**Техническое задание**

на разработку веб-приложения для организации личного портфолио

Команда: ИП Шадрин

*Шадрин Андрей Алексеевич (тимлид,)*

*Савина Анастасия Олеговна (аналитик)*

*Зубова Екатерина Александровна (дизайнер)*

*Джобиров Владимир Джамшедович (backend-разработчик)*

*Ворожцов Антон Александрович (frontend-разработчик)*

**Содержание**

[**1. Общие положения 3**](#_39wcdho8sr73)

[1.1 Полное наименование системы 3](#_30j0zll)

[1.2 Определения, обозначения и сокращения 3](#_1fob9te)

[**2. Назначения и цели создания системы 3**](#_srn2od30phh)

[2.1 Назначение системы 3](#_2et92p0)

[2.2 Цель и задачи системы 4](#_tyjcwt)

[**3. Требования к системе 5**](#_2s8eyo1)

[3.1 Нефункциональные требования 5](#_17dp8vu)

[3.1.1 Требования к интерфейсу 6](#_3rdcrjn)

[3.1.2 Требования к обеспечению безопасности персональных данных 7](#_26in1rg)

[3.1.3 Требование к производительности системы 7](#_lnxbz9)

[3.2 Функциональные требования 7](#_1ksv4uv)

[3.2.1 Требования к страницам 8](#_og1dzw4aab9h)

[3.2.2 Требования к разграничению доступа 12](#_vdljxr1a2td0)

[3.2.3 Требования к регистрации и авторизации 13](#_g6hu91gopa3n)

[3.2.4 Требования к комментированию постов 13](#_phthvom96azn)

[3.2.5 Требования к реакциям на пост 13](#_og1dzw4aab9h)

# 1. Общие положения

## Полное наименование системы

Социальный блог Шадрины Татьяны Геннадьевны

## Определения, обозначения и сокращения

БД – PostgreSQL.

ОС – Windows.

Система – приложение будет разработано при помощи веб-фреймворка Django, на языке Python.

СУБД – реляционная СУБД на основе PostgreSQL

# Назначения и цели создания системы

## Назначение системы

Создаваемая система предназначена для управления социальным блогом заказчика, часть которого будет собой представлять личное портфолио с обеспечением следующей функциональности:

1. Управление постами личного блога: публикация поста; удаление поста; удаление комментария к посту
2. Возможности пользователей: оставлять комментарии на посты; ставить реакции на посты
3. Модерация контента: добавление и удаление пользователей; блокировка сообщений.

## Цель и задачи системы

Цель разработки Системы – повышение качества и обеспечение эффективности управления личным блогом заказчика путем:

* Автоматизации процессов для управления контентом блога;
* Повышения защиты и безопасности данных
* Брендинга и идентичности
* Повышения вовлеченности аудитории
* Персонализации и адаптации контента

Система должна решать следующие задачи:

1. Обеспечить возможность администратору создавать и управлять контентом блога, включая добавление, редактирование и удаление постов;
2. предоставить механизм управления доступом к блогу для администратора, позволяющий управлять правами пользователей на чтение и комментирование;
3. Обеспечить интеграцию с социальными сетями для расширения аудитории и увеличения присутствия контента;
4. Предоставить статистику посещаемости и взаимодействия с контентом для администратора блога

# Требования к системе

### Нефункциональные требования

Функционирование системы должно осуществляться в интернет-браузере. Для этого будет применяться подход, который позволит пользователям получать доступ к функциональным возможностям системы через веб-интерфейс, который будет оптимизирован для работы в различных браузерах.

База данных должна быть размещена в системе управления базами данных семейства PostgreSQL. Это позволит обеспечить эффективное хранение и управление данными проекта, обеспечивая при этом высокую производительность и надежность работы.

Интерфейс системы должен быть простым в освоении и удобным для пользователей. Он должен предоставлять необходимый функционал для выполнения задач пользователей, при этом минимизируя сложность взаимодействия с системой. Во время работы пользователю должны быть доступны только те функции, которые соответствуют их должностным полномочиям и потребностям, чтобы обеспечить оптимальную производительность и удобство использования.

