|  |  |
| --- | --- |
| 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 лабораторной работе №** | **12** |

**Название:**

Сессии. Выполнение авторизации. Интеграционные тесты.

**Дисциплина:** Языки Интернет-программирования



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-35Б |  | 10.11.2023 | И.А. Буханцев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Е.Ю. Гаврилова |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Задание:**

Модифицировать код приложения ЛР 8 таким образом, чтобы вычисление было невозможно без регистрации пользователя и аутентификации при помощи логина/пароля.

* Сгенерировать при помощи генератора scaffold ресурс для регистрации пользователей.
* Создать БД и выполнить миграцию соответствующим запросом rake.
* Проверить возможность добавления, редактирования информации и получения списка пользователей.
* Удалить отображение поля пароля при просмотре списка пользователей.
* Добавить контроллер сессий.
* Реализовать форму для ввода логина/пароля при обращении по адресу /. Добавить ссылку на регистрацию нового пользователя. При успешном вводе логина/пароля должно осуществляться перенаправление на страницу ввода параметров для вычисления.
* Реализовать при помощи контроллера сессий во всех действиях контроллера проверку о того, прошел ли пользователь аутентификацию или нет (с выдачей соответствующей отладочной информации).
* Вставить фильтры для запроса аутентификации.
* Подготовить интеграционный тест, позволяющий проверить регистрацию нового пользователя, вход под его именем и выполнение вычислений.
* Подготовить интеграционный тест для проверки невозможности выполнения вычислений без ввода логина/пароля.
* Проверить маршруты приложения с помощью rake routes и убрать лишние. Обеспечить доступ при обращении по адресу /.

**Цель:** Получение практических навыков в создании веб-приложений, использующих аутентификацию. Получить навыки написания интеграционных тестов приложений.

Запросы на добавление, изменение и вывод данных:

**INSERT** **INTO** users (email, password\_digest, remember\_token, created\_at, updated\_at)

**VALUES** ('test@email.com', '$2a$12$fPPfThxkvqZwkVqpNbc3u.ehzJv6v3NanqPkzQyPK5TiCx3vr/joO', 'b95a184d03b82f167ef75d6668096eb1e6dd4d2a', **CURRENT\_TIMESTAMP**, **CURRENT\_TIMESTAMP**);

**UPDATE** users

**SET** password\_digest='123456789'

**WHERE** id = 1;

**SELECT** \*

**FROM** users;

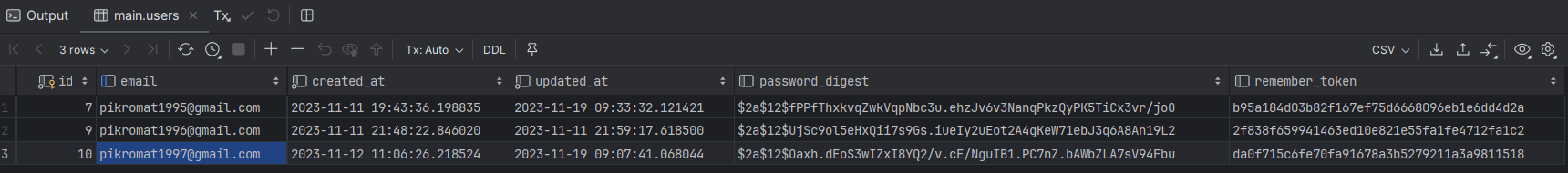


Рисунок 1 – вывод таблицы БД

Демонстрация таблицы без пароля:

**CREATE** **VIEW** SHOW\_USERS **AS**

**SELECT** id, email, created\_at, updated\_at

**FROM** users;

**SELECT** \*

**FROM** SHOW\_USERS

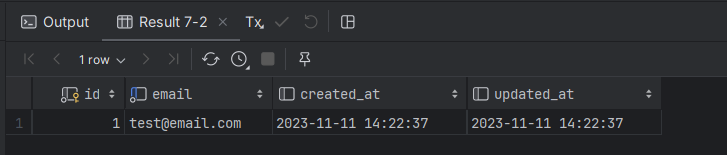
****

Рисунок 2 – вывод запроса

**stylesheets/application.scss**

*/\**

*\*= require\_tree .*

*\*= require\_self*

*\*/*

@import "bootstrap";

table {

**border**: 3px solid black;

**border-collapse**: collapse;

**width**: 20%;

}

td {

**border**: 1px solid #333;

**text-align**: center;

}

**stylesheets/custom.css.scss**

@import "bootstrap";

html, body {

**height**: 100%;

}

html {

**position**: relative;

}

footer {

**position**: fixed;

**width**: 100%;

**bottom**: 0;

}

input, textarea, select, .uneditable-input {

**border**: 1px solid #bbb;

**width**: 100%;

**margin-bottom**: 15px;

}

input {

**height**: auto !important;

}

#error\_explanation {

**color**: #f00;

ul {

**list-style**: none;

**margin**: 0 0 18px 0;

}

}

.field\_with\_errors {

**@extend** .control-group !optional;

**@extend** .error !optional;

}

**controllers/application\_controller.rb**

**class** ApplicationController **<** **ActionController::Base**

protect\_from\_forgery with: **:exception**

**include** SessionsHelper

**end**

**controllers/sessions\_controller.rb**

**class** SessionsController **<** ApplicationController

**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 = 'Пользователя не существует'

msg\_status = **:danger**

**elsif** !user.authenticate**(**password**)**

msg\_text = 'Неверный пароль'

msg\_status = **:danger**

**end**

**if** msg\_status == **:success**

sign\_in user

msg\_text = 'Вы успешно вошли'

