|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**Отчет**

|  |  |
| --- | --- |
| **по зачетной работе** |  |

**Тема:**

Список задач и запланированных дел

**Дисциплина:** Языки интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-35 Б |  |  | Дулина И.А. |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Маняшев Э.Р. |
|  |  |  | (Подпись, дата) | (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

**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>

**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 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

return

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

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

# 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

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

# 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 }

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: %i[ 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 зарегестрирован!')

msg\_status = :danger

elsif password != password\_confirmation

msg\_text = t('Пароль для подтверждения введен неверно')

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

# 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

# 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

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'

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 зарегестрирован!: "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: "Закрыть"

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 зарегестрирован!: "Пользователь уже зарегестрирован!"

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

**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: '#00000',

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: '#00000',

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

#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

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

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

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 зарегестрировать 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 зарегестрировать

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')

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 проверочка, что не залогинились 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' => '#00000',

'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::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 "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)

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

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::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

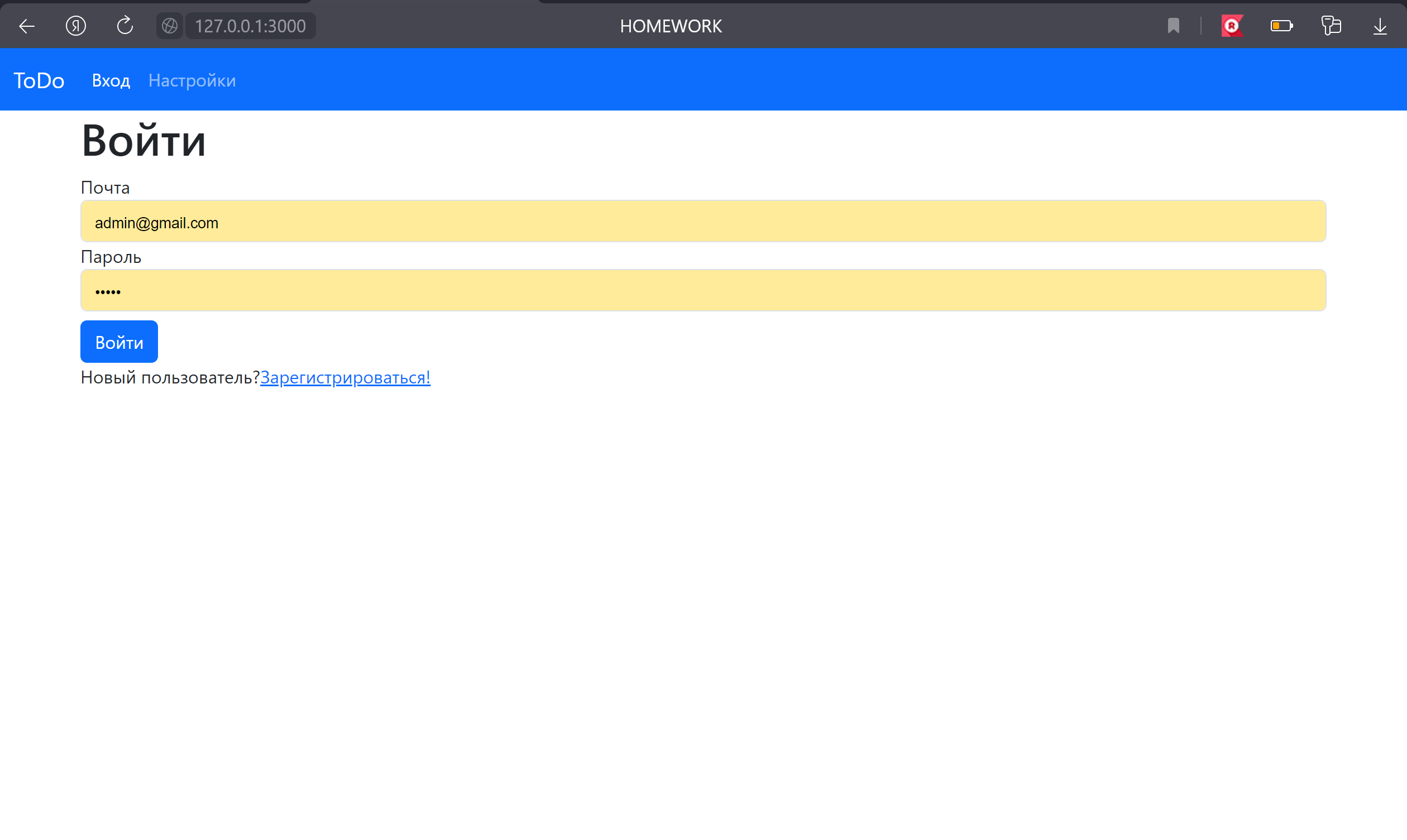


Рисунок 1 – страница входа

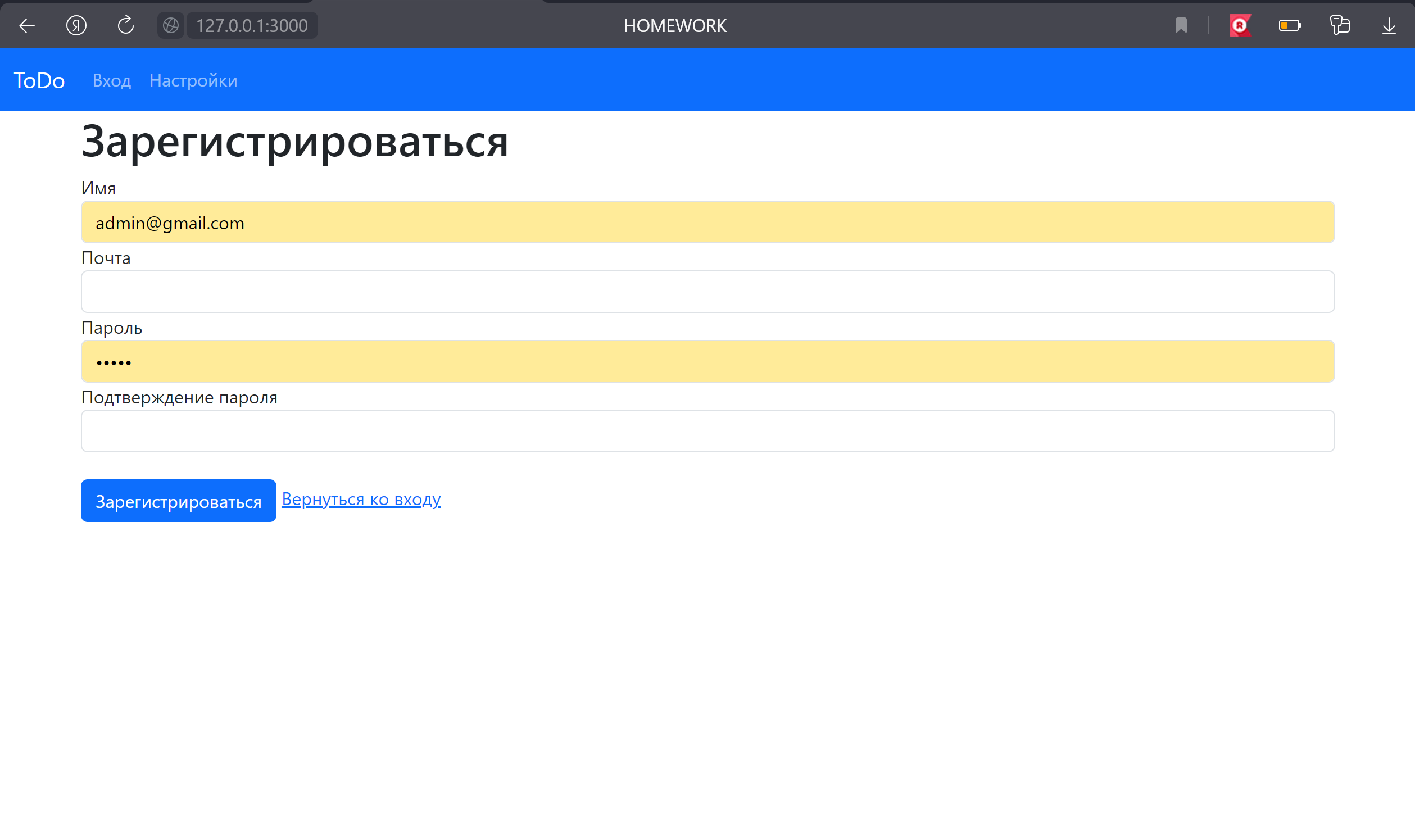


Рисунок 2 – страница регистрации

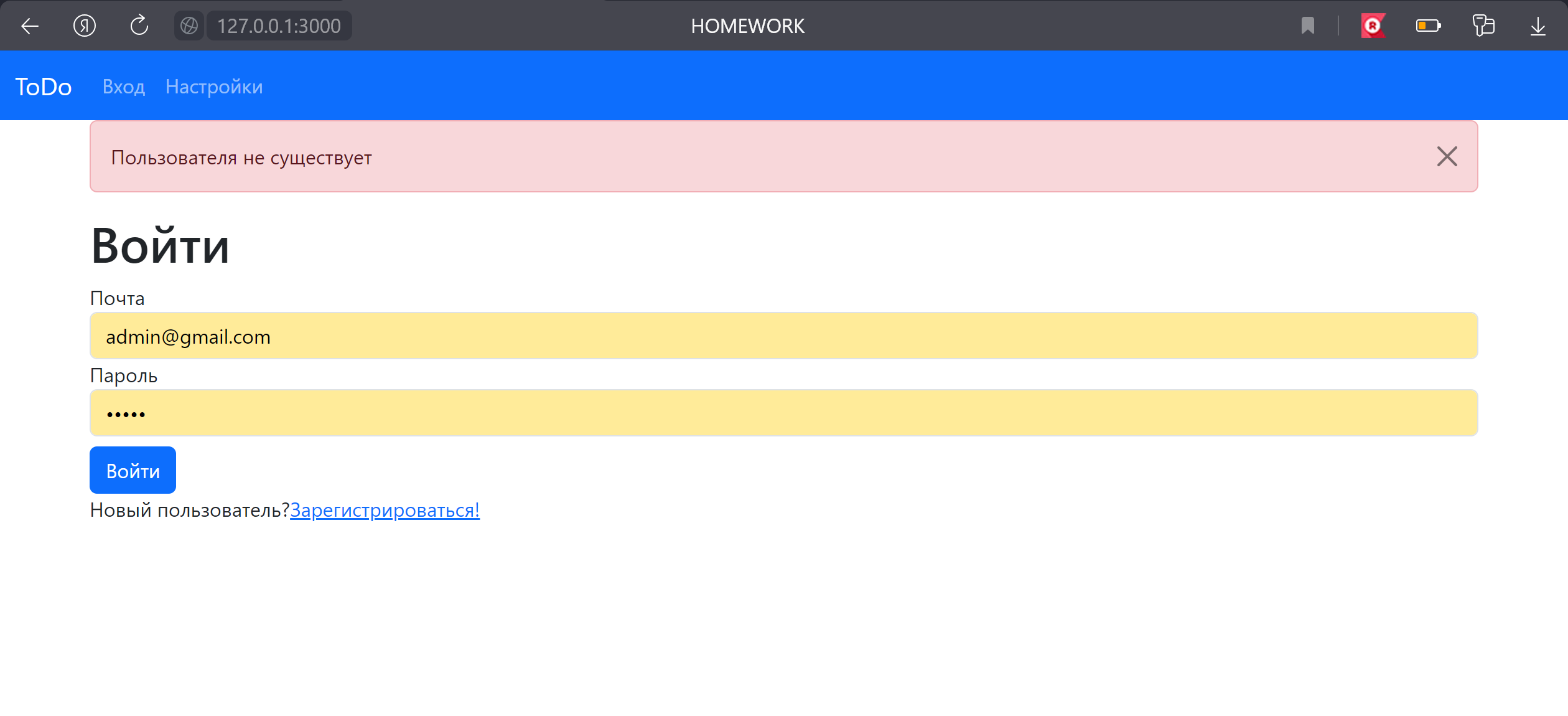


Рисунок 3 – вывод ошибки о несуществующем пользователе при входе

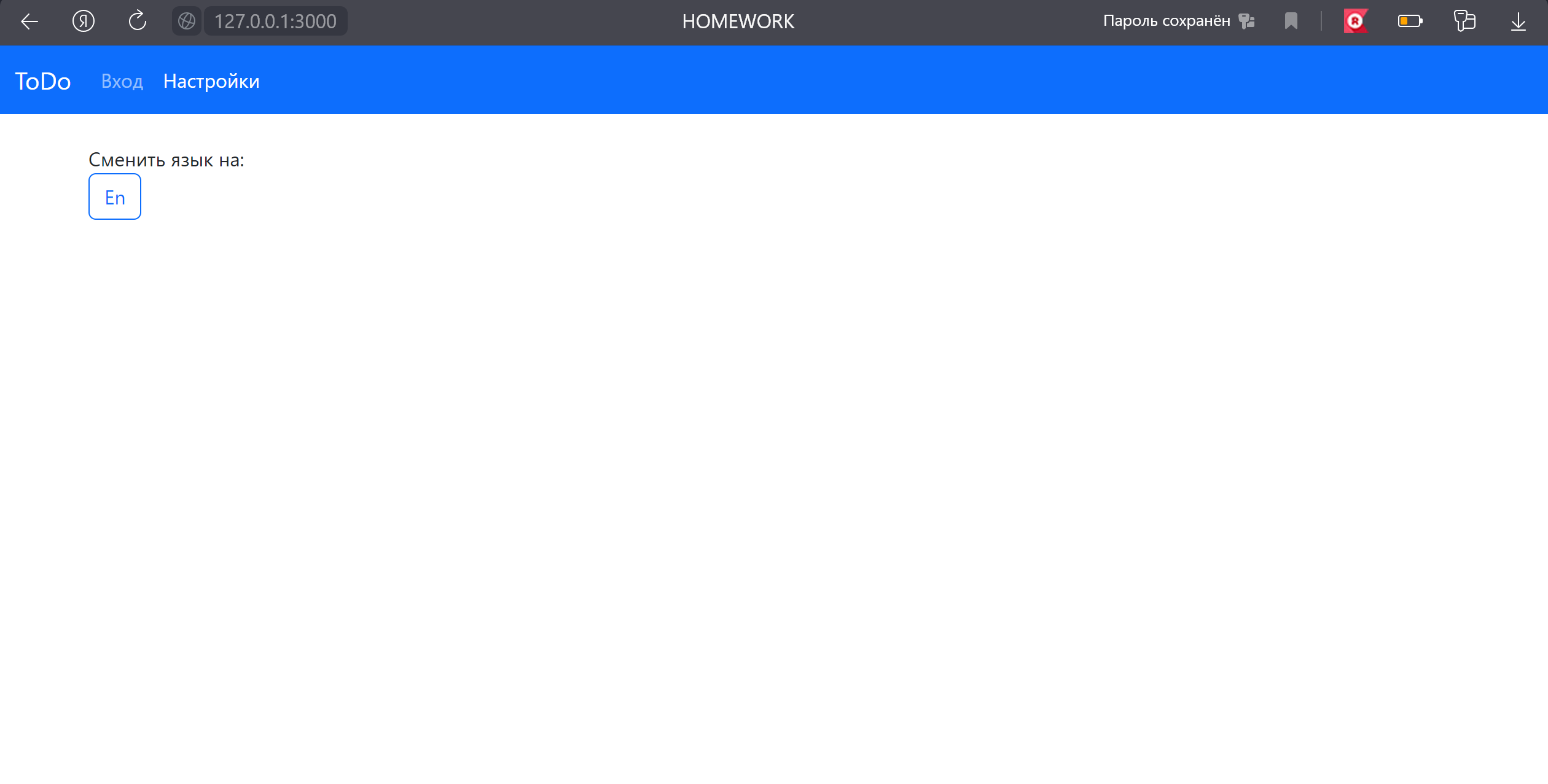


Рисунок 4 – страница настроек при входе

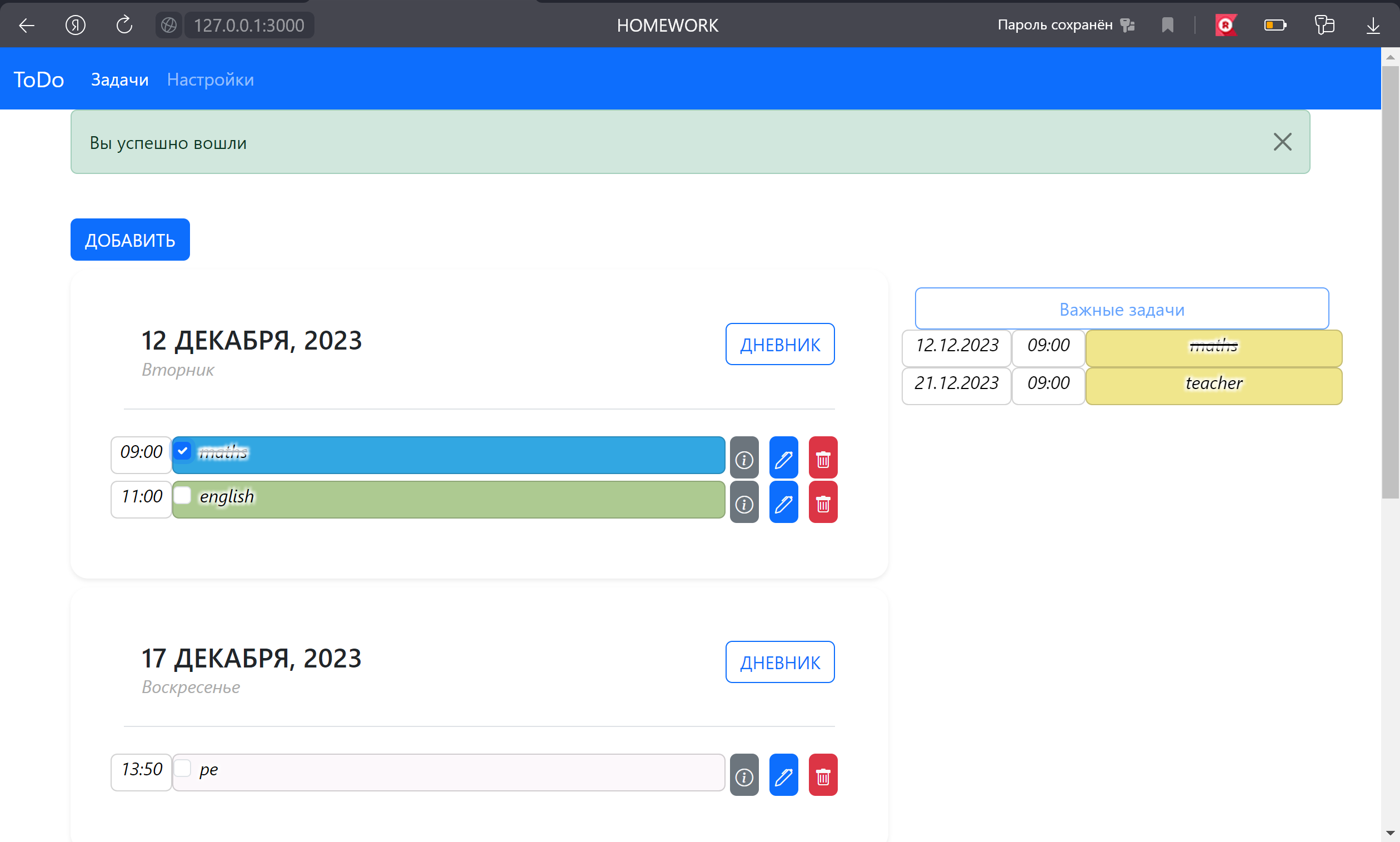


Рисунок 5 – главная страница (успешный вход)