Для настройки сервера и обеспечения его эффективной работы будут использованы следующие технологии:

* Git: для контроля версий кода, управления изменениями и совместной разработки. Это позволит организовать удобное и безопасное ведение разработки проекта.
* Nginx: веб-сервер, который будет использоваться для обработки HTTP-запросов, обеспечивая масштабируемость и быстродействие системы.
* Gunicorn: веб-сервер, который будет отвечать за обработку запросов к приложению Django и его поддержку. Он обеспечит стабильную работу веб-приложения и эффективное использование серверных ресурсов.
* PostgreSQL: система управления базами данных, которая будет использоваться для хранения и организации данных проекта. PostgreSQL обеспечит высокую производительность и надежность работы с данными, а также широкие возможности по их анализу и обработке.
* Для обеспечения безопасности передачи данных между клиентом и сервером будет установлен SSL-сертификат и использован протокол HTTPS. Это обеспечит защищенное соединение и обмен информацией между пользователями и сервером, предотвращая возможные атаки и утечки данных.
* Для обеспечения доступности системы по сети будет зарегистрировано доменное имя, что позволит пользователям легко обращаться к системе по известному адресу. Это повысит удобство использования и доступность системы для пользователей.

### Требования к интерфейсу

Интерфейс Системы должен быть прост, нагляден, интуитивно понятен и легок в освоении, и должен отвечать следующим требованиям:

1. Доступность функций регистрации и авторизации;
2. Возможность просмотра постов заказчика с возможностью проставления реакций и комментирования;
3. Понятное меню навигации, позволяющее быстро перемещаться между разделами блога заказчика

## Требования к обеспечению безопасности персональных данных

Система должна обеспечивать реализацию следующих мер по обеспечению безопасности персональных данных:

* Применение механизма валидации форм при помощи сторонних библиотек Python;
* Внедрение защиты от основных атак, таких как CSRF (межсайтовая подделка запроса) и XSS (межсайтовый скриптинг), с использованием встроенных инструментов Django;
* Применение HTTPS протокола для защищенной передачи данных между клиентом и сервером, что обеспечивает шифрование информации во время передачи по сети;
* Защита от инъекций SQL путем использования параметризованных запросов и механизмов ORM (Object-Relational Mapping) в Django;

### Требование к производительности системы

* Операционная система: Ubuntu 22.04 LTS 64-bit
* Оперативная память (RAM): 512 МБ
* Универсальный SSD диск: 5 ГБ
* Конфигурация сервера: 1 vCPU – 1 ядро (аренда 10% мощности облачного сервера с последующим увеличением)

### Функциональные требования

Каждая страница системы должна отображать вкладку, содержащую навигационное меню для быстрого доступа к основным разделам и функциям приложения. Это меню должно включать в себя следующие элементы:

* мероприятия и праздники;
* образовательная деятельность;
* методическая копилка;
* конкурсы и награды;
* фото и видео

### Требования к страницам

* + - 1. Главная страница сайта (index.html)

На главной странице сайта пользователь встречает информацию о заказчике, расположенную по середине страницы. Слева отображается фотография заказчика, а справа сверху вниз представлена следующая информация:

Информация о заказчике:

* ФИО: Шадрина Татьяна Геннадьевна
* Город: Невьянск

“4 главных мои качества”:

* Порядочность
* Доброжелательность
* Сдержанность
* Честность

В самом низу должна быть размещена цитата:

“Воспитатель сам должен быть тем, кем он хочет сделать воспитанников”

Справа и слева от этой информации находятся стрелочки, позволяющие переключаться между различными секциями с дополнительной информацией о заказчике.

* + - 1. Мероприятия и праздники.

Навигационная кнопка перенаправляет пользователя на страницу с описанием предстоящих событий и праздников, а также предоставляет доступ к галерее фотографий и видео.

Посты на страницу загружаются Администратором при помощи встроенной админ-панели в Django. При публикации поста заказчик вводит следующие данные:

* заголовок: название мероприятия или праздника;
* содержимое: подробное описание мероприятия, включая даты, время, место проведения и другие важные детали;
* изображение: фотография, связанная с мероприятием или праздником

Переключение постов осуществляется путем нажатия на стрелку, располагающейся справа от поста.