**flash[**msg\_status**]** = msg\_text

**format**.html **{** **redirect\_to** input\_path **}**

**format**.json **{** **render** :**show**, status: **:created**, location: input\_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**

**def** **destroy**

sign\_out

**redirect\_to** root\_url

**end**

**end**

**controllers/twins\_controller.rb**

*# frozen\_string\_literal: true*

**class** TwinsController **<** ApplicationController

**def** **input**

**unless** signed\_in?

**redirect\_to** signin\_path

**end**

**end**

**def** view

**unless** signed\_in?

**redirect\_to** signin\_path

**end**

n = params**[:n]**.**to\_i**

**@result** = twins**(**n**)**

**end**

**def** is\_prime**(**num**)**

**return** **false** **if** num **<**= 1

**Math**.sqrt**(**num**)**.**to\_i**.downto**(**2**)**.**each** **{|**i**|** **return** **false** **if** num **%** i == 0**}**

**true**

**end**

**def** twins**(**n**)**

pairs = **{}**

**(**n..2 **\*** n**)**.**each** **do** **|first|**

**(first**..2 **\*** n**)**.**each** **do** **|**second**|**

**if** second **-** **first** == 2 **and** is\_prime**(first)** **and** is\_prime**(**second**)**

pairs**[first]** = second

**end**

**end**

**end**

**[**pairs, to\_table**(**pairs**)]**

**end**

**def** to\_table**(**pairs = **@result**, table\_class = 'table table-striped'**)**

**@table** = 'Unknown!'

**unless** pairs.empty?

rows = ''

pairs.**each** **do** **|**key, value**|**

rows **+**= "<tr><td>#{key}</td><td>#{value}</td></tr>"

**end**

**@table** = "<table class=\"#{table\_class}\"><tbody>#{rows}</tbody></table>"

**end**

**@table**

**end**

**end**

**controllers/users\_controller.rb**

**class** UsersController **<** ApplicationController

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

*# GET /users or /users.json*

**def** index

**@users** = User.all

**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 http://127.0.0.1:3000/signup*

**def** create

msg\_text = ''

msg\_status = **:success**

email = params**[:user][:email]**

password = params**[:user][:password]**

password\_confirmation = params**[:user][:password\_confirmation]**

**puts** password

**@user** = User.**new(**user\_params**)**

**respond\_to** **do** **|format|**

**if** **@user**

**if** User.find\_by\_email**(**email**)**

msg\_text = 'Пользователь уже зарегестрирован!'

msg\_status = **:danger**

**elsif** password != password\_confirmation

msg\_text = '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 = 'Введите почту корректно'

msg\_status = **:danger**

**end**

**if** msg\_status == **:success** **and** **@user**.**save**

sign\_in **@user**

msg\_text = 'Спасибо за регистрацию'

**flash[**msg\_status**]** = msg\_text

**format**.html **{** **redirect\_to** input\_path **}**

**format**.json **{** **render** :**show**, status: **:created**, location: input\_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: "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**

**@user**.**destroy**

**respond\_to** **do** **|format|**

**format**.html **{** **redirect\_to** users\_url, notice: "User was successfully destroyed." **}**

**format**.json **{** head **:no\_content** **}**

**end**

**end**

private

*# Use callbacks to share common setup or constraints between actions.*

**def** set\_user

**@user** = User.find**(**params**[:id])**

**end**

*# Only allow a list of trusted parameters through.*

**def** user\_params

params.**require(:user)**.permit**(:email**, **:password)**

**end**

**end**

**helpers/sessions\_helper.rb**

**module** SessionsHelper

**def** sign\_in**(**user**)**

remember\_token = User.new\_remember\_token

cookies.permanent**[:remember\_token]** = remember\_token

user.update\_attribute**(:remember\_token**, User.encrypt**(**remember\_token**))**

**self**.current\_user = user

**end**

**def** sign\_out

current\_user.update\_attribute**(:remember\_token**,

User.encrypt**(**User.new\_remember\_token**))**

cookies.**delete(:remember\_token)**

**self**.current\_user = **nil**

**end**

**def** current\_user=**(**user**)**

**@current\_user** = user

**end**

*# Пользователь является вошедшим если в сессии существует текущий пользователь, т.e., если current\_user не является nil*

**def** signed\_in?

!current\_user.**nil**?

**end**

*# Поиск текущего пользователя с помощью remember\_token.*

**def** current\_user

remember\_token = User.encrypt**(**cookies**[:remember\_token])**

**@current\_user** **||**= User.find\_by**(**remember\_token: remember\_token**)**

**end**

**end**

**javascript/controllers/application.js**

**//** Configure your import map **in** config**/**importmap.rb. Read more: https:**//**github.com**/**rails**/**importmap**-**rails

import "@hotwired/turbo-rails"

import "controllers"

import "jquery"

import "jquery\_ujs"

import "popper"

import "bootstrap"

import "src/main"

**javascript/src/main.js**

$(document).on('click', '.btn-close', **function** () {

$('.alert').fadeOut();

});

**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**

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

*require* "test\_helper"  
*# https://www.softcover.io/read/28fdb94f/ruby\_on\_rails\_tutorial\_3rd\_edition/sign\_up*class *AuthenticationPagesTest* < *ActionDispatch*::*IntegrationTest* def *add\_record*(*email*, *password*)  
 *record* = *User*.new(:email *=> email*, :password *=> password*)  
 *record*.save  
 *record* end  
  
 *################################## 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.com', '123456')  
  