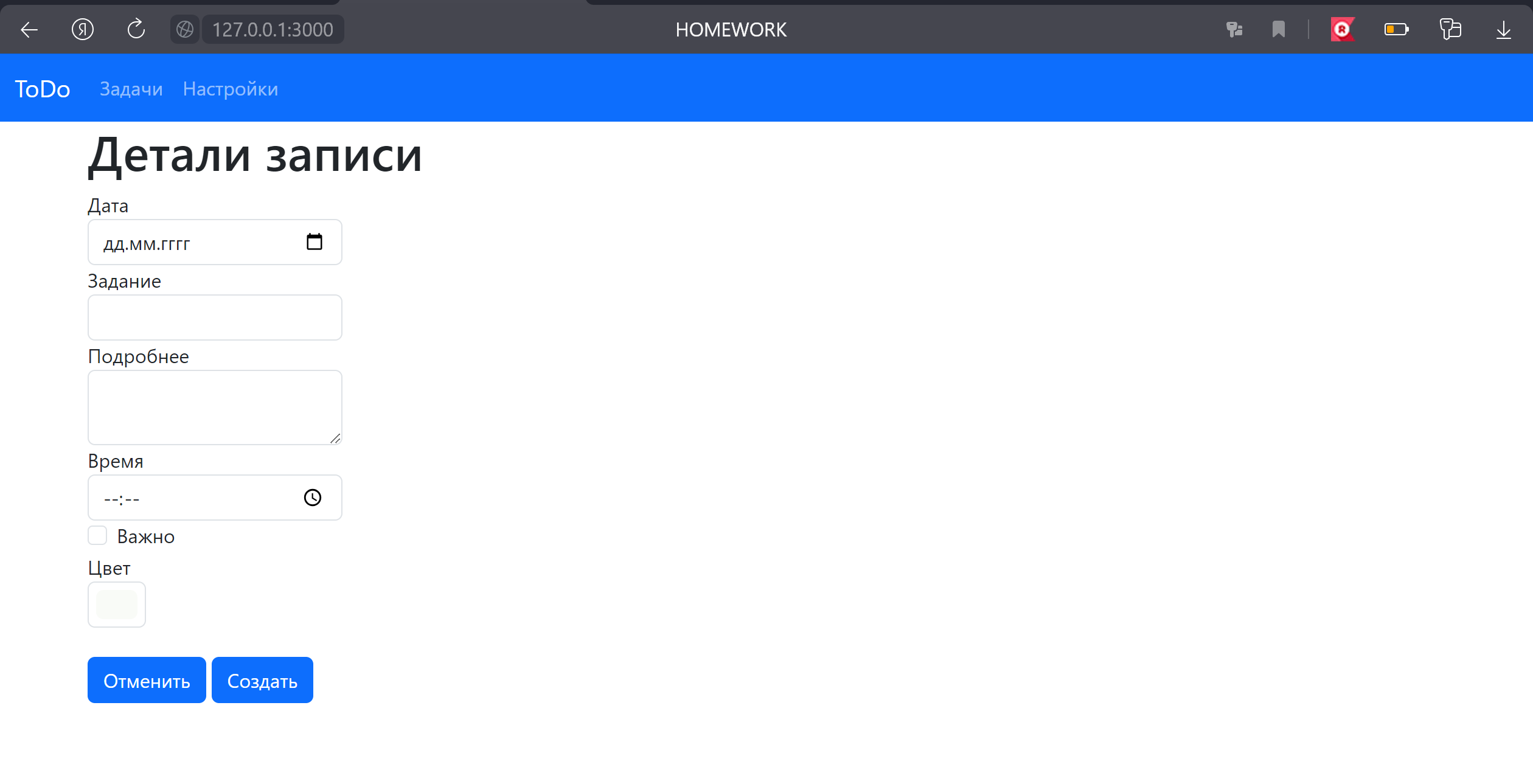


Рисунок 6 – страница создания новой записи

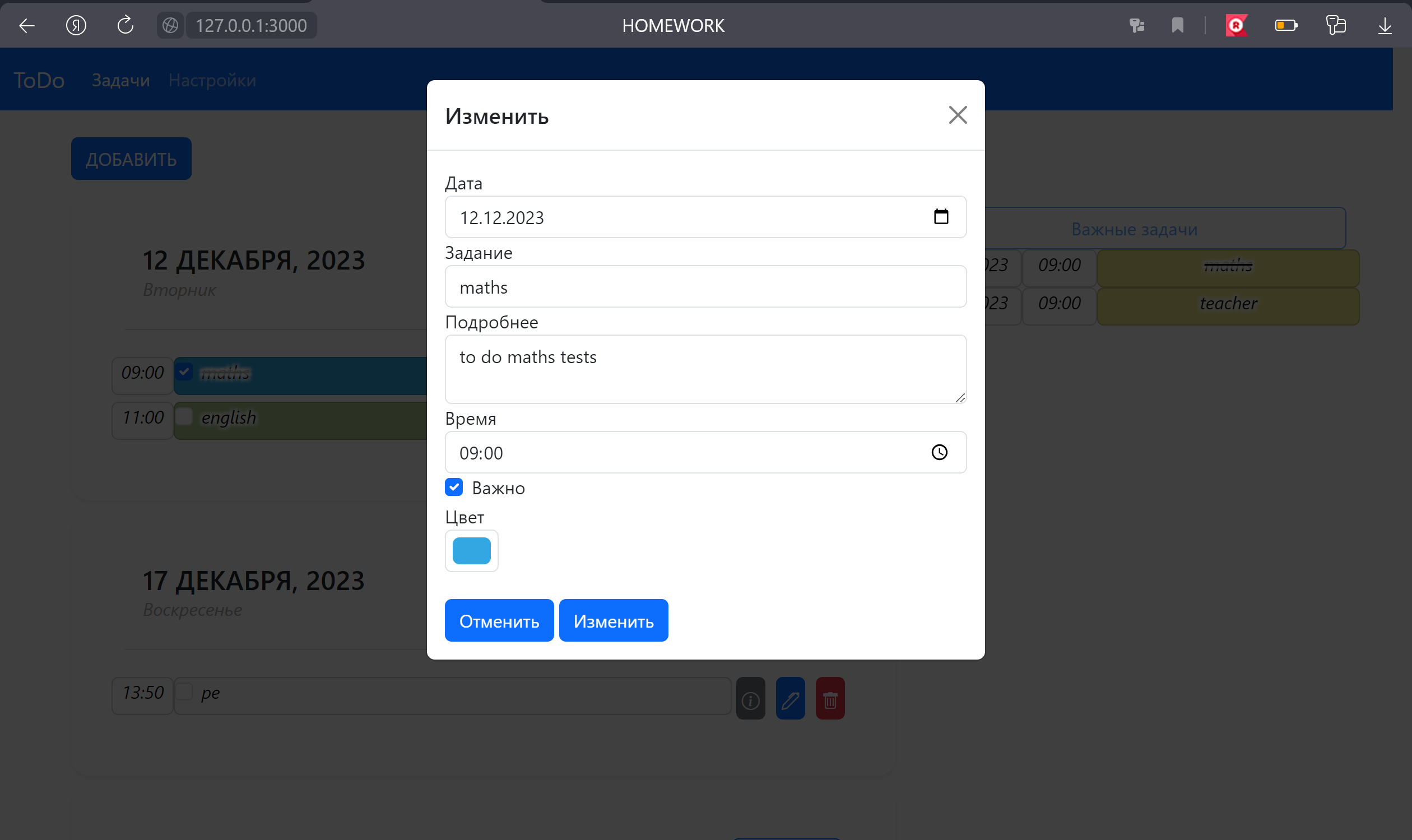


Рисунок 7 – вывод модульного окна для изменения записи на главной странице

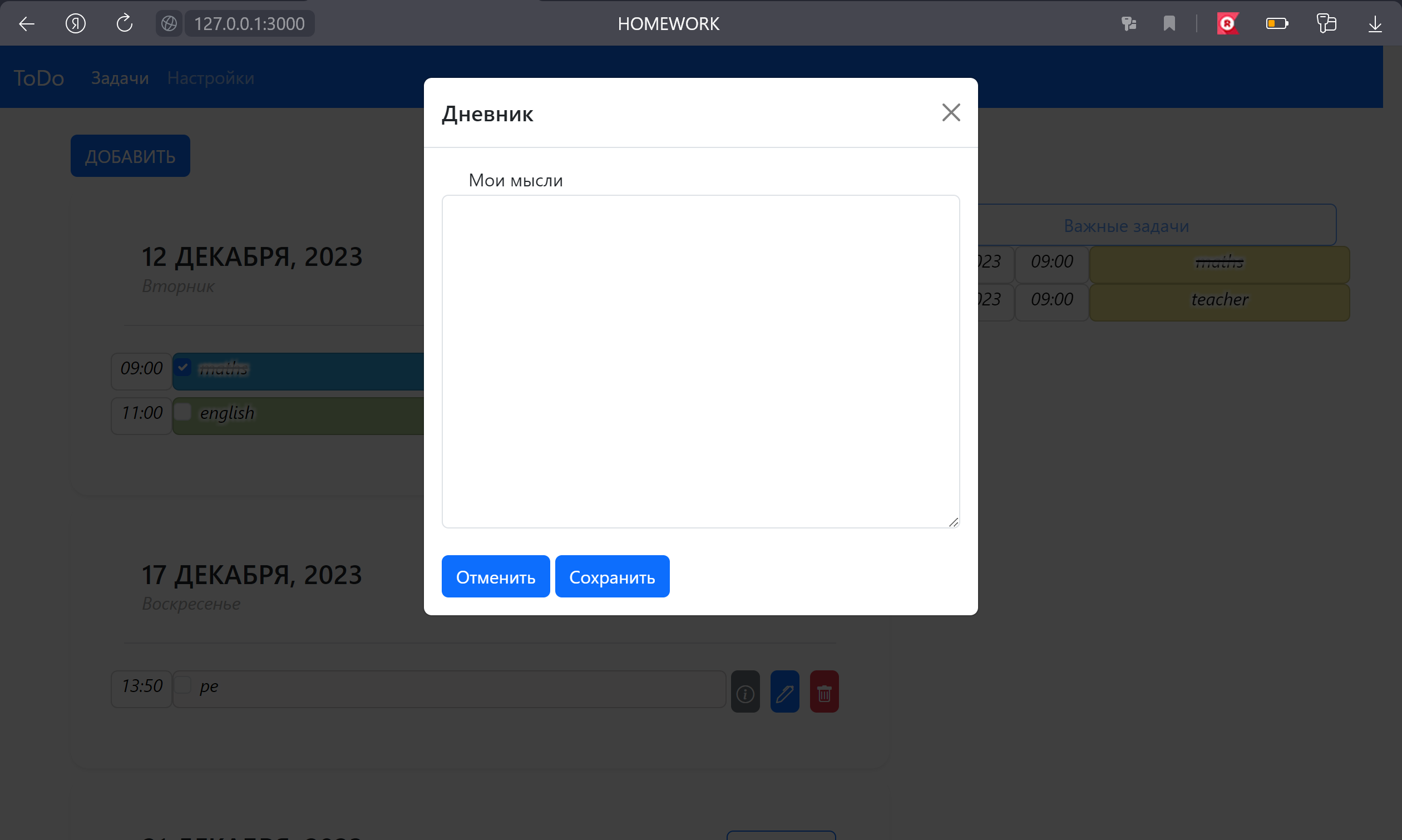


Рисунок 8 – вывод модульного окна для внесения мыслей в «дневник» на главной странице

