



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 09.03.03 Прикладная информатика

**О Т Ч Е Т**

по зачетной работе

Тема: Список задач и запланированных дел

Дисциплина: Языки интернет-программирования

Студент

ИУ6-35 Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Дулина И.А.

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Маняшев Э.Р.

(И.О. Фамилия)

Москва, 2023

### **Требования:**

- 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).

**Цель:** написать приложение, в котором авторизованные пользователи смогут создавать свой ежедневник, заполнять его своими задачами

### **Задачи:**

- Создать страницу с выводом всех записанных задач в зависимости от того, какой пользователь совершил вход
- Спроектировать БД для пользователей и задач, которая будет содержать в себе внешний ключ на id пользователя
- Реализовать авторизацию
- Добавить возможность добавлять дату, время, важность, а также процесс выполнения (сделано или нет) для каждой задачи
- Добавить возможность удалять задачу
- Добавить динамический вывод окна с изменением записи
- Добавить динамический вывод подробной информации о каждой записи
- Добавить динамический вывод модульного окна, где можно вписывать свои мысли и размышления о прошедшем дне
- Добавить вывод самых важных задач на другой части страницы
- Реализовать интернационализацию приложения
- Написать интеграционные тесты и тесты в браузере при помощи Katalon Recorder / Selenium Webdriver.

### Создание приложения:

```
$rails new diary_task
$cd diary_task
$ rails g controller Sessions --no-test-framework
$rails g controller Main menu settings
$rails g scaffold User name:string email:string:uniq password_digest:string
remember_token:string:index admin:Boolean
$ rails g scaffold Note User:references day_date:date time:string task:string
weekday:string colour:string diary:text info:text import:boolean if_done:boolean
$ rake db:create
$rake db:migrate
$rake db:migrate RAILS_ENV=test
```

### Зависимости приложения:

**Gemfile** добавить

```
gem "jquery-rails"
gem "bootstrap"
gem "sassc-rails"
gem 'momentjs-rails'
gem 'bootstrap-datetimepicker-rails'
gem 'i18n'
#для selenium
gem 'rspec-rails'
```

### Сценарии javascript:

```
app/javascript/application.js
import "@hotwired/turbo-rails"
import "controllers"
import "jquery"
import "jquery_ujs"
import "popper"
```

```

import "bootstrap"
import "src/main"

//= require moment
//= require bootstrap-datetimepicker
//= require pickers

app/javascript/src/main.js

$(document).on('click', '.btn-close', function () {
    $('.alert').fadeOut();
});

//ФУНКЦИЯ ВЫВОДА СБОКУ
function show_result(data) {
    // document.getElementById("right").value="";
    console.log(data);
    const result = document.getElementById("right");
    result.innerHTML = "<div >" + data + "</div>";
    // Создать кнопку
    let newButton = document.createElement('button');
    newButton.classList.add('btn', 'btn-primary');
    newButton.textContent = Hide_Button;
    newButton.addEventListener('click', function () {
        document.getElementById('right').innerHTML = "";
        // Очищаем содержимое целевого элемента
    });
    // Добавить кнопку на страницу
    document.getElementById("right").appendChild(newButton);
}

$(document).ajaxSuccess(function (event, request, options, data) {
    console.log(options)

```

```

let form_url = locale+options.url
console.log('main_right_form success', data, 'form url', form_url);

if (form_url.includes('main/adding/')) {
  show_result(data.value);
}

});

```

### **Таблицы стилей:**

#### **app/assets/stylesheets/application.scss**

```

@import "bootstrap";
@import "bootstrap-datetimepicker";

```

### **Другие настройки:**

#### **config/initializers/assets.rb**

```

Rails.application.config.assets.version = "1.0"
Rails.application.config.assets.precompile +=
  %w( jquery.min.js jquery_ujs.js bootstrap.min.js popper.js)

```

#### **config/importmap.rb**

```

pin_all_from 'app/javascript/src', under: 'src'
pin "jquery", to: "jquery.min.js", preload: true
pin "jquery_ujs", to: "jquery_ujs.js", preload: true
pin "popper", to: "popper.js", preload: true
pin "bootstrap", to: "bootstrap.min.js", preload: true

```

### **Модели:**

#### **app/models/user.rb**

```

class User < ApplicationRecord
  before_save { self.email = email.downcase }
  before_create :create_remember_token

```

```

VALID_EMAIL_REGEX = /\A[\w+\-\.]+\@[a-z\d\-\.\.][a-z]+\z/i

```

```

has_secure_password
validates :email, presence: true,
          format: { with: VALID_EMAIL_REGEX },
          uniqueness: { case_sensitive: false }
validates :password, length: { minimum: 4 }

```

```

def User.new_remember_token
  SecureRandom.urlsafe_base64
end

```

```

def User.encrypt(token)
  Digest::SHA1.hexdigest(token.to_s)
end

```

```

private

```

```

def create_remember_token
  self.remember_token = User.encrypt(User.new_remember_token)
end
end

```

#### **app/models/note.rb**

```

class Note < ApplicationRecord
  belongs_to :User
end

```

#### **Миграции:**

#### **db/migrate/20231130195519\_create\_users.rb**

```

class CreateUsers < ActiveRecord::Migration[7.1]
  def change
    create_table :users do |t|
      t.string :name

```

```

      t.string :email
      t.string :password_digest
      t.string :remember_token

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

```

#### **db/migrate/20231130203343\_create\_notes.rb**

```

class CreateNotes < ActiveRecord::Migration[7.1]
  def change
    create_table :notes do |t|
      t.references :User, null: false, foreign_key: true
      t.date :day_date
      t.string :time
      t.string :task
      t.string :week_day
      t.string :colour

      t.timestamps
    end
  end
end

```

#### **db/migrate/20231130213255\_add\_admin\_to\_users.rb**

```

class AddAdminToUsers < ActiveRecord::Migration[7.1]
  def change
    add_column :users, :admin, :boolean
  end
end

```

end

**db/migrate/20231209094310\_add\_diary\_to\_notes.rb**

class AddDiaryToNotes < ActiveRecord::Migration[7.1]

def change

add\_column :notes, :diary, :text

add\_column :notes, :add\_info\_to\_notes, :string

add\_column :notes, :info, :text

end

end

**db/migrate/20231209094536\_remove\_add\_info\_to\_notes\_from\_notes.rb**

class RemoveAddInfoToNotesFromNotes < ActiveRecord::Migration[7.1]

def change

remove\_column :notes, :add\_info\_to\_notes, :text

end

end

**db/migrate/20231216091406\_add\_import\_to\_notes.rb**

class AddImportToNotes < ActiveRecord::Migration[7.1]

def change

add\_column :notes, :import, :boolean

end

end

**db/migrate/20231216101638\_add\_if\_done\_to\_notes.rb**

class AddIfDoneToNotes < ActiveRecord::Migration[7.1]

def change

add\_column :notes, :if\_done, :boolean

end

end

**Представления:**

**app/views/layouts/\_each\_diary.html.erb**



```

<!--модульное окно-->
<div class="modal fade" id="dModal<%= @note.id %>" tabindex="-1"
    aria-labelledby="dModalLabel<%= @note.id %>" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="dModalLabel<%= @note.id
%>"><%=t("DIARY") %></h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"
                    aria-label="<%=t("Close")%>"></button>
            </div>
            <!--тело модульного окна-->
            <div class="modal-body">
                <div class="row">
                    <div class="col">
                        <form action="/main/diary/<%= @note.id %>" accept-charset="UTF-8"
method="get">
                            <input type="hidden" name="_method" value="patch"
autocomplete="off">
                            <div>
                                <label class="form-check" for="note_дневник"><%=t("Мои мысли")
%></label>
                                <textarea class="form-control" style="min-height:300px"
name="note[diary]" id="note_diary"><%= @note.diary %></textarea>
                            </div>
                            <br>
                            <div>
                                <!--#formnovalidate - отмена проверки на required-->

```

```

        <input class="btn btn-primary" type="submit" value="<%=t("Cancel")
%>" data-disable-with="Cancel Note">

        <input id="ChangeDiary<%=@note.id%>" class="btn btn-primary"
type="submit" value="<%=t("Save") %>"

        data-disable-with="Change Diary" name="ChangeDiary_button">

    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

### **app/views/layouts/\_each\_note.html.erb**

```

<div class="row ">
  <div class="col-1 ps-0 pe-0 ">
    <div class="card">
      <div class="form ps-0 pe-0 mx-auto">
        <label class="form-label mx-auto " for="flexCheck<%=@note.id %>">
          <span class="fst-italic mx-auto "><%= @note.time %></span>
        </label></div>
      </div>
    </div>
  <div class="col-9 ps-0 pe-0">
    <div class="card" style="background-color:<%= @note.colour %>;">
      <div class="form-check mb-2">
        <input class="form-check-input input_main"
          type="checkbox"
          name="checkbox_for_note"
          value="<%= @note.id %>"

```

```

        <%= "checked" if @note.if_done %>
        onclick="updateCheckbox(<%= @note.id %>, this.checked)">
    <label class="form-check-label" for="flexCheck<%= @note.id %>">
        <span class="fst-italic "><%= @note.task %></span></label></div>
    </div>
</div>

<!--кнопка для подробной инфы о задаче-->

<div class="col ps-1 pe-0">
    <div class="form-check ps-0 pe-0">
        <form id="main_right_form<%= @note.id %>"
action="main/adding/<%= @note.id %>" accept-charset="UTF-8" data-remote="true"
        method="get" data-type="json">
            <button type="submit" class="btn btn-secondary ps-1 pe-1">
                <i class="bi bi-info-circle text-light"></i>
            </button>
        </form>
    <!-- <form action="/notes/<%= @note.id %>" method="get">-->
    <!-- <input type="hidden" name="_method" value="get"/>-->

    <!-- </form>-->
    </div>
</div>

<div class="col ps-0 pe-0">
    <div class="form-check ps-0 pe-0 mx-auto">
        <!--кнопка для модульного окна-->
        <a class="btn btn-primary ps-1 pe-1 mx-auto"
            data-bs-toggle="modal"
            data-bs-target="#exampleModal<%= @note.id %>"
            href="/notes/<%= @note.id %>/edit"

```

```

        role="button"
        name="menu_change_note_btn">
        <i class="bi bi-pen text-light"></i></a>
<!--модульное окно-->
<div class="modal fade" id="exampleModal<%= @note.id %>" tabindex="-1"
    aria-labelledby="exampleModalLabel<%= @note.id %>" aria-
hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel<%= @note.id
%>"><%=t("Change Note") %></h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"
                    aria-label=<%=t("Close") %>></button>
            </div>
            <!--тело модульного окна-->
            <div class="modal-body">
                <div class="row">
                    <div class="col">
                        <form action="/notes/<%= @note.id %>" accept-charset="UTF-8"
method="patch">
                            <input type="hidden" name="_method" value="patch"
autocomplete="off">
                            <div>
                                <label style="display: block"
for="note_Дата"><%=t("Дата")%></label>
                                <input class="form-control" required="required" value="<%=
@note.day_date %>" type="date" name="note[day_date]" id="note_day_date">
                                </div>

```

```

        <div>
            <label style="display: block"
for="note_Задание"><%=t("Задание")%></label>
                <input class="form-control" required="required" type="text"
value="<%= @note.task %>" name="note[task]" id="note_task">
            </div>
            <div>
                <label style="display: block"
for="note_Подробнее"><%=t("Подробнее")%></label>
                <textarea class="form-control" name="note[info]"
id="note_info"><%= @note.info %></textarea>
            </div>
            <div>
                <label style="display: block"
for="note_Время"><%=t("Время")%></label>
                <input class="form-control" value="<%= @note.time %>"
type="time" name="note[time]" id="note_time">
            </div>
            <div class="form-check">
                <input class="form-check-input" type="checkbox" <%= "checked" if
@note.if_done %>
                    name="note[import]" id="note_import"
onclick="updateCheckbox(<%= @note.id %>, this.checked)">
                <label style="display: block" class="form-check-label"
for="note_Important"><%=t("Важно")%></label>
            </div>
            <div>
                <label style="display: block"
for="note_Цвет"><%=t("Цвет")%></label>

```

```

        <input class="form-control form-control-color" value="<%=
@note.colour %>" type="color" name="note[colour]"
        id="note_colour">
    </div>
    <br>
    <div>
        <!--#formnovalidate - отмена проверки на required-->
        <input class="btn btn-primary" type="submit" name="commit"
value="<%=t("Cancel")%>"
        data-disable-with="Cancel Note" formnovalidate=" ">
        <input id="ChangeNote<%=@note.id%>" class="btn btn-primary"
type="submit" name="commit" value="<%=t("Change Note") %>"
        data-disable-with="Change Note">
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

```

<div class="col ps-0 pe-0">
    <div class="form-check ps-0 pe-0">
        <form action="/notes/<%= @note.id %>" method="delete">
            <input type="hidden" name="_method" value="delete"/>
            <button type="submit" class="btn btn-danger ps-1 pe-1">
                <i class="bi bi-trash text-light"></i>

```

```

    </button>
  </form>
</div>
</div>
</div>
app/views/layouts/_header.html.erb
<header>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">ToDo</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-
bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <% if signed_in? %>
            <li class="nav-item">
              <%= link_to "#{t("Задачи")}", menu_path, class: "nav-link #{
request.path == menu_path || request.path == '/' ? 'active' : "}" %>
            </li>
            <li class="nav-item">
              <%= link_to "#{t("Настройки")}", settings_path, class: "nav-link #{
request.path == settings_path ? 'active' : "}" %>
            </li>

          <% else %>
            <li class="nav-item navbar-right">