 *get* signup\_url  
 assert\_response :success  
  
 *post* users\_url, params: { "authenticity\_token" *=>* "token", "user" *=>* { "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" *=>* { "email" *=>* "test@test.com", "password" *=>* "123456", "password\_confirmation" *=>* "123456" } }  
 follow\_redirect!  
 end  
  
 assert\_template 'input'  
 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.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  
  
 assert\_template 'input'  
 assert\_response 200  
 end  
  
 *test* 'login of a non-existent user' do  
 *post* sessions\_url, params: { "authenticity\_token" *=>* "token", "session" *=>* { "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" *=>* { "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.com', '123456')  
  
 assert\_difference 'User.count', 0 do  
 *# login  
 post* sessions\_url, params: { "authenticity\_token" *=>* "token", "session" *=>* { "email" *=>* "test@test.com", "password" *=>* "123456" } }  
 follow\_redirect!  
 end  
  
 assert\_difference 'User.count', 0 do  
 *# Logout  
 delete* signout\_url  
 follow\_redirect! *# перенаправлены в input, там проверочка, что не залогинились и иедм в login* follow\_redirect! *# из input в login* end  
  
 assert\_template 'sessions/new'  
 assert\_response 200  
 end  
  
 *################################## Вычисления не возможны без входа ######################################  
 test* "Calculations are impossible without sign in" do  
 *# view  
 get* view\_url, params: { n: 10 }  
  
 *# Если не вошли, значит редиректимся в signin* assert\_response 302  
  
 *# input  
 get* input\_url  
 assert\_response 302  
 end  
end

**layouts/\_footer.html.erb**

<footer class="footer bg-dark text-center text-lg-start">

<**div** class="text-center p-3 text-light">

© 2023 Copyright

</**div**>

</footer>

**layouts/\_header.html.erb**

<header>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<**div** class="container-fluid">

<**a** class="navbar-brand" href="#">ЛР12</**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 'Ввод', input\_path, class: "nav-link #{ request.path == input\_path || request.path == '/' ? 'active' : '' }" %>

</**li**>

<**li** class="nav-item">

<%= link\_to 'Вывод', view\_path, class: "nav-link #{ request.path == view\_path ? 'active' : '' }" %>

</**li**>

<**li** class="nav-item">

<%= link\_to 'Выход', signout\_path, method: "delete", class: "nav-link #{ request.path == signout\_path ? 'active' : '' }" %>

</**li**>

<% else %>

<**li** class="nav-item">

<%= link\_to 'Вход', signin\_path, class: "nav-link #{ request.path == signin\_path ? 'active' : '' }" %>

<% end %>

</**ul**>

</**div**>

</**div**>

</nav>

</header>

**layouts/\_shim.html.erb**

*<!--[if lt IE 9]>*

*<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>*

*<![endif]-->*

**layouts/application.html.erb**

<!DOCTYPE html>

<**html** lang="ru">

<**head**>

<**title**>Twins</**title**>

<**meta** name="viewport" content="width=device-width,initial-scale=1">

<**link** rel="icon" type="image/png" href="/assets/logo.png">

<%= csrf\_meta\_tags %>

<%= csp\_meta\_tag %>

<%= stylesheet\_link\_tag "application", "data-turbo-track": "reload" %>

<%= javascript\_importmap\_tags %>

<%= render 'layouts/shim' %>

</**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**>

<%= render 'layouts/footer' %>

</**body**>

</**html**>

**sessions/new.html.erb**

<% provide(:title, "Sign in") %>

<**h1**>Вход</**h1**>

<**div** class="row">

<**div** class="span6 offset3">

<%= form\_for(:session, url: sessions\_path) do |f| %>

<%= f.label :email %>

<%= f.text\_field :email, class: "form-control", required: true %>

<%= f.label 'Пароль' %>

<%= f.password\_field :password, minlength: 4, class: "form-control", required: true %>

<%= f.submit "Войти", class: "btn btn-large btn-primary mt-4" %>

<% end %>

<**p**>Еще не зарегистрированы? <%= link\_to "Зарегестрироватья сейчас!", signup\_path %></**p**>

</**div**>

</**div**>

**twins/input.html.erb**

<**h1**>Twins#input</**h1**>

<**div** class="row">

<**div** class="span6 offset3">

<**p**>

<**a** href="<%= url\_for(only\_path: false) + '.html.erb' %>">Мы находимся по

адресу: <%= url\_for(only\_path: false) + '.html.erb' %></**a**>

</**p**>

<**form** action="<%= view\_path %>" method="get" accept-charset="UTF-8">

<**div** class="form-group">

<**label** for="n"></**label**>

<**input** type="number" id="n" name="n" min="0" minlength="1" oninput="this.value = Math.abs(this.value);" class="form-control" required/>

</**div**>

<**button** type="submit" class="btn btn-large btn-primary mt-4">Ввести</**button**>

</**form**>

</**div**>

</**div**>

**twins/view.html.erb**

<**h1**>Twins#view</**h1**>

<**div**>

<**p**>

<**a** href="<%= url\_for(only\_path: false) + '.html.erb' %>">Мы находимся по

адресу: <%= url\_for(only\_path: false) + '.html.erb' %></**a**>

</**p**>

<%= @table.html\_safe %>

<**br**/>

<%= link\_to "Рассчитать заново", input\_path %>

</**div**>

**users/new.html.erb**

<% provide(:title, 'Sign up') %>

<**h1**>Регистрация</**h1**>

<**div** class="row">

<**div** class="span6 offset3">

<%= form\_for(@user) do |f| %>

<%= f.label :email %>

<%= f.text\_field :email, class: "form-control", required: true %>

<%= f.label 'Пароль' %>

<%= f.password\_field :password, class: "form-control", required: true %>

<%= f.label 'Подтверждение пароля', 'Confirmation' %>

<%= f.password\_field :password\_confirmation, minlength: 4, class: "form-control", suggested: "new-password", required: true %>

