



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

О Т Ч Е Т

по зачётной работе

Дисциплина: Языки интернет программирования

Студент	<u>ИУ6-31Б</u> (Группа)	<u>28.12.22</u> (Подпись, дата)	<u>М.Р. Уторов</u> (И.О. Фамилия)
Преподаватель		<u></u> (Подпись, дата)	<u>Д.В. Малахов</u> (И.О. Фамилия)

Цель: целью моей работы было сделать сайт для подготовки учеников школ к ЕГЭ по информатике, где можно было бы решить экзамен полностью за отведённое время, или же решить первую или вторую часть по отдельности. Также там присутствует рейтинговая система между учениками и система уровней (которые повышаются при получении опыта за решение заданий).

Ход работы: сначала я спроектировал базу данных, выделил все сущности и определил роли по бизнес-процессам, которые хотел видеть в своей работе. Итак углубимся немного глубже в структуру моей работы. Есть 3 роли: ученик, администратор и супер-администратор.

1) Ученик:

Имеет базовые возможности, решать тесты, просматривать уже решённые тесты, редактировать свои данные для авторизации, просматривать свой профиль и уровень, а также рейтинговую таблицу

2) Администратор:

Имеет почти те же возможности, что и ученик, за исключением того, что не может участвовать в рейтинге между пользователями, но имеет дополнительные возможности: добавление заданий на сайт

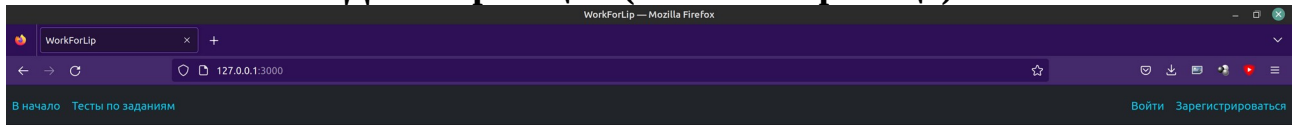
3) Супер-Администратор:

Имеет все вышеописанные возможности (как и обычный администратор не участвует в рейтинговой таблице), а также возможность назначать обычных учеником администраторами, и наоборот администраторов делать учениками. Может удалить любого пользователя с сайта, предполагается, что супер-администратор будет один.

Стоит отметить, что сессия на сайте ограничена по времени, по истечении которого пользователю нужно будет снова пройти авторизацию.

Проект в скриншотах:

До авторизации (Главная страница)



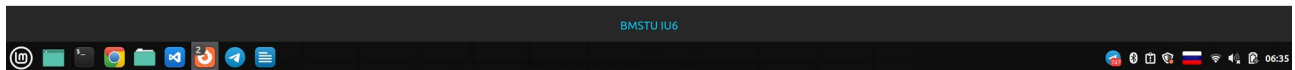
На сайте зарегистрировано: 0

Рейтинг

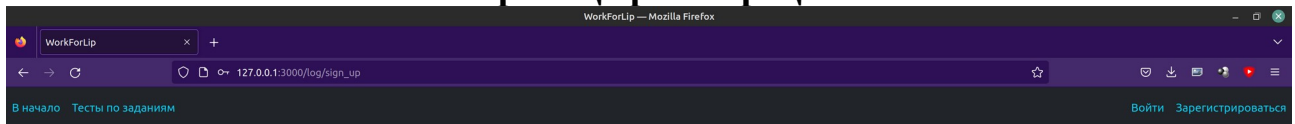
Место

Пользователь

Баллы



Страница регистрации



Sign Up

Email address:

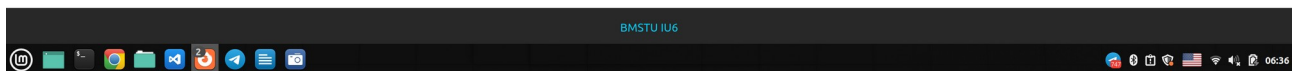
user@user.ru

Password:

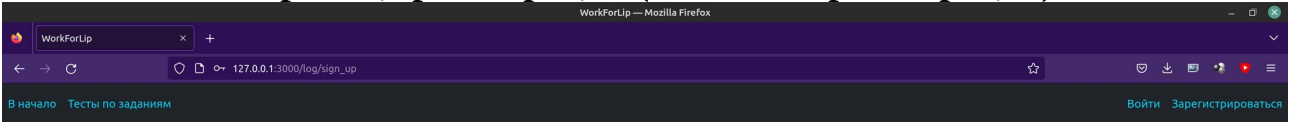
Retype password:

Sign Up

Homepage



Страница регистрации (Успешная регистрация)



Sign Up

Email address:

Password:

Retype password:

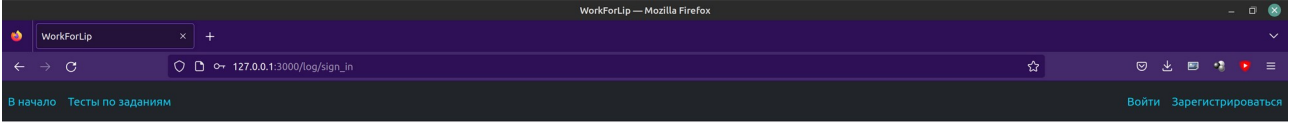
[Sign Up](#)

[Homepage](#)

Successfully
registration!



Страница авторизации



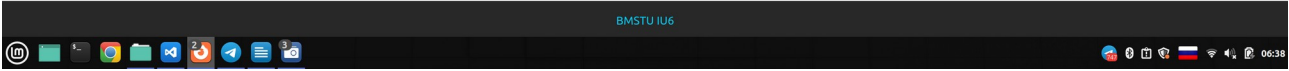
Log In

Email address:

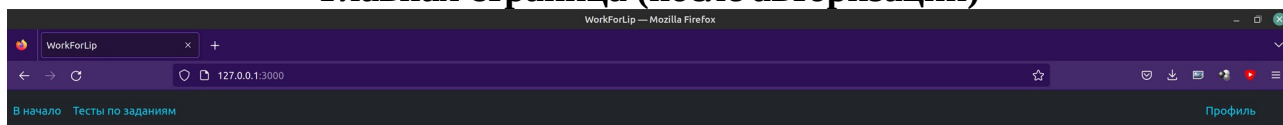
Password:

[Log In](#)

[Homepage](#)



Главная страница (после авторизации)



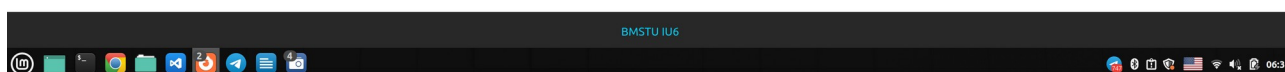
На сайте зарегистрировано: 1

Рейтинг

Место

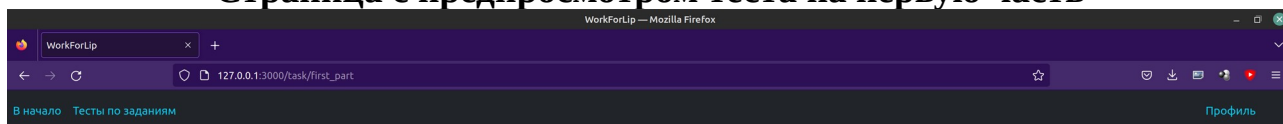
Пользователь

Баллы



Замечание: отсутствие пользователя в рейтинговой таблице обусловлено отсутствием опыта.

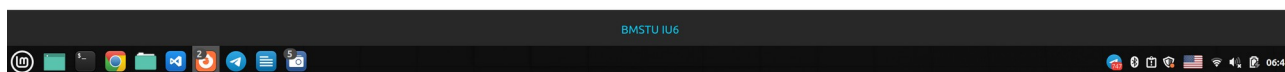
Страница с предпросмотром теста на первую часть



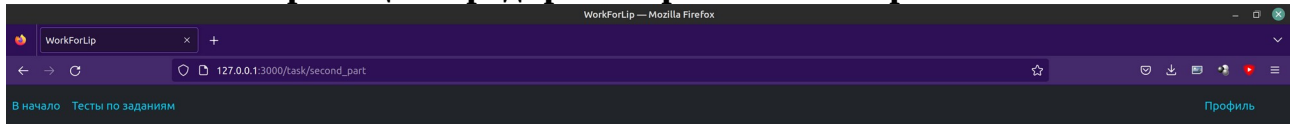
Первая часть

Первая часть состоит из 25 простых заданий на комбинаторику, алгебру логики, базовые алгоритмические задачи, электронные таблицы и работу с текстовым редактором.

Начать



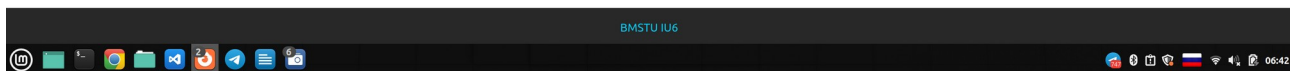
Страница с предпросмотром теста второй части



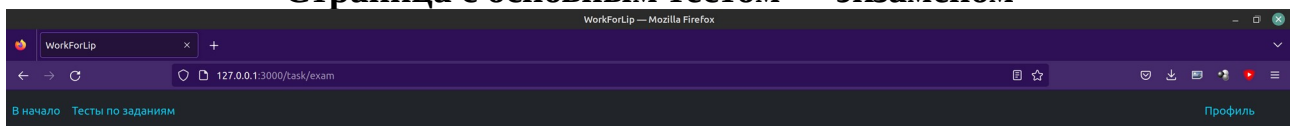
Вторая часть

Вторая часть это два последних задания с экзамена, то есть 26 и 27. Это считаются сложные задачи на алгоритмическое программирование, за эти задания ставятся по два первичных балла

[Начать](#)



Страница с основным тестом — экзаменом



Экзамен

В ЕГЭ по информатике 2022 года — девять практико-ориентированных заданий, тогда как в прошлом году таких заданий было только восемь. В них проверяются навыки работы с прикладным программным обеспечением и навыки написания собственных программ для решения различных задач. В отличие от ЕГЭ прошлых лет, в нынешней структуре экзамена содержатся задания только с кратким ответом базового, повышенного и высокого уровней сложности. По сравнению с ЕГЭ 2021 года изменились задания 3, 9 и 17.

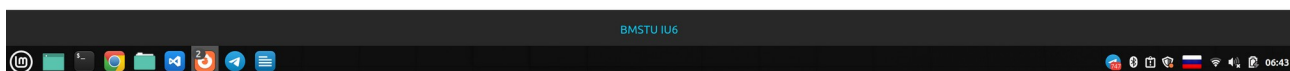
Длительность экзамена – 3 часа 55 минут (255 минут), полтора часа из которых рекомендовано уделить выполнению заданий первой части КИМов.

Задания в билетах разделены на блоки:

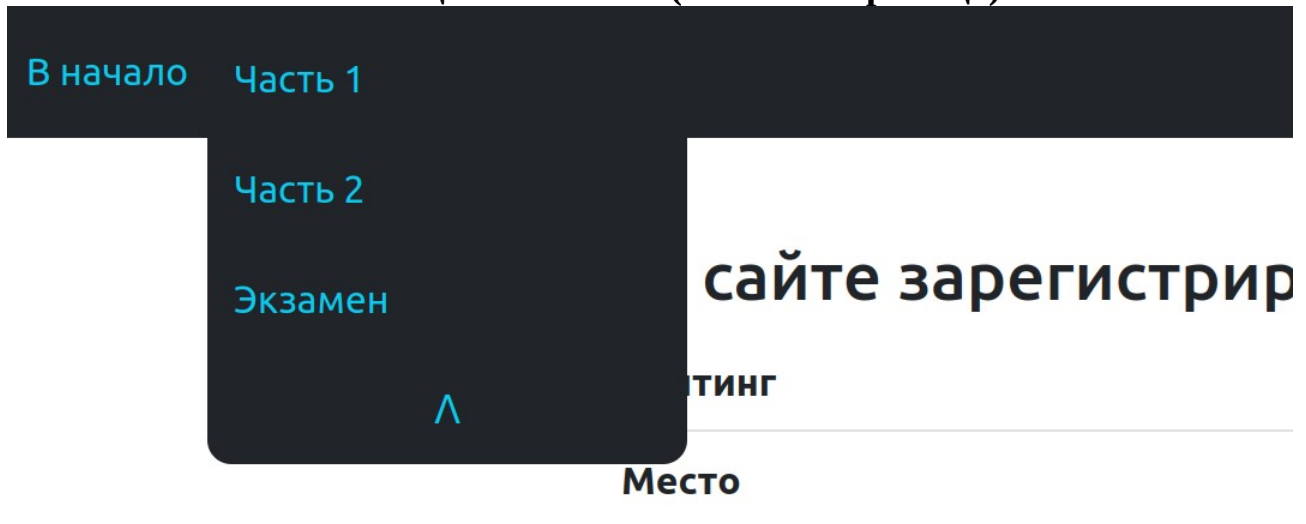
- **Часть 1** - 25 задания с кратким ответом.
- **Часть 2** - 2 задачи с развернутым ответом.

