

Это шаблон документа с требованиями от лаборатории Smart Education Lab ИТИС КФУ

Пояснения написаны красными буквами и курсивом, обычным текстом написаны примеры формулировок.

Скопируйте этот документ в папку своего проекта и начинайте писать требования, удаляя *пояснения*. Удачи!

Notes (Название проекта)

Напишите краткое описание проекта в 2-3 предложениях: Нужно реализовать веб-сервис, который будет хранить заметки с тегами и комментариями. Веб-сервис должен быть доступен для регистрации для всех пользователей.

Технологии: *Опишите ключевые технологии, используемые в проекте. Не нужно вставлять полный список библиотек, хватит названий основных компонентов (языки программирования, фреймворки, базы данных и другие сервисы)* Python 3, Django 3, PostgreSQL, Celery, JavaScript, Vue

Заинтересованные лица: *опишите представителей со стороны заказчика, к кому можно будет обратиться за уточнением требований*

- Иванов Иван Иванович, заместитель директора такого-то института КФУ

Команда проекта: *опишите команду проекта и распределение ролей (если оно есть)*

- Сидорова Марина Владимировна, 11-501 - тимлид
- Петров Александр Геннадьевич, 11-502 - бекенд разработчик
- Гаврилов Олег Петрович, 11-503 - фронтенд разработчик

Глоссарий

Чаще всего проект разрабатывается в специфической предметной области. Термины, используемые в этой области, могут быть непонятны людям, которые не знакомы с этим проектом. В этом разделе нужно описать такие термины и аббревиатуры, чтобы читать документ и работать с проектом было проще.

Тег – ключевое слово, которое привязывается к заметке. В дальнейшем по тегу можно сделать фильтрацию списка заметок.

Заметка – документ с названием и текстом, содержащий определенную информацию.

Пользователи

Опишите пользователей вашего проекта: кто эти люди, чем занимаются и с какой целью будут использовать проект.

Проектом может пользоваться любой желающий после прохождения регистрации.

Если есть авторизация и/или регистрация, опишите эти процессы. Если есть авторизация/регистрация через внешние сервисы, укажите это.

Во время регистрации пользователь вводит email и пароль. После регистрации на email отправляется письмо с подтверждением. Войти на сайт пользователь может только после подтверждения email. Пароль должен храниться в базе данных в зашифрованном виде.

Опишите роли пользователей в проекте.

Роль	Описание	Доступный функционал
Обычный пользователь	Любой человек, зарегистрированный на сайте	Добавляет свои заметки в личном кабинете и может посмотреть информацию о них
Администратор	Специальные пользователи с доступом к административной панели	Видит всех пользователей, количество их заметок, и может блокировать пользователей.

Дизайн

Опишите общие пожелания по дизайну: цвета, шрифты, раскладка интерфейса... Запросите с заказчика ссылки на проекты похожей тематики и проекты, дизайн которых нравится.

Пример для другого проекта: Дизайн должен быть выполнен в стиле КФУ с использованием фирменных цветов.

Меню и информация об авторизации пользователя должна располагаться в шапке сайта. Внизу нужно указать название проекта, копирайт и пометку, что проект был разработан в КФУ ИТИС.

При разработке дизайна используйте готовые решения (например, Bootstrap). Если нужно использовать цвета/логотипы/шрифты каких-нибудь брендов, погуглите брендбуки. Например, у КФУ есть свой брендбук.

Функциональные требования

Здесь нужно детали реализации каждого раздела проекта. Разделение может быть по страницам, по экранам...

Если что-то проще описать рисунком или диаграммой, нужно вставлять эти элементы.

Даже если в проекте будут разрабатываться отдельные бекенд и фронтенд части, не стоит описывать их функционал по отдельности. API на бекенде будет разрабатываться под страницы на фронтенде, и их функционал будет близким.

Регистрация и авторизация

Зарегистрироваться и авторизоваться может любой пользователь.

В форме регистрации должны быть следующие поля (все поля обязательные):

- электронная почта
- пароль
- повтор пароля
- имя

После регистрации нужно отправить пользователю письмо со ссылкой для подтверждения почты. Пока пользователь не подтвердил почту, он не сможет авторизоваться.

В форме входа должны быть следующие поля (все поля обязательные):

- электронная почта
- пароль

Если электронная почта или пароль неверные (не нашелся пользователь в базе данных), вывести сообщение: Неверные электронная почта или пароль. Пароли должны храниться в хешированном виде.

Управление заметками

Создание заметки

На главной странице есть возможность перейти к созданию заметки.

В форме создания заметки есть следующие поля:

- название (обязательное поле)
- описание (обязательное поле)
- файл
- теги (может быть несколько тегов у одной заметки).

После создания заметки пользователя должно перебросить на страницу заметки.

Редактирование заметки

Форма редактирования заметки идентична форме создания заметки. В текстовые поля предварительно заполнена информация из существующей заметки.

После сохранения заметки пользователя должно перебросить на страницу заметки.

Страница заметки

Вывод информацию о заметке для просмотра. Содержит следующие блоки информации:

- название
- описание
- файл
- теги

Сверху страницы должны быть кнопки для перехода к

- списку заметок
- редактированию заметки
- удалению заметки

Удаление заметки

К удалению можно перейти с страницы заметки. Пользователю должно показываться подтверждение: Вы уверены, что хотите удалить заметку? Если пользователь нажимает да, заметка удаляется, пользователь переходит на главную страницу.

Список заметок (главная страница)

Показывает список заметок в табличном виде

Столбцы таблицы:

- название
- описание
- файл
- теги

При клике на строку должен происходить переход на страницу заметки.

Должна быть фильтрация по дате создания и поиск по названию.

Административная панель

Административная панель разрабатывается отдельно от основных страниц сайта, поэтому описывается отдельно.

Административная панель должна располагаться на адресе /admin. Попасть в нее могут только администраторы.

В административной панели представлены списки пользователей. В каждом списке есть возможность редактировать, добавить и удалить соответствующую сущность.

API

Если требуется разработать API для внешних приложений или какие-либо интеграции с внешними ресурсами, нужно описать их отдельно. API для фронтенд-части вашего проекта не стоит описывать отдельно, потому что оно является частью основного функционала

Используется внешним сервисом. Для авторизации API-запросов используется токен, который устанавливается для каждого пользователя вручную через административную панель. Этот токен передается в заголовках всех запросов следующим образом:

Authorization: Token {token}

В API должны быть следующие методы:

здесь API спроектировано вплоть до адресов и методов. Если на данном этапе разработки нет информации о конкретике, можно описать функционал общими словами

- GET /notes/ – по получить все заметки этого пользователя
- POST /notes/ - создать заметку
- GET /status/ - узнать статус работы системы (возвращает статус код 200, если сервис работает, и другие коды в случае ошибок).

Приложения

Также в проекте должны быть артефакты, которые будут дополнять описание требований. Ссылки на эти артефакты должны быть указаны здесь.

- Структура базы данных – диаграмма таблиц в базе данных, можно сделать с помощью draw.io
- Макеты интерфейсов – наброски можно сделать с помощью draw.io, более проработанные варианты можно сделать в [Figma](https://www.figma.com/)