### **«Приложение для ведения дневника с интеграцией календаря»» (аналог Notion)**

#### **1. ВВЕДЕНИЕ (INTRODUCTION)**

**Название проекта:** «Персональный цифровой дневник» (рабочее название)

**Краткое описание:**  
Этот проект представляет собой веб-приложение для ведения личного дневника, позволяющее пользователям записывать мысли, планы, заметки и напоминания. Система включает возможность организации записей по категориям и тегам, интеграцию с календарем для планирования событий и задач, а также защиту личных данных.

**Цели проекта:**

1. **Разработать удобный и безопасный инструмент для ведения личного дневника.**
2. **Интегрировать систему календаря и напоминаний для планирования событий.**
3. **Обеспечить возможность поиска и фильтрации записей.**.
4. **Реализовать функцию экспорта и импорта записей.**
5. **Разработать мобильную адаптацию и возможность использования с разных устройств.**.

**Задачи проекта:**

* Создать систему регистрации и аутентификации пользователей..
* Реализовать функционал создания, редактирования и удаления записей.
* Интегрировать календарь и возможность привязки событий к записям..
* Обеспечить поддержку форматирования текста и загрузки изображений..
* Внедрить механизм защиты данных и резервного копирования.

**Целевая аудитория:**

* Люди, которые ведут дневники и записывают личные мысли.
* Студенты и профессионалы, использующие записи для планирования.
* Люди, которым важно фиксировать воспоминания и события.

#### **2. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ**

**Общий функционал и бизнес-логика:**

1. **Регистрация и вход: пользователи могут зарегистрироваться через email или социальные сети.**
2. **Создание записей: возможность форматировать текст, прикреплять изображения и файлы.**
3. **Календарь: просмотр записей в формате календаря, возможность планирования событий.**
4. **Напоминания: настройка уведомлений о записях и событиях.**
5. **Защита данных: шифрование записей, двухфакторная аутентификация**.
6. **Поиск и фильтрация: сортировка записей по тегам, дате, содержимому.**
7. **Экспорт и импорт: возможность сохранять записи в PDF, TXT, JSON.**.

**Основные роли пользователей:**

* **Обычный пользователь** – ведет личный дневник, планирует события.
* **Администратор** – управляет системой, модерирует контент.

**3. ТРЕБОВАНИЯ К СИСТЕМЕ (FUNCTIONAL REQUIREMENTS)**

Для реализации платформы необходимо обеспечить следующие функциональные и технические требования:

**1. Функциональные требования**

**1.1 Регистрация и аутентификация**

* Регистрация через email и социальные сети.
* Двухфакторная аутентификация (2FA)
* Восстановление пароля через email или SMS.

**1.2 Управление пользователями**

* Личный кабинет с настройками и списком записей.
* Возможность установки приватности записей (личные, публичные, с доступом по ссылке)...

**1.3 Работа с записями**

* Создание, редактирование и удаление записей..
* Поддержка форматирования текста (Markdown, WYSIWYG)..
* Загрузка изображений, аудиозаписей и файлов.
* Категоризация записей (по темам, тегам, датам).

**1.4 Календарь и напоминания**

* Интерактивный календарь с возможностью привязки записей.
* Настройка напоминаний (push, email).
* Синхронизация с Google Calendar.

**1.5 Поиск и фильтрация**

* Поиск по содержимому и тегам..
* Фильтрация по дате, категории, наличию файлов.

**1.6 Экспорт и импорт данных**

* Экспорт записей в PDF, TXT, JSON.
* Импорт данных из других приложений.

**2. Нефункциональные требования**

* Производительность: поддержка большого количества записей без задержек.
* **Безопасность: шифрование данных, защита от атак.**
* Масштабируемость: возможность добавления новых функций..
* **Доступность: адаптивный дизайн для мобильных устройств.**

**Ссылка на GitHub:**

**Разделение работы:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Неделя** | **Тема** | **Задание** | **Сложности(во время отчета)** | **Выполнено(в процентах)** |
| 1.Оба | Подготовка и настройка проекта | ✅ Определение требований к платформе (основные функции: создание записей, редактирование, интеграция с календарем, поиск и фильтрация, регистрация пользователей).  ✅ Разработка структуры базы данных (модели: Пользователь, Администратор, Дата, Сообщения и т. д.).  ✅Установка Django и необходимых библиотек (Django REST Framework, Django ORM, Pillow для изображений и т. д.).  ✅ Настройка виртуального окружения и создание базового проекта Django.  ✅ Подключение базы данных (SQLite/PostgreSQL).  ✅ Подготовка начальной документации (README). |  |  |
| 2.Алдияр | Работа с шаблонами и статикой | ✅Реализация регистрации, авторизации и аутентификации пользователей.   * Использование Django Authentication. * Подключение аутентификации через e-mail и пароль.   ✅Реализация CRUD-операций.  ✅Создание календарного интерфейса  ✅Настройка загрузки изображений |  |  |
| 3.Жасулан | Работа с ORM и данными  (напоминаниями и интеграцией календаря) | ✅Поиск и фильтрация записей..  ✅Добавление напоминаний.  ✅Интеграция FullCalendar.js для отображения записей  ✅ Реализация синхронизации с Google Calendar.  ✅ Добавление сообщений об ошибках и исключений. |  |  |
| 4.Алдияр | Личный кабинет, тестирование | ✅Разработка личного кабинета пользователя  ✅ Внедрение системы избранного  ✅Отображение записей в календаре.  ✅Тестирование безопасности и производительности.. |  |  |
| 5.Оба | Подготовка к файналу, завершение, документация | ✅ Улучшение кода (рефакторинг, чистка ненужных файлов, комментарии).  ✅Написание документации (README, описание API, инструкции по развертыванию).  ✅Подготовка презентации проекта.  ✅ Деплой проекта (Heroku, Render, Vercel и т. д.). Презентация проекта. |  |  |