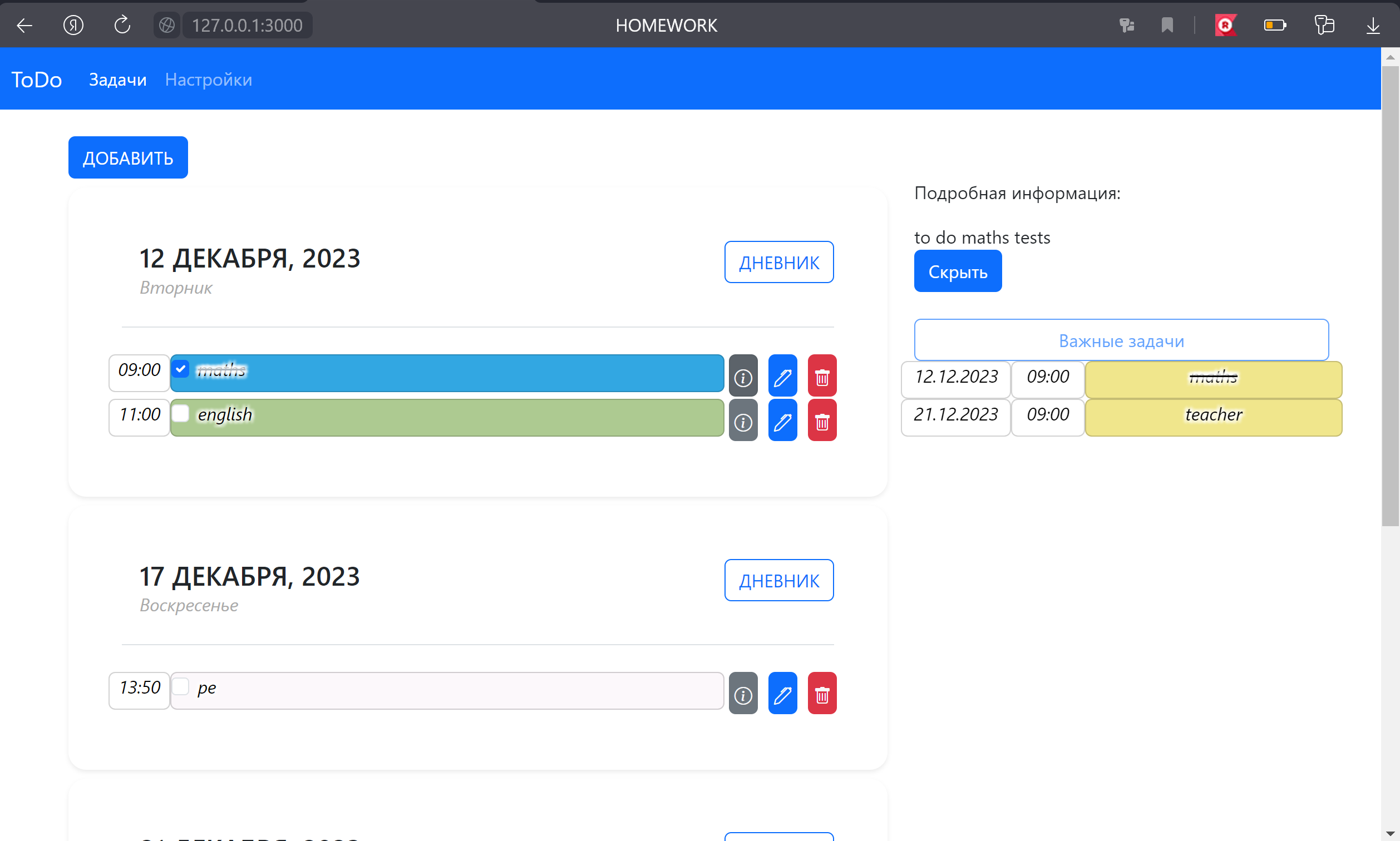


Рисунок 9 – динамический вывод подробной информации о задаче на главной странице

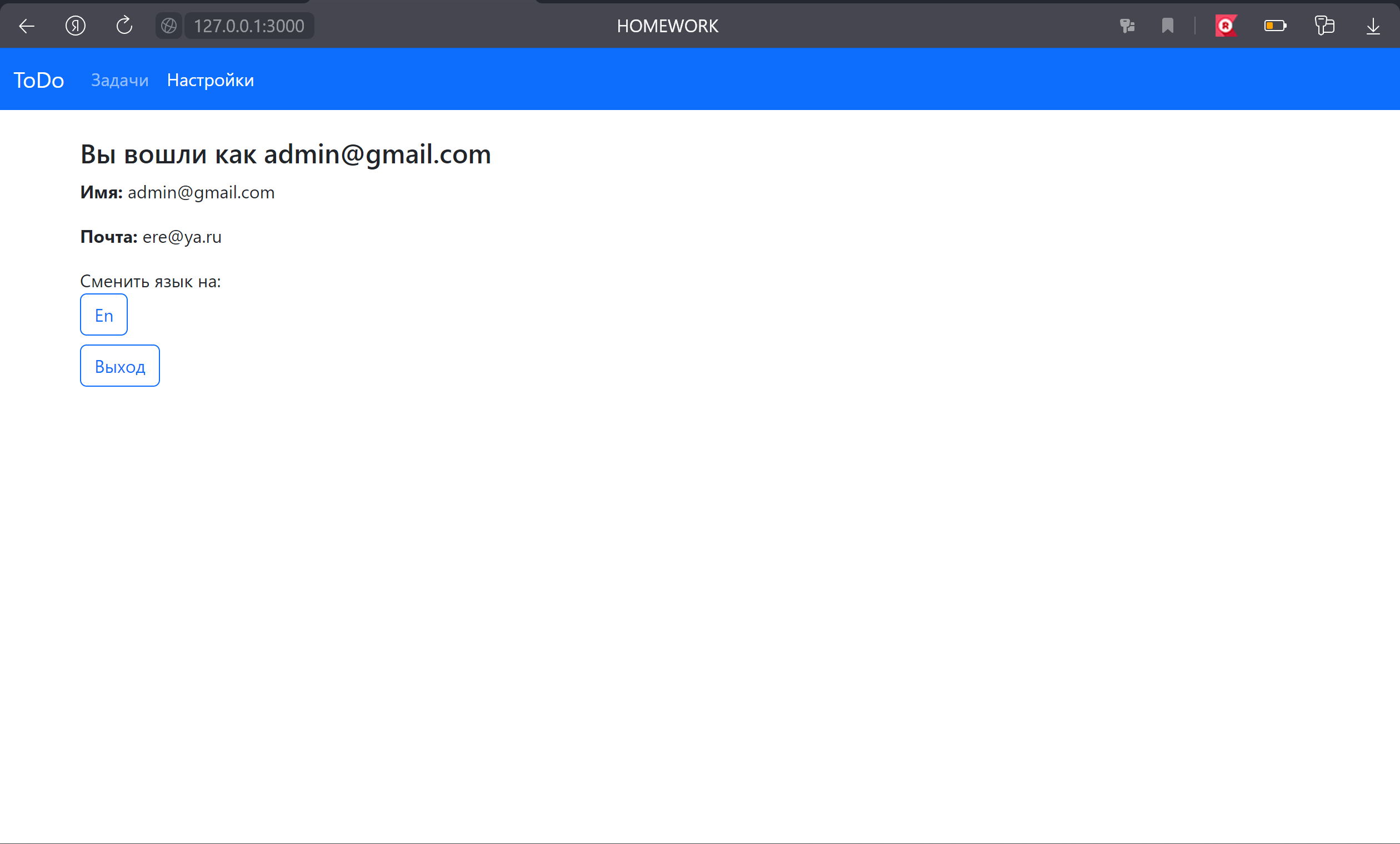


Рисунок 10 – страница настроек

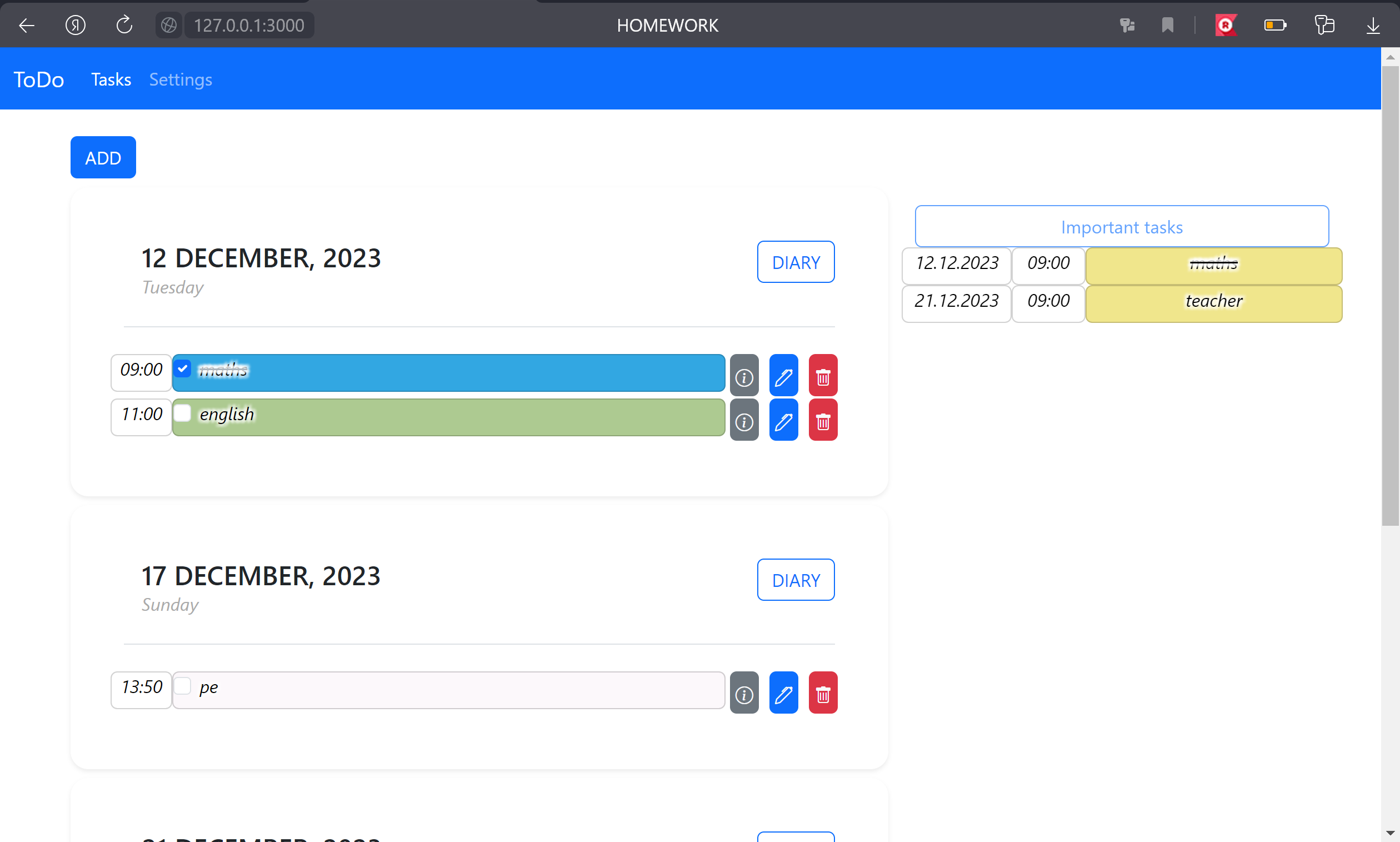