```

```

        <%= link_to "#{t("Вход")}", signin_path, class: "nav-link #{
request.path == signin_path ? 'active' : " }" %>
    </li>
    <li class="nav-item">
        <%= link_to "#{t("Настройки")}", settings_path, class: "nav-link #{
request.path == settings_path ? 'active' : " }" %>
    </li>
    <% end %>

</ul>

<!--      <ul class="navbar-nav ms-auto mb-2 mb-lg-0 ">-->
    <%# if signed_in? %>
    <!--      <a class="navbar-brand float-right" href="#">Вы вошли как <%=
current_user.name %></a>-->
        <%# end %>
    <!--      </ul>-->
    </div>
</div>
</nav>
</header>
app/views/layouts/_import_task.html.erb
<div class="row import-card">
    <div class="col-3 ps-0 pe-0 ">
        <div class="card">
            <div class="form ps-0 pe-0 mx-auto">
                <label class="form-label mx-auto " for="flexCheck1">
                    <span class="fst-italic mx-auto "><%=
@note.day_date.strftime("%d.%m.%Y") %></span>
                </label></div>
            </div>

```



```

</div>
<div class="col-2 ps-0 pe-0 ">
  <div class="card">
    <div class="form ps-0 pe-0 mx-auto">
      <label class="form-label mx-auto " for="flexCheck1">
        <span class="fst-italic mx-auto "><%= @note.time %></span>
      </label></div>
    </div>
  </div>
</div>
<div class="col ps-0 pe-0 ">
  <div class="card" style="background-color: #F0E68C">
    <label class="form-label mx-auto" for="flexCheck1">
      <span class="fst-italic " id="task_<%= @note.id %>" style="<%= 'text-decoration: line-through;' if @note.if_done %>"><%= @note.task %></span>
    </label>
  </div>
</div>
</div>

```

#### **app/views/layouts/\_shin.html.erb**

```

<!--[if lt IE 9]>
<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->

```

#### **app/views/layouts/\_timetable.html.erb**

```

<%= @result.html_safe%>

```

#### **app/views/layouts/application.html.erb**

```

<!DOCTYPE html>
<html>
<head>
  <title>Chisla12</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 %>

<%= stylesheet_link_tag "application", media: "all", "data-turbolinks-track" =>
true %>
<%= javascript_include_tag "application", "data-turbolinks-track" => true %>
<%= csrf_meta_tags %>
<%= render 'layouts/shin' %>
<%= javascript_importmap_tags %>
</head>

<body>
<%= render 'layouts/header' %>
<div class="container">
  <% flash.each do |key, value| %>
    <div class="alert alert-<%= key %> alert-dismissible fade show"
role="alert"><%= value %>
      <button id="flash-close" type="button" class="btn-close" data-
dismiss="alert">
        </button>
      </div>
    <% end %>
  <%= yield %>
</div>
</body>
</html>

```

## **app/views/main/menu.html.erb**

```
<head>
  <link
    rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-
icons.css"
  />
</head>
<style>
  .holder {
    background: #f0efe9;
    min-height: 100vh;
  }

  .rounded {
    border-radius: 1rem !important;
  }

  .text-gray {
    color: #aaa;
  }

  input:checked + label {
    text-decoration: line-through;
    color: #adb5bd;
  }

  label {
    cursor: pointer;
  }
```

```

a {
    text-decoration: none;
}
#right_main {
    position: fixed; /* Фиксированное положение */
    float:right;
    width: 30%;
}
span {
    color: #000;
    text-shadow:
        0px 0px 2px #ffffff,
        -2px -2px 2px #ffffff,
        2px -2px 2px #ffffff,
        -2px 2px 2px #ffffff,
        2px 2px 2px #ffffff;
}
.import{
    width: 100%;
}
.import-card:hover{
    transform: scale(1.01);
    filter: drop-shadow(5px 5px 5px rgba(0,0,0,0.3));
}
</style>
<script type="text/javascript">
    var locale = "<%= I18n.locale %>";
    var Hide_Button="<%= t("Hide") %>"
    function updateCheckbox(id, status) {

```

```

console.log("Данные", id, status);
$.ajax({
  url: "/notes/update_checkbox/"+id,
  type: "patch",
  dataType: "json",
  data: { if_done: status },
  success: function(response) {
    console.log("Данные в базе данных обновлены", id, status);
    updateTaskDisplay(id, status); // вызываем функцию для
динамического обновления зачеркивания
  },
  error: function(xhr, status, error) {
    console.error(error);
  }
});
}

function updateTaskDisplay(id, status) {
  console.log('dynamic update through')
  let taskElement = $('#task_' + id);
  if (status) {
    taskElement.css('text-decoration', 'line-through');
  } else {
    taskElement.css('text-decoration', 'none');
  }
}
</script>
<br>
<a class="btn btn-primary" href="adding" role="button" ><%=t(:ADD)%></a>

<!--сюда мои карточки-->

```

```

<div class="row" id="notes">
  <div class="col-8">
    <div id="timetable-list">
      <%= render 'layouts/timetable' %>
    </div>
  </div>
  <div class="col-3" style="word-wrap: break-word; overflow-wrap: break-
word;">
    <div id=right_main>
      <div id="right"></div>
      <br>
      <a href="#" class="import btn btn-outline-primary disabled" tabindex="-1"
aria-disabled="true"><%=t("Важные задачи") %></a>
      <%= @import.html_safe %>
    </div>
  </div>
</div>

<!-- Modal -->

app/views/main/menu.js.erb
document.getElementById('timetable-list').innerHTML = "<%=
escape_javascript(render 'timetable') %>";

app/views/main/settings.html.erb
<div>
  <br>
  <%= if signed_in? %>
    <h4><a class="navbar-brand float-right" href="#"><%= t("Вы вошли как") %>
<%= current_user.name %></a></h4>

    <%= render @current_user %>

```

```

<% end %>

<%= t("Сменить язык на") %>:

<br>

<% I18n.available_locales.each do |locale| %>

  <% if I18n.locale != locale %>

    <%= link_to t(".lang.#{locale}"), url_for(locale: locale), class: "btn btn-
outline-primary mb-2" %>

  <% end %>

<% end %>

<br>

<% if signed_in? %>

  <div>

    <%= link_to "#{t('Выход')}", signout_path, method: "delete", class: "btn btn-
outline-primary mb-2 #{ request.path == signout_path ? 'active' : " }" %>

  </div>

<% end %>

</div>

```

### **app/views/notes/\_form.html.erb**

```

<div class="row">

  <div class="col-md-auto">

    <%= form_with(model: note, :id => 'add_new_task') do |form| %>

      <%# if user.errors.any? %>

      <!-- <div style="color: red">-->

      <!-- <h2><%# pluralize(user.errors.count, "error") %> prohibited this user
from being saved:</h2>-->

      <!-- <ul>-->

      <%# user.errors.each do |error| %>

      <!-- <li><%# error.full_message %></li>-->

      <%# end %>

```

```

<!--    </ul>-->

<!--    </div>-->

<%# end %>

<div>
    <%= form.label "#{t('Дата')}" , style: "display: block" %>
    <%= form.date_field :day_date, class: "form-control", required: true %>
</div>

<div>
    <%= form.label "#{t('Задание')}" , style: "display: block" %>
    <%= form.text_field :task, class: "form-control", required: true %>
</div>

<div>
    <%= form.label "#{t('Подробнее')}" , style: "display: block" %>
    <%= form.text_area :info, class: "form-control"%>
</div>

<div>
    <%= form.label "#{t('Время')}" , style: "display: block" %>
    <%= form.time_field :time, class: "form-control" %>
</div>

<div class="form-check">
    <%= form.check_box :import, {class: "form-check-input"}, true, false %>
    <%= form.label "#{t('Важно')}" , style: "display: block", class: "form-check-
label" %>
</div>

<div>
    <%= form.label "#{t('Цвет')}" , style: "display: block" %>
    <%= form.color_field :colour, class: "form-control form-control-color",
value: "#f9fbf7" %>

```



</div>

<br>

<div>

<!--#formnovalidate - отмена проверки на required-->

<input class="btn btn-primary" type="submit" name="commit"  
value="<%=t("Cancel") %>" data-disable-with="Create Note" formnovalidate>

<input class="btn btn-primary" type="submit" name="commit"  
value="<%=t("Create Note") %>" data-disable-with="Create Note">

</div>

<% end %>

</div>

</div>

**app/views/notes/\_note.html.erb**

<div id="<%= dom\_id note %>">

<p>

<strong><%=t("Пользователь") %>:</strong>

<%= note.User\_id %>

</p>

<p>

<strong><%=t("Дата") %>:</strong>

<%= note.day\_date %>

</p>

<p>

<strong><%=t("Время") %>:</strong>

<%= note.time %>

</p>

```
<p>
  <strong><%=t("Задание") %>:</strong>
  <%= note.task %>
</p>
```

```
<p>
  <strong><%=t("День недели") %>:</strong>
  <%= note.week_day %>
</p>
```

```
<p>
  <strong><%=t("Цвет") %>:</strong>
  <%= note.colour %>
</p>
```

```
</div>
```

#### **app/views/notes/edit.html.erb**

```
<h1><%=t("Editing note") %></h1>
<%= render "form", note: @note%>
<br>
```

#### **app/views/notes/index.html.erb**

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

```
<h1>Notes</h1>
```

```
<div id="notes">
  <% @notes.each do |note| %>
    <%= render note %>
  <p>
```

```

    <%= link_to "#{t("Show this note")}" , note %>

  </p>

<% end %>

</div>

```

```

<%= link_to "#{t("New Note")}" , new_note_path %>

```

### **app/views/notes/new.html.erb**

```

<h1><%=t("Детали записи") %></h1>

```

```

<%= render "form", note: @note %>

```

### **app/views/notes/show.html.erb**

```

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

```

```

<%= render @note %>

```

```

<div>

```

```

  <%= link_to "#{t("Edit this note")}" , edit_note_path(@note) %> |

```

```

  <%= link_to "#{t("Back to notes")}" , notes_path %>

```

```

  <%= button_to "#{t("Destroy this note")}" , @note, method: :delete %>

```

```

</div>

```

### **app/views/sessions/new.html.erb**

```

<h1><%=t("Sign in") %></h1>

```

```

<%= form_for(:session, :url => sessions_path) do |f| %>

```

```

  <div class="field">

```

```

    <%= f.label t("Email") %><br />

```

```

    <%= f.email_field :email, class:"form-control"%>

```

```

  </div>

```

```

  <div class="field">

```

```

    <%= f.label t("Password") %><br />

```

```

    <%= f.password_field :password, class:"form-control" %>
  </div>

  <div class="actions">
    <%= f.submit "#{t("Sign in")}", name: "login_btn", class: "btn btn-primary mt-2"%>
  </div>

<% end %>

```

```

<p><%=t("New user?") %><%= link_to "#{t("Sign up now!")}", signup_path %></p>

```

### **app/views/users/\_form.html.erb**

```

<%= form_with(model: user) do |form| %>
  <div>
    <%= form.label t("Name"), style: "display: block" %>
    <%= form.text_field :name, class:"form-control" %>
  </div>

  <div>
    <%= form.label t("Email"), style: "display: block" %>
    <%= form.text_field :email, class:"form-control", required:true %>
  </div>

  <div>
    <%= form.label t("Password"), style: "display: block" %>
    <%= form.password_field :password, class:"form-control", required:true %>
  </div>

  <div>
    <%= form.label t("Password confirmation"), style: "display: block" %>

```

```
<%= form.password_field :password_confirmation, minlength: 4, suggested:
"new-password", required: true,class:"form-control" %>
```

```
</div>
```

```
<br>
```

```
<div>
```

```
<%= form.submit t("Sign up"), class:"btn btn-primary"%>
```

```
<%= link_to t("Back to sign in"), signin_path %>
```

```
</div>
```

```
<% end %>
```

**app/views/users/\_user.html.erb**

```
<div id="<%= dom_id user %>">
```

```
<p>
```

```
<strong><%=t("Name") %>:</strong>
```

```
<%= user.name %>
```

```
</p>
```

```
<p>
```

```
<strong><%=t("Email") %>:</strong>
```

```
<%= user.email %>
```

```
</p>
```

```
</div>
```

**app/views/users/edit.html.erb**

```
<h1><%=t("Editing user") %></h1>
```

```
<%= render "form", user: @user %>
```

```
<br>
```

```

<div>
  <%= link_to t("Show this user"), @user %> |
  <%= link_to t("Back to users"), users_path %>
</div>

```

### **app/views/users/index.html.erb**

```

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

```

```

<h1>Users</h1>

```

```

<div id="users">
  <% @users.each do |user| %>
    <%= render user %>
    <p>
      <%= link_to t("Show this user"), user %>
    </p>
  <% end %>
</div>

```

```

<%= link_to t("New user"), new_user_path %>

```

### **app/views/users/new.html.erb**

```

<h1><%=t("Sign up") %></h1>

```

```

<%= render "form", user: @user %>

```

### **app/views/users/show.html.erb**

```

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

```

```

<%= render @user %>

```

```

<div>
  <%= link_to t("Edit this user"), edit_user_path(@user) %> |

```

```

<%= link_to t("Back to users"), users_path %>

<% if @current_user && @current_user.admin == true %>
  <%= button_to t("Destroy this user"), @user, method: :delete %>
<% end %>

</div>

