span></td>
      <td><a href="/register" style="font-size: small;">Зарегистрироваться</a></td>
    </tr>
  </table>
<% end %>
```

Код файла register.html.erb:

```
<%= form_tag("/create_new_user", :method => "post") do %>
  <table id="user">
    <caption><h1>Регистрация нового пользователя</h1></caption>
    <tr>
      <td colspan="2"><a class="change" onclick="to_org()">Сменить
аккаунт</a></td>
    </tr>
    <tr>
      <td width="40%">Введите имя</td>
      <td><%= text_field_tag :name, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td width="40%">Введите фамилию</td>
      <td><%= text_field_tag :surname, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите email</td>
      <td><%= email_field_tag :email, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите телефон</td>
      <td><%= text_field_tag :phone, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите пароль</td>
      <td><%= password_field_tag :password, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Повторите пароль</td>
      <td><%= password_field_tag :another_password, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td><%= submit_tag "Зарегистрироваться", class: "success" %></td>
      <td><span style="color: red;"><%= @error1 %></span></td>
    </tr>
  </table>
<% end %>
```

Продолжение кода файла register.html.erb:

```
</tr>
<tr style="font-style: italic;">
  <td><span style="font-size: small;">Уже зарегистрированы?</span></td>
  <td><a href="/login" style="font-size: small;">Войти</a></td>
</tr>
<!--<tr>
  <td colspan="2"><%= hidden_field_tag :type, "user" %></td>
</tr-->
</table>
<% end %>
<%= form_tag("/create_new_organisation", :method => "post") do %>
  <table id="organisation" hidden>
    <caption><h1>Регистрация новой организации</h1></caption>
    <tr>
      <td colspan="2"><a class="change" onclick="to_user()">Сменить
аккаунт</a></td>
    </tr>
    <tr>
      <td width="40%">Введите название</td>
      <td><%= text_field_tag :name, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите email</td>
      <td><%= email_field_tag :email, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите телефон</td>
      <td><%= text_field_tag :phone, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите url сайта</td>
      <td><%= text_field_tag :website_url, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите пароль</td>
      <td><%= password_field_tag :password, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Повторите пароль</td>
      <td><%= password_field_tag :another_password, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите описание</td>
      <td><%= text_area_tag :description, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td><%= submit_tag "Зарегистрироваться", class: "success" %></td>
      <td><span style="color: red;"><%= @error2 %></span></td>
    </tr>
    <tr style="font-style: italic;">
      <td><span style="font-size: small;">Уже зарегистрированы?</span></td>
      <td><a href="/login" style="font-size: small;">Войти</a></td>
    </tr>
    <!--<tr>
      <td colspan="2"><%= hidden_field_tag :type, "organisation" %></td>
    </tr-->
  </table>
<% end %>
```


Код файла routes.rb:

```
Rails.application.routes.draw do
  root 'base#index'
  get '/organisation/:id' => 'organisation_interaction#organisation'
  get '/job/:id' => 'organisation_interaction#job'
  post '/create_job' => 'organisation_interaction#create_job'
  post '/add_worker' => 'organisation_interaction#add_worker'
  get '/login' => 'user_interaction#login'
  get '/logout' => 'user_interaction#logout'
  get '/register' => 'user_interaction#register'
  post '/auth' => 'user_interaction#auth'
  post '/create_new_user' => 'user_interaction#create_new_user'
  post '/create_new_organisation' => 'user_interaction#create_new_organisation'
end
```

Код файла base_controller_test.rb:

```
require 'test_helper'
class BaseControllerTest < ActionDispatch::IntegrationTest
  test "should get index" do
    get "/"
    assert_response :success
  end
end
```

Код файла organization_interaction_controller_test.rb:

```
require 'test_helper'
class OrganisationInteractionControllerTest < ActionDispatch::IntegrationTest
  test "should get organisation/:id" do
    @organisation = Organisation.new(name: "Example Name", description: "Something
can be here", website_url:"example.com", email: "organisation@example.com",
phone:"89150051555", password: "00001")
    @organisation.save
    get "/organisation/#{@organisation.id}"
    assert_response :success
  end
  test "should get job/:id" do
    @organisation = Organisation.new(name: "Example Name", description: "Something
can be here", website_url:"example.com", email: "organisation@example.com",
phone:"89150051555", password: "00001")
    @organisation.save
    @job = @organisation.jobs.create(name: "Example Job", description: "Something can
be here", need_to_do: "let's do it")
    get "/job/#{@job.id}"
    assert_response :success
  end
end
```

Код файла user_interaction_controller_test.rb:

```
require 'test_helper'
class UserInteractionControllerTest < ActionDispatch::IntegrationTest
  test "should get login" do
    get "/login"
    assert_response :success
  end
  test "should get register" do
    get "/register"
    assert_response :success
  end
end
```

Продолжение кода файла `user_interaction_controller_test.rb`:

```
test "should create a new user" do
  post "/create_new_user", params: { name: "Example Name", surname: "Example Surname", email: "user@example.com", phone:"89150051550", password: "12345", another_password: "12345" }
  assert_response :redirect
end
test "should create a new organisation" do
  post "/create_new_organisation", params: { name: "Example Name", description: "Something can be here", website_url:"example.com", email: "organisation@example.com", phone:"89150051555", password: "00001", another_password: "00001" }
  assert_response :redirect
end
end
```

Код файла `website_capabilities_test.rb`:

```
require 'test_helper'
class WebsiteCapabilitiesTest < ActionDispatch::IntegrationTest
  test "can register a new user and create a summary from him" do
    post "/create_new_user", params: { name: "Example Name", surname: "Example Surname", email: "user@example.com", phone:"89150051550", password: "12345", another_password: "12345" }
    assert_response :redirect
    @organisation = Organisation.new(name: "Example Name", description: "Something can be here", website_url:"example.com", email: "organisation@example.com", phone:"89150051555", password: "00001")
    @organisation.save
    @job = @organisation.jobs.create(name: "Example Job", description: "Something can be here", need_to_do: "let's do it")
    post "/add_worker", params: {job_id: @job.id, summary: "example summary"}
    assert_response :redirect
  end
  test "can register a new organisation and create a job from it" do
    post "/create_new_organisation", params: { name: "Example Name", description: "Something can be here", website_url:"example.com", email: "organisation@example.com", phone:"89150051555", password: "00001", another_password: "00001" }
    assert_response :redirect
    post "/create_job", params: {name: "Example Job", need_to_do: "Some needs", description: "Here is a new job"}
    assert_response :redirect
  end
end
```

Код файла `account_test.rb`:

```
require 'test_helper'
class AccountTest < ActiveSupport::TestCase
  def setup
    @organisation = Organisation.new(name: "Example Name", description: "Something can be here", website_url:"example.com", email: "organisation@example.com", phone:"89150051555", password: "00001")
    @user = User.new(name: "Example Name", surname: "Example Surname", email: "user@example.com", phone:"89150051550", password: "12345")
    @account = @organisation.accounts.new(session_id: "c6116f7395b34412ad0aea3c9b8d4bfa", user_id: @user_id)
  end
  test "should be valid" do
```

Продолжение кода файла account_test.rb:

```
    assert @account.valid?
  end
  test "session_id should be present" do
    @account.session_id = ""
    assert_not @account.valid?
  end
end
```

Код файла job_test.rb:

```
require 'test_helper'
class JobTest < ActiveSupport::TestCase
  def setup
    @organisation = Organisation.new(name: "Example Name", description: "Something
can be here", website_url:"example.com", email: "organisation@example.com",
phone:"89150051555", password: "00001")
    @job = @organisation.jobs.new(name: "Example Job", description: "Something can be
here", need_to_do: "let's do it")
  end
  test "should be valid" do
    assert @job.valid?
  end
  test "name should be present" do
    @job.name = ""
    assert_not @job.valid?
  end
  test "description should be present" do
    @job.description = ""
    assert_not @job.valid?
  end
  test "need_to_do should be present" do
    @job.need_to_do = ""
    assert_not @job.valid?
  end
  test "name should not be too long" do
    @job.name = "a" * 51
    assert_not @job.valid?
  end
  test "need_to_do should not be too long" do
    @job.need_to_do = "a" * 101
    assert_not @job.valid?
  end
end
```

Код файла middle_test.rb:

```
require 'test_helper'
class MiddleTest < ActiveSupport::TestCase
  def setup
    @user = User.create(name: "Example Name", surname: "Example Surname", email:
"user@example.com", phone:"89150051550", password: "12345")
    @organisation = Organisation.new(name: "Example Name", description:
"Something can be here", website_url:"example.com", email:
"organisation@example.com", phone:"89150051555", password: "00001")
    @job = @organisation.jobs.new(name: "Example Job", description: "Something
can be here", need_to_do: "let's do it")
    @middle = @job.middles.new(summary: "Hello, i can do this", user_id:
@user.id)
  end
```

Продолжение кода файла middle_test.rb:

```
test "should be valid" do
  assert @middle.valid?
end
test "summary should be present" do
  @middle.summary = ""
  assert_not @middle.valid?
end
end
```

Код файла organisation_test.rb:

```
require 'test_helper'
class OrganisationTest < ActiveSupport::TestCase
  def setup
    @organisation = Organisation.new(name: "Example Name", description: "Something
can be here", website_url:"example.com", email: "organisation@example.com",
phone:"89150051555", password: "00001")
  end
  test "should be valid" do
    assert @organisation.valid?
  end
  test "name should be present" do
    @organisation.name = ""
    assert_not @organisation.valid?
  end
  test "description should be present" do
    @organisation.description = ""
    assert_not @organisation.valid?
  end
  test "email should be present" do
    @organisation.email = ""
    assert_not @organisation.valid?
  end
  test "phone should be present" do
    @organisation.phone = ""
    assert_not @organisation.valid?
  end
  test "website_url should be present" do
    @organisation.website_url = ""
    assert_not @organisation.valid?
  end
  test "phone validation should accept valid phones" do
    valid_phones = %w[89170004045 89161115522 89159998765]
    valid_phones.each do |valid_phone|
      @organisation.phone = valid_phone
      assert @organisation.valid?, "#{valid_phones.inspect} should be valid"
    end
  end
  test "email validation should accept valid addresses" do
    valid_addresses = %w[organisation@example.com ORGANISATION@foo.COM A_US-
ER@foo.bar.org first.last@foo.jp alice+bob@baz.cn]
    valid_addresses.each do |valid_address|
      @organisation.email = valid_address
      assert @organisation.valid?, "#{valid_address.inspect} should be valid"
    end
  end
  test "emails and logins should be unique" do
    duplicate_organisation = @organisation.dup
```

Продолжение кода файла organisation_test.rb:

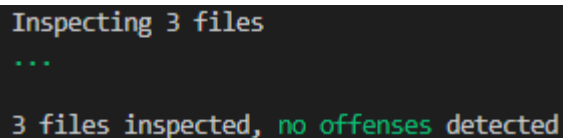
```
@organisation.save
assert_not duplicate_organisation.valid?
end
test "name should not be too long" do
  @organisation.name = "a" * 51
  assert_not @organisation.valid?
end
test "website_url should not be too long" do
  @organisation.website_url = "a" * 51
  assert_not @organisation.valid?
end
test "email should not be too long" do
  @organisation.email = "a" * 244 + "@example.com"
  assert_not @organisation.valid?
end
test "phone should not be too long" do
  @organisation.phone = "1" * 100
  assert_not @organisation.valid?
end
end
```

Код файла user_test.rb:

```
require 'test_helper'
class UserTest < ActiveSupport::TestCase
  def setup
    @user = User.new(name: "Example Name", surname: "Example Surname", email:
"user@example.com", phone:"89150051550", password: "12345")
  end
  test "should be valid" do
    assert @user.valid?
  end
  test "name should be present" do
    @user.name = ""
    assert_not @user.valid?
  end
  test "surname should be present" do
    @user.surname = ""
    assert_not @user.valid?
  end
  test "email should be present" do
    @user.email = ""
    assert_not @user.valid?
  end
  test "phone should be present" do
    @user.phone = ""
    assert_not @user.valid?
  end
  test "phone validation should accept valid phones" do
    valid_phones = %w[89170004045 89161115522 89159998765]
    valid_phones.each do |valid_phone|
      @user.phone = valid_phone
      assert @user.valid?, "#{valid_phones.inspect} should be valid"
    end
  end
  test "email validation should accept valid addresses" do
    valid_addresses = %w[user@example.com USER@foo.COM A_US-ER@foo.bar.org
first.last@foo.jp alice+bob@baz.cn]
    valid_addresses.each do |valid_address|
```

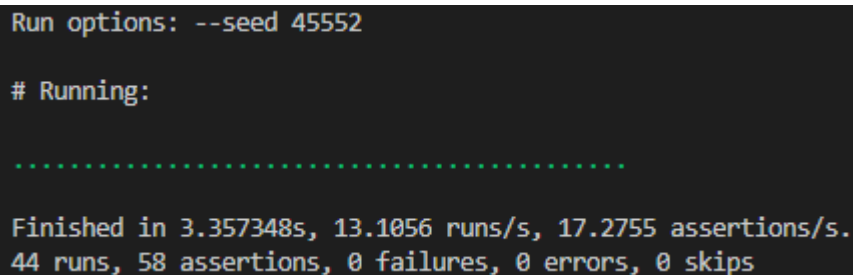
Продолжение кода файла user_test.rb:

```
@user.email = valid_address
assert @user.valid?, "#{valid_address.inspect} should be valid"
end
end
test "emails, names and surnames should be unique" do
  duplicate_user = @user.dup
  @user.save
  assert_not duplicate_user.valid?
end
test "name should not be too long" do
  @user.name = "a" * 51
  assert_not @user.valid?
end
test "surname should not be too long" do
  @user.surname = "a" * 51
  assert_not @user.valid?
end
test "email should not be too long" do
  @user.email = "a" * 244 + "@example.com"
  assert_not @user.valid?
end
test "phone should not be too long" do
  @user.phone = "1" * 100
  assert_not @user.valid?
end
end
```



```
Inspecting 3 files
...
3 files inspected, no offenses detected
```

Рисунок 1 «Результаты тестирования Rubocop»



```
Run options: --seed 45552

# Running:

.....

Finished in 3.357348s, 13.1056 runs/s, 17.2755 assertions/s.
44 runs, 58 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 2 «Результаты тестов»

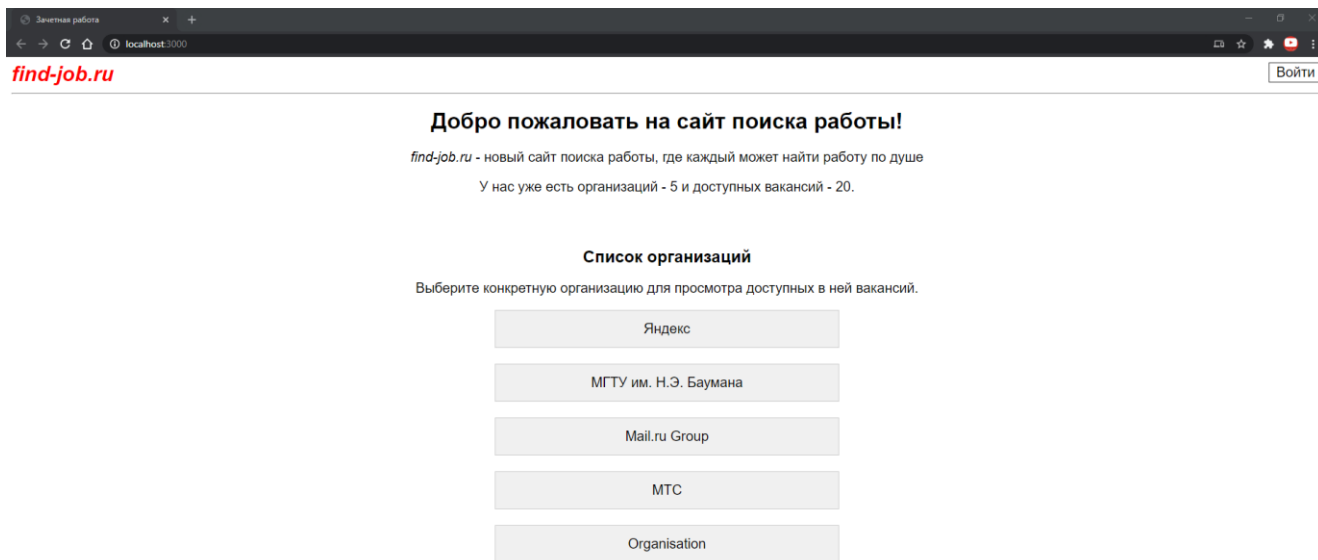


Рисунок 3 «Главная страница сайта»