<%= f.submit "Зарегестрироваться", class: "btn btn-large btn-primary mt-4" %>

<% end %>

</**div**>

</**div**>

**environments/development.rb**

**require** "active\_support/core\_ext/integer/time"

Rails.application.configure **do**

*# Settings specified here will take precedence over those in config/application.rb.*

*# In the development environment your application's code is reloaded any time*

*# it changes. This slows down response time but is perfect for development*

*# since you don't have to restart the web server when you make code changes.*

config.cache\_classes = **false**

*# Do not eager load code on boot.*

config.eager\_load = **false**

*# Show full error reports.*

config.consider\_all\_requests\_local = **true**

*# Enable server timing*

config.server\_timing = **true**

*# Enable/disable caching. By default caching is disabled.*

*# Run rails dev:cache to toggle caching.*

**if** Rails.root.join**(**"tmp/caching-dev.txt"**)**.exist?

config.action\_controller.perform\_caching = **true**

config.action\_controller.enable\_fragment\_cache\_logging = **true**

config.cache\_store = **:memory\_store**

config.public\_file\_server.headers = **{**

"Cache-Control" **=>** "public, max-age=#{2.days.to\_i}"

**}**

**else**

config.action\_controller.perform\_caching = **false**

config.cache\_store = **:null\_store**

**end**

*# Store uploaded files on the local file system (see config/storage.yml for options).*

config.active\_storage.service = **:local**

*# Don't care if the mailer can't send.*

config.action\_mailer.raise\_delivery\_errors = **false**

config.action\_mailer.perform\_caching = **false**

*# Print deprecation notices to the Rails logger.*

config.active\_support.deprecation = **:log**

*# Raise exceptions for disallowed deprecations.*

config.active\_support.disallowed\_deprecation = :**raise**

*# Tell Active Support which deprecation messages to disallow.*

config.active\_support.disallowed\_deprecation\_warnings = **[]**

*# Raise an error on page load if there are pending migrations.*

config.active\_record.migration\_error = **:page\_load**

*# Highlight code that triggered database queries in logs.*

config.active\_record.verbose\_query\_logs = **true**

*# Suppress logger output for asset requests.*

config.assets.quiet = **true**

*# Raises error for missing translations.*

*# config.i18n.raise\_on\_missing\_translations = true*

*# Annotate rendered view with file names.*

*# config.action\_view.annotate\_rendered\_view\_with\_filenames = true*

*# Uncomment if you wish to allow Action Cable access from any origin.*

*# config.action\_cable.disable\_request\_forgery\_protection = true*

*# config.action\_mailer.default\_url\_options = { host: 'localhost', port: 3000 }*

*# config.action\_view.preload\_links\_header = false*

*# config.action\_view.automatically\_disable\_submit\_tag = false*

**end**

**initializers/assets.rb**

*# Be sure to restart your server when you modify this file.  
  
# Version of your assets, change this if you want to expire all your assets.  
Rails*.application.config.assets.version = "1.0"  
*Rails*.application.config.assets.precompile += %w( jquery.min.js jquery\_ujs.js bootstrap.min.js popper.js )

**routes.rb**

*Rails*.application.routes.draw do  
 *resources* :users  
 *resources* :sessions, only: [:new, :create, :destroy]  
  
 *match* '/input', to: 'twins#input', via: 'get'  
 *match* '/view', to: 'twins#view', via: 'get'  
  
 *match* '/signup', to: 'users#new', via: 'get'  
 *match* '/signin', to: 'sessions#new', via: 'get'  
 *match* '/signout', to: 'sessions#destroy', via: 'delete'  
  
 *root* 'twins#input'  
end

**importmap.rb**

*# Pin npm packages by running ./bin/importmap  
  
pin* "application", preload: true  
*pin* "@hotwired/turbo-rails", to: "turbo.min.js", preload: true  
*pin* "@hotwired/stimulus", to: "stimulus.min.js", preload: true  
*pin* "@hotwired/stimulus-loading", to: "stimulus-loading.js", preload: true  
*pin\_all\_from* "app/javascript/controllers", under: "controllers"  
  
*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

**Gemfile**

*# frozen\_string\_literal: true  
  
source* 'https://rubygems.org'  
git\_source(:github) { |*repo*| "https://github.com/#{*repo*}.git" }  
  
*gem* "jquery-rails"  
*gem* "bootstrap"  
*gem* "sassc-rails"  
  
*gem* 'rails-controller-testing'  
  
*# Bundle edge Rails instead: gem "rails", github: "rails/rails", branch: "main"  
gem* 'rails', '~> 7.1.0'  
  
*# The original asset pipeline for Rails [https://github.com/rails/sprockets-rails]  
gem* 'sprockets-rails'  
  
*# Use sqlite3 as the database for Active Record  
gem* 'sqlite3'  
  
*# Use the Puma web server [https://github.com/puma/puma]  
gem* 'puma'  
  
*# Use JavaScript with ESM import maps [https://github.com/rails/importmap-rails]  
gem* 'importmap-rails'  
  
*# Hotwire's SPA-like page accelerator [https://turbo.hotwired.dev]  
gem* 'turbo-rails'  
  
*# Hotwire's modest JavaScript framework [https://stimulus.hotwired.dev]  
gem* 'stimulus-rails'  
  