```

### Контроллеры:

#### **app/controllers/ application\_controller.rb**

```

class ApplicationController < ActionController::Base
  protect_from_forgery with: :exception
  include SessionsHelper
  before_action :set_locale
  private
  def set_locale
    I18n.locale = params[:locale] || I18n.default_locale
  end
  def default_url_options
    { locale: I18n.locale }
  end
end

```

#### **app/controllers/ main\_controller.rb**

```

class MainController < ApplicationController
  def menu
    unless signed_in?
      respond_to do |format|
        format.html { redirect_to signin_path }
      end
    end
    return
  end
end

```

```

@import=""
@result = html_creation
respond_to do |f|
  f.html
  f.json do
    combined_json = {
      result: { type: @result.class.to_s, value: @result },
      import: { type: @import.class.to_s, value: @import }
    }
    render json: combined_json
  end
end
end

def html_creation
  notes = sort_notes
  current_date = '12345'
  result = ""
  output_begin = ""
  output_end = ""
  notes.each do |note|
    @note = note
    if @current_user && note.User_id == @current_user.id
      month_trans = t(note.day_date.strftime("%B"))
      if note.day_date != current_date
        result += output_begin + output_end
        output_begin, output_end = add_card(", ", note.day_date.strftime("%d
#{month_trans}, %Y"), note.week_day)
        current_date = note.day_date
      end
    end
  end
end

```



```

        output_begin += add_task

        @import += add_import_task if @note.import == true
    end
end

result += output_begin + output_end

result

end

def add_task

  # Путь к файлу шаблона
  template_path = Rails.root.join('app', 'views', 'layouts', '_each_note.html.erb')

  # Считываем содержимое файла
  template_content = File.read(template_path)

  # Заменяем переменную в шаблоне
  ERB.new(template_content).result(binding)
end

def add_card(output_begin, output_end, day_date, week_day)

  output_begin += "<div class=\"card card_hover rounded border-0 shadow-sm
position-relative mt-2\">
  <div class=\"card-body p-5\">
  <div class=\"align-items-center mb-4 pb-4 border-bottom\">
  <i class=\"far fa-calendar-alt fa-3x\"></i>
  <div class=\"ms-3\">
  <h4 class=\"text-uppercase fw-weight-bold mb-0\">#{day_date}
  <a class=\"btn btn-outline-primary float-end\"
    data-bs-toggle=\"modal\"

```

```

data-bs-target="#dModal#{@note.id}"
href="/main/diary/#{@note.id}"
role="button" name="diary_button">#{t("DIARY")}</a></h4>

# Путь к файлу шаблона
template_path = Rails.root.join('app', 'views', 'layouts', '_each_diary.html.erb')

# Считываем содержимое файла
template_content = File.read(template_path)

# Заменяем переменную в шаблоне
output_begin += ERB.new(template_content).result(binding)
output_begin += "<p class='text-gray fst-italic mb-0'>#{t(week_day)}</p></div></div>"
output_end += "</div></div>"
[output_begin, output_end]
end

def add_import_task
  # Путь к файлу шаблона
  template_path = Rails.root.join('app', 'views', 'layouts', '_import_task.html.erb')

  # Считываем содержимое файла
  template_content = File.read(template_path)

  # Заменяем переменную в шаблоне
  ERB.new(template_content).result(binding)
end

def sort_notes
  Note.all.sort_by { |x| [x.User_id, x.day_date, x.time.empty? ? 1 : 0, x.time] }
end

```

```
# ФУНКЦИИ ДОБАВЛЕНИЯ ИНФОРМАЦИИ СПРАВА
```

```
def adding
  unless signed_in?
    respond_to do |format|
      format.html { redirect_to signin_path }
    end
    return
  end
  @add_right = task_info
  respond_to do |f|
    f.html
    f.json do
      render json: { type: @add_right.class.to_s, value: @add_right }
    end
  end
end
```

```
def task_info
  pp 'params id', params[:id]
  @note = Note.find_by_id(params[:id])
  pp "note", @note.info
  "<p>#{t(:Подробная_информация)}: </p><div >" + @note.info + "</div>"
end
```

```
def diary
  data=Note.find_by_id(params[:id]).day_date
  diary = params[:note][:diary]
  Note.all.each do |note|
    @note=note
    if @note.day_date==data
```

```

        @note.update_attribute(:diary, diary)
      end
    end
  end

end

app/controllers/ notes_controller.rb
require 'date'

class NotesController < ApplicationController
  before_action :check_for_cancel, :only => [:create, :update]
  before_action :set_note, only: %i[ show edit update destroy ]

  # GET /notes or /notes.json
  def index
    @notes = Note.all
  end

  # GET /notes/1 or /notes/1.json
  def show
  end

  # GET /notes/new
  def new
    @note = Note.new
  end

  # GET /notes/1/edit
  def edit
  end
end

```

```

# POST /notes or /notes.json
def create
  unless signed_in?
    respond_to do |format|
      format.html { redirect_to signin_path }
    end
    return
  end

  days = {0 => "Воскресенье",
          1 => "Понедельник",
          2 => "Вторник",
          3 => "Среда",
          4 => "Четверг",
          5 => "Пятница",
          6 => "Суббота"}

  msg_text = "
  msg_status = :success
  params[:note][:User_id] = current_user.id
  day_date = params[:note][:day_date]
  params[:note][:week_day] = days[Date.strptime(day_date, "%Y-%m-
%d").wday]

  unless params[:note][:time]
    params[:note][:time] = "
  end

  unless params[:note][:diary]
    params[:note][:diary] = "
  end

  unless params[:note][:info]
    params[:note][:info] = t('Нет информации')
  end
end

```

```

unless params[:note][:import]
  params[:note][:import] = false
end
params[:note][:if_done]=false
@note = Note.new(note_params)

respond_to do |format|

  if @note
    pp 'check 3'
    if msg_status == :success and @note.save
      #menu_path @note
      msg_text = t('Запись была успешно создана')
      flash[msg_status] = msg_text
      format.html { redirect_to menu_path}
      format.json { render :show, status: :created, location: menu_path }
    else
      flash.now[msg_status] = msg_text
      format.html { render :new, status: :unprocessable_entity }
      format.json { render json: @user.errors, status: :unprocessable_entity }
    end
  end
end
end

# PATCH/PUT /notes/1 or /notes/1.json
def update
  pp 'update', params[:id], params[:if_done]
  respond_to do |format|
    if @note.update(note_params)

```

```

    Note.all.sort_by { |x| [x.User_id, x.day_date, x.time.empty? ? 1 : 0, x.time] }
    format.html { redirect_to menu_path}
    # format.json { render :show, status: :ok, location: @note }
    format.js
  else
    format.html { render :edit, status: :unprocessable_entity }
    format.json { render json: @note.errors, status: :unprocessable_entity }
  end
end

end

end

# DELETE /notes/1 or /notes/1.json
def destroy
  @note=Note.find(params[:id])
  @note.destroy!
  @notes = Note.all # обновляем список записей
  respond_to do |format|
    format.js
    format.html { redirect_to menu_url }
    # format.json { head :no_content }

  end
end

#для кнопочки отмена записи
def check_for_cancel
  if params[:commit] == t("Cancel")
    redirect_to menu_path
  end
end

def update_checkbox

```

```

id = params[:id] # предполагая, что id элемента передается в параметрах
status = params[:if_done] # статус true/false из чекбокса

# Здесь используйте вашу модель для обновления записи в базе данных
@note = Note.find(id)
if @note.update_attribute(:if_done, status)
  render json: { status: "success" }
else
  render json: { status: "error" }
end
end
private

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

# Only allow a list of trusted parameters through.
def note_params
  params.require(:note).permit(:day_date, :User_id, :time, :task, :week_day,
:colour, :diary, :info, :import, :if_done)
end
end

app/controllers/ sessions_controller.rb
class SessionsController < ApplicationController
  #skip_before_action :authenticate, only: [:new, :create]

  def new
  end

```



```

def create
  msg_text = "
  msg_status = :success

  email = params[:session][:email]
  password = params[:session][:password]

  puts password

  respond_to do |format|
    user = User.find_by(email: email.downcase)

    if !user
      msg_text = t('Пользователя не существует')
      msg_status = :danger
    elsif !user.authenticate(password)
      msg_text = t('Неверный пароль')
      msg_status = :danger
    end

    if msg_status == :success
      sign_in user
      msg_text = t('Вы успешно вошли')
      flash[msg_status] = msg_text
      format.html { redirect_to menu_path }
      format.json { render :show, status: :created, location: menu_path }
    else
      flash.now[msg_status] = msg_text
      format.html { render :new, status: :unprocessable_entity }
    end
  end
end

```

```

        format.json { render json: @user.errors, status: :unprocessable_entity }
      end
    end

    #user =
User.authenticate(params[:session][:email],params[:session][:password])
    end

    #убрать пользователя из сессии
    def destroy
      sign_out
      redirect_to root_url
    end

  end

app/controllers/ users_controller.rb
require './app/controllers/notes_controller'
class UsersController < ApplicationController
  before_action :set_user, only: [:show, :edit, :update, :destroy]
  include NotesHelper

  # GET /users or /users.json
  def index
    @users = User.all
    #added

    @user_name = current_user ? current_user.name : "unknown"
  end

  # GET /users/1 or /users/1.json
  def show
  end

  # GET /users/new

```

```

def new
  @user = User.new
end

# GET /users/1/edit
def edit
end

# POST /users or /users.json
def create
  #added
  msg_text = "
  msg_status = :success

  email = params[:user][:email]
  password = params[:user][:password]
  password_confirmation = params[:user][:password_confirmation]
  params[:user][:admin]=false
  puts password

  @user = User.new(user_params)
  pp 'check 2'
  respond_to do |format|

    if @user
      if User.find_by_email(email)
        msg_text = t('Пользователь уже зарегистрирован!')
        msg_status = :danger
      elsif password != password_confirmation
        msg_text = t('Пароль для подтверждения введен неверно')
      end
    end
  end
end

```

```

        msg_status = :danger
        elsif !email.match?('[a-z0-9]+[_a-z0-9\.-]*[a-z0-9]+@[a-z0-9-]+\.[a-z0-9-
]+)*(\.[a-z]{2,4})')
            msg_text = t('Введите почту корректно')
            msg_status = :danger
        end
        pp 'check 3'
        if msg_status == :success and @user.save
            sign_in @user
            msg_text = t('Спасибо за регистрацию')
            flash[msg_status] = msg_text

            format.html { redirect_to menu_path, notice: t('User was successfully
created') }
            format.json { render :show, status: :created, location: menu_path }
        else
            flash.now[msg_status] = msg_text
            format.html { render :new, status: :unprocessable_entity }
            format.json { render json: @user.errors, status: :unprocessable_entity }
        end
    end
end
end
end

# PATCH/PUT /users/1 or /users/1.json
def update
    respond_to do |format|
        if @user.update(user_params)
            format.html { redirect_to user_url(@user), notice: t("User was successfully
updated") }

```

```

    format.json { render :show, status: :ok, location: @user }
  else
    format.html { render :edit, status: :unprocessable_entity }
    format.json { render json: @user.errors, status: :unprocessable_entity }
  end
end
end
end

# DELETE /users/1 or /users/1.json
def destroy

  #удаление из notes сначала
  destroy_User_id(@user.id)
  @user.destroy!

  respond_to do |format|
    format.html { redirect_to users_url, notice: t("User was successfully
destroyed") }
    format.json { head :no_content }
  end
end

private

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

# Only allow a list of trusted parameters through.
def user_params

```

```

        params.require(:user).permit(:name, :email, :password,
:password_confirmation, :admin)
    end
end

```

### **app/helpers/notes\_helper.rb**

```

module NotesHelper
  #для удаления успешного удаления пользователя
  def destroy_User_id(_user_id)
    while @note=Note.find_by(User_id: _user_id)
      @note.destroy!
    end
  end
end

```

### **app/helpers/sessions\_helper.rb**

```

module SessionsHelper
  #added
  def sign_in(user)
    remember_token = User.new_remember_token
    pp 'check 1 in def sign_in user'
    cookies.permanent[:remember_token] = remember_token
    user.update_attribute(:remember_token, User.encrypt(remember_token))
    pp 'check 2'
    self.current_user = user
  end
  def current_user=(user)
    @current_user = user
  end
  def signed_in?
    !current_user.nil?
  end
end

```

```

def sign_out
  current_user.update_attribute(:remember_token,
                               User.encrypt(User.new_remember_token))
  cookies.delete(:remember_token)
  session[:current_user_id] = nil
  self.current_user = nil
end

def current_user
  remember_token = User.encrypt(cookies[:remember_token])
  pp 'check def current user in session_helper'
  @current_user ||= User.find_by(remember_token: remember_token)

  # @_current_user ||= session[:current_user_id] &&
  # User.find_by_id(session[:current_user_id])
end
end

```

### Маршруты:

#### config/routes.rb

```

Rails.application.routes.draw do
  scope "(:locale)", locale: /en|ru/ do
    resources :notes
    resources :users
    resources :sessions, :only => [:new, :create, :destroy]
    match '/menu', to: 'main#menu', via: 'get'
    match '/settings', to: 'main#settings', via: 'get'
    match '/main/adding/:id', to: 'main#adding', via: 'get'
    match '/main/diary/:id', to: 'main#diary', via: 'get'
    match '/notes/update_checkbox/:id', to: 'notes#update_checkbox', via: 'patch'
    match '/adding', to: 'notes#new', via: [:get, :post]
    match '/signup', to: 'users#new', via: 'get'
  end
end