Всего: 35 баллов.

[Начать](#)



Навигация по тестам (Главная страница)



Пример страницы с одним из задания теста

Задание 3

(А. Порог) В файле приведён фрагмент базы данных «Города и страны», описывающей различные страны, города и языки. База данных состоит из трех таблиц. Таблица «Страны» (код, название, континент, регион, площадь, год получения независимости, население, ОПЖ – ожидаемая продолжительность жизни, ВНД – валовый национальный доход, предыдущее значение ВНД, форма правления, идентификатор столицы). Таблица «Города» (идентификатор, название, код страны, район, население). Таблица «Языки» (код языка, код страны, название, является ли официальным, процент использования в стране). По некоторым значениям данных нет, в этом случае в таблице внесено значение NULL. На рисунке приведена схема базы данных.

Страны	Языки	Города
Код	Код	Код
Название	Название	Название
Континент	Официальный	Код страны
Регион	Процент	Район
Площадь		Население
Год независимости		
Население		
ОПЖ		
ВНД		
ВНД_пред		
Форма правления		
Код столицы		

Используя информацию из приведённой базы данных, определите среднюю площадь стран Южной Америки, в которых население столицы не превышает 150 000. Ответ округлите до целого числа.

3.xls

Закончить тест 3:39

Замечание: справа можно заметить навигацию по заданиям, а также кнопку предварительного завершения теста и таймер, который отсчитывает время теста и по его истечению автоматически нажимает кнопку завершения теста. Ещё может присутствовать у заданий кнопка скачивания файла, с которым

нужно работать в ходе решения задания.

Таблица с результатом теста (вверх)

WorkForLip

WorkForLip — Mozilla Firefox

127.0.0.1:3000/task/result?current_number=3&answer=

В начало Тесты по заданиям Профиль

Номер	Ваш ответ	Правильный ответ	Первичный балл
1		41	0
2		ухwz	0
3		88479	0
4			0
5			0
6			0
7			0
8			0
9			0
10			0
11			0
12			0
13			0
14			0
15			0
16			0
17			0

BMSTU IU6

Таблица с результатом теста (низ)

WorkForLip

WorkForLip — Mozilla Firefox

127.0.0.1:3000/task/result?current_number=3&answer=

В начало

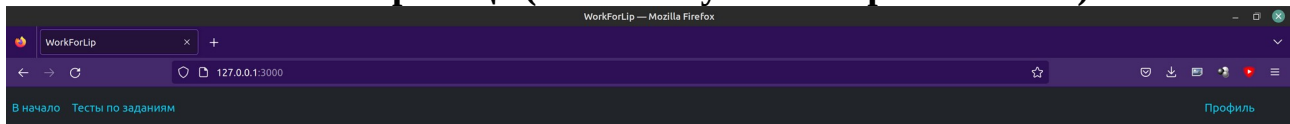
9		0
10		0
11		0
12		0
13		0
14		0
15		0
16		0
17		0
18		0
19		0
20		0
21		0
22		0
23		0
24		0
25		0
26		0
27		0

Ваш первичный балл: 0

В начало

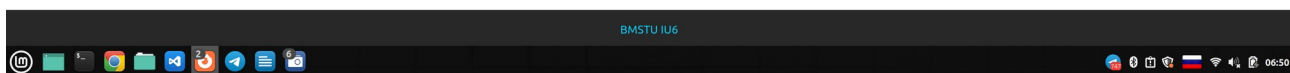
BMSTU IU6

Главная страница (после получения первого опыта)

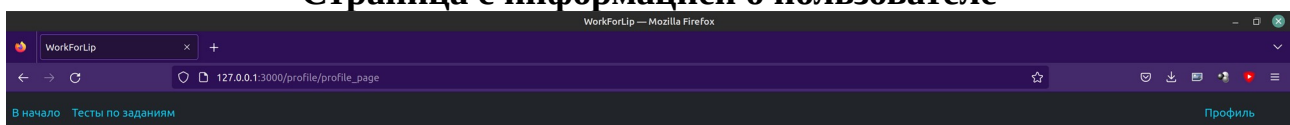


На сайте зарегистрировано: 1

Рейтинг		
Место	Пользователь	Баллы
1	user@user.ru	2



Страница с информацией о пользователе



Email:

user@user.ru

Роль:

user

Дата регистрации:

10.1.2023

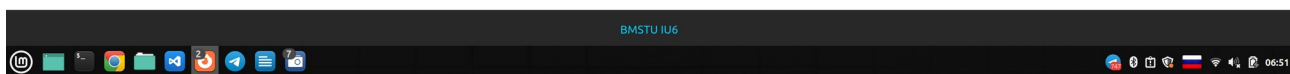
Количество пройденных тестов:

2

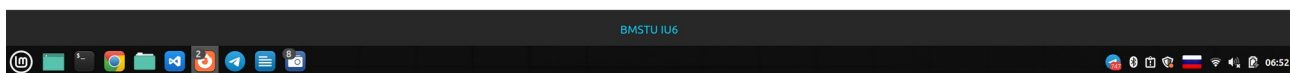
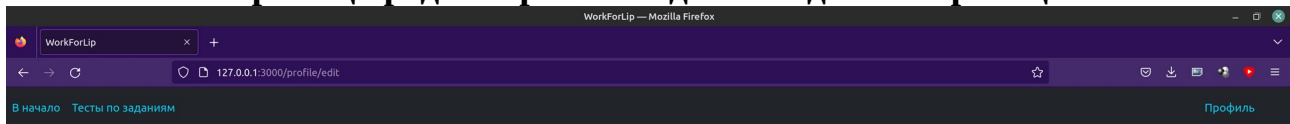
Уровень: 1

1

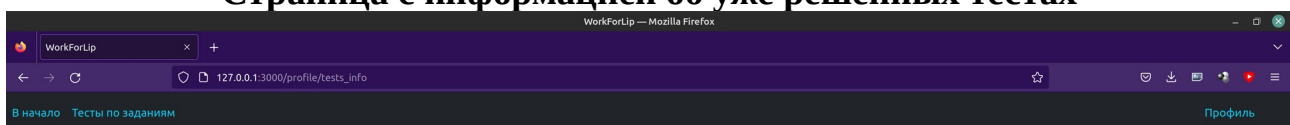
2



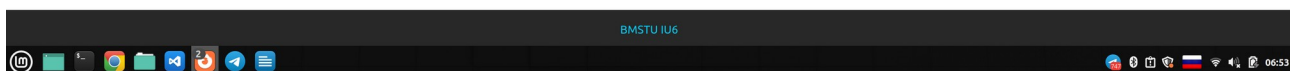
Страница редактирования данных для авторизации

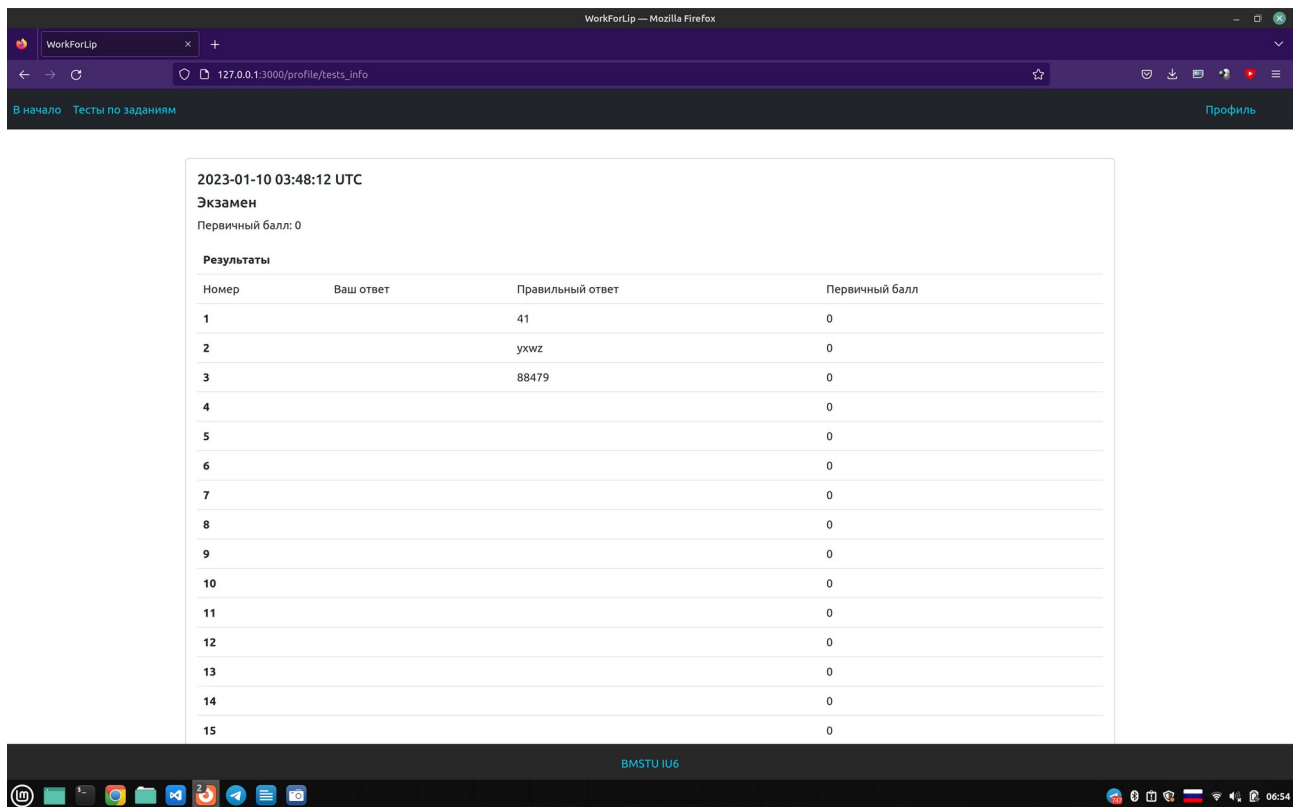


Страница с информацией об уже решённых тестах

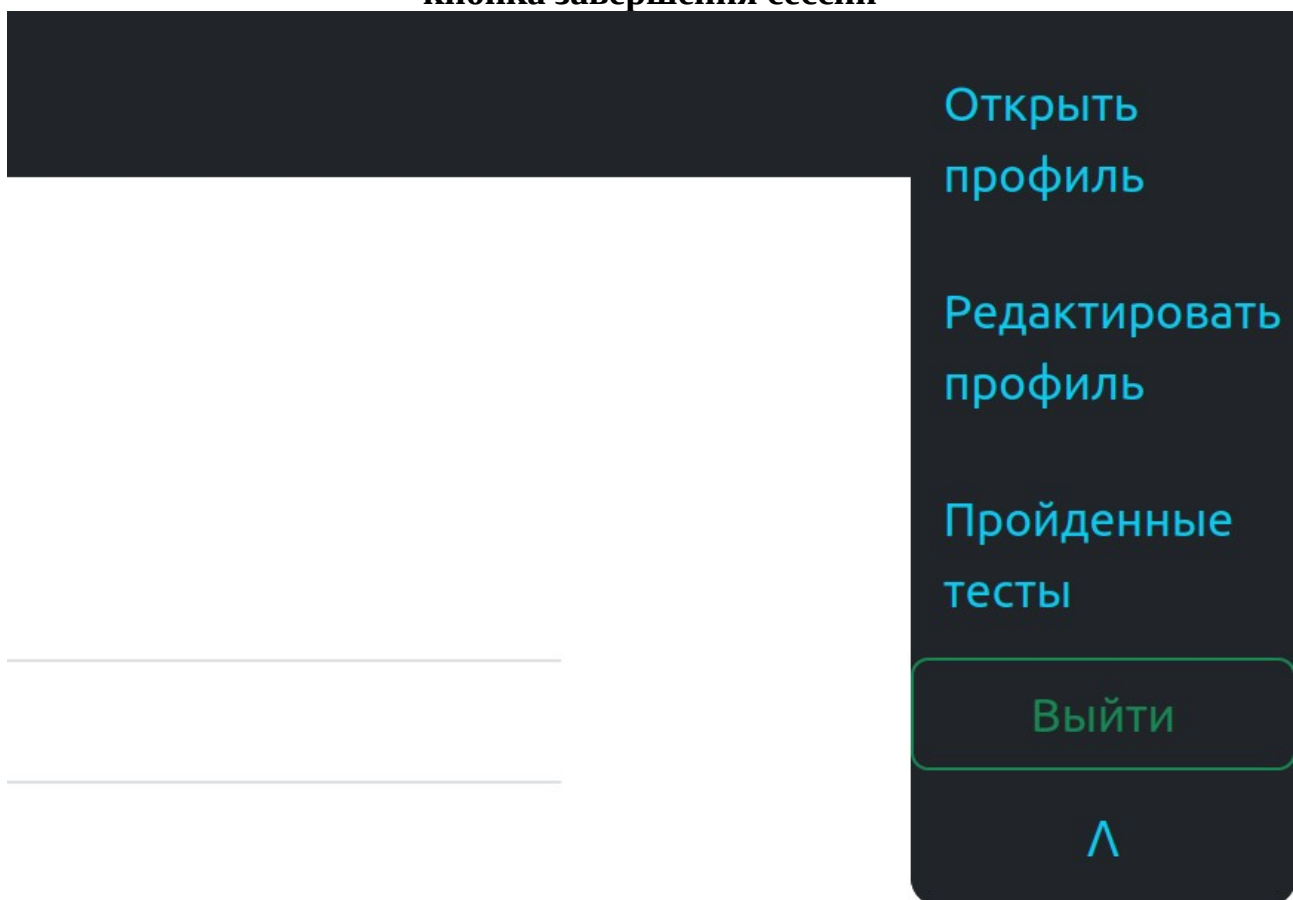


2023-01-10 03:48:12 UTC Экзамен Первичный балл: 0 Подробнее
2023-01-10 03:49:53 UTC Экзамен Первичный балл: 2 Подробнее