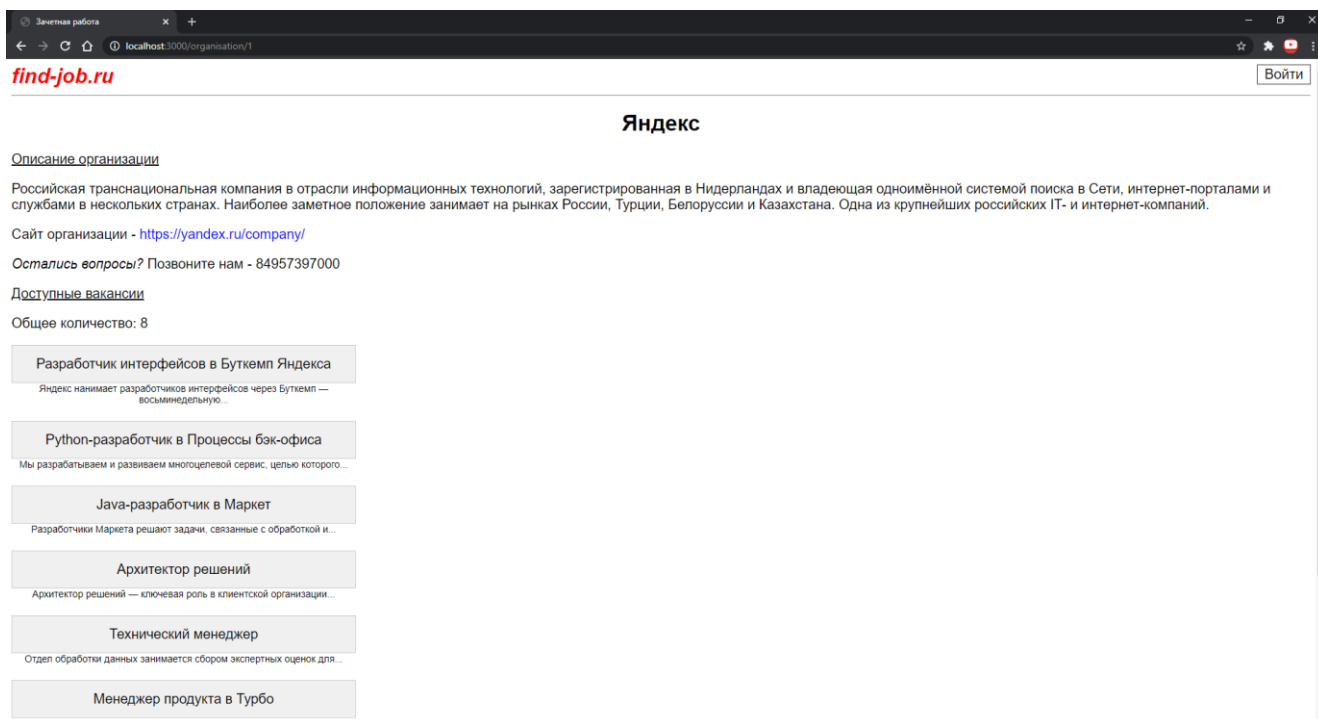


Рисунок 4 «Страница организации Яндекс»