```

```
match '/signin', to: 'sessions#new', via: 'get'
match '/set_locale', to: 'application#set_locale', via: 'get'
match '/signout', to: 'sessions#destroy', via: 'delete'
get "up" => "rails/health#show", as: :rails_health_check
delete '/notes/:id' => 'notes#destroy'

root 'main#menu'

end

end
```

### **Локализационные файлы:**

#### **config/locales/en.yml**

```
en:
  hello: "Hello world"
  January: "January"
  February: "February"
  March: "March"
  April: April
  June: June
  July: July
  August: August
  September: September
  October: October
  November: November
  December: "December"
  DIARY: DIARY
  ADD: "ADD"
  Close: "Close"
  Cancel: "Cancel"
  Change Note: "Change Note"
  Save: "Save"
  Задачи: "Tasks"
```



Настройки: "Settings"  
Вход: "Entry"  
Вы вошли как: "You signed up as"  
Выход: "Exit"  
Сменить язык на: "Change language to"  
Дата: "Date"  
Задание: "Task"  
Подробнее: "Details"  
Время: "Time"  
Важно: "Important"  
Цвет: "Color"  
Пользователь: "User"  
День недели: "Day of the week"  
Show this note: "Show this note"  
New Note: "New Note"  
Edit this note: "Edit this note"  
Back to notes: "Back to notes"  
Destroy this note: "Destroy this note"  
Email: "Email"  
Password: "Password"  
Sign in: "Sign in"  
Sign up now!: "Sign up now!"  
Name: "Name"  
Password confirmation: "Password confirmation"  
Sign up: "Sign up"  
Show this user: "Show this user"  
Back to users: "Back to users"  
Edit this user: "Edit this user"  
Destroy this user: "Destroy this user"  
Подробная\_информация: "In details"

Мои мысли: "My thoughts"

Воскресенье: "Sunday"

Понедельник: "Monday"

Вторник: "Tuesday"

Среда: "Wednesday"

Четверг: "Thursday"

Пятница: "Friday"

Суббота: "Saturday"

Нет информации: "No information"

Запись была успешно создана: "Note was successfully created"

Пользователя не существует: "User doesn't exist"

Неверный пароль: "Wrong password"

Вы успешно вошли: "You signed up successfully "

Пользователь уже зарегистрирован!: "User already exists"

Пароль для подтверждения введен неверно: "Password confirmation is wrong"

Введите почту корректно: "Wrong email format"

Спасибо за регистрацию: "Thank you for registration"

User was successfully created: "User was successfully created"

User was successfully updated: "User was successfully updated"

User was successfully destroyed: "User was successfully destroyed"

Важные задачи: "Important tasks"

Create Note: "Create Note"

Editing note: "Editing note"

Детали записи: "Note details"

Editing user: "Editing user"

Hide: "Hide"

New user?: "New user?"

Back to sign in: "Back to sign in"

**config/locales/ru.yml**

ru:

DIARY: Дневник  
January: "Января"  
February: "Февраля"  
March: "Марта"  
April: Апреля  
June: Июня  
July: Июля  
August: Августа  
September: Сентября  
October: Октября  
November: Ноября  
December: "Декабря"  
ADD: "ДОБАВИТЬ"  
Close: "Заккрыть"  
Cancel: "Отменить"  
Change Note: "Изменить"  
Save: "Сохранить"  
Задачи: "Задачи"  
Настройки: "Настройки"  
Вход: "Вход"  
Вы вошли как: "Вы вошли как"  
Выход: "Выход"  
Сменить язык на: "Сменить язык на"  
Дата: "Дата"  
Задание: "Задание"  
Подробнее: "Подробнее"  
Время: "Время"  
Важно: "Важно"  
Цвет: "Цвет"  
Пользователь: "Пользователь"

День недели: "День недели"  
Show this note: "Показать запись"  
New Note: "Новая запись"  
Edit this note: "Редактировать запись"  
Back to notes: "Ко всем записям"  
Destroy this note: "Удалить запись"  
Email: "Почта"  
Password: "Пароль"  
Sign in: "Войти"  
Sign up now!: "Зарегистрироваться!"  
Name: "Имя"  
Password confirmation: "Подтверждение пароля"  
Sign up: "Зарегистрироваться"  
Show this user: "Показать пользователя"  
Back to users: "Ко всем пользователям"  
Edit this user: "Редактировать пользователя"  
Destroy this user: "Удалить пользователя"  
Подробная\_информация: "Подробная информация"  
Мои мысли: " Мои мысли"  
Воскресенье: "Воскресенье"  
Понедельник: "Понедельник"  
Вторник: "Вторник"  
Среда: "Среда"  
Четверг: "Четверг"  
Пятница: "Пятница"  
Суббота: "Суббота"  
Нет информации: "Нет информации"  
Запись была успешно создана: "Запись была успешно создана"  
Пользователя не существует: "Пользователя не существует"  
Неверный пароль: "Неверный пароль"

Вы успешно вошли: "Вы успешно вошли"

Пользователь уже зарегистрирован!: "Пользователь уже зарегистрирован!"

Пароль для подтверждения введен неверно: "Пароль для подтверждения введен неверно"

Введите почту корректно: " Введите почту корректно"

Спасибо за регистрацию: "Спасибо за регистрацию"

User was successfully created: "Пользователь был успешно зарегистрирован"

User was successfully updated: "Пользователь был успешно изменен"

User was successfully destroyed: "Пользователь был успешно удален"

Важные задачи: "Важные задачи"

Create Note: "Создать"

Editing note: "Редактировать запись"

Детали записи: "Детали записи"

Editing user: "Редактировать пользователя"

Hide: "Скрыть"

New user?: "Новый пользователь?"

Back to sign in: "Вернуться ко входу"

### **Тестирование приложения:**

**test/controllers/main\_controller\_test.rb**

```
require "test_helper"

# rake test TEST=test/controllers/main_controller_test.rb - запуск теста
class MainControllerTest < ActionDispatch::IntegrationTest
  test "should get menu" do
    get menu_url
    assert_response 302 #так как не вошли
  end

  test "should get settings" do
    get settings_url
    assert_response :success
  end
end
```

```

end
end
test/controllers/notes_controller_test.rb
require "test_helper"

# rake test TEST=test/controllers/notes_controller_test.rb - запуск теста
class NotesControllerTest < ActionDispatch::IntegrationTest
  setup do
    @user = User.new(id:1, email: 'test1@gmail.com', password: '123456',
password_confirmation: '123456', remember_token: 'test_token', admin: false)

    @user.save

    @test_note1 = Note.new(day_date: '2012-12-12',
      User_id:1,
      time: '4:19',
      task: 'write',
      week_day: 'Понедельник',
      colour: '#000000',
      diary: 'sick',
      info: 'no',
      import: false,
      if_done: false)

    @test_note1.save

    @test_note2 = { day_date: '2012-12-13',
      User_id:1,
      time: '4:20',
      task: 'finish',
      week_day: 'Понедельник',
      colour: '#000000',
      diary: 'tired',
      info: 'no',
      import: false,

```

```

        if_done: false}
    end
    def user_params(email, password)
      { 'email' => email, 'password' => password, 'password_confirmation' =>
password, 'admin' => false }
    end

    def add_record(email, password)
      record = User.new(user_params(email, password))
      record.save
      record
    end

    test "should get index" do
      get notes_url
      assert_response :success
    end

    test "should get new" do
      get new_note_url
      assert_response :success
    end

    test "should create note" do
      #регистрация
      add_record('test@test.com', '123456')

      assert_difference 'User.count', 0 do
        post sessions_url, params: { 'authenticity_token' => 'token', 'session' => {
'email' => 'test@test.com', 'password' => '123456' } }
        follow_redirect!
      end
    end
  end
end

```

```

end

#проверка перенаправления
assert_template 'main/menu'
assert_response 200
assert_difference("Note.count") do
  post notes_url, params: { note: @test_note2 }
end

assert_redirected_to menu_path
end

test "should show note" do
  get note_url(locale='en',@test_note1)
  assert_response :success
end

test "should get edit" do
  get edit_note_url(locale='en',@test_note1)
  assert_response :success
end

test "should update note" do
  patch note_url(locale='en',@test_note1), params: { note: @test_note2 }
  assert_redirected_to '/main/menu'
end

test "should destroy note" do
  assert_difference("Note.count", -1) do
    delete note_url(locale='en', @test_note1)
  end
end

```



```

    assert_redirected_to '/main/menu'
  end
end

test/controllers/users_controller_test.rb

require "test_helper"

# rake test TEST=test/controllers/users_controller_test.rb - запуск теста
class UsersControllerTest < ActionDispatch::IntegrationTest
  setup do
    @user = User.new(email: 'test1@gmail.com', password: '123456',
password_confirmation: '123456', remember_token: 'test_token', admin: false)
    @user.save
    @test_user = { email: 'test2@gmail.com', password: '123456',
password_confirmation: '123456', remember_token: 'test_token', admin: false }
  end

  test 'should get index' do
    get users_url
    assert_response :success
  end

  test 'should get new' do
    get new_user_url
    assert_response :success
  end

  test 'should create user' do
    assert_difference('User.count') do
      post users_url, params: { user: @test_user }
    end
  end
end

```

```
    assert_redirected_to menu_path
  end
```

```
test 'should show user' do
  get user_url(locale = 'en', @user)
  assert_response :success
end
```

```
test 'should get edit' do
  get edit_user_url(locale = 'en', @user)
  assert_response :success
end
```

```
test 'should update user' do
  patch user_url(locale = 'en', @user), params: { user: @test_user }
  assert_redirected_to user_url(@user)
end
```

```
test 'should destroy user' do
  assert_difference('User.count', -1) do
    delete user_url(locale = 'en', @user)
  end
end
```

```
    assert_redirected_to users_url
  end
end
```

### **test/integration/authentication\_pages\_test.rb**

```
require "test_helper"
```

```
# rake test:integration
```

```
# rake test TEST=test/integration/authentication_pages_test.rb
```

```

class AuthenticationPagesTest < ActionDispatch::IntegrationTest
  def user_params(name, email, password)
    { 'name' => name, 'email' => email, 'password' => password,
'password_confirmation' => password, 'admin' => false }
  end

  def add_record(name, email, password)
    record = User.new(user_params(name, email, password))
    record.save
    record
  end

  # тест,позволяющий проверить регистрацию нового пользователя,
  # вход под его именем и выполнение вычислений
  ##### Sign up
#####

  # Проверяем доступность страницы регистрации
  test "test registration page access" do
    get signup_url
    assert_response :success
  end

  # Проверяем, что нельзя зарегистрировать того же пользователя
  test 'attempt to register with existing user details' do
    # Создаем пользователя
    add_record('test', 'test@test.com', '123456')
    get signup_url
    assert_response :success
    post users_url, params: { 'authenticity_token' => 'token',
                              'user' => { 'name' => 'test',
                                          'email' => 'test@test.com',

```

```

        'password' => '123456',
        'password_confirmation' => '123456' } }

    assert_response 422
  end

  # Проверяем, что пользователя можно зарегистрировать
  test 'successfully user registration' do
    get signup_url
    assert_response :success

    # Смотрим, что такой пользователь только 1
    assert_difference 'User.count', 1 do
      post users_url, params: { 'authenticity_token' => 'token',
                                'user' => { 'name' => 'test',
                                              'email' => 'test@test.com',
                                              'password' => '123456',
                                              'password_confirmation' => '123456' } }

      follow_redirect!
    end

    assert_template 'main/menu'
    assert_response 200
  end

  ##### Sign in
  #####

  # Проверяем доступность страницы входа
  test "test login page access" do
    get signin_url
    assert_response :success
  end

  test 'successfully user login' do
    add_record('test', 'test@test.com', '123456')
  end

```

```

assert_difference 'User.count', 0 do
  post sessions_url, params: { 'authenticity_token' => 'token',
                                "session" => { "name" => 'test',
                                                "email" => "test@test.com",
                                                "password" => "123456" } }

  follow_redirect!
end

assert_template 'main/menu'
assert_response 200
end

test 'login of a non-existent user' do
  post sessions_url, params: { 'authenticity_token' => 'token',
                                "session" => { "name" => 'test',
                                                "email" => "test@test.com",
                                                "password" => "123456" } }

  assert_template 'sessions/new'
  assert_response 422
end

test 'login without password' do
  post sessions_url, params: { 'authenticity_token' => 'token',
                                "session" => { "name" => 'test',
                                                "email" => "test@test.com",
                                                "password" => "" } }

  assert_template 'sessions/new'
  assert_response 422
end

##### Sign out
#####

test "test logout success" do

```

```

# Добавляем тестового юзера в БД
add_record('test', 'test@test.com', '123456')
assert_difference 'User.count', 0 do
  # login

  post sessions_url, params: { 'authenticity_token' => 'token',
                                "session" => { "name" => 'test',
                                                "email" => "test@test.com",
                                                "password" => "123456" } }

  follow_redirect!
end
assert_difference 'User.count', 0 do
  # Logout

  delete signout_url

  follow_redirect! # перенаправлены в menu, там проверка, что не
зalogинились и идем в login

  follow_redirect! # из menu в login
end
assert_template 'sessions/new'
assert_response 200
end

# Подготовить интеграционный тест для проверки невозможности
# выполнения вычислений без ввода логина/пароля
test "Post is impossible without sign in" do
  post notes_url, params: { 'authenticity_token' => 'test',
                            'test_note' => {
                              'day_date' => '2012-12-12',
                              'User_id' => 1,
                              'time' => '4:19',
                              'task' => 'write',
                              'week_day' => 'Понедельник',