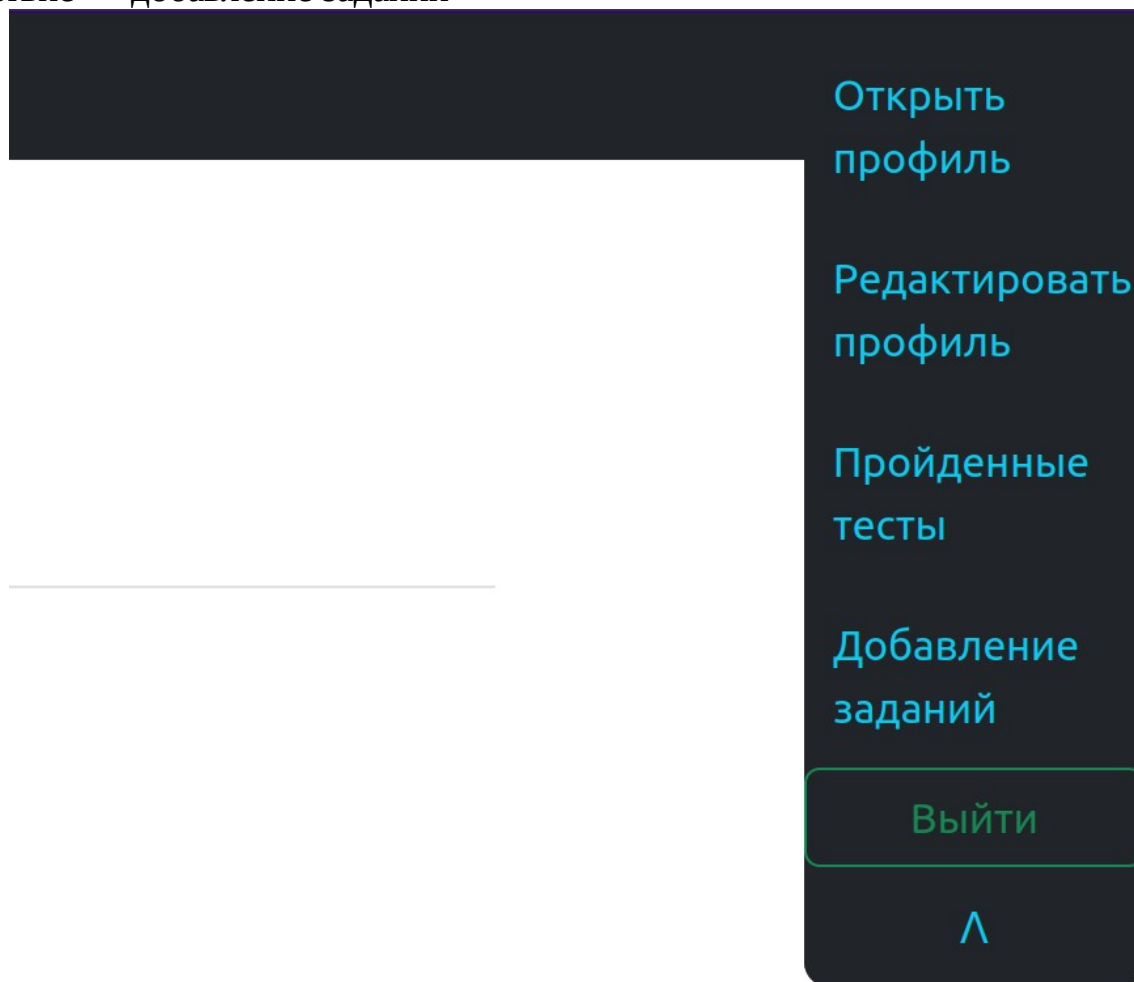




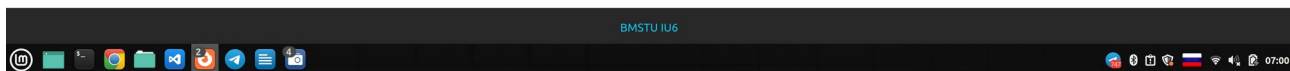
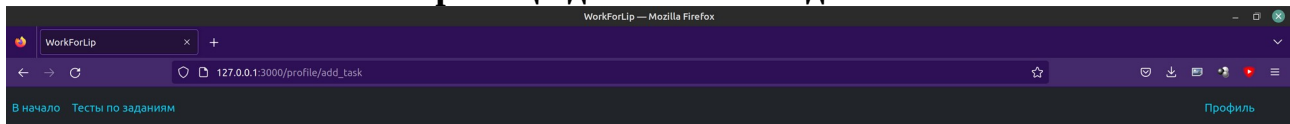
**Навигация по страницам связанных с профилем пользователя, а также
кнопка завершения сессии**



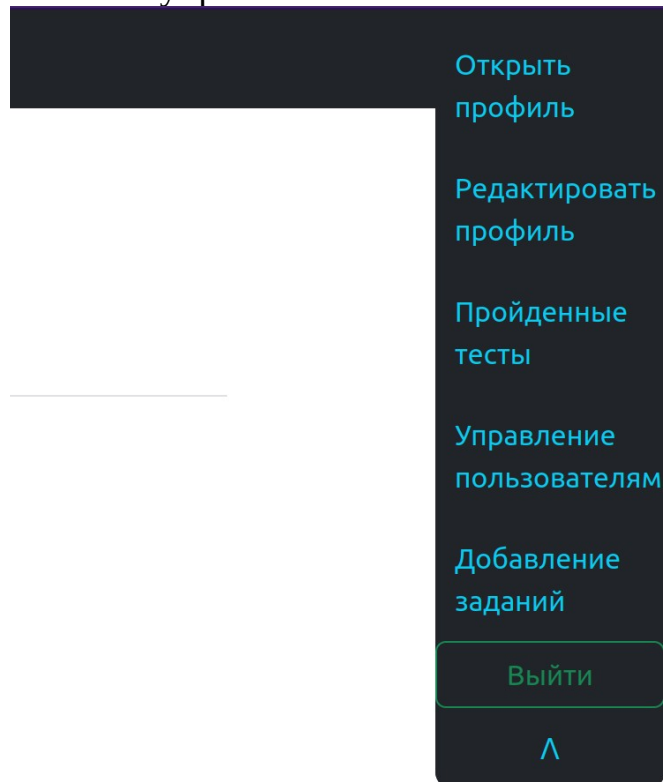
Теперь рассмотрим ситуацию, когда текущий пользователь администратор, в навигации действий связанных с пользователем появляется дополнительное действие — добавление заданий



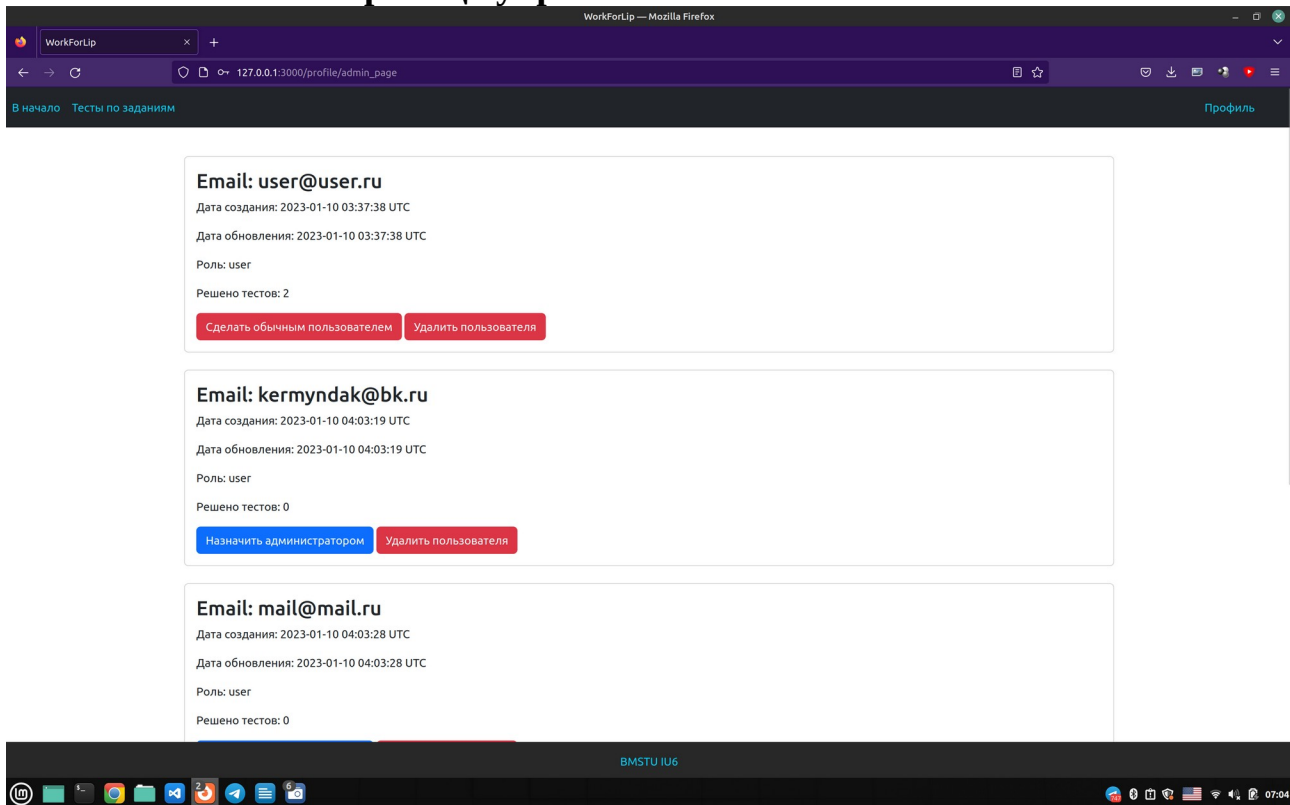
Страница добавления заданий



Теперь рассмотрим ситуацию, когда текущий пользователь супер-администратор, в навигации действий связанных с пользователем появляется дополнительное действие — управление пользователями



Страница управления пользователями



Маршруты:

routes.rb

```
Rails.application.routes.draw do
  root 'homepage#index'
  get 'task/first_part'
  get 'task/second_part'
  get 'task/exam'
  post 'log/log'
  get 'log/sign_in'
  get 'log/sign_out'
  get 'log/sign_up'
  get 'homepage/edit'
  get 'homepage/profile'
  get 'homepage/up'
  get 'task/task'
  get 'task/change_task'
  get '/task/downloader'
  get '/task/result'
  get '/profile/profile_page'
  get '/profile/admin_page'
  get '/profile/edit'
  post '/profile/log'
  get '/profile/tests_info'
  get '/profile/full_info'
```

```

get '/profile/up'
get '/profile/set_admin'
get '/profile/destroy'
get '/profile/add_task'
# Define your application routes per the DSL in
https://guides.rubyonrails.org/routing.html

