### План проекта "Виртуальный дневник для школьников"

#### **1. Цель проекта:**

* Создание виртуального дневника для школьников, предоставляющего функции отслеживания расписания, управления домашними заданиями и предоставления готовых решений для ДЗ.

#### **2. Задачи проекта:**

* Разработка сервиса для отслеживания расписания (Schedule Service).
* Разработка сервиса для управления домашними заданиями (Homework Service).
* Разработка сервиса предоставления готовых решений (Solutions Service) с обращением к внешним ресурсам по типу gdz.ru.
* Разработка сервиса распознавания текста с фотографий (OCR Service), определения, являются ли данные примерами, и, в зависимости от результата, обращение к сервису Photomath.
* Создание микросервиса для работы с текстовыми данными (Text Service) с использованием OpenAI GPT-3.
* Тестирование всех микросервисов (unit-тестирование, интеграционное тестирование, API-тестирование).
* Разворачивание приложения на сервере или облачной платформе.
* Создание репозитория с README-файлом, содержащим документацию проекта.
* Разработка стратегии маркетинга и продвижения проекта среди целевой аудитории.

#### **3. Стек технологий:**

* Язык программирования: Java.
* Фреймворк для микросервисов: Spring Boot.
* Базы данных: MySQL или PostgreSQL.
* Искусственный интеллект: OpenAI GPT-3.
* Распознавание текста с фото: Google Vision API или Tesseract OCR (по выбору).

#### **4. Микросервисы:**

* **Schedule Service**:
  + Описание: Хранит и управляет расписанием уроков.
  + Задача: Предоставляет возможность пользователю просматривать, редактировать и удалять занятия в расписании.
  + Технологии: Java, Spring Boot, база данных.
* **Homework Service**:
  + Описание: Хранит и управляет домашними заданиями.
  + Задача: Позволяет пользователю добавлять, редактировать и удалять домашние задания, а также просматривать список ДЗ.
  + Технологии: Java, Spring Boot, база данных.
* **Solutions Service**:
  + Описание: Предоставляет готовые решения для ДЗ, обращаясь к внешним ресурсам по типу gdz.ru.
  + Задача: Взаимодействует с внешними ресурсами для получения решений и сохраняет их в базе данных.
  + Технологии: Java, Spring Boot, веб-скрапинг библиотеки (например, Jsoup).
* **OCR Service**:
  + Описание: Распознает текст с фотографий, определяет, являются ли данные на фотографии примерами, и, в зависимости от результата, обращается к сервису Photomath.
  + Задача:
    - Использует Google Vision API или Tesseract OCR для извлечения текста с фотографий.
    - Обращается к OpenAI GPT-3, чтобы определить, являются ли извлеченные данные примерами.
    - Если данные на фотографии не являются примерами, сохраняет текстовую информацию.
    - Если данные на фотографии являются примерами, идет в сервис Photomath для получения решений.
  + Технологии: Java, Spring Boot, Google Vision API, Tesseract OCR, интеграция с OpenAI GPT-3.
* **Text Service**:
  + Описание: Обрабатывает текстовые данные с помощью OpenAI GPT-3 для генерации решений на основе текстовых заданий.
  + Задача: Генерирует решения на основе текстовых заданий.
  + Технологии: Java, Spring Boot, интеграция с OpenAI GPT-3.

#### **5. Тестирование:**

* Применение TDD (Test-Driven Development) для обеспечения качества кода.
* Unit-тестирование, интеграционное тестирование и тестирование API для всех микросервисов.
* Тестирование функциональности AI и OCR.
* Нагрузочное тестирование для масштабируемости.

#### **6. Деплой:**

* Развертывание приложения на сервере или облачной платформе для обеспечения доступности для пользователей.

#### **7. README-файл:**

* Создание репозитория с README-файлом, содержащим документацию проекта, описание проекта, инструкции по установке и использованию, а также информацию о разработчиках и лицензии.

#### **8. Маркетинг:**

* Разработка стратегии маркетинга и продвижения проекта среди целевой аудитории, включая выбор каналов продвижения и создание маркетинговых материалов.

Это полный план вашего проекта с учетом всех корректив. Если у вас есть дополнительные вопросы или нужна дополнительная информация, пожалуйста, дайте знать. Удачи в разработке вашего виртуального дневника!

### 1. Структура базы данных:

#### **Таблица** Users**:**

* **Поля**:
  + **user\_id** (PRIMARY KEY): Уникальный идентификатор пользователя.
  + **username**: Имя пользователя.
  + **password**: Хэш пароля пользователя.
  + Другие поля, такие как электронная почта, дополнительные данные о пользователе.