```

```

        'colour' => '#000000',
        'diary' => 'sick',
        'info' => 'no',
        'import' => false,
        'if_done' => false },
        'locale' => 'en' }

    follow_redirect!

    # Если не вошли, значит редиректимся в signin
    assert_response 200
    assert_template 'sessions/new'
  end
end

```

### **Тесты Rspec и Selenium:**

```
$rails generate rspec:install
```

```
$rails s
```

*В spec/ добавляем файл test\_chisla\_spec.rb*

*Записываем туда код, сгенерированный Katalon Recorder в формате Ruby*

*Добавляем require 'rails\_helper'*

*Заменяем везде `\${receiver}` на `@driver` в связи с ошибкой генерации*

*Katalon.*

```
$bundle exec rspec spec/test_chisla_spec.rb
```

```
spec/test_diary_spec.rb
```

```
require 'rails_helper'
```

```
require "json"
```

```
require "selenium-webdriver"
```

```
require "rspec-rails"
```

```
include RSpec::Expectations
```

```
describe "NewNote" do
```

```
before(:each) do
```

```
  @driver = Selenium::WebDriver.for :edge
```

```
  @base_url = "https://www.google.com/"
```

```
  @accept_next_alert = true
```

```
  @driver.manage.timeouts.implicit_wait = 30
```

```
  @verification_errors = []
```

```
end
```

```
after(:each) do
```

```
  @driver.quit
```

```
  @verification_errors.should == []
```

```
end
```

```
it "test_new_note" do
```

```
  @driver.get "http://127.0.0.1:3000/ru/signin"
```

```
  @driver.find_element(:id, "session_email").click
```

```
  @driver.find_element(:id, "session_email").clear
```

```
  @driver.find_element(:id, "session_email").send_keys "admin@gmail.com"
```

```
  @driver.find_element(:xpath, "//body").click
```

```
  @driver.find_element(:id, "session_password").click
```

```
  @driver.find_element(:id, "session_password").clear
```

```
  @driver.find_element(:id, "session_password").send_keys "12345"
```

```
  @driver.find_element(:xpath, "//body").click
```

```
  @driver.find_element(:name, "login_btn").click
```

```
  sleep 2.0
```

```
  @driver.get "http://127.0.0.1:3000/ru/menu"
```

```
  @driver.find_element(:link, "ДОБАВИТЬ").click
```

```
  @driver.get "http://127.0.0.1:3000/ru/adding"
```

```
  @driver.find_element(:id, "note_day_date").click
```

```
  @driver.find_element(:id, "note_day_date").clear
```



```

    @driver.find_element(:id, "note_day_date").send_keys "0002-12-12"
    @driver.find_element(:id, "note_day_date").clear
    @driver.find_element(:id, "note_day_date").send_keys "0020-12-12"
    @driver.find_element(:id, "note_day_date").clear
    @driver.find_element(:id, "note_day_date").send_keys "0200-12-12"
    @driver.find_element(:id, "note_day_date").clear
    @driver.find_element(:id, "note_day_date").send_keys "2002-12-12"
    @driver.find_element(:id, "note_task").click
    @driver.find_element(:id, "note_task").clear
    @driver.find_element(:id, "note_task").send_keys "test"
    @driver.find_element(:xpath,
    "-//form[@id='add_new_task']/div[7]/input[2]").click
end

```

```

def element_present?(how, what)
  @driver.find_element(how, what)
  true
rescue Selenium::WebDriver::Error::NoSuchElementException
  false
end

```

```

def alert_present?()
  @driver.switch_to.alert
  true
rescue Selenium::WebDriver::Error::NoAlertPresentError
  false
end

```

```

def verify(&blk)
  yield

```

```
rescue ExpectationNotMetError => ex
  @verification_errors << ex
end
```

```
def close_alert_and_get_its_text(how, what)
  alert = @driver.switch_to().alert()
  alert_text = alert.text
  if (@accept_next_alert) then
    alert.accept()
  else
    alert.dismiss()
  end
  alert_text
ensure
  @accept_next_alert = true
end
end
```

```
describe "NewDiary" do
```

```
  before(:each) do
    @driver = Selenium::WebDriver.for :edge
    @base_url = "https://www.google.com/"
    @accept_next_alert = true
    @driver.manage.timeouts.implicit_wait = 30
    @verification_errors = []
  end
```

```
  after(:each) do
```

```

@driver.quit
@verification_errors.should == []
end

it "test_new_diary" do
  @driver.get "http://127.0.0.1:3000/ru/signin"
  @driver.find_element(:id, "session_email").click
  @driver.find_element(:id, "session_email").clear
  @driver.find_element(:id, "session_email").send_keys "admin@gmail.com"
  @driver.find_element(:xpath, "//body").click
  @driver.find_element(:id, "session_password").click
  @driver.find_element(:id, "session_password").clear
  @driver.find_element(:id, "session_password").send_keys "12345"
  @driver.find_element(:xpath, "//body").click
  @driver.find_element(:name, "login_btn").click
  sleep 2.0
  @driver.get "http://127.0.0.1:3000/ru/menu"
  sleep 2.0
  @driver.find_element(:name, "diary_button").click
  sleep 2.0
  #@driver.find_element(:xpath, "//div[@id='dModal90']/div/div/div[2]").click
  @driver.find_element(:id, "note_diary").clear
  @driver.find_element(:id, "note_diary").send_keys "привет ещё раз"
  sleep 2.0
  @driver.find_element(:name, "ChangeDiary_button").click
  sleep 2.0
end

def element_present?(how, what)
  @driver.find_element(how, what)
end

```

```

    true
  rescue Selenium::WebDriver::Error::NoSuchElementError
    false
  end

```

```

def alert_present?()
  @driver.switch_to.alert
  true
rescue Selenium::WebDriver::Error::NoAlertPresentError
  false
end

```

```

def verify(&blk)
  yield
rescue ExpectationNotMetError => ex
  @verification_errors << ex
end

```

```

def close_alert_and_get_its_text(how, what)
  alert = @driver.switch_to().alert()
  alert_text = alert.text
  if (@accept_next_alert) then
    alert.accept()
  else
    alert.dismiss()
  end
  alert_text
ensure
  @accept_next_alert = true
end

```

```

end

describe "CheckboxAndChangeNote" do

  before(:each) do
    @driver = Selenium::WebDriver.for :edge
    @base_url = "https://www.google.com/"
    @accept_next_alert = true
    @driver.manage.timeouts.implicit_wait = 30
    @verification_errors = []
  end

  after(:each) do
    @driver.quit
    @verification_errors.should == []
  end

  it "test_checkbox_and_change_note" do
    @driver.get "http://127.0.0.1:3000/ru/signin"
    @driver.find_element(:id, "session_email").click
    @driver.find_element(:id, "session_email").clear
    @driver.find_element(:id, "session_email").send_keys "admin@gmail.com"
    @driver.find_element(:xpath, "//body").click
    @driver.find_element(:id, "session_password").click
    @driver.find_element(:id, "session_password").clear
    @driver.find_element(:id, "session_password").send_keys "12345"
    @driver.find_element(:xpath, "//body").click
    @driver.find_element(:name, "login_btn").click
    sleep 2.0
    @driver.get "http://127.0.0.1:3000/ru/menu"
    @driver.find_element(:name, "checkbox_for_note").click
  end
end

```

```

sleep 2.0
@driver.find_element(:name, "checkbox_for_note").click
sleep 2.0
@driver.find_element(:name, "menu_change_note_btn").click
@driver.find_element(:name, "note[colour]").click
@driver.find_element(:name, "note[colour]").clear
@driver.find_element(:name, "note[colour]").send_keys "#9cce69"
sleep 2.0
@driver.find_element(:name, "note[info]").clear
sleep 2.0
@driver.find_element(:name, "note[info]").send_keys "new test"
sleep 2.0
@driver.find_element(:name, "commit").click
sleep 2.0
end

```

```

def element_present?(how, what)
  @driver.find_element(how, what)
  true
rescue Selenium::WebDriver::Error::NoSuchElementException
  false
end

```

```

def alert_present?()
  @driver.switch_to.alert
  true
rescue Selenium::WebDriver::Error::NoAlertPresentError
  false
end

```

```

def verify(&blk)
  yield
rescue ExpectationNotMetError => ex
  @verification_errors << ex
end

def close_alert_and_get_its_text(how, what)
  alert = @driver.switch_to().alert()
  alert_text = alert.text
  if (@accept_next_alert) then
    alert.accept()
  else
    alert.dismiss()
  end
  alert_text
ensure
  @accept_next_alert = true
end
end

```

← 127.0.0.1:3000 HOMEWORK

ToDo Вход Настройки

## Войти

Почта  
admin@gmail.com

Пароль  
.....

Войти

Новый пользователь? [Зарегистрироваться!](#)

Рисунок 1 – страница входа

← 127.0.0.1:3000 HOMEWORK

ToDo Вход Настройки

## Зарегистрироваться

Имя  
admin@gmail.com

Почта

Пароль  
.....

Подтверждение пароля

Зарегистрироваться [Вернуться ко входу](#)

Рисунок 2 – страница регистрации



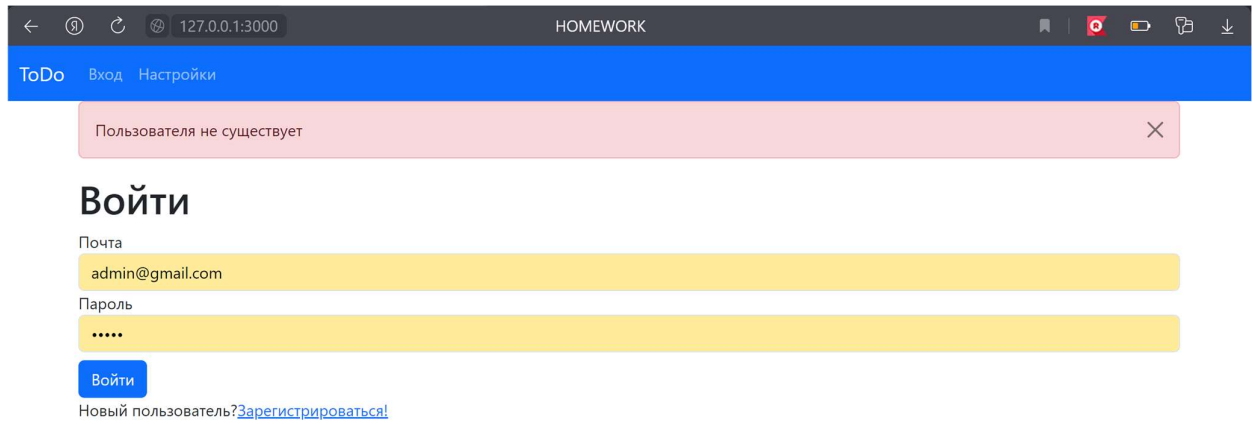


Рисунок 3 – вывод ошибки о несуществующем пользователе при входе

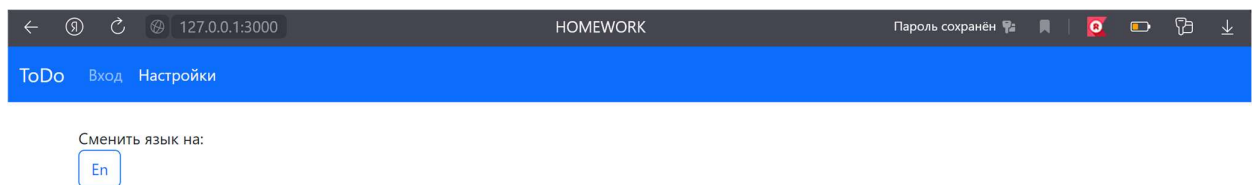


Рисунок 4 – страница настроек при входе

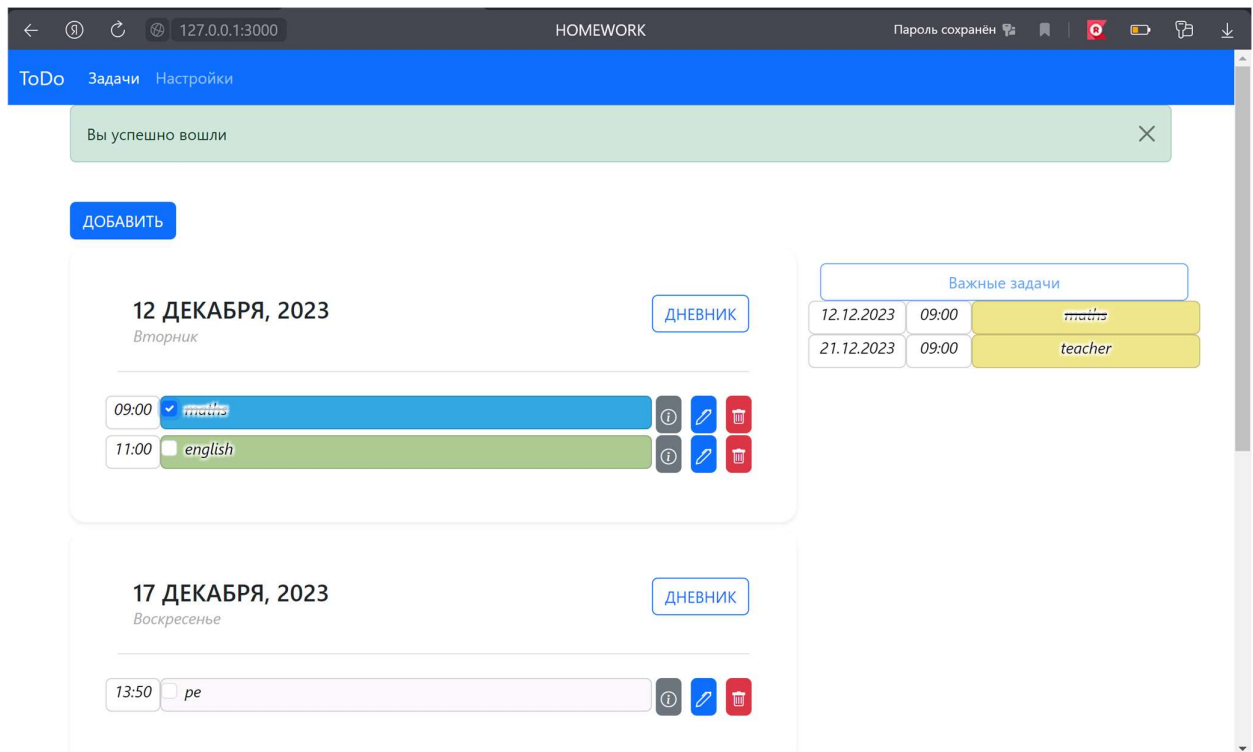


Рисунок 5 – главная страница (успешный вход)