```

```

# Defines the root path route ("/")
# root "articles#index"
end

```

Код контроллеров:

application_controller.rb

```

# frozen_string_literal: true

# This class for comon methods in all controllers
class ApplicationController < ActionController::Base
  def set_cookie
    if cookies[:login]
      if session[:current_user_id]
        cookies[:login] = { value: session[:current_user_id], expires: (Time.now +
600) } # 10 minute
      end
    else
      session[:current_user_id] = nil
    end
  end

  def redirect_to_sign_in
    redirect_to '/log/sign_in' unless cookies[:login]
  end
end

```

homepage_controller.rb

```

# frozen_string_literal: true

# This class for actions and other methods in homepage controller
class HomepageController < ApplicationController
  before_action :set_cookie
  before_action :redirect_to_sign_in, only: %i[profile edit up]
  def index
    @users = User.where(role: 'user')
    @users = @users.map { |user| [user, Test.where(user_id:

```

```

user.id).map(&:result).sum] }
      .sort_by(&:last).reverse
end

def profile
  @role = User.find(session[:current_user_id]).role
  render partial: 'profile_list'
end

def edit
  render partial: 'frame'
end

def up
  render partial: 'up_frame'
end
end

```

log_controller.rb

```

# frozen_string_literal: true

# This class for actions and methods in log controller
class LogController < ApplicationController
  before_action :set_params
  before_action :set_cookie, only: :sign_out
  def sign_in
    return unless check

    user = User.find_by(email: @email)
    if user.authenticate(@password)
      session[:current_user_id] = user.id
      cookies[:login] = { value: user.id, expires: (Time.now + 60) } # 1 minute
      redirect_to root_path
    else
      @msg << 'Incorrect password!'
    end
  end

  def sign_up
    user = User.new(email: @email, password: @password,
                    password_confirmation: @password_confirmation, role: 'user')
    if user.valid?
      user.save
      @msg_s = true
    else

```



```

      user.errors.objects.map(&:full_message).each { |msg| @msg << msg }
    end
  end

  def sign_out
    session[:current_user_id] = nil
    cookies.delete :login
    redirect_to root_path
  end

  def log
    case @commit
    when 'Sign Up'
      sign_up
    when 'Log In'
      sign_in
    end
  end

  private

  def set_params
    @email = params[:email]
    @password = params[:password]
    @password_confirmation = params[:password_confirmation]
    @commit = params[:commit]
    @msg = []
    @msg_s = false
  end

  def check
    if User.find_by(email: @email).nil?
      @msg << 'This email is not registered'
      return false
    end
    true
  end
end

```

profile_controller.rb

```

# frozen_string_literal: true

# This class for actions and methods in profile controller
class ProfileController < ApplicationController
  before_action :set_cookie

```

```
before_action :redirect_to_sign_in
before_action :set_profile, only: :profile_page
before_action :set_edit, only: :log
```

```
def profile_page
  @balls = @tests.map(&:result).sum
  @level = get_level(@balls)
  @procent = ((@balls - @level[1]) / (@level[2] - @level[1]).to_f * 100).to_i
end
```

```
def admin_page
  redirect_to root_path if User.find(session[:current_user_id]).role != 'superadmin'
  @users = User.where.not(role: 'superadmin')
end
```

```
def set_admin
  @user_id = params[:user].to_i
  @user = User.find(@user_id)
  if @user.role == 'user'
    @user.update_column(:role, 'admin')
  else
    @user.update_column(:role, 'user')
  end
  render partial: 'admin'
end
```

```
def destroy
  @user_id = params[:user].to_i
  User.destroy(@user_id)
  redirect_to '/profile/admin_page'
end
```

```
def edit; end
```

```
def up
  @id = params[:test].to_i
  render partial: 'up'
end
```

```
def full_info
  @tests = Test.where(user_id: session[:current_user_id])
  @current = Test.find(params[:test].to_i)
  render partial: 'full_info'
end
```

```

def tests_info
  @tests = Test.where(user_id: session[:current_user_id])
end

def log
  @current.update!(email: @email, password: @password,
    password_confirmation: @password_confirmation)
end

private

def set_edit
  @email = params[:email]
  @password = params[:password]
  @password_confirmation = params[:password_confirmation]
  @current = User.find(session[:current_user_id])
  messages
end

def set_profile
  @current = User.find(session[:current_user_id])
  @born = @current.created_at
  @born = "#{@born.day}.#{@born.month}.#{@born.year}"
  @tests = Test.where(user_id: session[:current_user_id])
end

def messages
  @msg = []
  @msg_s = false
  user = User.new(email: @email, password: @password,
    password_confirmation: @password_confirmation)
  if user.valid?
    @msg_s = true
  else
    user.errors.objects.map(&:full_message).each { |msg| @msg << msg }
  end
end

def get_level(ball)
  case ball
  when (0..10)
    [1, 0, 10]
  when (11..35)
    [2, 11, 35]
  when (36..75)

```

```

    [3, 36, 75]
  when (76..120)
    [4, 76, 120]
  else
    [5, 0, 5]
  end
end
end
end

```

task_controller.rb

```

# frozen_string_literal: true

```

```

# This class for actions and methods in task controller

```

```

class TaskController < ApplicationController

```

```

  before_action :redirect_to_sign_in

```

```

  before_action :set_params_test, only: %i[change_task task downloader result]

```

```

  before_action :set_params, only: %i[change_task task downloader]

```

```

  before_action :set_params_result, only: :result

```

```

  before_action :set_test_cookie, only: :task # set cookie time for test pages

```

```

  before_action :set_cookie, only: :result # set cookie time for result page

```

```

  def first_part; end

```

```

  def second_part; end

```

```

  def exam; end

```

```

  def add_task

```

```

  end

```

```

  def task

```

```

    @current.update_answer(@number, @id_task)

```

```

    @user_answer = @current.decode_user[@number - 1]

```

```

    case @type_test

```

```

      when 'first'

```

```

        render 'task_first'

```

```

      when 'second'

```

```

        render 'task_second'

```

```

      end

```

```

  end

```

```

  def change_task

```

```

    @current.update_answer(@number, @id_task) unless @current.check(@number)

```

```

    render turbo_stream: turbo_stream.update('task-block', partial: 'frame')

```

```

  end

```

```

def downloader
  send_file(
    "#{Rails.root}/public/files/#{@file}",
    filename: "#{@number}#{@file[@file.index('.')..]}",
    type: "application/#{@file.index('.')}"
  )
end

```

```

def result
  @current.update(end: true)
  @balls = @current.decode.map { |_id_task, ball| ball }.reduce(:+)
end

```

private

```

def set_params
  @number = params[:number].nil? ? 1 : params[:number].to_i
  @number = 26 if @type_test == 'second'
  @current_number = params[:current_number].to_i
  @answer = params[:answer]
  @user_answer = @current.decode_user[@number - 1]
  @current.update_result(@current_number, @answer) if @answer # check answer
on task
  return unless check_current_task

```

```

  generate_task
end

```

```

def generate_task
  where = Task.where(number: @number)
  @task = where[rand(where.size)]
  @formulation = @task.formulation
  @path_image = @task.path_image
  @file = @task.file
  @id_task = @task.id
end

```

```

def check_current_task
  if @current.check(@number)
    @id_task = @current.check(@number)
    @task = Task.find(@id_task)
    @formulation = @task.formulation
    @path_image = @task.path_image
    @file = @task.file

```

```
    return false
  end
  true
end
```

```
def set_params_result
  @current_number = params[:current_number].to_i
  @answer = params[:answer]
  @current.update_result(@current_number, @answer) unless @answer.empty?
end
```

```
def set_params_test
  @type_test = params[:test]
  @current = if Test.find_by(user_id: session[:current_user_id], end: 0).nil?
    Test.create(user_id: session[:current_user_id],
                type_test: @type_test)
  else
    Test.find_by(user_id: session[:current_user_id], end: 0)
  end
end
```

```
def set_test_cookie
  if cookies[:login]
    case @type_test
    when 'exam'
      cookies[:login] = { value: session[:current_user_id], expires: (Time.now +
14_100 + 600) } # 10 minute 14_600
      cookies[:time] = { value: ActiveSupport::JSON.encode(Time.now.to_a[..-
4].reverse), expires: (Time.now + 236) } unless cookies[:time]
    when 'first'
      cookies[:login] = { value: session[:current_user_id], expires: (Time.now + 5400
+ 600) } # 1 hour, 30 min + 10 minute
      cookies[:time] = { value: ActiveSupport::JSON.encode(Time.now.to_a[..-
4].reverse), expires: (Time.now + 5400) } unless cookies[:time]
    when 'second'
      cookies[:login] = { value: session[:current_user_id], expires: (Time.now + 8700
+ 600) } # 2 hour, 25 min + 10 minute
      cookies[:time] = { value: ActiveSupport::JSON.encode(Time.now.to_a[..-
4].reverse), expires: (Time.now + 8700) } unless cookies[:time]
    end
  else
    session[:current_user_id] = nil
  end
end
end
```

Представления и „куски“ рендеринга

Partial _frame.html.erb (homepage controller)

```
<%= turbo_frame_tag 'frame' do %>
  <div id="task-tests">
    <li class="nav-item active task-part">
      <%= link_to 'Часть 1', '/task/first_part', class: 'nav-link link-info' %>
    </li>
    <li class="nav-item active task-part">
      <%= link_to 'Часть 2', '/task/second_part', class: 'nav-link link-info' %>
    </li>
    <li class="nav-item active task-part">
      <%= link_to 'Экзамен', '/task/exam', class: 'nav-link link-info', id: 'exam' %>
    </li>
    <li class="nav-item active task-part">
      <%= link_to '^', '/homepage/up', class: 'nav-link link-info link-up' %>
    </li>
  </div>
<% end %>
```

Partial _profile.html.erb (homepage controller)

```
<%= turbo_frame_tag 'profile' do %>
  <div id="profile">
    <li class="nav-item active task-part">
      <%= link_to 'Открыть профиль', '/profile/profile_page', class: 'nav-link link-
info', id: 'open-profile' %>
    </li>
    <li class="nav-item active task-part">
      <%= link_to 'Редактировать профиль', '/profile/edit', class: 'nav-link link-
info' %>
    </li>
    <li class="nav-item active task-part">
      <%= link_to 'Пройденные тесты', '/profile/tests_info', class: 'nav-link link-
info', id: 'tests-info' %>
    </li>
    <% if @role == 'superadmin' %>
      <li class="nav-item active task-part">
        <%= link_to 'Управление пользователями', '/profile/admin_page', class:
'nav-link link-info' %>
      </li>
      <li class="nav-item active task-part">
        <%= link_to 'Добавление заданий', '/profile/add_task', class: 'nav-link
link-info' %>
      </li>
    </li>
  </div>
<% end %>
```

```

    <% elsif @role == 'admin' %>
      <li class="nav-item active task-part">
        <%= link_to 'Добавление заданий', '/profile/add_task', class: 'nav-link
link-info' %>
      </li>
    <% end %>
  <li>
    <%= button_to 'Выйти', '/log/sign_out', method: :get, class: "btn btn-outline-
success", id: 'sign-out-button' %>
  </li>
  <li class="nav-item active task-part">
    <%= link_to '^', '/homepage/up', class: 'nav-link link-info link-up' %>
  </li>
</div>
<% end %>

```

Partial _up_frame.html.erb (homepage controller)

```

<%= turbo_frame_tag 'profile' do %>
  <div id="profile">
    <li class="nav-item active">
      <%= link_to 'Профиль', '/homepage/profile', class: 'nav-link link-info'%>
    </li>
  </div>
<% end %>

<%= turbo_frame_tag 'frame' do %>
  <div id="task-tests">
    <li class="nav-item active">
      <%= link_to 'Тесты по заданиям', '/homepage/edit', class: 'nav-link link-info'%>
    </li>
  </div>
<% end %>

```

index.html.erb (homepage controller)

```

<div class="container-md content"><h3>На сайте зарегистрировано: <%=
User.count - User.count_by_sql('Select count(*) from Users where role = "admin" or
role = "superadmin"') %></h3>
<table class="table">
  <tr><th>Рейтинг</th></tr>
  <tr><th>Место</th><th>Пользователь</th><th>Баллы</th></tr>
  <% count = 1 %>
  <% @users.each do |user, balls| %>
    <% unless balls.zero? %>

```



```

    <tr>
      <td><%= count %></td>
      <td><%= user.email %></td>
      <td><%= balls %></td>
    </tr>
    <% count += 1 %>
  <% end %>
<% end %>
</table>
</div>