#### **Таблица** Subjects**:**

* **Поля**:
  + **subject\_id** (PRIMARY KEY): Уникальный идентификатор предмета.
  + **name**: Название предмета.

#### **Таблица** Classes**:**

* **Поля**:
  + **class\_id** (PRIMARY KEY): Уникальный идентификатор класса.
  + **name**: Название класса (например, 10А, 11Б).
  + **user\_id** (FOREIGN KEY): Ссылка на пользователя (классный руководитель).

#### **Таблица** Schedules**:**

* **Поля**:
  + **schedule\_id** (PRIMARY KEY): Уникальный идентификатор расписания.
  + **class\_id** (FOREIGN KEY): Ссылка на класс.
  + **subject\_id** (FOREIGN KEY): Ссылка на предмет.
  + **day\_of\_week**: День недели (понедельник, вторник и т.д.).
  + **lesson\_number**: Номер урока.
  + **homework\_id** (FOREIGN KEY, NULLABLE): Ссылка на домашнее задание (если есть).

#### **Таблица** Homework**:**

* **Поля**:
  + **homework\_id** (PRIMARY KEY): Уникальный идентификатор домашнего задания.
  + **subject\_id** (FOREIGN KEY): Ссылка на предмет.
  + **class\_id** (FOREIGN KEY): Ссылка на класс.
  + **description**: Описание домашнего задания.
  + **due\_date**: Дата сдачи.
  + **solutions\_id** (FOREIGN KEY, NULLABLE): Ссылка на готовые решения (если есть).

#### **Таблица** Solutions**:**

* **Поля**:
  + **solutions\_id** (PRIMARY KEY): Уникальный идентификатор решения.
  + **homework\_id** (FOREIGN KEY): Ссылка на домашнее задание.
  + **solution\_text**: Текст решения.

#### **Таблица** TextData**:**

* **Поля**:
  + **text\_data\_id** (PRIMARY KEY): Уникальный идентификатор текстовых данных.
  + **user\_id** (FOREIGN KEY): Ссылка на пользователя.
  + **text**: Текстовые данные (например, текст задачи для решения).
  + **is\_example**: Флаг, указывающий, являются ли данные примерами.

### 2. Классы и объекты:

#### **Класс** User**:**

* **Поля**:
  + **userId**: Уникальный идентификатор пользователя.
  + **username**: Имя пользователя.
  + **password**: Хэш пароля пользователя.
  + Другие поля для дополнительных данных о пользователе.

#### **Класс** Subject**:**

* **Поля**:
  + **subjectId**: Уникальный идентификатор предмета.
  + **name**: Название предмета.

#### **Класс** Class**:**

* **Поля**:
  + **classId**: Уникальный идентификатор класса.
  + **name**: Название класса.
  + **classTeacher**: Ссылка на объект пользователя (классный руководитель).

#### **Класс** Schedule**:**

* **Поля**:
  + **scheduleId**: Уникальный идентификатор расписания.
  + **class**: Ссылка на объект класса.
  + **subject**: Ссылка на объект предмета.
  + **dayOfWeek**: День недели (понедельник, вторник и т.д.).
  + **lessonNumber**: Номер урока.
  + **homework**: Ссылка на объект домашнего задания (если есть).

#### **Класс** Homework**:**

* **Поля**:
  + **homeworkId**: Уникальный идентификатор домашнего задания.
  + **subject**: Ссылка на объект предмета.
  + **class**: Ссылка на объект класса.
  + **description**: Описание домашнего задания.
  + **dueDate**: Дата сдачи.
  + **solutions**: Ссылка на объект готовых решений (если есть).

#### **Класс** Solutions**:**

* **Поля**:
  + **solutionsId**: Уникальный идентификатор решения.
  + **homework**: Ссылка на объект домашнего задания.
  + **solutionText**: Текст решения.

#### **Класс** TextData**:**

* **Поля**:
  + **textDataId**: Уникальный идентификатор текстовых данных.
  + **user**: Ссылка на объект пользователя.
  + **text**: Текстовые данные (например, текст задачи для решения).
  + **isExample**: Флаг, указывающий, являются ли данные примерами.