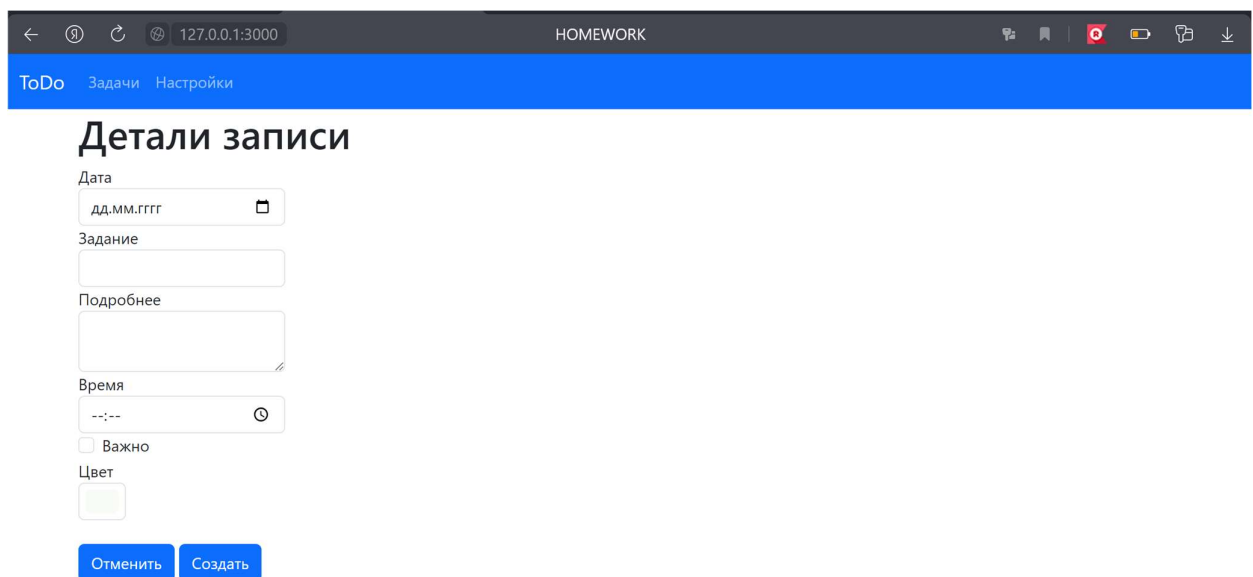


Рисунок 6 – страница создания новой записи

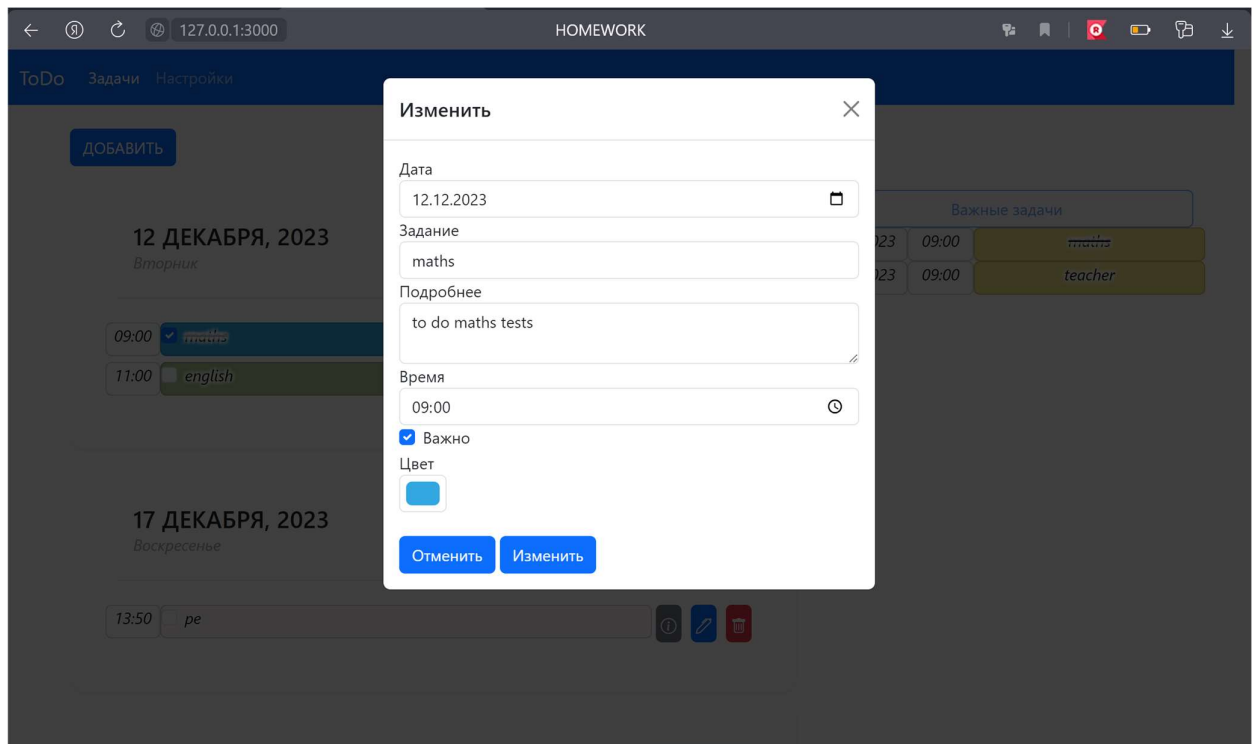


Рисунок 7 – вывод модульного окна для изменения записи на главной странице

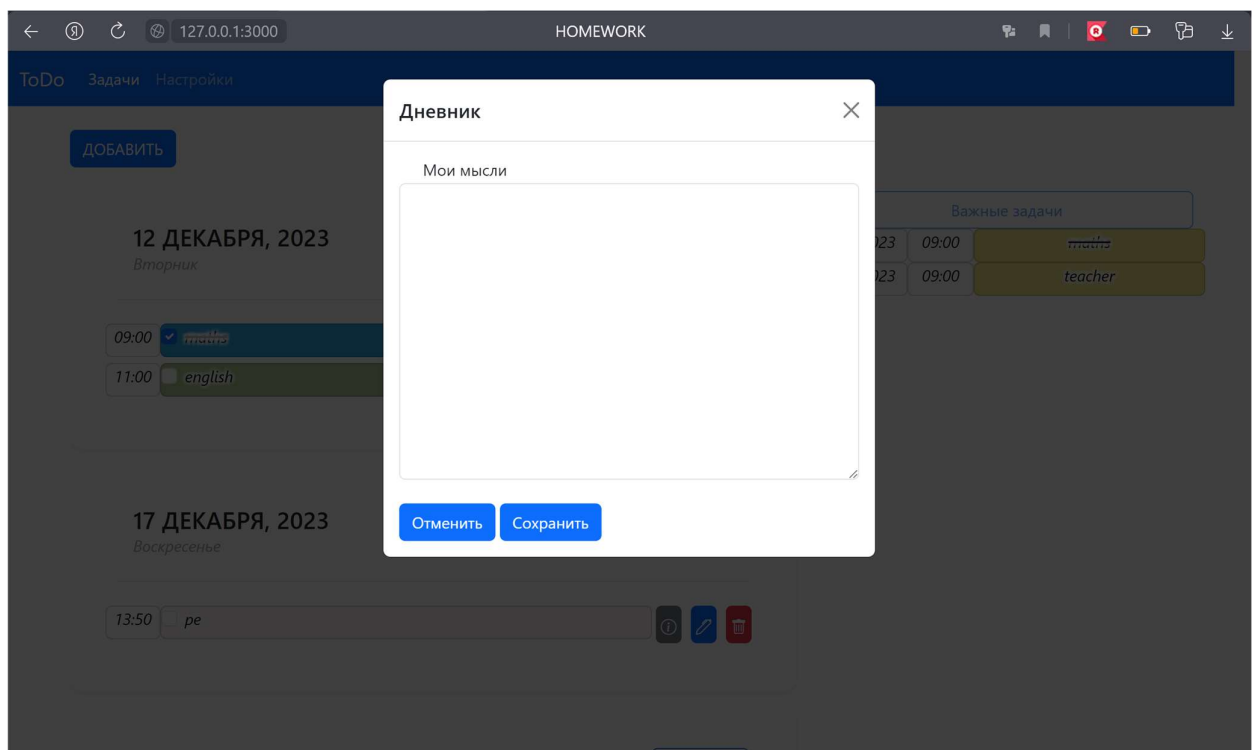


Рисунок 8 – вывод модульного окна для внесения мыслей в «дневник» на главной странице