```

application.html.erb

```

<!DOCTYPE html>
<html>
  <head>
    <title>WorkForLip</title>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <%= csrf_meta_tags %>
    <%= csp_meta_tag %>

    <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
    <%= javascript_importmap_tags %>
    <%= turbo_frame_tag "modal" %>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></
script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.mi
n.js"></script>
    <!-- CSS only -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F
9CUG65" crossorigin="anonymous">
    </head>

  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a href="/" class="nav-link link-info">В начало</a>
        </li>
        <%= turbo_frame_tag 'frame' do %>
          <div id="task-tests">
            <li class="nav-item active">

```

```

        <%= link_to 'Тесты по заданиям', '/homepage/edit', class: 'nav-link link-
info', id: 'tests-link-down'%>
    </li>
</div>
<% end %>
</ul>
<div class="log">
    <ul class="navbar-nav mr-auto" id="log">
        <% unless User.find_by_id(session[:current_user_id]).nil? %>
            <%= turbo_frame_tag 'profile' do %>
                <div id="profile">
                    <li class="nav-item active">
                        <%= link_to 'Профиль', '/homepage/profile', class: 'nav-link link-info',
id: 'profile-link-down' %>
                    </li>
                </div>
            <% end %>
        <% else %>
            <li class="nav-item active">
                <a href="/log/sign_in" class="nav-link link-info">Войти</a>
            </li>
            <li class="nav-item active">
                <a href="/log/sign_up" class="nav-link link-info">Зарегистрироваться</a>
            </li>
        <% end %>
    </ul>
</div>
</nav>
<main>
    <%= yield %>
    <script>
        $('input[type=text]').focus()
    </script>
</main>
<div class="footer">
    <div class="bmstu-link">
        <a href="https://e-learning.bmstu.ru/iu6/" class="nav-link link-info">BMSTU
IU6</a>
    </div>
</div>
</body>
</html>

```

Partial _messages.html.erb (log controller)

```
<% @msg.each do |message| %>
  <p><%= message %></p>
<% end %>
```

log.turbo_stream.erb (log controller)

```
<%= turbo_stream.update 'msg', partial: 'messages' %>
<% if @msg_s %>
  <%= turbo_stream.update 'msg-s', 'Successfully registration!' %>
<% end %>
```

sign_in.html.erb (log controller)

```
<div class="sign-in">
  <h1>Log In</h1>
  <%= form_with url: "/log/log", method: :post do |form| %>
    <div class="form-group">
      <%= form.label :email, "Email address:" %>
      <%= form.email_field :email, placeholder: "Enter email", class: "form-control"
    %>
    </div>
    <div class="form-group">
      <%= form.label :password, "Password:" %>
      <%= form.password_field :password, placeholder: 'password', class: "form-control" %>
    </div>
    <%= form.submit "Log In", class: "btn btn-success" %>
  <% end %>
  <%= button_to 'Homepage', root_path, method: :get, class: "btn btn-primary btn-sign-in" %>

  <div class="errors-class">
    <div id="msg"></div>
  </div>
</div>
```

sign_up.html.erb (log controller)

```
<div class="sign-out">
  <h1>Sign Up</h1>
  <%= form_with url: "/log/log", method: :post do |form| %>
    <div class="email">
      <%= form.label :email, "Email address:" %>
      <%= form.text_field :email, placeholder: "Enter email", class: 'form-control' %>
    </div>
    <div class="password">
      <%= form.label :password, "Password:" %>
```

```

    <%= form.password_field :password, placeholder: 'password', class: "form-
control" %>
  </div>
  <div class="password-confirmation">
    <%= form.label :password_confirmation, "Retype password:" %>
    <%= form.password_field :password_confirmation, placeholder: 'Password',
class: "form-control" %>
  </div>
  <%= form.submit "Sign Up", class: "btn btn-success" %>
<% end %>
<%= button_to 'Homepage', root_path, method: :get, class: "btn btn-primary btn-
sign-out" %>

<div class="errors-class">
  <div id="msg"></div>
</div>
<div id="msg-s"></div>
</div>

```

Partial _admin.html.erb (profile controller)

```

<%= turbo_frame_tag 'set_admin' do %>
  <% if @user.role == 'user' %>
    <a href="/profile/set_admin?user=<%= @user_id %>" class="btn btn-
primary">Назначить администратором</a>
  <% else %>
    <a href="/profile/set_admin?user=<%= @user_id %>" class="btn btn-
danger">Сделать обычным пользователем</a>
  <% end %>
<% end %>

```

Partial _full_info.html.erb (profile controller)

```

<%= turbo_frame_tag 'full' do %>
  <table class="table">
    <thead class="thead-light">
      <tr>
        <th colspan="4">Результаты</th>
      </tr>
      <tr>
        <td>
          Homep
        </td>
        <td>
          Ваш ответ
        </td>
        <td>

```

```

        Правильный ответ
      </td>
    </td>
    Первичный балл
  </td>
</tr>
</thead>
<% decode_results = @current.decode %>
<% answers = decode_results.map { |id_task, _ball| (id_task.zero?) ? nil :
Task.find(id_task).answer } %>
<% @current.decode_user.each_with_index do |user_answer, index| %>
  <tbody>
    <tr>
      <th scope="row">
        <%= index + 1 %>
      </th>
      <td>
        <%= user_answer%>
      </td>
      <td>
        <%= answers[index] %>
      </td>
      <td>
        <%= decode_results[index][1] %>
      </td>
    </tr>
  </tbody>
<% end %>
</table>
<a href="/profile/up?test=<%= @current.id %>" class='btn btn-primary'>^</a>
<% end %>

```

Partial _messages.html.erb (profile controller)

```

<% @msg.each do |message| %>
  <p><%= message %></p>
<% end %>

```

Partial _up.html.erb (profile controller)

```

<%= turbo_frame_tag 'full' do %>
  <a href="/profile/full_info?test=<%= @id %>" class="btn btn-
primary">Подробнее</a>
<% end %>

```

add_task.html.erb (profile controller)

```
<div class="container-sm content">
  <%= form_with url: "/profile/log", method: :post do |form| %>
    <div class="number">
      <%= form.label :number, "Number email:" %>
      <%= form.number_field :number, in: 1..27, placeholder: "number", class: 'form-
control' %>
    </div>
    <br>
    <div class="formulation">
      <%= form.label :formulation, "Formulation:" %>
      <%= form.text_area :formulation, placeholder: 'formulation', class: "form-
control" %>
    </div>
    <br>
    <div class="answer">
      <%= form.label :answer, "Answer:" %>
      <%= form.text_field :answer, placeholder: 'answer', class: "form-control" %>
    </div>
    <div class="type-test">
      <%= form.label :type_exam, "Type test" %><br>
      <%= form.select :type_exam, ['Examination', 'First part', 'Second Part'], class:
'form-control' %>
    </div>
    <br>
    <%= form.submit "Save", class: "btn btn-success" %>
  <% end %>
  <%= button_to 'Homepage', root_path, method: :get, class: "btn btn-primary btn-
edit" %>

  <div class="errors-class">
    <div id="msg"></div>
  </div>
  <div id="msg-s"></div>
</div>
```

admin_page.html.erb (profile controller)

```
<div class="container-md content">
  <% @users.each do |user| %>
    <div class="card">
      <div class="card-body">
        <h3 class="card-title">Email: <%= user.email %></h3>
        <p>Дата создания: <%= user.created_at %></p>
        <p>Дата обновления: <%= user.updated_at %></p>
        <p>Роль: <%= user.role %></p>
      </div>
    </div>
  </div>
```

```

    <p>Решено тестов: <%= Test.where(user_id: user.id).size %></p>
    <%= turbo_frame_tag 'set_admin' do %>
      <% if user.role == 'user' %>
        <a href="/profile/set_admin?user=<%= user.id %>" class="btn btn-
primary">Назначить администратором</a>
      <% else %>
        <a href="/profile/set_admin?user=<%= user.id %>" class="btn btn-
danger">Сделать обычным пользователем</a>
      <% end %>
    <% end %>
    <a href="/profile/destroy?user=<%= user.id %>" class="btn btn-
danger">Удалить пользователя</a>
  </div>
</div>
<br>
<% end %>
</div>

```

edit.html.erb (profile controller)

```

<div class="container-sm content">
  <%= form_with url: "/profile/log", method: :post do |form| %>
    <div class="email">
      <%= form.label :email, "Email address:" %>
      <%= form.text_field :email, placeholder: "Enter email", class: 'form-control' %>
    </div>
    <br>
    <div class="password">
      <%= form.label :password, "Password:" %>
      <%= form.password_field :password, placeholder: 'password', class: "form-
control" %>
    </div>
    <br>
    <div class="password-confirmation">
      <%= form.label :password_confirmation, "Retype password:" %>
      <%= form.password_field :password_confirmation, placeholder: 'password',
class: "form-control" %>
    </div>
    <br>
    <%= form.submit "Save", class: "btn btn-success" %>
  <% end %>
  <%= button_to 'Homepage', root_path, method: :get, class: "btn btn-primary btn-
edit" %>

  <div class="errors-class">
    <div id="msg"></div>
  </div>

```

```

</div>
<div id="msg-s"></div>
</div>

```

log.turbo_stream.erb (profile controller)

```

<%= turbo_stream.update 'msg', partial: 'messages' %>
<% if @msg_s %>
  <%= turbo_stream.update 'msg-s', 'Successfully update!' %>
<% end %>

```

profile_page.html.erb (profile controller)

```

<div class="container-md">
  <h1>
    Email:
  </h1>
  <p class="info-profile"> <%= @current.email %></p>
  <% if @current.role %>
    <h1>
      Роль:
    </h1>
    <p class="info-profile"> <%= @current.role %></p>
  <% end %>
  <h1>
    Дата регистрации:
  </h1>
  <p class="info-profile"> <%= @born %></p>
  <h1>
    Количество пройденных тестов:
  </h1>
  <p class="info-profile"> <%= @tests.size %> </p>
  <h1>
    Уровень: <%= @level[0] %>
  </h1>
  <br>
  <div class="row">
    <div class="col" id="level-left"><h1><%= @level[0] %></h1></div>
    <div class="col-10">
      <div class="progress" role="progressbar" aria-label="Basic example" aria-
valuenow="25" aria-valuemin="0" aria-valuemax="100">
        <div class="progress-bar" id="progress-bar" style="width: <%=
@procent %>%"></div>
      </div>
    </div>
    <div class="col" id="level-right"><h1><%= @level[0] + 1%></h1></div>
  </div>

```


</div>

tests_info.html.erb (profile controller)

```
<div class="container-md content">
  <% @tests.each_with_index do |test, index| %>
    <div class="card">
      <div class="card-body">
        <h5 class="card-title"><%= test.updated_at %></h5>
        <% if test.type_test == 'exam' %>
          <h5>Экзамен</h5>
        <% elsif test.type_test == 'first'%>
          <h5>Первая часть</h5>
        <% elsif test.type_test == 'second' %>
          <h5>Вторая часть</h5>
        <% end %>
        <p>Первичный балл: <%= test.result %></p>
        <%= turbo_frame_tag 'full' do %>
          <a href="/profile/full_info?test=<%= test.id %>" class="btn btn-
primary">Подробнее</a>
        <% end %>
      </div>
    </div>
  <br>
<% end %>
<% if @tests.empty? %>
  <h2>У вас ещё нет решённых тестов</h2>
<% end %>
</div>
```