Рисунок 11 – главная страница на английском языке

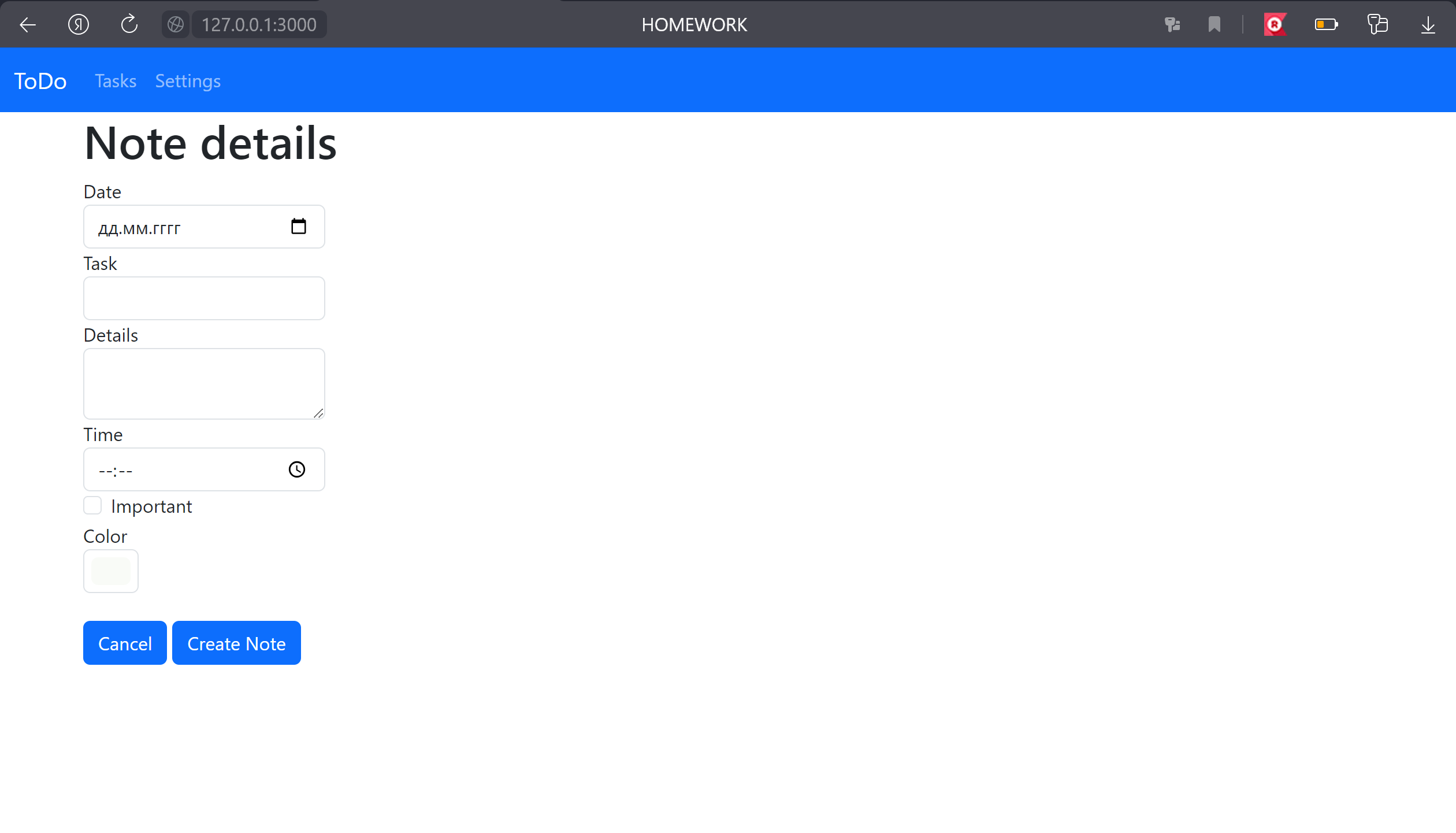


Рисунок 12 – страница создания новой записи на английском языке

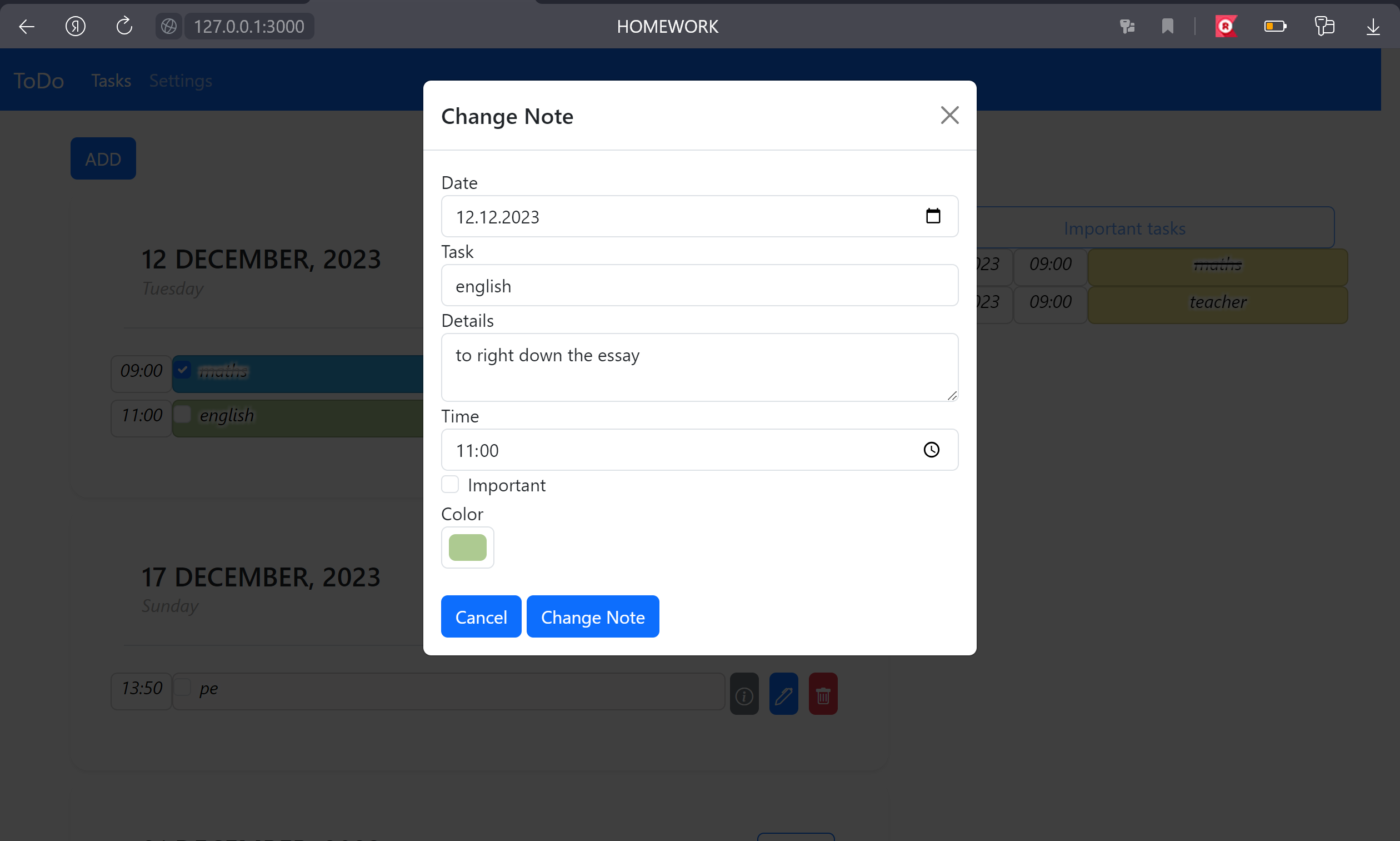