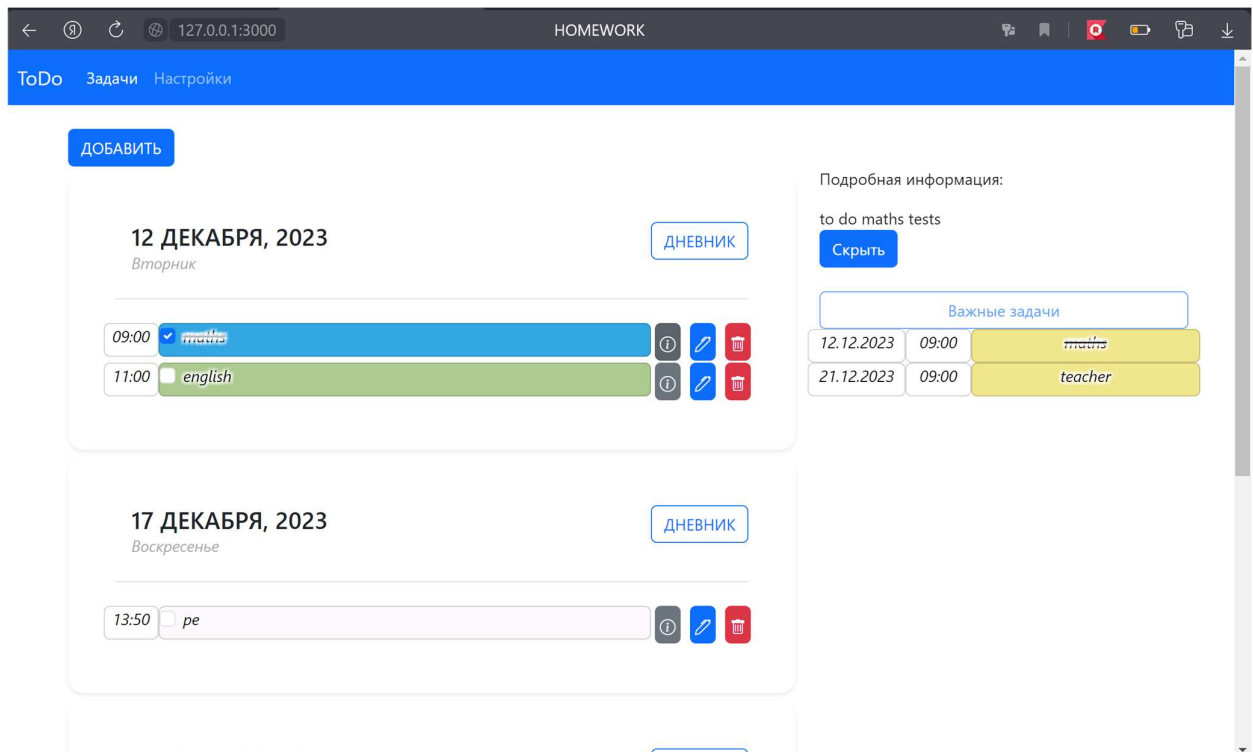


Рисунок 9 – динамический вывод подробной информации о задаче на главной странице

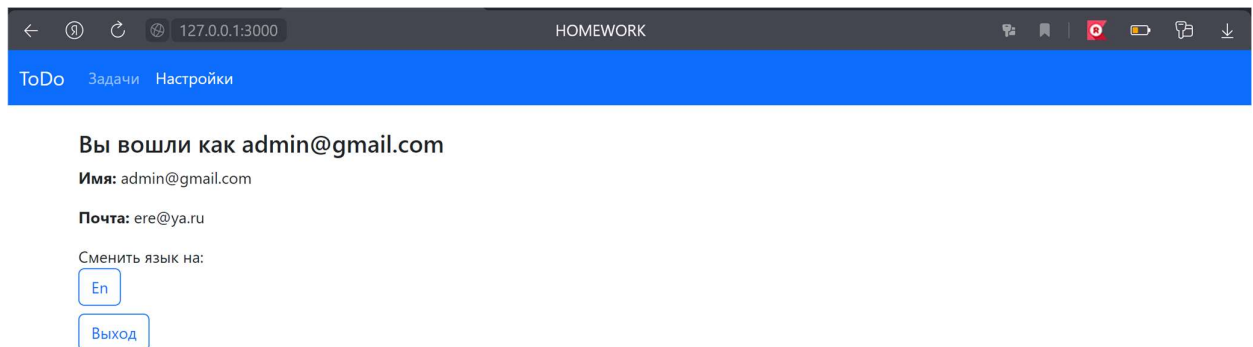


Рисунок 10 – страница настроек

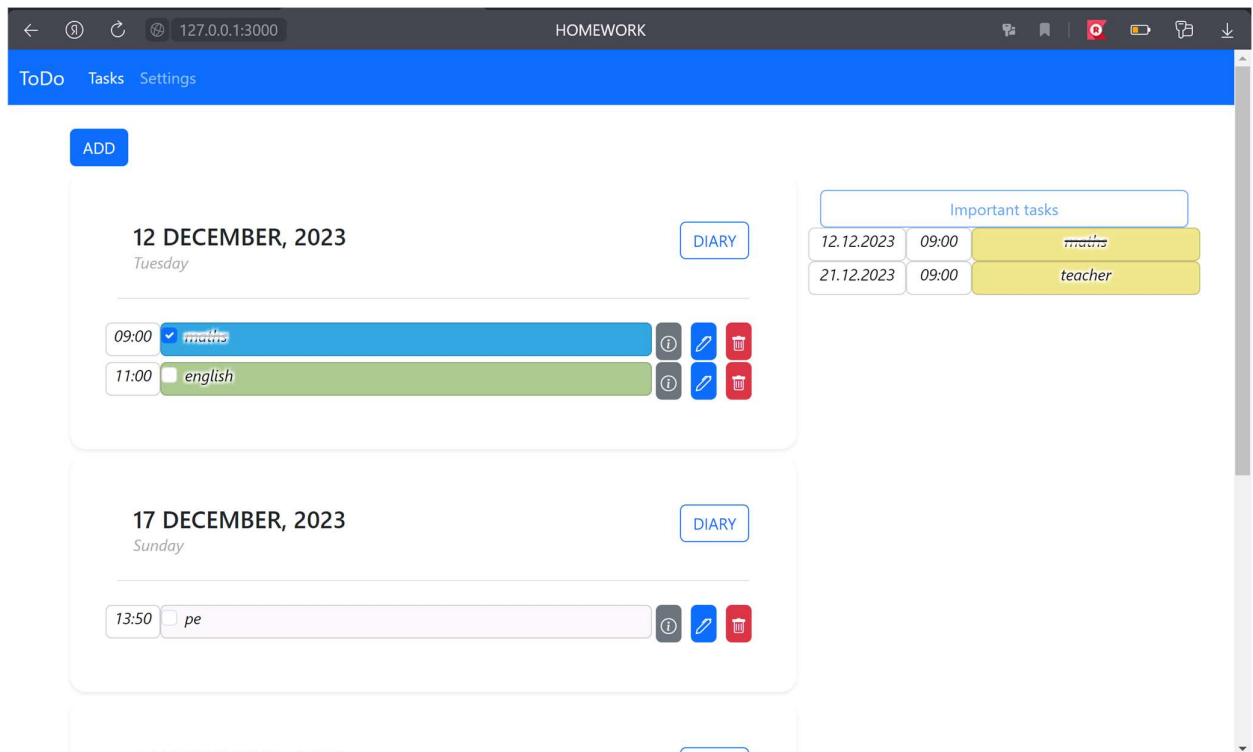


Рисунок 11 – главная страница на английском языке

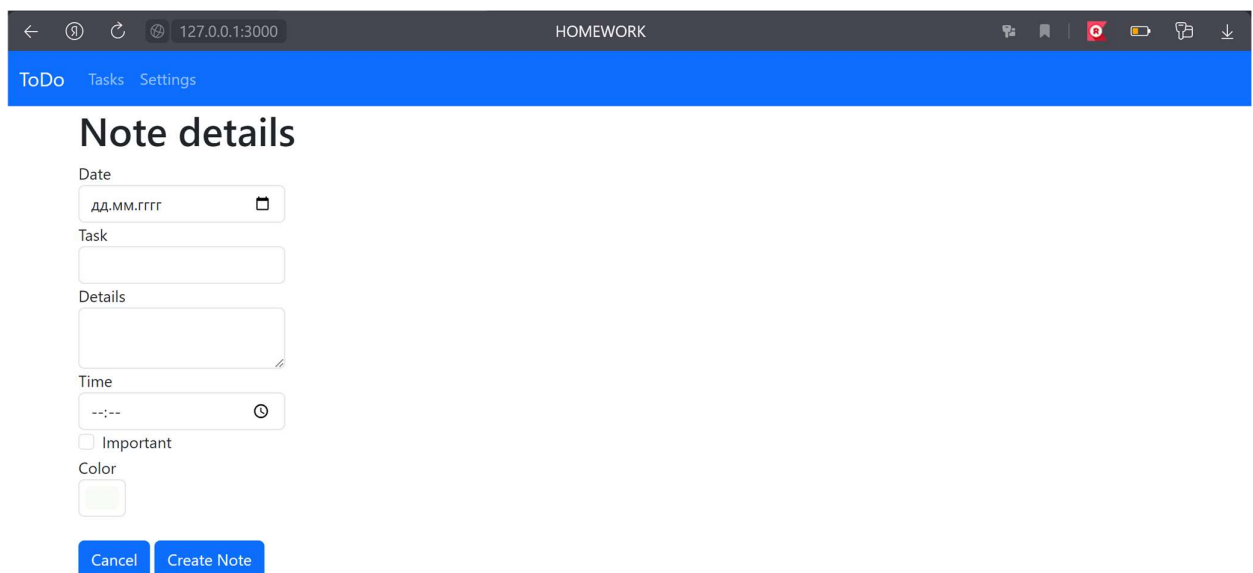


Рисунок 12 – страница создания новой записи на английском языке

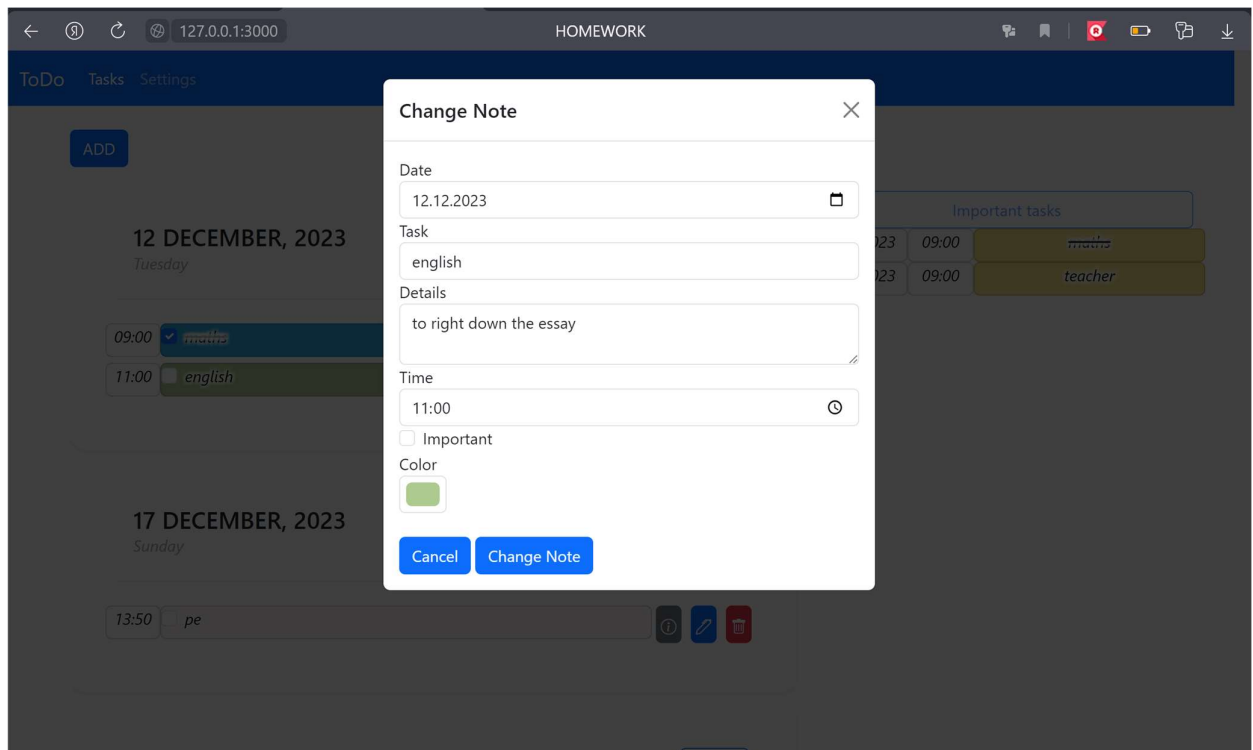


Рисунок 13 – вывод модульного окна для изменения записи на главной странице на английском языке

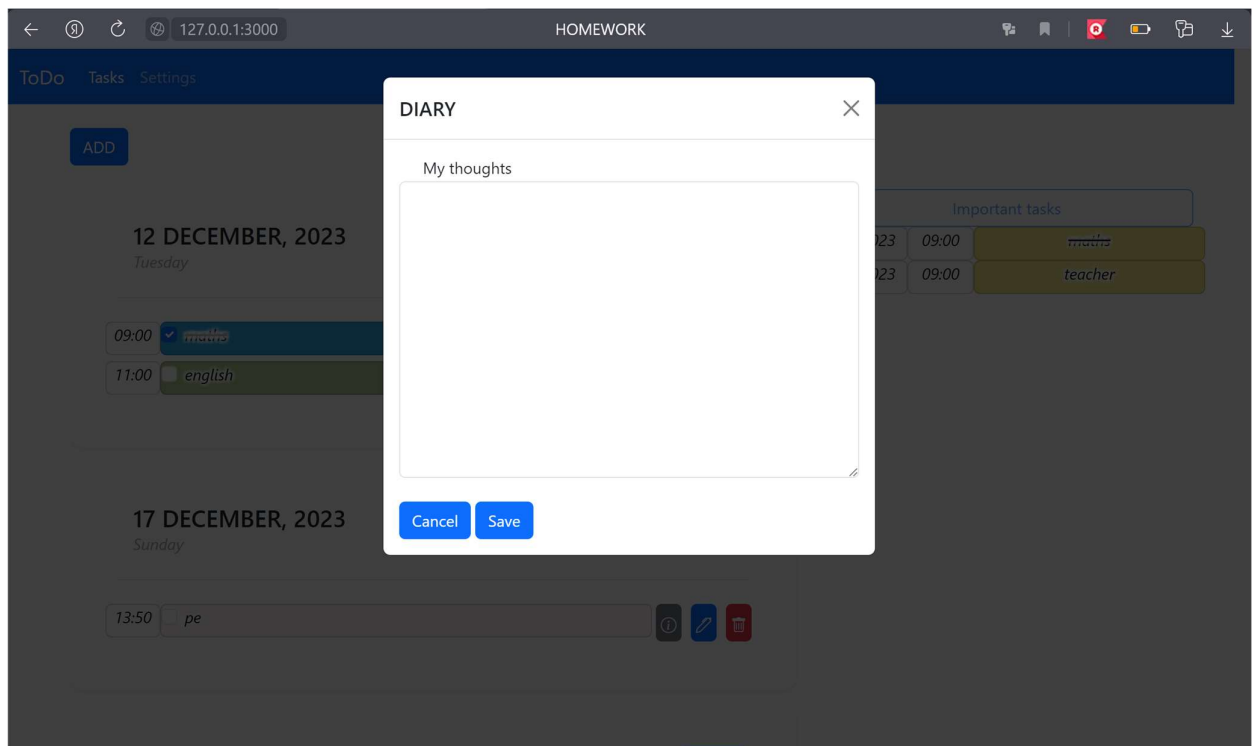


Рисунок 14 – вывод модульного окна для внесения мыслей в «дневник» на главной странице на английском языке

```
PS D:\education\3 semester\ipl\lab\TASK\diary\diary_task> rubocop app/controllers --config rubocop.yaml
Inspecting 5 files
.....

5 files inspected, no offenses detected
```

Рисунок 15 – результат работы rubocop app/controllers --config rubocop.yaml

```
"check 1 in def sign_in user"
"check 2"
"check def current user in session_helper"
"check def current user in session_helper"
.123456
"check 1 in def sign_in user"
"check 2"
"check def current user in session_helper"
"check def current user in session_helper"
."check def current user in session_helper"
.123456
"check def current user in session_helper"
.
"check def current user in session_helper"
.123456
"check 1 in def sign_in user"
"check 2"
"check def current user in session_helper"
"check def current user in session_helper"
"check def current user in session_helper"
"check def current user in session_helper"
"check def current user in session_helper"
."check def current user in session_helper"
."check def current user in session_helper"
"check def current user in session_helper"
.

Finished in 5.720687s, 1.5732 runs/s, 4.3701 assertions/s.
9 runs, 25 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 16 – результат работы интеграционных тестов

```

PS D:\education\3 semester\ipl\lab\TASK\diary\diary_task> rake test TEST=test/controllers/users_controller_test.rb
Running 7 tests in a single process (parallelization threshold is 50)
Run options: --seed 56

# Running:

.. "check def current user in session_helper"
"check def current user in session_helper"
.. "check def current user in session_helper"
.123456
"check 2"
"check 3"
"check 1 in def sign_in user"
"check 2"
.. "check def current user in session_helper"
.. "check def current user in session_helper"
.