Partial _frame.html.erb (task controller)

```
<h2 class="title-task">Задание <%= @number %></h2>
<pre class="task-info">
  <% if @path_image %>
    <%= image_tag @path_image, class: 'task-image'%>
  <% end %>
  <%= @formulation %>
</pre>
<% if @file %>
  <div class="file">
    <%= link_to "/task/downloader?number=#{@number}", class: 'file-downloader'
do %>
      <%= image_tag 'logo-file.png' %>
    <% end %>
    <%= "#{@number}#{@file[@file.index('.')..]}" %>
  </div>
```

```
<% end %>
<input type="text" name="answer" id="answer" class="answer-task", value="<%=
@user_answer %>">
<script>$('input[type=text]').focus()</script>
```

exam.html.erb (task controller)

```
<div class="container-task">
  <h2>Экзамен</h2>
  <p class="test-description">В ЕГЭ по информатике 2022 года — девять
  практико-ориентированных заданий, тогда как в прошлом году таких заданий
  было только восемь. В них проверяются навыки работы с прикладным
  программным обеспечением и навыки написания собственных программ для
  решения различных задач.
  В отличие от ЕГЭ прошлых лет, в нынешней структуре экзамена содержатся
  задания только с кратким ответом базового, повышенного и высокого уровней
  сложности. По сравнению с ЕГЭ 2021 года изменились задания 3, 9 и 17.
  </p>
  <p class="test-description">
    Длительность экзамена – 3 часа 55 минут (255 минут), полтора часа из
    которых рекомендовано уделить выполнению заданий первой части
    КИМов.<br>
    Задания в билетах разделены на блоки:
    <ul>
      <li class="default-font"><b>Часть 1</b> - 25 задания с кратким ответом.</li>
      <li class="default-font"><b>Часть 2</b> - 2 задачи с развернутым
    ответом.</li>
    </ul>
  </p>
  <b class="default-font">Всего: 35 баллов.</b><br>
  <br>
  <%= link_to 'Начать', '/task/task?test=exam', class: 'start-test', id: "start-test" %>
</div>
```

first_part.html.erb (task controller)

```
<div class="container-task">
  <h2>Первая часть</h2>
  <p class="test-description">Первая часть состоит из 25 простых заданий на
  комбинаторику, алгебру логики, базовые алгоритмические задачи, электронные
  таблицы и работу с текстовым редактором. </p>
  <%= link_to 'Начать', '/task/task?test=first', class: 'start-test' %>
</div>
```

result.html.erb (task controller)

```
<div class="container-task">
  <table class="table table-dark">
    <thead class="thead-light">
      <tr>
        <th colspan="4">Результаты</th>
      </tr>
      <tr>
        <td>
          Номер
        </td>
        <td>
          Ваш ответ
        </td>
        <td>
          Правильный ответ
        </td>
        <td>
          Первичный балл
        </td>
      </tr>
    </thead>
    <% decode_results = @current.decode %>
    <% answers = decode_results.map { |id_task, _ball| (id_task.zero?) ? nil :
Task.find(id_task).answer } %>
    <% @current.decode_user.each_with_index do |user_answer, index| %>
      <tbody>
        <tr>
          <th scope="row">
            <%= index + 1 %>
          </th>
          <td>
            <%= user_answer%>
          </td>
          <td>
            <%= answers[index] %>
          </td>
          <td>
            <%= decode_results[index][1] %>
          </td>
        </tr>
      </tbody>
    <% end %>
  </table>
  <% @current.update(result: @balls) %>
</div>
```

```

    <p id="p-result">Ваш первичный балл: <%= @balls %></p>
    <a href="/" class="btn btn-success">В начало</a>
</div>

```

second_part.html.erb (task controller)

```

<div class="container-task">
  <h2>Вторая часть</h2>
  <p class="test-description">Вторая часть это два последних задания с экзамена,
  то есть 26 и 27. Это считаются сложные задачи на алгоритмическое
  программирование, за эти задания ставятся по два первичных балла </p>
  <%= link_to 'Начать', '/task/task?test=second', class: 'start-test' %>
</div>

```

task_first.html.erb (task controller)

```

<div class="container-task">
  <div class="task">
    <div class="task-block" id="task-block">
      <h2 class="title-task">Задание <%= @number %></h2>
      <pre class="task-info">
        <% if @path_image %>
          <%= image_tag @path_image, class: 'task-image' %>
        <% end %>
        <%= @formulation %>
      </pre>
      <% if @file %>
        <div class="file">
          <%= link_to "/task/downloader?number=#{@number}", class: 'file-
downloader' do %>
            <%= image_tag 'logo-file.png' %>
          <% end %>
          <%= "#{@number}#{@file[@file.index('.')..]}" %>
        </div>
      <% end %>
      <input type="text" name="answer" id="answer" class="answer-task",
value="<%= @user_answer %>">
    </div>
    <div class="list-tasks">
      <div class="row-navigate">
        <li><a href="/task/change_task" class="small-number">1</a></li>
        <li><a href="/task/change_task" class="small-number">2</a></li>
        <li><a href="/task/change_task" class="small-number">3</a></li>
        <li><a href="/task/change_task" class="small-number">4</a></li>
        <li><a href="/task/change_task" class="small-number">5</a></li>
      </div>
      <div class="row-navigate">

```

```

        <li><a href="/task/change_task" class="small-number">6</a></li>
        <li><a href="/task/change_task" class="small-number">7</a></li>
        <li><a href="/task/change_task" class="small-number">8</a></li>
        <li><a href="/task/change_task" class="small-number">9</a></li>
        <li><a href="/task/change_task">10</a></li>
    </div>
    <div class="row-navigate">
        <li><a href="/task/change_task">11</a></li>
        <li><a href="/task/change_task">12</a></li>
        <li><a href="/task/change_task">13</a></li>
        <li><a href="/task/change_task">14</a></li>
        <li><a href="/task/change_task">15</a></li>
    </div>
    <div class="row-navigate">
        <li><a href="/task/change_task">16</a></li>
        <li><a href="/task/change_task">17</a></li>
        <li><a href="/task/change_task">18</a></li>
        <li><a href="/task/change_task">19</a></li>
        <li><a href="/task/change_task">20</a></li>
    </div>
    <div class="row-navigate">
        <li><a href="/task/change_task">21</a></li>
        <li><a href="/task/change_task">22</a></li>
        <li><a href="/task/change_task">23</a></li>
        <li><a href="/task/change_task">24</a></li>
        <li><a href="/task/change_task">25</a></li>
    </div>
    <%= link_to 'Закончить тест', '/task/result', class: 'btn btn-danger', id: 'result'
%>
    </div>
</div>
</div>

<style>
    .list-tasks {
        height: 310px !important;
    }
</style>

<script>
    $('<div class="row-navigate li">').click(function(){
        let answer = document.getElementById('answer').value
        let current_number = document.getElementsByClassName('title-task')
[0].textContent
        current_number = current_number.substr(current_number.indexOf(' ') + 1,

```

```

current_number.length - 1)
  href = this.childNodes[0]
  href.href = '/task/change_task?number=' + href.text
  href.href += '&current_number=' + current_number + '&answer=' + answer
  console.log(href)
})

$('.list-tasks a#result').click(function(){
  let answer = document.getElementById('answer').value
  let current_number = document.getElementsByClassName('title-task')
[0].textContent
  current_number = current_number.substr(current_number.indexOf(' ') + 1,
current_number.length - 1)
  href = this
  href.href = '/task/result?'
  href.href += 'current_number=' + current_number + '&answer=' + answer
  console.log(href)
})
</script>

```

task_second.html.erb (task controller)

```

<div class="container-task">
  <div class="task">
    <div class="task-block" id="task-block">
      <h2 class="title-task">Задание <%= @number %></h2>
      <pre class="task-info">
        <% if @path_image %>
          <%= image_tag @path_image, class: 'task-image'%>
        <% end %>
        <%= @formulation %>
      </pre>
      <% if @file %>
        <div class="file">
          <%= link_to "/task/downloader?number=#{@number}", class: 'file-
downloader' do %>
            <%= image_tag 'logo-file.png' %>
          <% end %>
          <%= "#{@number}#{@file[@file.index('.')..]}"%>
        </div>
      <% end %>
      <input type="text" name="answer" id="answer" class="answer-task",
value="<%= @user_answer %>">
    </div>
    <div class="list-tasks">
      <div class="row-navigate">

```

```

        <li><a href="/task/change_task">26</a></li>
        <li><a href="/task/change_task">27</a></li>
    </div>
    <%= link_to 'Закончить тест', '/task/result', class: 'btn btn-danger', id: 'result'
%>
    </div>
</div>
</div>

<style>
.list-tasks {
    height: 87px !important;
    width: 150px !important;
}
</style>

<script>
    $('<div>
        .row-navigate li').click(function(){
            let answer = document.getElementById('answer').value
            let current_number = document.getElementsByClassName('title-task')
[0].textContent
            current_number = current_number.substr(current_number.indexOf(' ') + 1,
current_number.length - 1)
            href = this.childNodes[0]
            href.href = '/task/change_task?number=' + href.text
            href.href += '&current_number=' + current_number + '&answer=' + answer
            console.log(href)
        })

        $('<div>
        .list-tasks a#result').click(function(){
            let answer = document.getElementById('answer').value
            let current_number = document.getElementsByClassName('title-task')
[0].textContent
            current_number = current_number.substr(current_number.indexOf(' ') + 1,
current_number.length - 1)
            href = this
            href.href = '/task/result?'
            href.href += 'current_number=' + current_number + '&answer=' + answer
            console.log(href)
        })
    </div>
    </script>

```

task.html.erb (task controller)

```
<div class="container-task">
  <div class="task">
    <div class="task-block" id="task-block">
      <h2 class="title-task">Задание <%= @number %></h2>
      <pre class="task-info">
        <% if @path_image %>
          <%= image_tag @path_image, class: 'task-image'%>
        <% end %>
        <%= @formulation %>
      </pre>
      <% if @file %>
        <div class="file">
          <%= link_to "/task/downloader?number=#{@number}", class: 'file-
downloader' do %>
            <%= image_tag 'logo-file.png' %>
          <% end %>
          <%= "#{@number}#{@file[@file.index('.')..]}" %>
        </div>
      <% end %>
      <input type="text" name="answer" id="answer" class="answer-task",
value="<%= @user_answer %>">
    </div>
    <div class="list-tasks">
      <div class="row-navigate">
        <li><a href="/task/change_task" class="small-number"
id="link-1">1</a></li>
        <li><a href="/task/change_task" class="small-number"
id="link-2">2</a></li>
        <li><a href="/task/change_task" class="small-number"
id="link-3">3</a></li>
        <li><a href="/task/change_task" class="small-number">4</a></li>
        <li><a href="/task/change_task" class="small-number">5</a></li>
      </div>
      <div class="row-navigate">
        <li><a href="/task/change_task" class="small-number">6</a></li>
        <li><a href="/task/change_task" class="small-number">7</a></li>
        <li><a href="/task/change_task" class="small-number">8</a></li>
        <li><a href="/task/change_task" class="small-number">9</a></li>
        <li><a href="/task/change_task">10</a></li>
      </div>
      <div class="row-navigate">
        <li><a href="/task/change_task">11</a></li>
        <li><a href="/task/change_task">12</a></li>
        <li><a href="/task/change_task">13</a></li>
      </div>
    </div>
  </div>
</div>
```



```
<li><a href="/task/change_task">14</a></li>
<li><a href="/task/change_task">15</a></li>
</div>
<div class="row-navigate">
  <li><a href="/task/change_task">16</a></li>
  <li><a href="/task/change_task">17</a></li>
  <li><a href="/task/change_task">18</a></li>
  <li><a href="/task/change_task">19</a></li>
  <li><a href="/task/change_task">20</a></li>
</div>
<div class="row-navigate">
  <li><a href="/task/change_task">21</a></li>
  <li><a href="/task/change_task">22</a></li>
  <li><a href="/task/change_task">23</a></li>
  <li><a href="/task/change_task">24</a></li>
  <li><a href="/task/change_task">25</a></li>
</div>
<div class="row-navigate">
  <li><a href="/task/change_task">26</a></li>
  <li><a href="/task/change_task">27</a></li>
</div>
<%= link_to 'Закончить тест', '/task/result', class: 'btn btn-danger', id: 'result'
%>

<% arr = ActiveSupport::JSON.decode(cookies[:time]) %>
<% t = 236 - (Time.now - Time.new(arr[1], arr[2], arr[3], arr[4], arr[5],
arr[6])) %>
  <div id="timer-time"> <%= "#{(t / 60).to_i}:#{(t % 60).to_i}" %> </div>
</div>
</div>
</div>

<script>
$( '.row-navigate li').click(function(){
  let answer = document.getElementById('answer').value
  let current_number = document.getElementsByClassName('title-task')
[0].textContent
  current_number = current_number.substr(current_number.indexOf(' ') + 1,
current_number.length - 1)
  href = this.childNodes[0]
  href.href = '/task/change_task?number=' + href.text
  href.href += '&current_number=' + current_number + '&answer=' + answer
  console.log(href)
})

$( '.list-tasks a#result').click(function(){
```

```

    let answer = document.getElementById('answer').value
    let current_number = document.getElementsByClassName('title-task')
[0].textContent
    current_number = current_number.substr(current_number.indexOf(' ') + 1,
current_number.length - 1)
    href = this
    href.href = '/task/result?'
    href.href += 'current_number=' + current_number + '&answer=' + answer
    console.log(href)
})

time = document.getElementById('timer-time')
let min = 0;
let set_min = time.innerHTML.substr(time.innerHTML.indexOf(':') + 1, ) - 1
let hours = time.innerHTML.substr(0, time.innerHTML.indexOf(':'))
let timer = setInterval(() => {
    if (hours == 0 && (set_min - min) < 0) {clearInterval(timer),
document.getElementById('result').click()};
    if (set_min - min < 0){
        time.innerHTML = --hours + ':59';
        min = 0;
        set_min = 58;
    }
    else {
        time.innerHTML = hours + ':' + ((set_min - min < 10) ? '0' + (set_min - min+
+) : (set_min - min++))
    }
}, 1000);
</script>