### 3. Связи между классами и таблицами:

* Пользователь (User) связан с Классом (Class) и данными (TextData).
* Класс (Class) связан с Расписанием (Schedule) и Домашними заданиями (Homework).
* Расписание (Schedule) связано с Предметом (Subject) и Домашними заданиями (Homework).
* Домашнее задание (Homework) связано с Предметом (Subject) и Готовыми решениями (Solutions).
* Готовые решения (Solutions) связаны с Домашними заданиями (Homework).
* Текстовые данные (TextData) связаны с Пользователем (User).

Это общая структура базы данных, объектов и их связей для вашего проекта. Разработка функциональности и логики приложения будет зависеть от конкретных требований и бизнес-логики.

#### **День 1: Подготовка и начало разработки**

**Задача 1:** Создать репозиторий на GitHub для проекта.

* Инициализировать пустой репозиторий.
* Добавить файл README.md для документации.

#### **День 2: Настройка среды разработки**

**Задача 2:** Настроить среду разработки.

* Установить и настроить выбранную IDE (например, Visual Studio Code или IntelliJ IDEA).
* Установить необходимые плагины и расширения для работы.

#### **День 3-4: Разработка базовой структуры**

**Задача 3:** Создать базу данных и таблицы.

* Создать таблицы Users, Subjects, Classes, Schedules, Homework, Solutions, TextData.
* Определить структуру каждой таблицы.

**Задача 4:** Разработать классы и связи.

* Создать классы User, Subject, Class, Schedule, Homework, Solutions, TextData.
* Определить связи между классами.

#### **День 5-6: Создание механизма миграции**

**Задача 5:** Создать механизм миграции для базы данных.

* Использовать выбранный инструмент для миграции (например, Flask-Migrate для Python).
* Подготовить миграции для созданных таблиц.

#### **День 7-8: Разработка системы авторизации и регистрации**

**Задача 6:** Создать систему авторизации и регистрации пользователей.

* Разработать страницу входа.
* Реализовать регистрацию с сохранением данных в базе данных.

#### **День 9-10: Создание страницы профиля**

**Задача 7:** Создать страницу профиля пользователя.

* Отобразить информацию о пользователе (имя, класс, предметы).
* Реализовать возможность изменения информации.

#### **День 11-12: Разработка управления расписанием**

**Задача 8:** Разработать функциональность для добавления и редактирования расписания.

* Создать страницу для добавления занятий.
* Реализовать возможность редактирования расписания.

#### **День 13-14: Отображение расписания на графическом интерфейсе**

**Задача 9:** Создать страницу для отображения расписания.

* Отобразить расписание пользователя на графическом интерфейсе.

#### **День 15-16: Управление домашними заданиями**

**Задача 10:** Разработать функциональность для добавления и редактирования домашних заданий.

* Создать страницу для добавления домашних заданий.
* Реализовать возможность редактирования домашних заданий.

#### **День 17-18: Уведомления о сроках сдачи**

**Задача 11:** Реализовать уведомления о предстоящих сроках сдачи домашних заданий.

* Создать механизм отправки уведомлений (например, по электронной почте или через пуш-уведомления).

#### **День 19-20: Разработка сервиса предоставления готовых решений**

**Задача 12:** Разработать сервис для предоставления готовых решений.

* Интегрировать внешние ресурсы (например, веб-сайты с готовыми решениями).
* Создать API для получения решений.

#### **День 21-22: Разработка сервиса распознавания текста**

**Задача 13:** Разработать сервис распознавания текста с фотографий (OCR Service).

* Интегрировать Google Vision API для извлечения текста.
* Создать функцию определения, являются ли данные примерами.

#### **День 23-24: Создание сервиса работы с текстовыми данными**

**Задача 14:** Создать сервис для работы с текстовыми данными (Text Service).

* Интегрировать OpenAI GPT-3 для обработки текста.
* Реализовать функции по работе с текстовыми данными.

#### **День 25-26: Нагрузочное тестирование и оптимизация**

**Задача 15:** Провести нагрузочное тестирование приложения.

* Создать тестовые сценарии для проверки производительности.
* Оптимизировать приложение для обработки большой нагрузки.

#### **День 27-28: Тестирование и отладка**

**Задача 16:** Провести тестирование приложения.

* Написать unit-тесты для каждого компонента.
* Провести интеграционное тестирование и тестирование API.
* Исправить найденные ошибки и недочеты.

#### **День 29-30: Документация и развертывание**

**Задача 17:** Создать README-файл с документацией.

* Описать процесс установки и использования приложения.
* Разработать инструкции для развертывания приложения на сервере или в облаке.

Этот план предоставляет конкретные задачи для каждого дня, чтобы поэтапно разработать идеальное приложение "Виртуальный дневник для школьников".