Рисунок 13 – вывод модульного окна для изменения записи на главной странице на английском языке

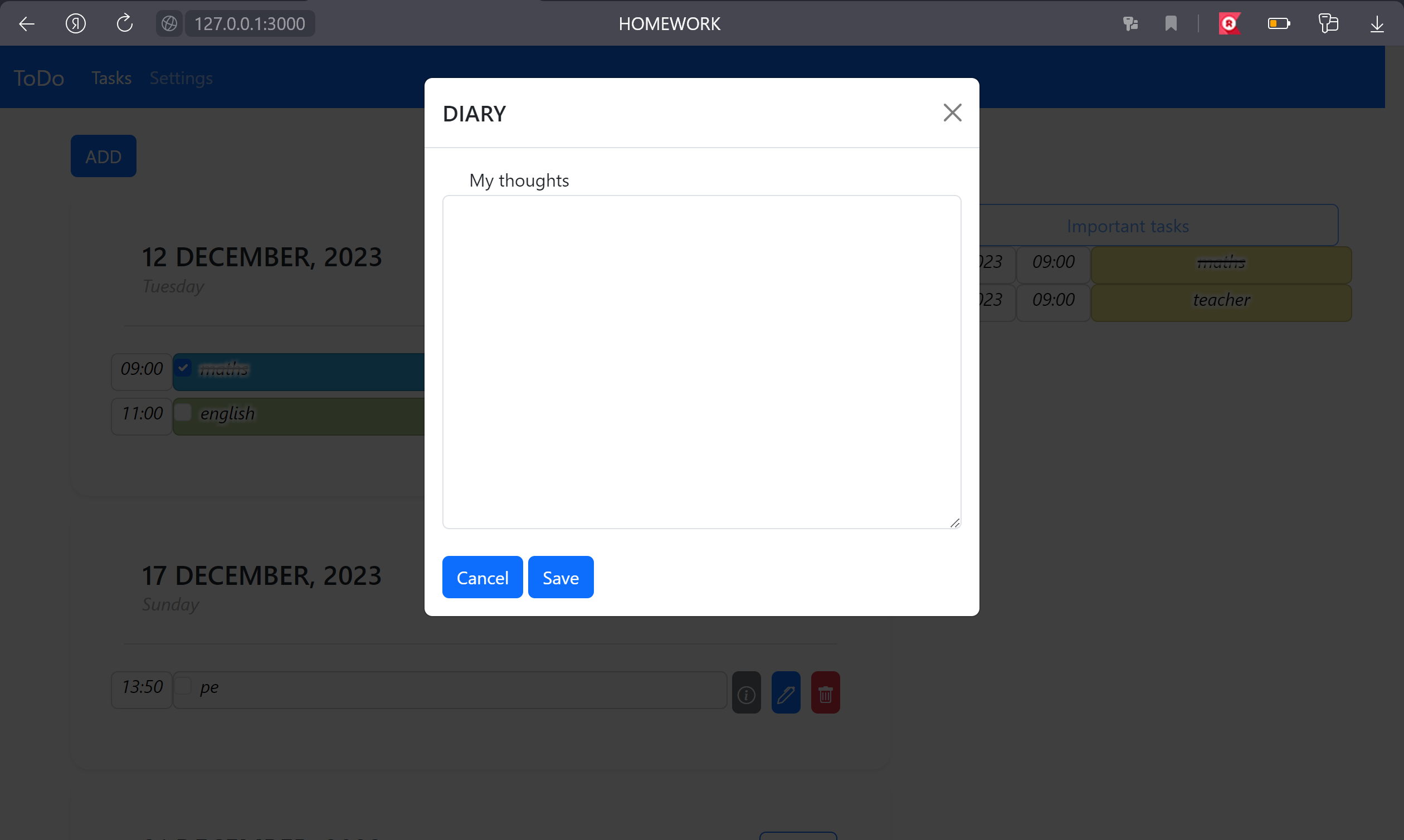


Рисунок 14 – вывод модульного окна для внесения мыслей в «дневник» на главной странице на английском языке