*# Build JSON APIs with ease [https://github.com/rails/jbuilder]  
gem* 'jbuilder'  
  
*# Use Active Model has\_secure\_password [https://guides.rubyonrails.org/active\_model\_basics.html#securepassword]  
gem* "bcrypt"  
  
*# Windows does not include zoneinfo files, so bundle the tzinfo-data gem  
gem* 'tzinfo-data', platforms: %i[ mingw mswin x64\_mingw jruby ]  
  
*# Reduces boot times through caching; required in config/boot.rb  
gem* 'bootsnap', require: false  
  
*group* :development, :test do  
 *# See https://guides.rubyonrails.org/debugging\_rails\_applications.html#debugging-with-the-debug-gem  
 gem* 'debug', platforms: %i[ mri mingw x64\_mingw ]  
 *gem* 'rspec-rails'  
end  
  
*group* :development do  
 *# Use console on exceptions pages [https://github.com/rails/web-console]  
 gem* 'web-console'  
end  
  
*group* :test do  
 *# Use system testing [https://guides.rubyonrails.org/testing.html#system-testing]  
 gem* 'capybara'  
 *gem* 'selenium-webdriver'  
  
end

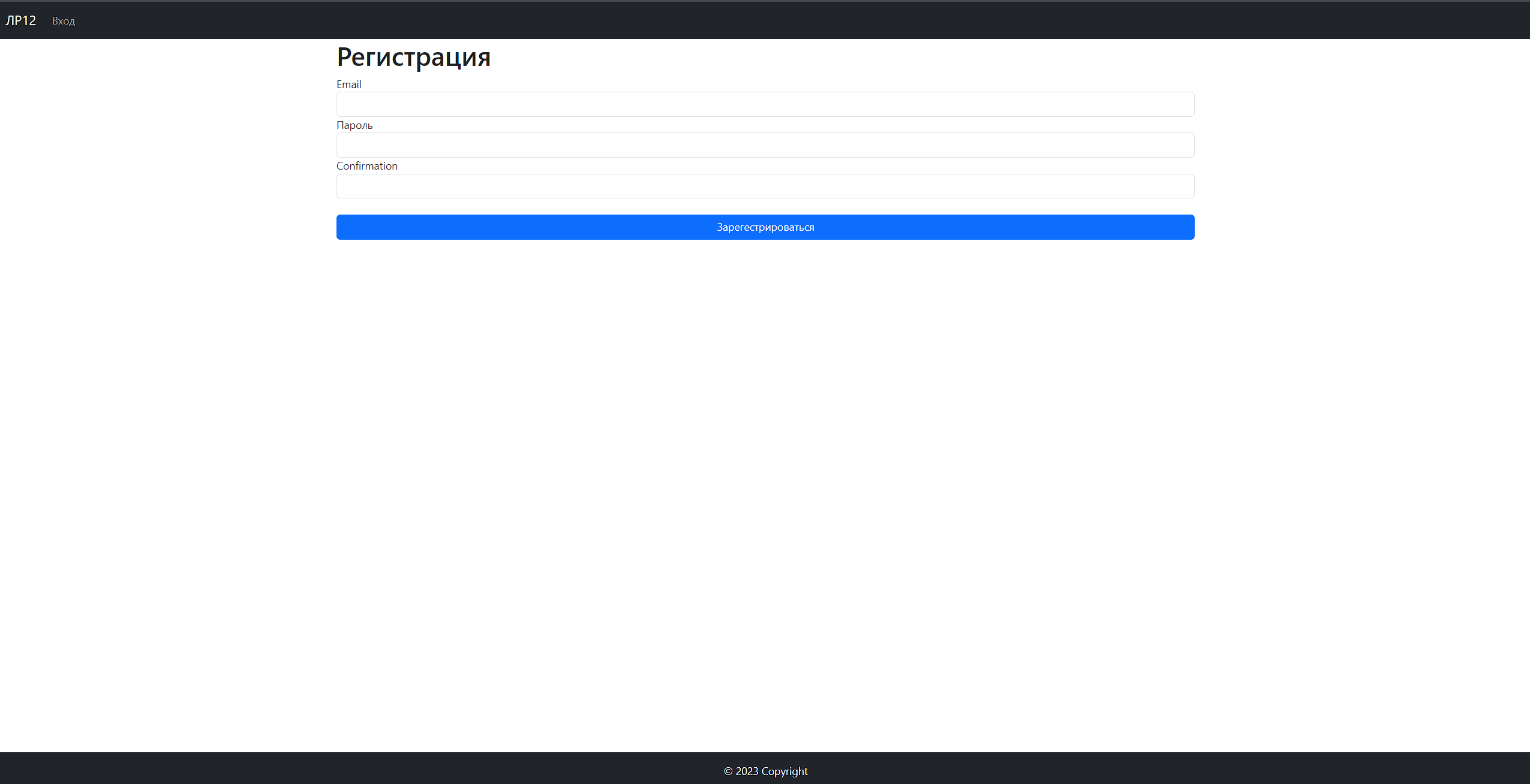
****

Рисунок 3 – форма регистрации

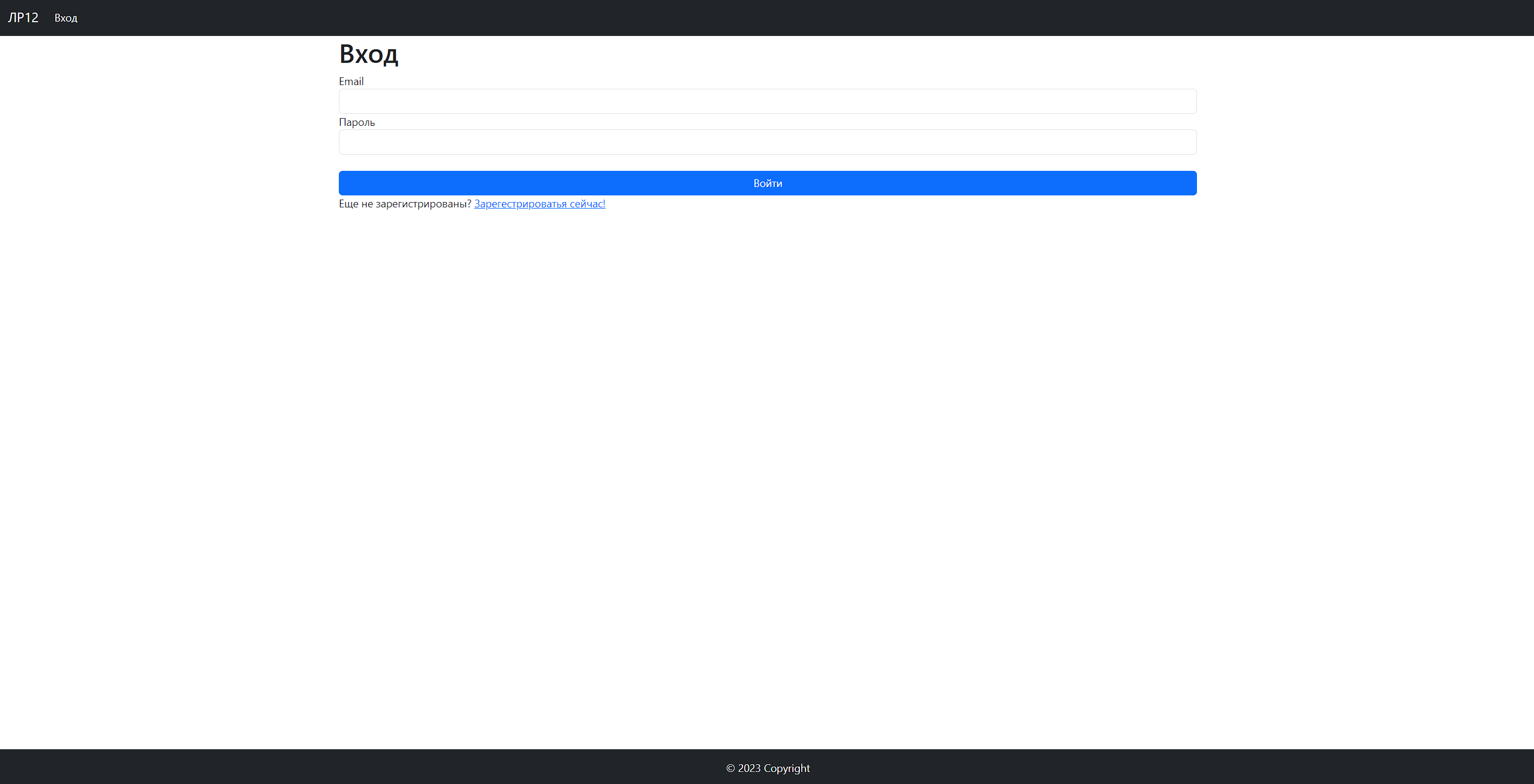
****

Рисунок 4 – форма входа

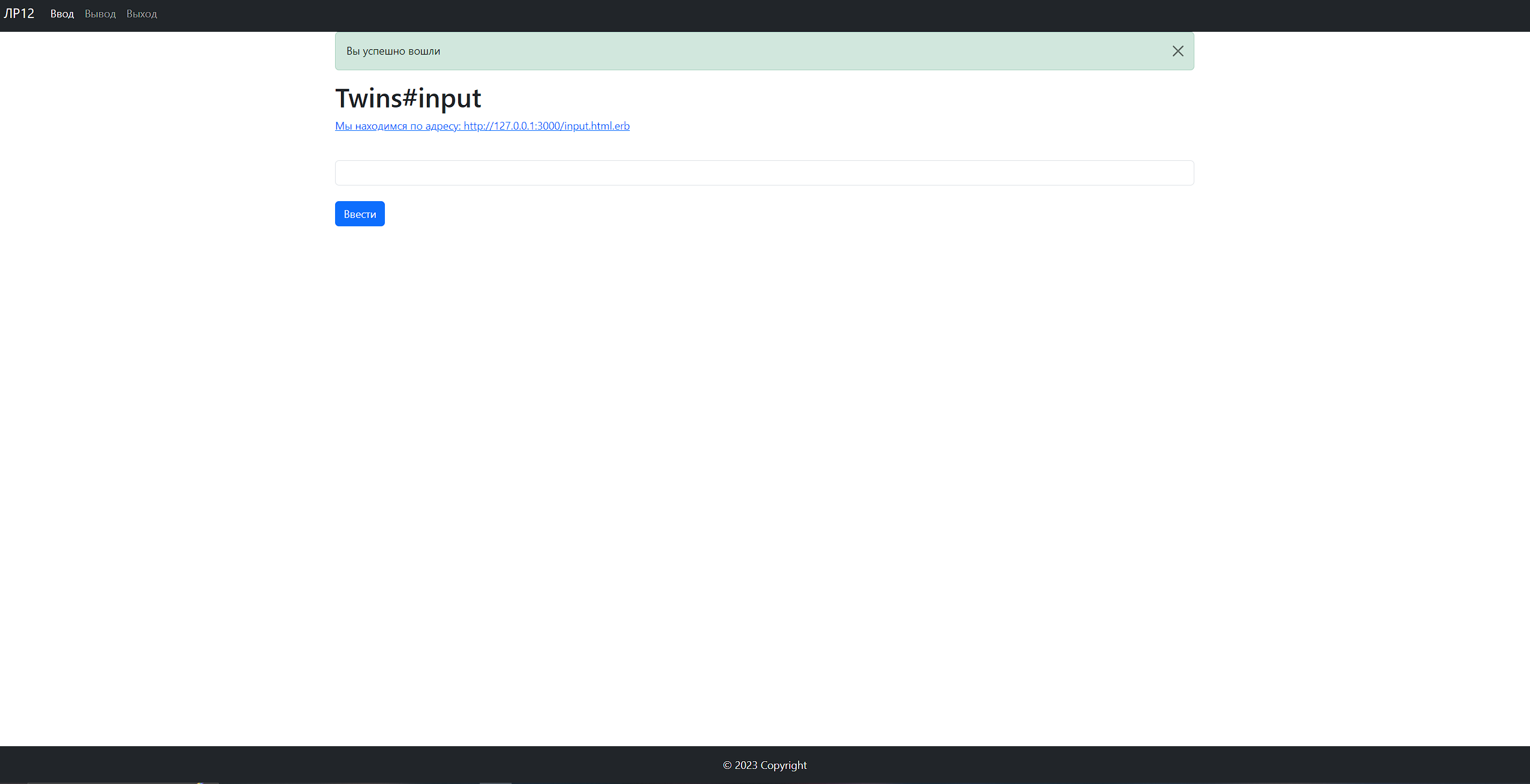
****

Рисунок 5 – успешный вход

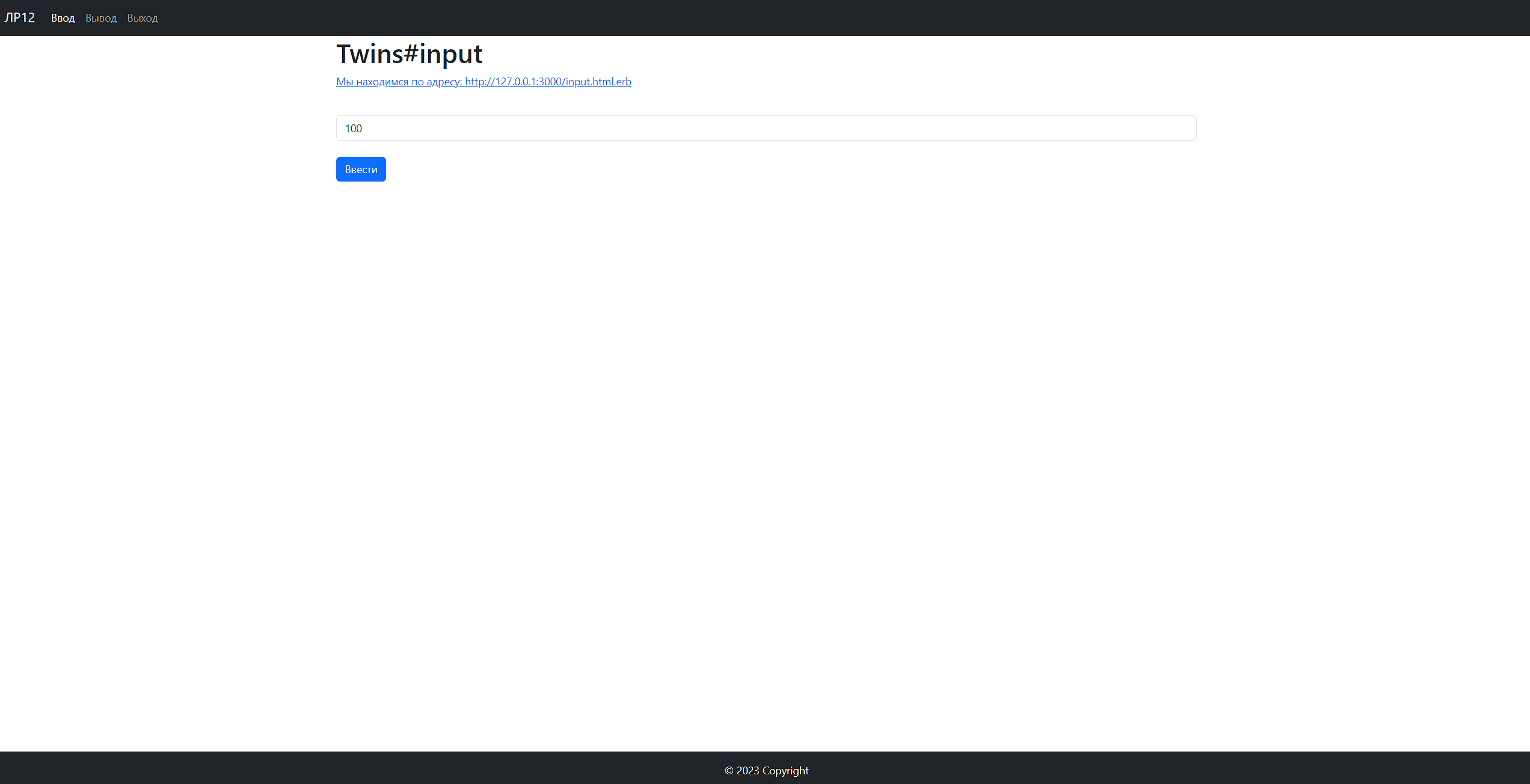
****

Рисунок 6 – ввод значения

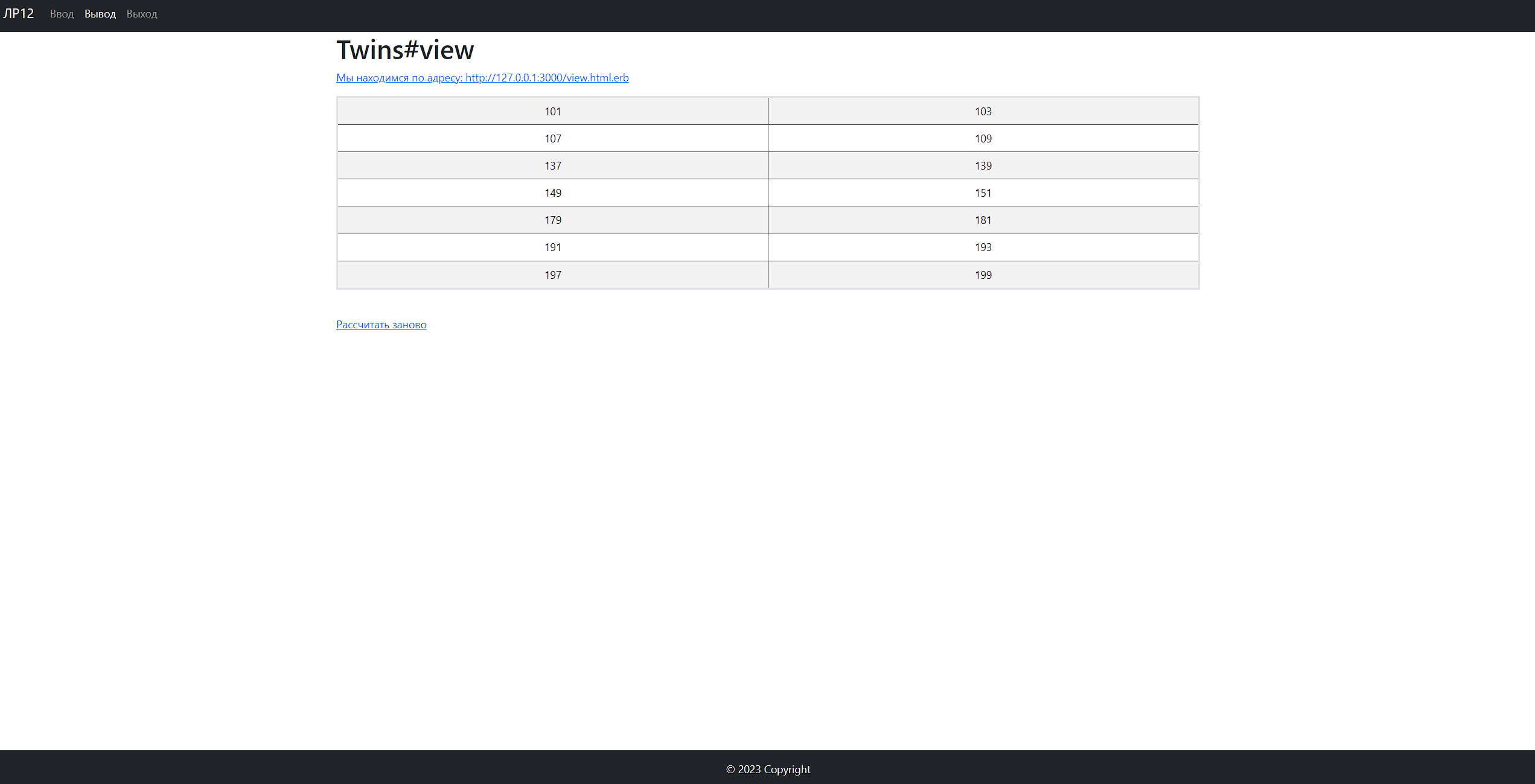
****

Рисунок 7 – вывод

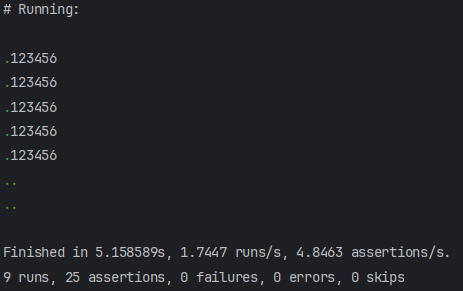
****

Рисунок 8 – тесты

**Распечатка БД:**

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); id integer PRIMARY KEY PRIMARY KEY AUTOINCREMENT

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); id integer NOT NULL NOT NULL

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); input integer

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); result json

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); created\_at datetime NOT NULL NOT NULL

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); updated\_at datetime NOT NULL NOT NULL

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 1 0

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 10 1

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); "{\"11\":13,\"17\":19}" 2

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 2023-11-08 16:33:04.789709 3

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 2023-11-08 16:33:04.789709 4

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 2 0

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 11 1

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); "{\"11\":13,\"17\":19}" 2

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 2023-11-08 16:33:07.187956 3

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 2023-11-08 16:33:07.187956 4

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 3 0

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 15 1

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); "{\"17\":19}" 2

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 2023-11-08 16:33:09.666411 3

results CREATE TABLE "results" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL, "input" integer, "result" json, "created\_at" datetime(6) NOT NULL, "updated\_at" datetime(6) NOT NULL); 2023-11-08 16:33:09.666411 4

**Вывод -** Получили практические навыки в создании веб-приложений, использующих аутентификацию. Получили навыки написания интеграционных тестов приложений.