```

Схема базы данных:

```

ActiveRecord::Schema[7.0].define(version: 2022_12_22_023017) do
  create_table "tasks", force: :cascade do |t|
    t.integer "number", null: false
    t.string "formulation", default: "", null: false
    t.string "path_image"
    t.string "file"
    t.string "answer", null: false
    t.integer "part", default: 1, null: false
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
  end

  create_table "tests", force: :cascade do |t|
    t.string "type_test", default: "Exam", null: false

```

```

t.integer "user_id"
t.string "user_answer", default:
"[null,null,null,null,null,null,null,null,null,null,null,null,null,null,nu
ll,null,null,null,null,null,null,null,null]", null: false
t.string "answer", default: "[[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],
[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],
[0,0],[0,0]]", null: false
t.integer "end", default: 0
t.integer "result", default: 0
t.datetime "created_at", null: false
t.datetime "updated_at", null: false
t.index ["user_id"], name: "index_tests_on_user_id"
end

create_table "users", force: :cascade do |t|
  t.string "email", null: false
  t.string "password_digest", null: false
  t.string "role"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["email"], name: "index_users_on_email", unique: true
end
end

```

Модели:

task.rb

```

class Task < ApplicationRecord
  validates :number, {
    numericality: {
      greater_than: 0,
      less_than_or_equal_to: 27,
      message: 'Number task must be in range 1-27'
    }
  }
  validates :answer, {
    presence: {
      message: 'Answer can\'t be empty'
    }
  }
end

```

test.rb

```
class Test < ApplicationRecord
  belongs_to :user

  def decode
    ActiveSupport::JSON.decode(self.answer)
  end

  def decode_user
    ActiveSupport::JSON.decode(self.user_answer)
  end

  def update_answer(number, id_task, result=0)
    arr = decode
    arr[number - 1] = [id_task, result]
    update(answer: encode(arr))
  end

  def check(number) # false when task is not update
    task = decode[number - 1]
    (task[0].zero?) ? false : task[0]
  end

  def update_result(number, value)
    arr = decode
    task = Task.find(arr[number - 1][0])
    answers = decode_user
    answers[number - 1] = value
    arr[number - 1][1] = (task.answer == value) ? 1 : 0
    update(answer: encode(arr))
    update(user_answer: encode(answers))
  end

  private

  def encode(array)
    ActiveSupport::JSON.encode(array)
  end
end
```

user.rb

```
class User < ApplicationRecord
  has_secure_password
  has_many :tests, dependent: :destroy, autosave: true

  validates :email, {
    presence: {
      message: "Email can't be empty"
    },
    uniqueness: {
      message: "This email address is already registered"
    },
    format: {
      with: /^(?:[a-z0-9!#$%&'*\+=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*\+=?^_`{|}~-]+)*|(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*)@(?:\.(?:[a-z0-9](?:[a-z0-9]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9]*[a-z0-9])?|(?:(?:2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9]))\.){3}(?:2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9])[a-z0-9]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\)/,
      message: "This incorrect email address"
    }
  }

  validates :password, {
    length: {
      minimum: 8,
      message: "Length password - min 8 symbols"
    }
  }
end
```

Файлы миграций:

2022121601_create_users.rb

```
class CreateUsers < ActiveRecord::Migration[7.0]
  def change
    create_table :users do |t|
      t.string :email, null: false
      t.string :password_digest, null: false
      t.string :role

      t.timestamps
    end
    add_index :users, :email, unique: true
  end
end
```

end

20221222014156_create_tasks.rb

```
class CreateTasks < ActiveRecord::Migration[7.0]
```

```
def change
```

```
create_table :tasks do |t|
```

```
t.integer :number, null: false
```

t.string :formulation, null: false, default: "

```
t.string :path_image
```

```
t.string :file
```

```
t.string :answer, null: false
```

t.integer :part, null: false, default: 1

t.timestamps

end

end

end

20221222023017 create tests.rb

```
class CreateTests < ActiveRecord::Migration[7.0]
```

```
def change
```

```
create table :tests do |t|
```

```
t.string :type test, null: false, default: 'Exam'
```

```
t.belongs_to :user
```

```
t.string :user_answer, null: false, default:
```

[illegible][illegible]

```
t.integer :end, default: 0
```

```
t.integer :result, default: 0
```

t.timestamps

end

end

end

Стили CSS:

application.css

```
.footer {
```

height: 50px;

width: 100%;

```
background-color: #282829;
```

bottom: 0;

```
    left: 0;
    position: fixed;
}
```

```
.bmstu-link {
    margin: 15px auto 0 auto;
    display: flex;
    justify-content: center;
}
```

```
.bmstu-link > a {
    color: white;
}
```

```
#log {
    position: absolute;
    right: 0;
    top: 8px;
}
```

```
.sign-in {
    max-width: 400px;
    margin: 185px auto;
    border: black solid 1px;
    padding: 70px;
}
```

```
.sign-in input, .sign-out input {
    margin-bottom: 10px;
}
```

```
#msg {
    color: red;
    font-size: 24px;
}
```

```
.task-part {
    padding: 3px;
}
```

```
.task-part:hover {
    background-color: rgb(129, 114, 114);
    border-radius: 10px;
}
```

```
.sign-out {
  max-width: 400px;
  margin: 185px auto;
  border: black solid 1px;
  padding: 70px;
}

#msg-s {
  color: green;
  font-size: 24px;
}

#task-tests {
  position: absolute;
  background-color: #212529;
  width: 200px;
  border-radius: 10px;
}

/* For task.html.erb */

.container-task {
  max-width: 1650px;
  background-color: rgb(174, 209, 239);
  margin: 30px auto;
  padding: 40px;
}

.task {
  display: flex;
}

.task-block{
  width: 1200px;
  border: solid black 1px;
  padding: 10px;
  margin-right: auto;
}

.list-tasks {
  width: 360px;
  border: solid black 1px;
  padding: 5px;
  height: 357px;
}
```



```
.list-tasks li {  
    margin: 0 10px;  
    list-style: none;  
    display: inline-block;  
}  
  
.list-tasks a {  
    text-decoration: none;  
    color: black;  
    border: solid black 1px;  
    padding: 12px;  
    background-color: rgb(204, 212, 220);  
    font-size: 13px;  
}  
  
.row-navigate {  
    margin: 32px 0;  
}  
  
.small-number {  
    padding: 12px 16px !important;  
}  
  
.task-info {  
    white-space: pre-wrap;  
    font-size: 18px;  
}  
  
.answer-task {  
    width: 400px;  
    height: 35px;  
    margin: 0 auto;  
    display: block;  
}  
  
.title-task {  
    text-align: center;  
}  
  
.list-tasks a:hover {  
    background-color: rgb(179, 200, 221);  
}  
  
.list-tasks li:hover {
```

```
    transform: scale(1.2);
}

.task-image {
    height: 550px;
    width: 1150px;
    display: block;
}

.file-downloader {
    text-decoration: none;
}

.file {
    display: flex;
    float: left;
    font-size: 20px;
    align-items: center;
}

.file a {
    margin-right: 5px;
}

#timer-time {
    border: 1px solid black;
    width: 65px;
    font-size: 24px;
    float: right;
    padding: 3px 10px;
}

/* For exam.html.erb */
.test-description {
    font-size: 24px;
}

.start-test {
    text-decoration: none;
    border: black solid 1px;
    padding: 7px 17px;
    background-color: rgb(241, 201, 201);
}

.start-test:hover {
```

```
padding: 10px 20px;
box-shadow: 0 0 15px rgb(129, 114, 114);
}
```

```
.default-font {
  font-size: 20px;
}
```

```
/* For _profile.html.erb */
#profile {
  background-color: #212529;
  width: 130px;
  border-radius: 10px;
}
```

```
#sign-out-button {
  width: 130px;
}
```

```
.link-up {
  text-align: center;
}
```

```
/* For profile_page.html.erb */
.info-profile {
  font-size: 22px;
}
```

```
#progress-bar {
  background-color: #B6137D;
}
```

```
/* For edit.html.erb */
.content {
  margin: 40px 0;
  max-width: 700px;
}
```

```
.btn-edit {
  margin-top: 10px;
}
```

```
/* For add_task.html.erb */
```

```
#formulation {
```

```

height: 300px;
}

#type_exam {
  height: 40px;
  width: 180px;
}

```

Проверка на Rubocop:

```

marat@marat-IdeaPad-5-Pro-14ACN6:~/labs/work_for_lip$ rubocop app/controllers/
Inspecting 5 files
...CC

Offenses:

app/controllers/profile_controller.rb:89:3: C: Metrics/MethodLength: Method has too many lines. [12/10]
  def get_level(ball) ...
  ~~~~~
app/controllers/task_controller.rb:4:1: C: Metrics/ClassLength: Class has too many lines. [108/100]
class TaskController < ApplicationController ...
~~~~~
app/controllers/task_controller.rb:50:3: C: Metrics/AbcSize: Assignment Branch Condition size for set_params is too high. [<5, 17, 5> 18.41/17]
  ]
  def set_params ...
  ~~~~~
app/controllers/task_controller.rb:99:3: C: Metrics/AbcSize: Assignment Branch Condition size for set_test_cookie is too high. [<7, 58, 8> 58.97/17]
  def set_test_cookie ...
  ~~~~~
app/controllers/task_controller.rb:99:3: C: Metrics/CyclomaticComplexity: Cyclomatic complexity for set_test_cookie is too high. [8/7]
  def set_test_cookie ...
  ~~~~~
app/controllers/task_controller.rb:99:3: C: Metrics/MethodLength: Method has too many lines. [27/10]
  def set_test_cookie ...
  ~~~~~
app/controllers/task_controller.rb:110:121: C: Layout/LineLength: Line is too long. [125/120]
  cookies[:login] = { value: session[:current_user_id], expires: (Time.now + 5400 + 600) } # 1 hour, 30 min + 10 minute
  ~~~~~
app/controllers/task_controller.rb:117:121: C: Layout/LineLength: Line is too long. [125/120]
  cookies[:login] = { value: session[:current_user_id], expires: (Time.now + 8700 + 600) } # 2 hour, 25 min + 10 minute
  ~~~~~

5 files inspected, 8 offenses detected

```

Стоит отметить, что исправление этих ошибок не целесообразно и повлечёт за собой ухудшение читаемости кода и целостности программы.