Комментирование постов осуществляется при помощи пустой формы, находящейся в нижней части каждого поста. Справа от формы будут доступны различные реакции, нажав на которые, будет осуществляться оценка.

3. Образовательная деятельность

Навигационная кнопка перенаправляет пользователя на страницу с информацией о различных программах обучения, курсах, лекциях и мастер-классах, предоставляемых заказчиком.

……

……

4. Методическая копилка

Навигационная кнопка перенаправляет пользователя на страницу, где доступны обзоры методических материалов, рекомендации и советы для специалистов соответствующей области.

На странице представлен список опубликованных заказчиком методических материалов. Каждый элемент списка отображает заголовок материала, а также имеет возможность прикрепить приложенный файл для дополнительной информации.

Для удобства пользователей на странице реализована функция сортировки по двум параметрам: "Дате" и "Популярности". Это позволяет пользователям быстро находить нужный материал и ознакомиться с ним в соответствии с их потребностями и предпочтениями.

5. Конкурсы и награды

Навигационная кнопка направляет пользователя на страницу, где доступна информация о текущих и прошедших конкурсах, а также о результатах и наградах.

На данной странице представлена следующая информация:

Текущие конкурсы: здесь пользователи могут ознакомиться с информацией о запланированных конкурсах, включая их тематику, правила участия и призовой фонд. Для добавления нового конкурса администратор в лице заказчика может использовать встроенную админ-панель в Django, где будет заполнять соответствующие поля, включая название конкурса, его описание, даты проведения и другие детали.

Прошедшие конкурсы: здесь пользователи могут просмотреть информацию о прошедших конкурсах, победителях и призовых местах. Администратор также имеет возможность добавить информацию о прошедших конкурсах через админ-панель в Django, указывая результаты и призовые места.

Награды и достижения: на этой странице представлена информация о полученных наградах и достижениях заказчика. Администратор может добавлять информацию о наградах через админ-панель, указывая название награды, дату получения и другие сопутствующие данные.

6. Фото и видео

Навигационная кнопка "Фото и видео" перенаправляет пользователя на страницу, где доступна коллекция фотографий и видеороликов, связанных с блогом заказчика.

На данной странице представлена следующая информация:

Фотографии: пользователи могут просматривать фотографии, загруженные администратором через админ-панель в Django. Каждая фотография сопровождается названием и кратким описанием, чтобы пользователи могли легко понять ее контекст и содержание. Администратор может добавлять новые фотографии, указывая их заголовок, описание и прикрепляя соответствующий файл.

Видеоролики: здесь пользователи могут просматривать видеоролики, связанные с тематикой блога заказчика. Видео загружаются и добавляются администратором через админ-панель в Django. Каждый видеоролик имеет название и описание, а также предоставляет возможность воспроизведения прямо на странице.

### Требования к разграничению доступа

Система должна подразделять функциональные возможности для двух ролей: Администратор и Пользователь.

Администратор в лице заказчика должен иметь возможность исполнять в системе следующие процессы:

* мониторинг постов:
  + публикация и удаление поста;
  + право на изменение содержимого поста;
  + удаление комментариев к посту
* создание и удаление пользователей

Обычные пользователи имеют следующие возможности:

* просмотр проектов: Пользователи должны иметь возможность просматривать информацию о проектах, оставленную на сайте заказчиком;
* оставление комментариев: Пользователи могут оставлять комментарии к проектам и другим постам, размещенным на сайте;
* управление своим профилем: Пользователи могут редактировать свои персональные данные, включая имя, контактную информацию и пароль;
* оценивание проектов (например, ставить лайки)

### Требования к регистрации и авторизации

Система должна предоставлять Администраторам и Пользователям возможность регистрироваться и авторизовываться в системе.

В процессе регистрации/авторизации заполняются следующие поля:

* имя;
* фамилия;
* email;
* password\_1;
* password\_2

Требования к созданию/публикации поста:

* тема поста;
* основное содержимое;
* изображение

### Требования к комментированию постов

* авторизация Пользователя/Администратора
* поле комментария;
* валидация нецензурных выражений

## Требования к реакциям на пост

* авторизация Пользователя/Администратора;
* дроп-лист с выпадающим списком реакций;
* валидация реакций