Finished in 4.107567s, 1.7042 runs/s, 2.9214 assertions/s.
7 runs, 12 assertions, 0 failures, 0 errors, 0 skips

```

Рисунок 17 – результат работы тестов таблицы users

```

PS D:\education\3 semester\ipl\lab\TASK\diary\diary_task> rake test TEST=test/controllers/notes_controller_test.rb
Running 7 tests in a single process (parallelization threshold is 50)
Run options: --seed 36217

# Running:

"update"
"1"
nil
.. "check def current user in session_helper"
.. "check def current user in session_helper"
.123456
"check 1 in def sign_in user"
"check 2"
"check def current user in session_helper"
"check def current user in session_helper"
"check def current user in session_helper"
"check def current user in session_helper"
"check 3"
.. "check def current user in session_helper"
.. "check def current user in session_helper"
.

Finished in 4.399017s, 1.5913 runs/s, 4.0918 assertions/s.
7 runs, 18 assertions, 0 failures, 0 errors, 0 skips

```

Рисунок 18 – результат работы тестов таблицы notes

```

PS D:\education\3 semester\ipl\lab\TASK\diary\diary_task> rake test TEST=test/controllers/main_controller_test.rb
Running 2 tests in a single process (parallelization threshold is 50)
Run options: --seed 58915

# Running:

"check def current user in session_helper"
.. "check def current user in session_helper"
"check def current user in session_helper"
"check def current user in session_helper"
.

Finished in 3.484375s, 0.5740 runs/s, 0.5740 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips

```



Рисунок 19 – результат работы тестов главной страницы (main)

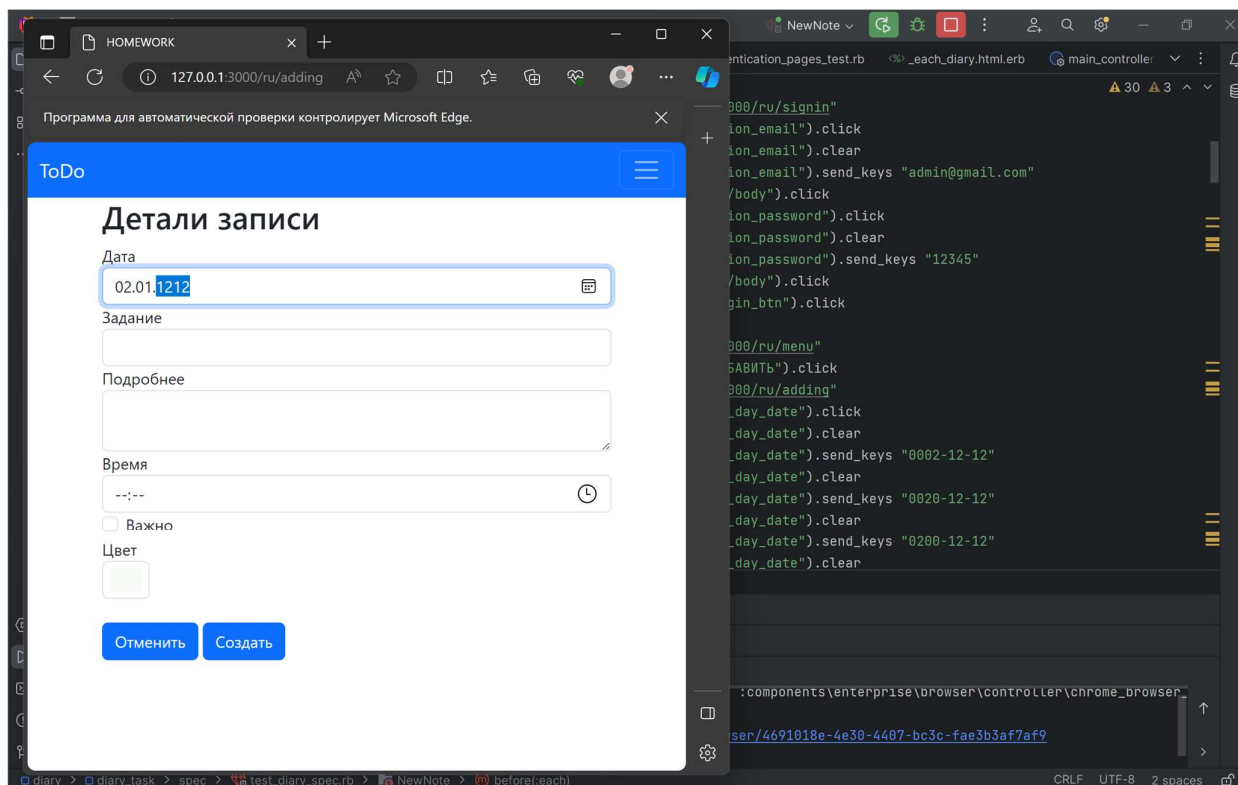


Рисунок 20 – тест Selenium на добавление новой задачи

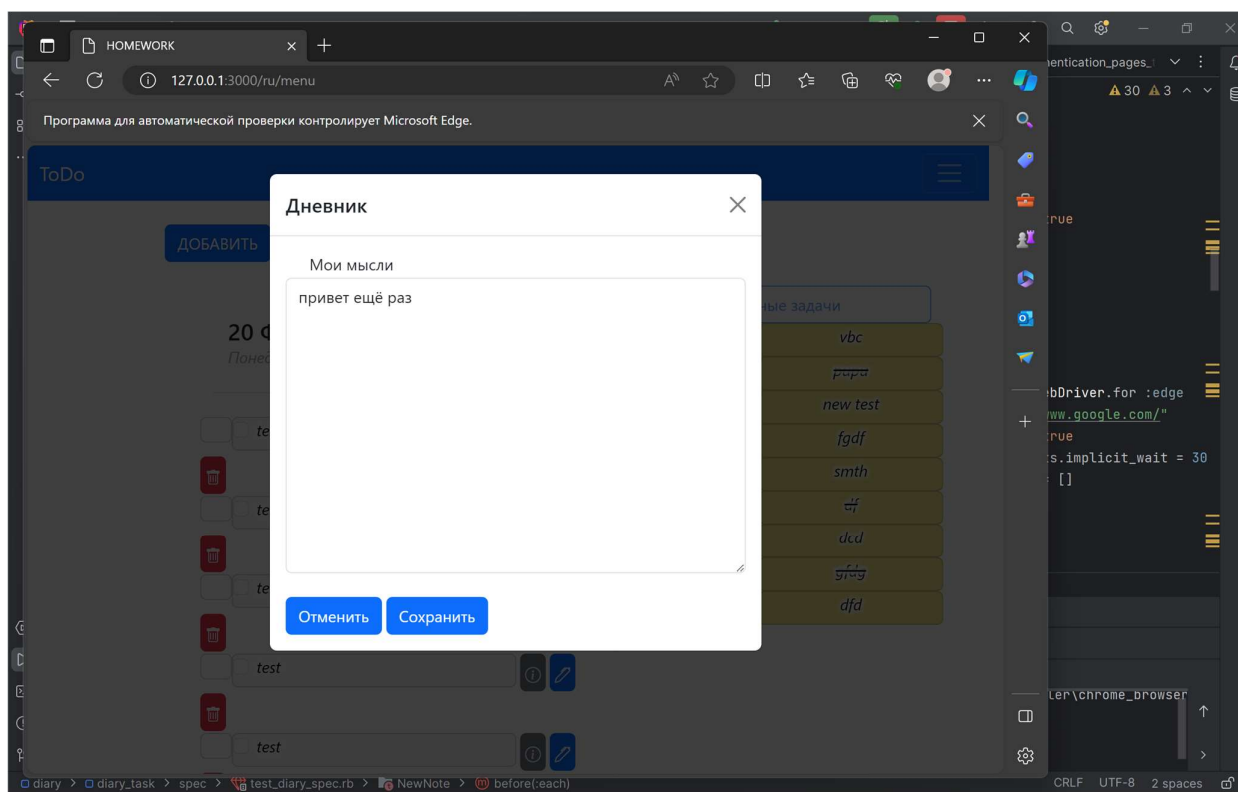


Рисунок 21 – тест Selenium на добавление записи в дневник

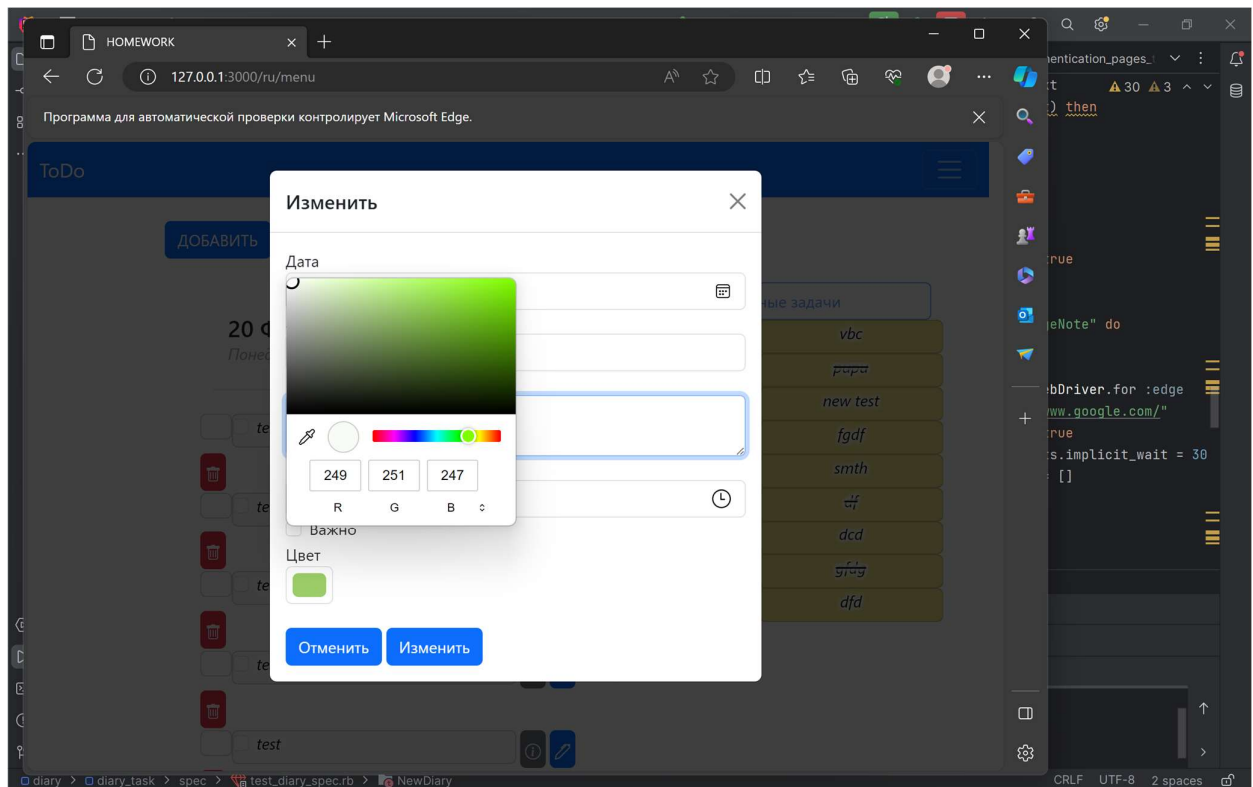


Рисунок 22 – тест Selenium на нажатие checkbox и изменения данных о записи

```
Finished in 1 minute 27.43 seconds (files took 6 seconds to load)
3 examples, 0 failures
```

Рисунок 23 – вывод в консоль об успешном прохождении всех тестов Selenium

**Вывод:** было сделано приложение для ежедневника, предоставляющее авторизованным пользователям возможность: вносить и просматривать свои задачи с учетом их важности и статуса выполнения, добавлять, удалять, редактировать каждую из них, сохранять мысли и размышления о прошедшем дне, просматривать наиболее важные задачи на отдельной части страницы при помощи удобного вывода и выбирать удобный язык для гибкости использования. С точки зрения разработки приложения была сформирована БД с таблицами users и notes со связью одно ко многим, реализована авторизация и разработаны интеграционные тесты и тесты в браузере.