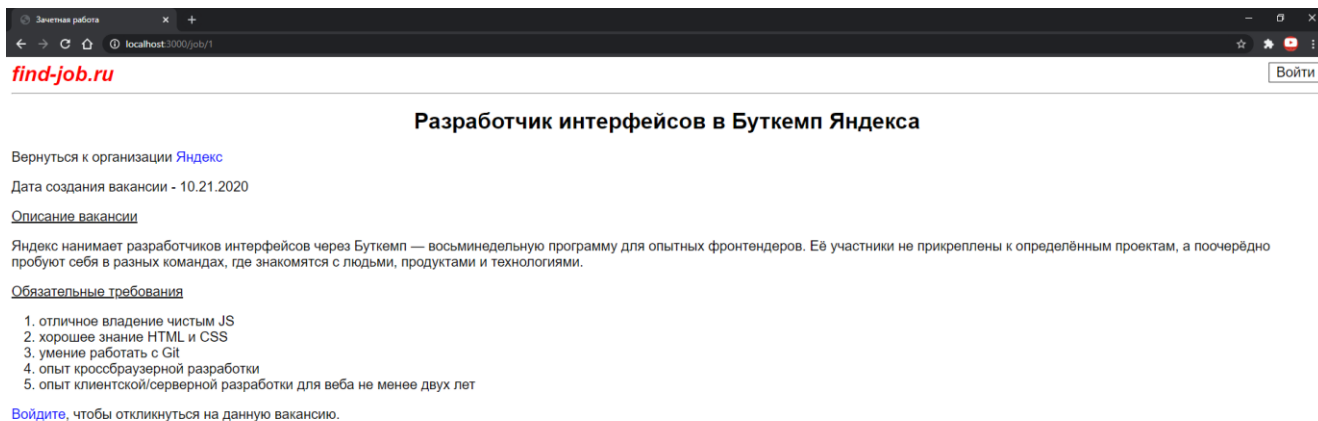


Рисунок 5 «Страница вакансии Яндекса»

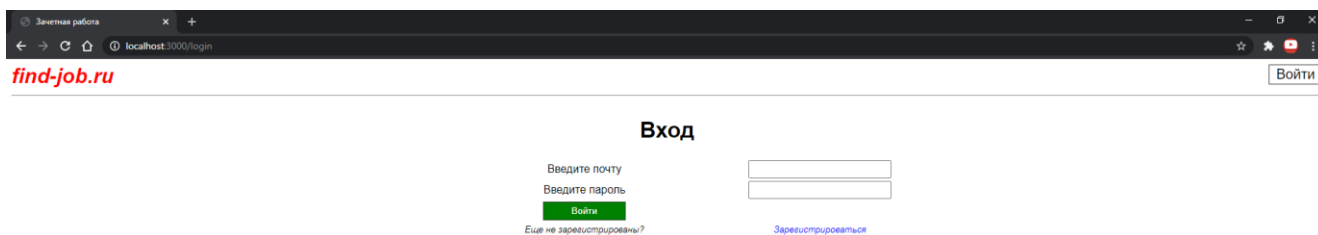


Рисунок 6 «Страница входа»

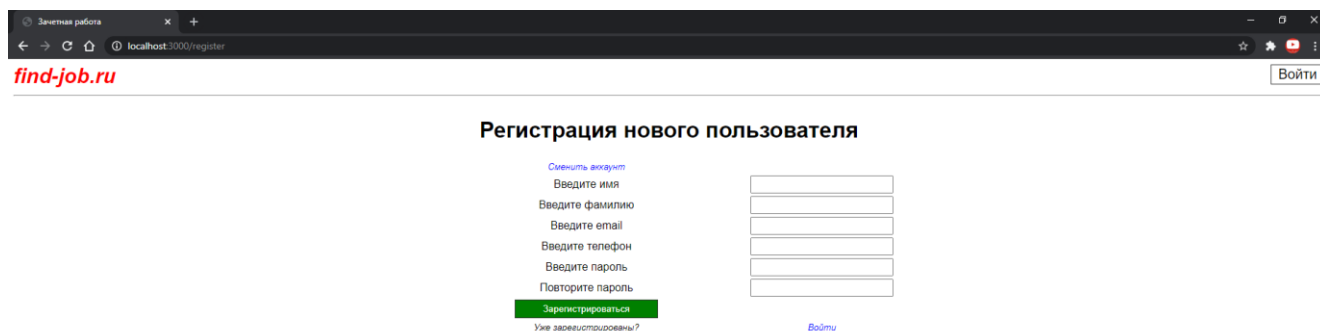


Рисунок 7 «Страница регистрации»

Вывод

В ходе данной зачетной работы были продемонстрированы основные навыки, полученные в данном курсе, в создании веб-приложений с использованием фреймворка Rails.