Оглавление

[Индивидуальная работа 2](#_Toc197098121)

[Тема: Разработка REST-API приложения "Блог" 2](#_Toc197098122)

[1. Описание работы 2](#_Toc197098123)

[2. Документация проекта 2](#_Toc197098124)

[2.1 Краткое содержание 2](#_Toc197098125)

[2.2 Основные сущности 2](#_Toc197098126)

[2.3 Компоненты приложения 3](#_Toc197098127)

[2.4 Структура базы данных: 4](#_Toc197098128)

[PostgreSQL 4](#_Toc197098129)

[MongoDB 4](#_Toc197098130)

[3. Примеры интерфейса 4](#_Toc197098131)

[1. Главная страница 5](#_Toc197098132)

[2. Регистрация 5](#_Toc197098133)

[3. Просмотр статьи 5](#_Toc197098134)

[4. Редактирование поста 5](#_Toc197098135)

[5. Редактирование комментария 5](#_Toc197098136)

[6. Удаление комментария 5](#_Toc197098137)

[7. Создание поста 5](#_Toc197098138)

[8. Работа формы поиска 5](#_Toc197098139)

[4. Безопасность 5](#_Toc197098140)

[5. Инструкция по запуску проекта 6](#_Toc197098141)

[6. Выводы 7](#_Toc197098142)

[7. Библиография 7](#_Toc197098143)

# Индивидуальная работа

# Тема: Разработка REST-API приложения "Блог"

# 1. Описание работы

Целью данной работы являлась разработка веб-приложения — блога, который поддерживает добавление, просмотр, редактирование, удаление статей, а также возможность оставлять комментарии.

В ходе проекта были закреплены навыки разработки REST API, взаимодействия с реляционными и документно-ориентированными базами данных, а также применены знания, полученные в течение учебного семестра.

# 2. Документация проекта

Приложение реализовано на PHP с использованием PostgreSQL и MongoDB.

## 2.1 Краткое содержание

Проект представляет собой личный блог, в котором администратор (автор) публикует статьи. Пользователи могут просматривать и комментировать статьи, а также искать их по категориям и тегам. Комментирование доступно только зарегистрированным пользователям.

## 2.2 Основные сущности

**1. Пользователь**

Пользователь может иметь одну из двух ролей:

* `user` — зарегистрированный пользователь
* `admin` — администратор, Царь и Бог данного блога (Гуцу Николай)

**2. Статья**

Только администратор может публиковать статьи. Статья содержит:

* заголовок
* дату публикации
* изображение
* содержание
* теги
* комментарии
* форму для добавления комментариев

## 2.3 Компоненты приложения

**1. Общедоступный (гость):**

* Просмотр постов
* Поиск по тегам и категориям
* Аутентификация

**2. Частично защищённый (`user`):**

* Возможность оставлять комментарии

**3. Защищённый (`admin`):**

* Создание, редактирование, удаление постов
* Редактирование, удаление комментариев
* Добавление новых администраторов

## 2.4 Структура базы данных:

Используются две СУБД:

### PostgreSQL

* `users` — информация о пользователях
* `posts` — статьи
* `categories` — статичные категории
* `tags` — теги
* `post\_tags` — связи между постами и тегами

### MongoDB

Комментарии хранятся в отдельной коллекции:

* `id` — ID комментария
* `post\_id` — ID поста
* `user\_id` — ID пользователя
* `content` — текст комментария
* `created\_at` — дата публикации

# 3. Примеры интерфейса

## 1. Главная страница

## 2. Регистрация

## 3. Просмотр статьи

## 4. Редактирование поста

## 5. Редактирование комментария

## 6. Удаление комментария

## 7. Создание поста

## 8. Работа формы поиска

# 4. Безопасность

* Использование `PDO` и подготовленных выражений
* Защита от XSS через `htmlspecialchars()`
* Контроль доступа по ролям
* Хэширование паролей
* Работа через сессии

# 5. Инструкция по запуску проекта

**1. Клонируйте репозиторий:**

*`git clone https://github.com/yourusername/blog-project.git`*

**2. Создайте структуру базы данных** PostgreSQL с помощью скрипта *`create\_db\_script.php*`.

**3. Установите MongoDB**: `https://www.mongodb.com/try/download/community`.

**4. Проверьте, что в `php.ini` включены расширения:**

* pdo\_pgsql
* pgsql
* mongodb

4.1. Если MongoDB-расширение не установлено:

* `*pecl install mongodb*`
* Или загрузите (https://windows.php.net/downloads/pecl/releases/mongodb/1.13.0/) и поместите *`php\_mongodb.dll`* в `*php/ext*` и подключите в `*php.ini*`. Убедитесь, что PHP версии ≤ 8.1

**5. Установите зависимости** через Composer командами `*composer install*` & `*composer require mongodb/mongodb*`

**6. Создайте файл `*config/db.php*`** со следующими параметрами:

* define('DB\_HOST', 'localhost');
* define('DB\_NAME', 'blog');
* define('DB\_USER', 'YOUR\_USERNAME');
* define('DB\_PASS', 'YOUR\_PASSWORD');
* define('DB\_PORT', '5432');

**7. Запустите сервер** командой `*php -S localhost:8000 -t public*`.

# 6. Выводы

В результате работы было создано функциональное REST-API веб-приложение, соответствующее целям индивидуального задания. Проект демонстрирует уверенное владение PHP, базами данных PostgreSQL и MongoDB, архитектурой MVC и основами безопасности.

# 7. Библиография

`https://pecl.php.net/package/mongodb`

`https://moodle.usm.md/course/view.php?id=7161`

`https://zudochkin.github.io/2011/08/mongodb-php/`

`https://dataguide.vercel.app/dataguide/mongodb/working-with-dates`