Тесты:

task_spec.rb (model task)

```
require 'rails_helper'
```

```

RSpec.describe Task, type: :model do
  describe 'If adding with similar params' do
    it 'should be bad message' do
      u = Task.new(number: 25)
      expect(u.valid?).to be false
      expect(u.errors.objects.map(&:message)).to eq(["Answer can\\t be empty"])
    end

    it 'Correct params for add task' do
      u = Task.new(number: 25, answer: '123')
      expect(u.valid?).to be true
      expect(u.errors.objects.map(&:message)).to eq([])
    end
  end
end

```

```
end
end
```

user_spec.rb (model user)

```
require 'rails_helper'
```

```
RSpec.describe User, type: :model do
  describe 'If adding with similar params' do
    before do
      User.create!(email: 'user@mail.ru', password: 'password', password_confirmation:
'password') if User.find_by(email: 'user@mail.ru').nil?
    end

    it 'should be error if input is not unique' do
      u = User.new(email: 'user@mail.ru', password: 'password',
password_confirmation: 'password')
      expect(u.valid?).to be false
      expect(u.errors.objects.map(&:message)).to eq(["This email address is already
registered"])
    end
  end

  describe 'If adding correct params' do
    before do
      User.destroy_by(email: 'user@mail.ru')
    end
    it 'Input correct values' do
      u = User.new(email: 'user@mail.ru', password: 'password',
password_confirmation: 'password')
      expect(u.valid?).to be true
      expect(u.errors.objects.map(&:message)).to eq([])
    end
  end
end
```

homepage_spec.rb (request test for homepage controller)

```
require 'rails_helper'
```

```
RSpec.describe "Homepages", type: :request do
  describe "GET /index" do
    it "returns http success" do
      get root_path
      expect(response).to have_http_status(:success)
    end
  end
end
```

```

describe 'GET edit' do
  it 'returns 302 code' do
    get '/homepage/edit'
    expect(response).to have_http_status(302)
  end

  it 'returns text' do
    get '/homepage/edit'
    expect(response.body).to eq("<html><body>You are being <a  
href=\"http://www.example.com/log/sign_in\">redirected</a>.</body></html>")
  end
end
end

```

log_spec.rb (request test for log controller)

```
require 'rails_helper'
```

```

RSpec.describe "Logs", type: :request do
  describe "POST /log" do
    before do
      User.destroy_by(email: 'user@mail.ru') if User.find_by(email: 'user@mail.ru')
    end
    it "returns http success" do
      post "/log/log", xhr: true
      expect(response).to have_http_status(:success)
    end
  end

```

```

    it 'returns successfully registration' do
      post "/log/log/", params: {commit: 'Sign Up', email: 'user@mail.ru', password:
'password', password_confirmation: 'password'}, xhr: true
      expect(response.body).to eq("<turbo-stream action=\"update\"  
target=\"msg\"><template></template></turbo-stream>\n  <turbo-stream  
action=\"update\" target=\"msg-s\"><template>Successfully  
registration!</template></turbo-stream>\n")
    end
  end
end

```

```

describe "GET /sign_in" do
  it "returns http success" do
    get "/log/sign_in"
    expect(response).to have_http_status(:success)
  end
end
end

```

```
describe "GET /sign_out" do
  it "returns http success" do
    get "/log/sign_out"
    expect(response).to have_http_status(302)
  end
end
```

```
describe "GET /sign_up" do
  it "returns http success" do
    get "/log/sign_up"
    expect(response).to have_http_status(:success)
  end
end
```

```
describe 'controller tests' do
  context 'controller tests for sign up' do
    before do
      User.destroy_by(email: 'user@mail.ru') if User.find_by(email: 'user@mail.ru')
    end
    it 'test @msg_s' do
      post "/log/log", params: {commit: 'Sign Up', email: 'user@mail.ru', password:
'password', password_confirmation: 'password'}, xhr: true
      expect(assigns(:msg_s)).to be true
    end
```

```
    it 'test @msg_s' do
      post '/log/log', params: {commit: 'Sign Up', email: 'user@mail.ru', password:
'password', password_confirmation: 'bad'}, xhr: true
      expect(assigns(:msg_s)).to be false
    end
```

```
    it 'test @msg' do
      post "/log/log", params: {commit: 'Sign Up', email: 'user@mail.ru', password:
'password', password_confirmation: 'password'}, xhr: true
      expect(assigns(:msg)).to eq([])
    end
```

```
    it 'test @msg' do
      post '/log/log', params: {commit: 'Sign Up', email: 'user@mail.ru', password:
'password', password_confirmation: 'bad'}, xhr: true
      expect(assigns(:msg)).to eq(["Password confirmation doesn't match Password"])
    end
```

```
    it 'test @msg' do
      post '/log/log', params: {commit: 'Sign Up', email: 'user', password: 'password',
```

```

password_confirmation: 'password'}, xhr: true
  expect(assigns(:msg)).to eq(["Email This incorrect email address"])
end

  it 'test @msg' do
    post '/log/log', params: {commit: 'Sign Up', email: 'user@mail.ru', password:
'pass', password_confirmation: 'pass'}, xhr: true
    expect(assigns(:msg)).to eq(["Password Length password - min 8 symbols"])
  end

  it 'test @msg' do
    post '/log/log', params: {commit: 'Sign Up', email: 'user', password: 'pass',
password_confirmation: 'passe'}, xhr: true
    expect(assigns(:msg)).to eq(["Password confirmation doesn't match Password",
"Email This incorrect email address", "Password Length password - min 8 symbols"])
  end
end

context 'controller tests for sign in' do
  before do
    User.create(email: 'user@mail.ru', password: 'password',
password_confirmation: 'password')
  end
  it 'test @msg' do
    post '/log/log', params: {commit: 'Log In', email: 'user@mail.ru', password:
'password'}, xhr: true
    expect(assigns(:msg)).to eq([])
  end

  it 'test @msg' do
    post '/log/log', params: {commit: 'Log In', email: 'user@mail.ru', password:
'pass'}, xhr: true
    expect(assigns(:msg)).to eq(["Incorrect password!"])
  end
end
end
end
end

```

profile_spec.rb (request test for profile controller)

```

require 'rails_helper'

RSpec.describe "Profiles", type: :request do
  describe "GET /profile_page" do
    it "returns http success" do
      get "/profile/profile_page"
    end
  end
end

```



```

    expect(response).to have_http_status(302)
  end
end

describe "GET /admin_page" do
  it "returns http success" do
    get "/profile/admin_page"
    expect(response).to have_http_status(302)
  end
end

describe 'Redirect without session activate' do
  it 'admin_page' do
    get '/profile/admin_page'
    expect(response).to redirect_to('/log/sign_in')
  end

  it 'edit' do
    get '/profile/edit'
    expect(response).to redirect_to('/log/sign_in')
  end

  it 'profile_page' do
    get '/profile/profile_page'
    expect(response).to redirect_to('/log/sign_in')
  end

  it 'tests_info' do
    get '/profile/tests_info'
    expect(response).to redirect_to('/log/sign_in')
  end
end
end
end

```

task_spec.rb (request test for task controller)

```

require 'rails_helper'

RSpec.describe "Tasks", type: :request do
  describe "GET /first_part" do
    it "returns http success" do
      get "/task/first_part"
      expect(response).to have_http_status(302)
    end
  end
end

```

```
describe "GET /second_part" do
  it "returns http success" do
    get "/task/second_part"
    expect(response).to have_http_status(302)
  end
end
```

```
describe "GET /exam" do
  it "returns http success" do
    get "/task/exam"
    expect(response).to have_http_status(302)
  end
end
```

```
describe 'Redirect without session activate' do
  it 'task' do
    get '/task/task', params: { number: 1 }
    expect(response).to redirect_to('/log/sign_in')
  end
end
```

```
  it 'exam' do
    get '/task/exam'
    expect(response).to redirect_to('/log/sign_in')
  end
end
```

```
  it 'first_part' do
    get '/task/first_part'
    expect(response).to redirect_to('/log/sign_in')
  end
end
```

```
  it 'second_part' do
    get '/task/second_part'
    expect(response).to redirect_to('/log/sign_in')
  end
end
```

```
  it 'result' do
    get '/task/result', params: { answer: "" }
    expect(response).to redirect_to('/log/sign_in')
  end
end
end
```

log_spec.rb (system test for log controller)

```
require 'rails_helper'

RSpec.describe 'Static content', type: :system do
  let(:email_tester) { 'tester@mail.ru' }
  let(:password_tester) { 'password' }

  scenario 'Sign up, Log in, Sign out' do
    visit '/log/sign_up'
    fill_in :email, with: email_tester
    fill_in :password, with: password_tester
    fill_in :password_confirmation, with: password_tester
    find('.btn-success').click
    sleep(1)
    expect(current_path).to eq('/log/sign_up')

    visit '/log/sign_in'
    sleep(1)
    fill_in :email, with: email_tester
    fill_in :password, with: password_tester
    find('.btn-success').click
    sleep(1)
    expect(current_path).to eq(root_path)

    visit root_path
    click_link('profile-link-down')
    click_button('sign-out-button')
    expect(current_path).to eq(root_path)
  end
end
```

profile_spec.rb (system test for profile controller)

```
require 'rails_helper'

RSpec.describe 'Static content', type: :system do
  let(:email_tester) { 'tester@mail.ru' }
  let(:password_tester) { 'password' }
  let(:admin_mail) { 'admin@admin.ru' }

  Task.create(number: 1, path_image: '1-1.png', answer: '41')
  Task.create(number: 2, path_image: '2-1.png', answer: 'yxwz')
  Task.create(number: 3, path_image: '3-1.png', answer: '88479', file: '3-1.xls')

  scenario 'Examination' do
    visit '/log/sign_up'
```

```
fill_in :email, with: email_tester
fill_in :password, with: password_tester
fill_in :password_confirmation, with: password_tester
find('.btn-success').click
sleep(0.5)
expect(current_path).to eq('/log/sign_up')
```

```
visit '/log/sign_in'
sleep(0.5)
fill_in :email, with: email_tester
fill_in :password, with: password_tester
find('.btn-success').click
sleep(0.5)
expect(current_path).to eq(root_path)
```

```
click_link('tests-link-down')
click_link('exam')
sleep(0.5)
expect(current_path).to eq('/task/exam')
```

```
click_link('start-test')
sleep(0.2)
fill_in :answer, with: '41'
find("a", class: 'small-number', text: "2").click
sleep(0.5)
fill_in :answer, with: 'yxwz'
sleep(0.7)
visit '/task/result?current_number=2&answer=yxwz'
expect(find('#p-result')).to have_text('Ваш первичный балл: 2')
visit root_path
click_link('profile-link-down')
sleep(0.2)
click_link('tests-info')
sleep(0.2)
expect(current_path).to eq('/profile/tests_info')
find('.btn-primary').click
expect(find('#full')).to have_text("Результаты\nНомер Ваш ответ Правильный
ответ Первичный балл\n1 41 41 1\n2 yxwz yxwz 1\n3 0\n4 0\n5 0\n6 0\n7 0\n8 0\
n9 0\n10 0\n11 0\n12 0\n13 0\n14 0\n15 0\n16 0\n17 0\n18 0\n19 0\n20 0\n21 0\n22
0\n23 0\n24 0\n25 0\n26 0\n27 0\n^")
sleep(0.2)
visit root_path
click_link('profile-link-down')
sleep(0.2)
find('#open-profile').click
```

```

    sleep(0.2)
    expect(find('.container-md')).to have_text("Email:\ntester@mail.ru\nРоль:\nuser\nДата регистрации:\n28.12.2022\nКоличество пройденных тестов:\n1\nУровень: 1\n\n1\n2")
  end

```

```

scenario 'Examination' do
  visit '/log/sign_up'
  fill_in :email, with: admin_mail
  fill_in :password, with: password_tester
  fill_in :password_confirmation, with: password_tester
  find('.btn-success').click
  sleep(0.5)
  expect(current_path).to eq('/log/sign_up')

```

```

  User.find(1).update_column(:role, 'admin')

```

```

  visit '/log/sign_in'
  sleep(0.5)
  fill_in :email, with: admin_mail
  fill_in :password, with: password_tester
  find('.btn-success').click
  sleep(0.5)
  expect(current_path).to eq(root_path)

```

```

  visit root_path
  click_link('profile-link-down')
  sleep(0.2)
  find('#open-profile').click
  sleep(0.2)
  expect(find('.container-md')).to have_text("Email:\nadmin@admin.ru\nРоль:\nadmin\nДата регистрации:\n28.12.2022\nКоличество пройденных тестов:\n0\nУровень: 1\n\n1\n2")
  end
end

```

task_spec.rb (system test for task controller)

```

require 'rails_helper'

```

```

RSpec.describe 'Static content', type: :system do
  let(:email_tester) { 'tester@mail.ru' }
  let(:password_tester) { 'password' }

```

```

  Task.create(number: 1, path_image: '1-1.png', answer: '41')
  Task.create(number: 2, path_image: '2-1.png', answer: 'yxwz')

```

```
Task.create(number: 3, path_image: '3-1.png', answer: '88479', file: '3-1.xls')
```

```
scenario 'Examination' do
  visit '/log/sign_up'
  fill_in :email, with: email_tester
  fill_in :password, with: password_tester
  fill_in :password_confirmation, with: password_tester
  find('.btn-success').click
  sleep(0.5)
  expect(current_path).to eq('/log/sign_up')
```

```
  visit '/log/sign_in'
  sleep(0.5)
  fill_in :email, with: email_tester
  fill_in :password, with: password_tester
  find('.btn-success').click
  sleep(0.5)
  expect(current_path).to eq(root_path)
```

```
  click_link('tests-link-down')
  click_link('exam')
  sleep(0.5)
  expect(current_path).to eq('/task/exam')
```

```
  click_link('start-test')
  sleep(0.2)
  fill_in :answer, with: '41'
  find("a", class: 'small-number', text: "2").click
  sleep(0.5)
  fill_in :answer, with: 'yxwz'
  sleep(0.7)
  visit '/task/result?current_number=2&answer=yxwz'
  expect(find('#p-result')).to have_text('Ваш первичный балл: 2')
```

```
end
```

```
end
```

Результаты тестов

```
marat@marat-IdeaPad-5-Pro-14ACN6:~/labs/work_for_lip$ bundle exec rspec spec/
**** * .....*****
```

Вывод: у меня получилось реализовать проект, который я и планировал.