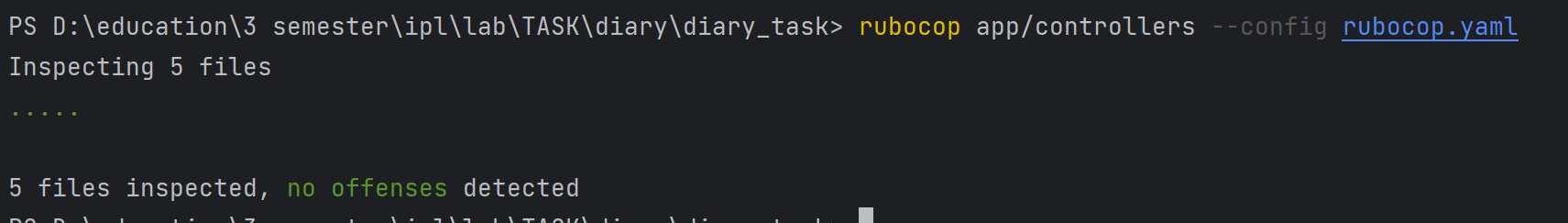


Рисунок 15 – результат работы rubocop app/controllers --config rubocop.yaml

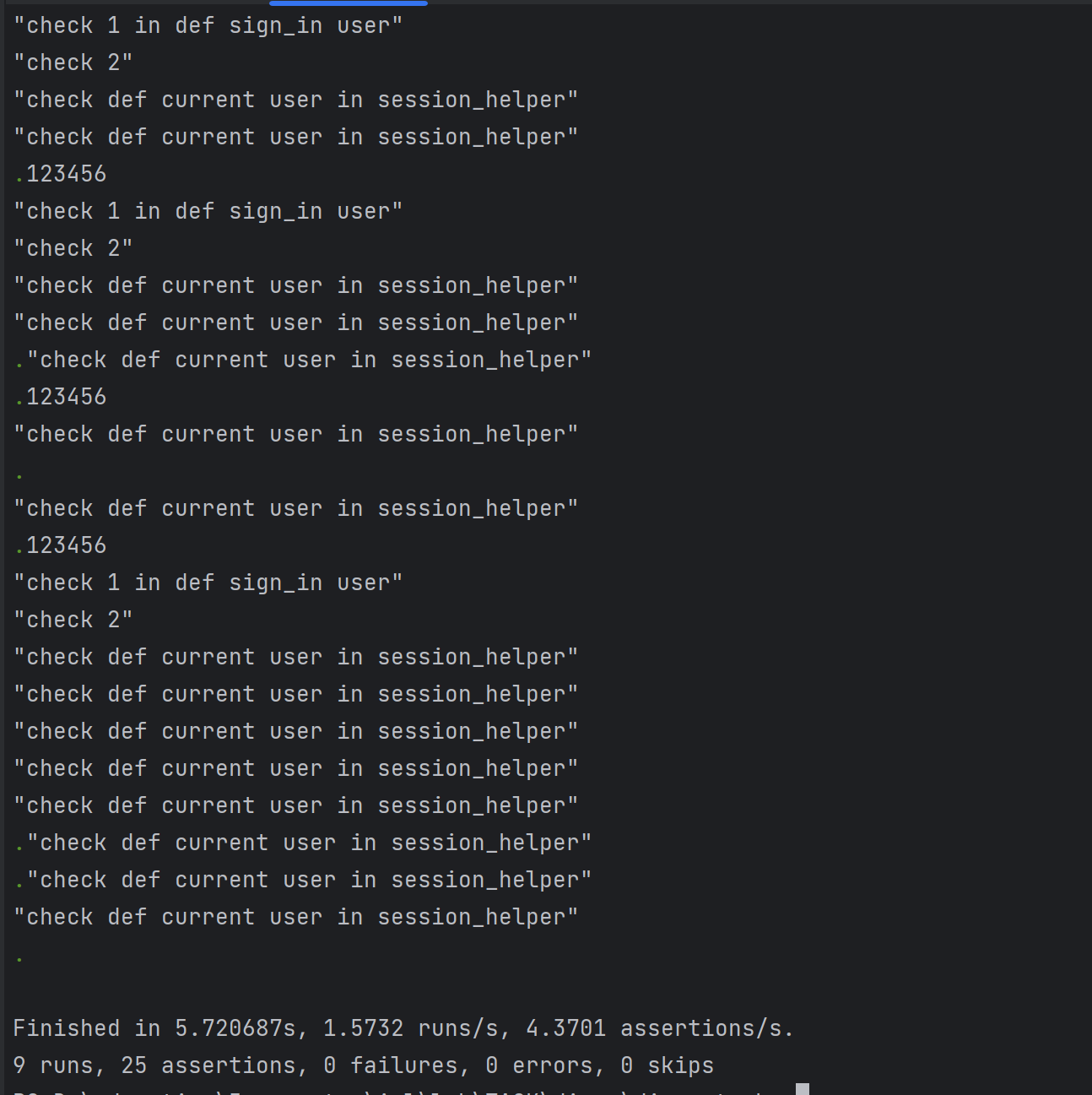


Рисунок 16 – результат работы интеграционных тестов



Рисунок 17 – результат работы тестов таблицы users



Рисунок 18 – результат работы тестов таблицы notes



Рисунок 19 – результат работы тестов главной страницы (main)