**План разработки дневника с интеграцией календаря(аналог Notion) на 5 недель по дням**

**Первая неделя:**

|  |  |  |  |
| --- | --- | --- | --- |
| **День** | **Задание** | **Выполнено** | **Сложности** |
| **Четверг** | Анализ требований проекта и проработка backend-схемы Осмотр верстки (макета) сайта Выбор редактора кода и версии Django, Python |  |  |
| **Пятница** | Установка Django и настройка проекта.Ознакомление с базовыми файлами Django.Проектирование структуры проекта |  |  |
| **Суббота** | Создание приложения main.Разбор шаблона проектирования MTV.Простая маршрутизация URL-адресов |  |  |
| **Воскресенье** | Параметр request в debug-режиме. Работа с HTML-шаблонами, placeholders .Первый шаблонный тег if-endif |  |  |
| **Понедельник** | Установка Git-клиента. Инициализация репозитория.Загрузка кода с GitHub, работа с коммитами |  |  |
| **Вторник** | Перенос HTML-верстки в Django templates Работа со статическими файлами Подключение load static |  |  |
| **Среда** | Организация хранения статики Настройка static url Наследование шаблонов (extends, базовый шаблон) |  |  |

**Неделя 2: Работа с шаблонами и статикой**

|  |  |  |  |
| --- | --- | --- | --- |
| **День** | **Задание** | **Выполнено** | **Сложности** |
| **Четверг** | Создание ссылок в шаблонах (url) Расширенная маршрутизация (include, namespace |  |  |
| **Пятница** | Разработка структуры записей дневника (шаблоны, вывод данных, циклы for-endfor) |  |  |
| **Суббота** | Работа с БД, миграции, основы ORM Создание Django-моделей Админка и регистрация моделей |  |  |
| **Воскресенье** | Модель записей дневника Начальная рабoта с ORM Установка IPython |  |  |
| **Понедельник** | Автогенерация слагов для URL Фикстуры Установка django-debug-toolbar |  |  |
| **Вторник** | Продвинутая работа с ORM ORM в контроллерах Работа с медиа-файлами |  |  |
| **Среда** | Дополнительные методы моделей Пользовательские шаблонные теги Конвертеры URL-адресов (int, slug) |  |  |

**Неделя 3: Работа с ORM и данными**

|  |  |  |  |
| --- | --- | --- | --- |
| **День** | **Задание** | **Выполнено** | **Сложности** |
| **Четверг** | Обратное разрешение URL-адресов Пагинация на странице Фильтры в шаблонах |  |  |
| **Пятница** | Пагинация через GET-запрос Получение GET-параметров Фильтр записей |  |  |
| **Суббота** | Объект request в шаблонах Пользовательские теги с параметрами, urlencode Поиск по сайту |  |  |
| **Воскресенье** | Q-объекты Установка PostgreSQL Подключение PostgreSQL к Django |  |  |
| **Понедельник** | Создание БД через pgadmin4 и терминал (psql) Полнотекстовый поиск Создание приложения Users |  |  |
| **Вторник** | Работа с формами, POST-запросы Формы, связанные с моделями Авторизация пользователя |  |  |
| **Среда** | Реальная работа с формами Регистрация пользователя Личный кабинет |  |  |

**Неделя 4: Личный кабинет, тестирование, документация**

|  |  |  |  |
| --- | --- | --- | --- |
| **День** | **Задание** | **Выполнено** | **Сложности** |
| **Четверг** | Декоратор login\_required Отправка messages Перенаправление после авторизации |  |  |
| **Пятница** | Рефакторинг кода, устранение багов Добавление уникальных функций Финальное тестирование (юнит-тесты, API, безопасность) |  |  |
| **Суббота** | Подготовка документации (README, API, развертывание) Финализация проекта |  |  |
| **Воскресенье** | Деплой на сервер (Heroku, Render, Vercel и т. д.) Презентация проекта |  |  |
| **Понедельник** | Анализ обратной связи и багов  Исправление критических ошибок |  |  |
| **Вторник** | Улучшение производительности (оптимизация запросов, кэширование)  Оптимизация работы с медиафайлами |  |  |
| **Среда** | Улучшение UI/UX (небольшие доработки интерфейса)  Проверка адаптивности |  |  |

**5 неделя: Подготовка к файналу, завершение, документация**

|  |  |  |  |
| --- | --- | --- | --- |
| **День** | **Задание** | **Выполнено** | **Сложности** |
| **1** | Финальное тестирование всей платформы  Подготовка презентации для демонстрации |  |  |
| **2** | Резервное копирование данных  Подготовка к развертыванию обновлений |  |  |
| **3** | Улучшение системы поиска и фильтрации записей в личном дневнике(Сортировка по релевантности, Рекомендации) |  |  |
| **4** | Оптимизация базы данных (индексы, анализ нагрузок)  Улучшение документации |  |  |
| **5** | Ментальная подготовка |  |  |
| **6** | **Результат** |  |  |