**Часть 1: Анализ текущего интерфейса**

1. **Описание задачи**:  
   * **Слабые места**:
     + Интерфейс выглядит не совсем современно и требует обновления цветовой схемы, шрифтов и элементов управления.
     + Обложки книг и их описание могут быть плохо размещены на экране, что ухудшает восприятие.
   * **Что требует улучшения**:
     + **Цветовая схема**: Современные цвета, чтобы интерфейс был визуально привлекательным и удобным для восприятия.
     + **Расположение кнопок**: Кнопки могут быть более заметными и удобными для пользователя, особенно при взаимодействии с различными окнами.
2. **Инструменты**:  
   * Использовал максимально современные инструменты для повышения качества дизайна “обёртки” продукта..
3. **Практическое задание**:  
   * Создал документ с описанием слабых мест и начал их улучшение.

### **Часть 1: Разработка нового дизайна**

1. **Цветовая палитра**: Я выберу гармоничные цвета для интерфейса с использованием шрифтов, чтобы создать контрастный и приятный цветовой баланс. Тёмная тема для хорошего восприятия
2. **Шрифты**:
   * **Для заголовков**: жирный шрифт, например, **"Roboto"** или **"Montserrat"**.
   * **Для текста**: стандартный шрифт без засечек, например, **"Open Sans"** или **"Lato"**.
3. **Макет интерфейса**:
   * Поля для ввода
   * Кнопки (логин, регистрация, восстановление пароля)
   * Логотип и заголовки
   * Местоположение иконок Всё будет ориентировано на удобство пользователя.

### **Часть 2: Реализация стилизованного интерфейса**

1. **Стилизация элементов интерфейса**: Я изменю внешний вид кнопок и полей в тёмных тонах
   * **Кнопки**: будут иметь тёмный фон с белым текстом, круглые углы и плавные анимации при наведении.
   * **Поля ввода**: оформлю с учетом контраста, добавлю маркеры ошибок и подсказки.
2. **Добавление иконок**: Использован индивидуальный логотип проекта.
   * Добавлю иконки в кнопки с помощью **Tkinter**.
3. **Реализация цветовой схемы и шрифтов**: После выбора шрифтов было принято решение сделать тёмную цветовую схему моментами разбавленная изображениями.

### **Документирование изменений**

1. **Обновление README**:
2. Book Manager
3. Описание
4. Book Manager — это приложение для управления книгами, которое предлагает пользователям:
5. Современную систему авторизации и регистрации.
6. Систему заметок — приложение запоминает, где пользователь остановился в чтении книги.
7. Систему рекомендаций книг, используя API Google Books.
8. Приложение также имеет современный и стильный интерфейс, с акцентом на удобство пользователя и современный дизайн.
9. Установка
10. Требования:
11. Убедитесь, что у вас установлен Python 3.13 или выше. Скачайте Python с официального сайта.
12. Скачайте проект.
13. Перейдите в папку с проектом и установите зависимости из файла requirements.txt:
14. src/
15. Запустите приложение:
16. main.py
17. Описание интерфейса
18. Дизайн
19. Цветовая палитра: Интерфейс использует гармоничную цветовую палитру для приятного восприятия и контрастности. Применены темные фоны с белым или светлым текстом для удобства чтения.
20. Шрифты: Для заголовков использован шрифт "Roboto" или "Montserrat", а для текста — "Open Sans" или "Lato", что обеспечивает высокую читаемость и современный вид.
21. Кнопки и поля ввода: Все кнопки имеют плавные анимации при наведении, с округленными углами и приятным темным фоном с белым текстом. Поля ввода удобно расположены, с подсказками и маркерами ошибок.
22. Иконки: Иконки для входа, регистрации и других действий были добавлены с использованием библиотеки FontAwesome, что улучшает визуальную идентификацию функционала.
23. Прогресс чтения: Для каждой книги отображается прогресс чтения, включая последнюю страницу и текущую страницу, на которой остановился пользователь.
24. Основные окна:
25. Главное окно: отображает список книг, позволяет переключаться между категориями (прочитанные/запланированные).
26. Окно добавления книги: включает поля для ввода данных о книге и кнопку для добавления.
27. Стилизация окон:
28. Каждое окно в приложении открывается в полноэкранном режиме с удобным интерфейсом. Шрифты крупные, и элементы дизайна плавно изменяются при взаимодействии пользователя.
29. Структура проекта
30. bookmanager/
31. ├── src/
32. │ ├── auth.py # Код для авторизации, регистрации и восстановления пароля
33. │ ├── database.py # Логика работы с базой данных
34. │ ├── osn.py # Интерфейс основного окна
35. │ ├── UI.py # Интерфейс авторизации
36. │ ├── func.py # Логика основного окна
37. │ ├── main.py # Главный файл запуска
38. ├── tests/ # Тесты
39. ├── docs/ # Документация
40. ├── requirements.txt # Зависимости
41. └── README.md # Описание проекта
42. Примечания
43. Для работы с данными пользователей и книг используется SQLite.
44. В проекте реализована локальная база данных для хранения информации о пользователях и их книгах.