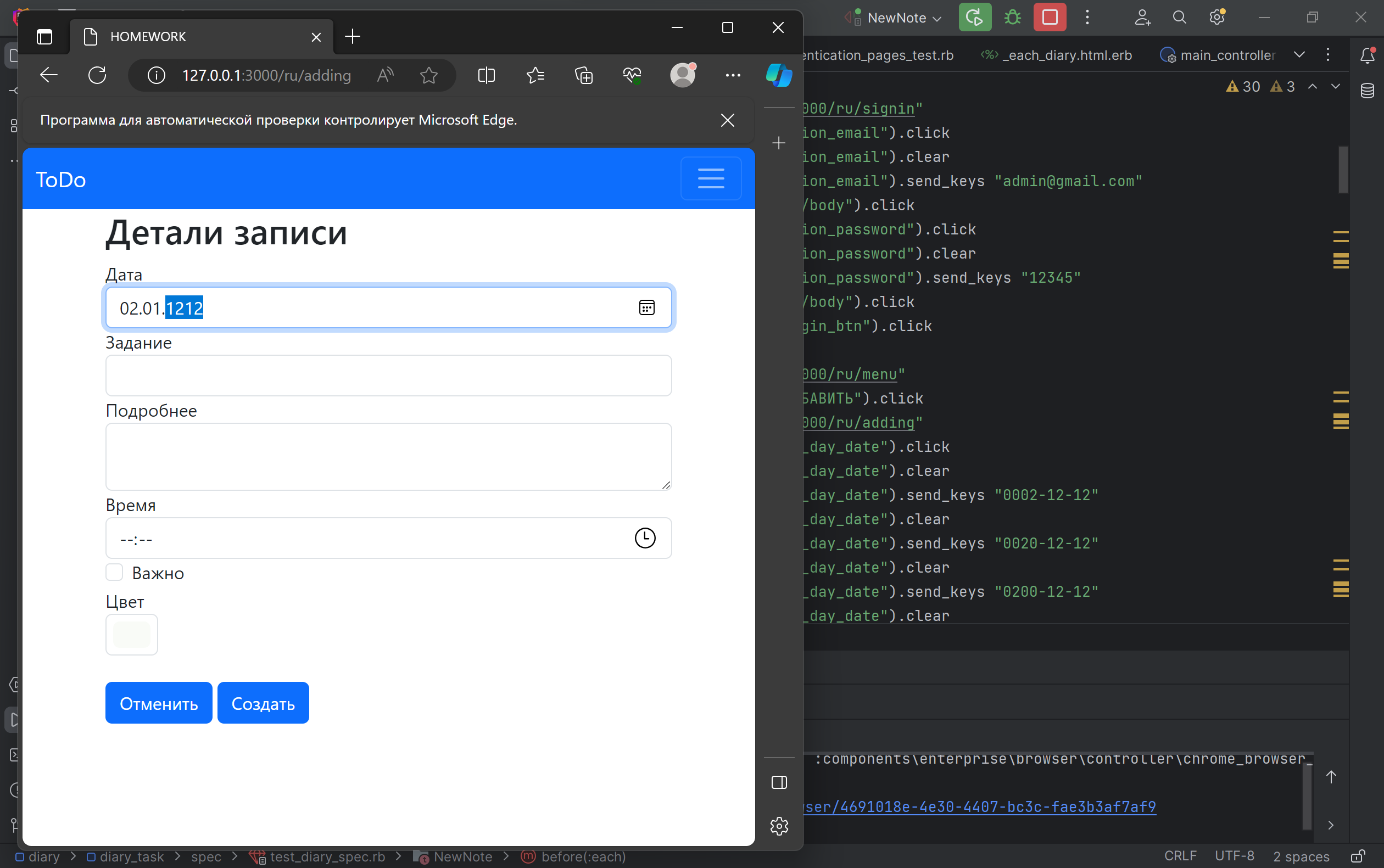


Рисунок 20 – тест Selenium на добавление новой задачи

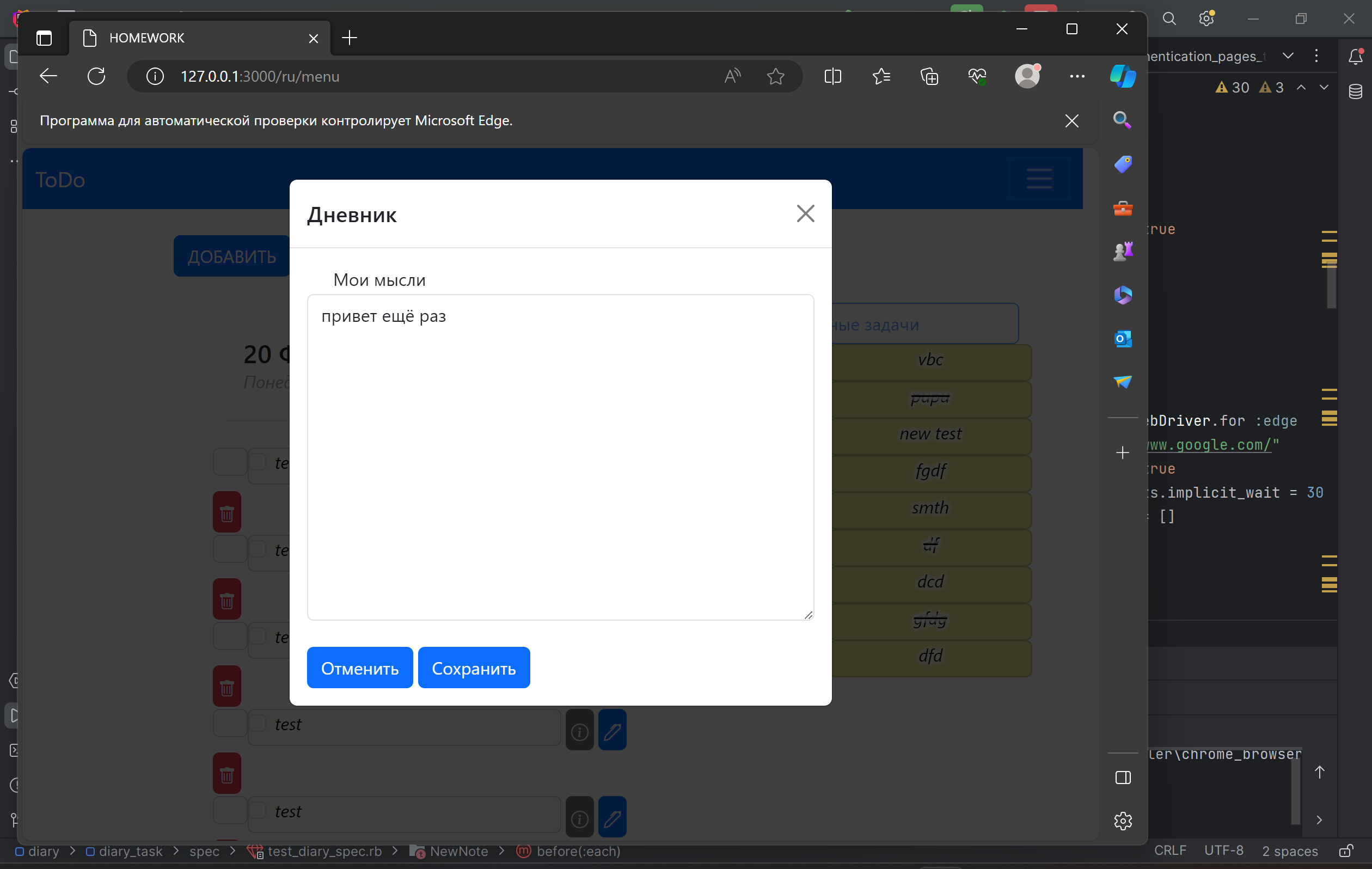


Рисунок 21 – тест Selenium на добавление записи в дневник

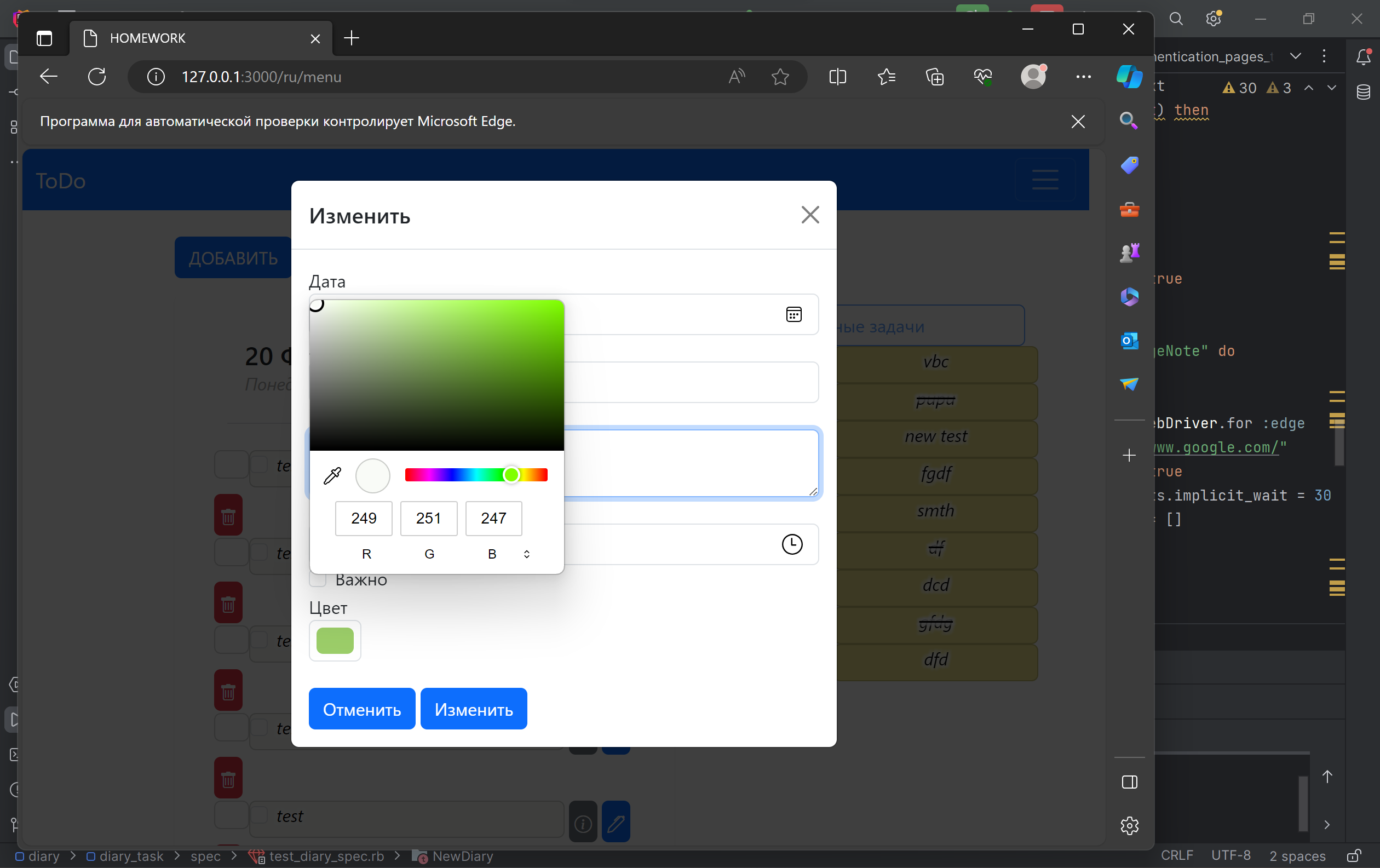


Рисунок 22 – тест Selenium на нажатие checkbox и изменения данных о записи

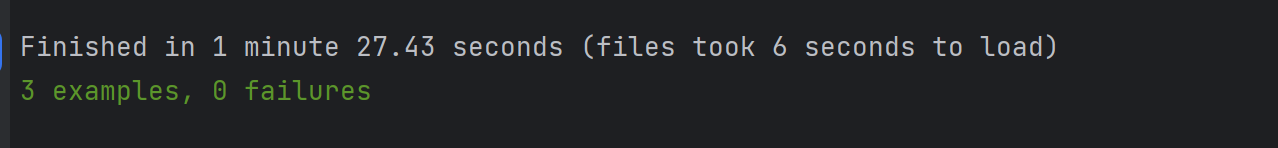


Рисунок 23 – вывод в консоль об успешном прохождении всех тестов Selenium

**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.