МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования «Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

**ОТЧЕТ**

по лабораторной работе №0

**Проектирование приложения**

Выполнили студенты

группы 6304-010302D

Красных Д.П.

Накашидзе Е.С.

Самара 2025

**Цель работы:**

Разработать концепцию и архитектуру веб-приложения, спроектировать API и создать Git-репозиторий.

**Проектирование архитектуры**:

**1.Разработать схему взаимодействия компонентов (сервер, БД, клиент);**

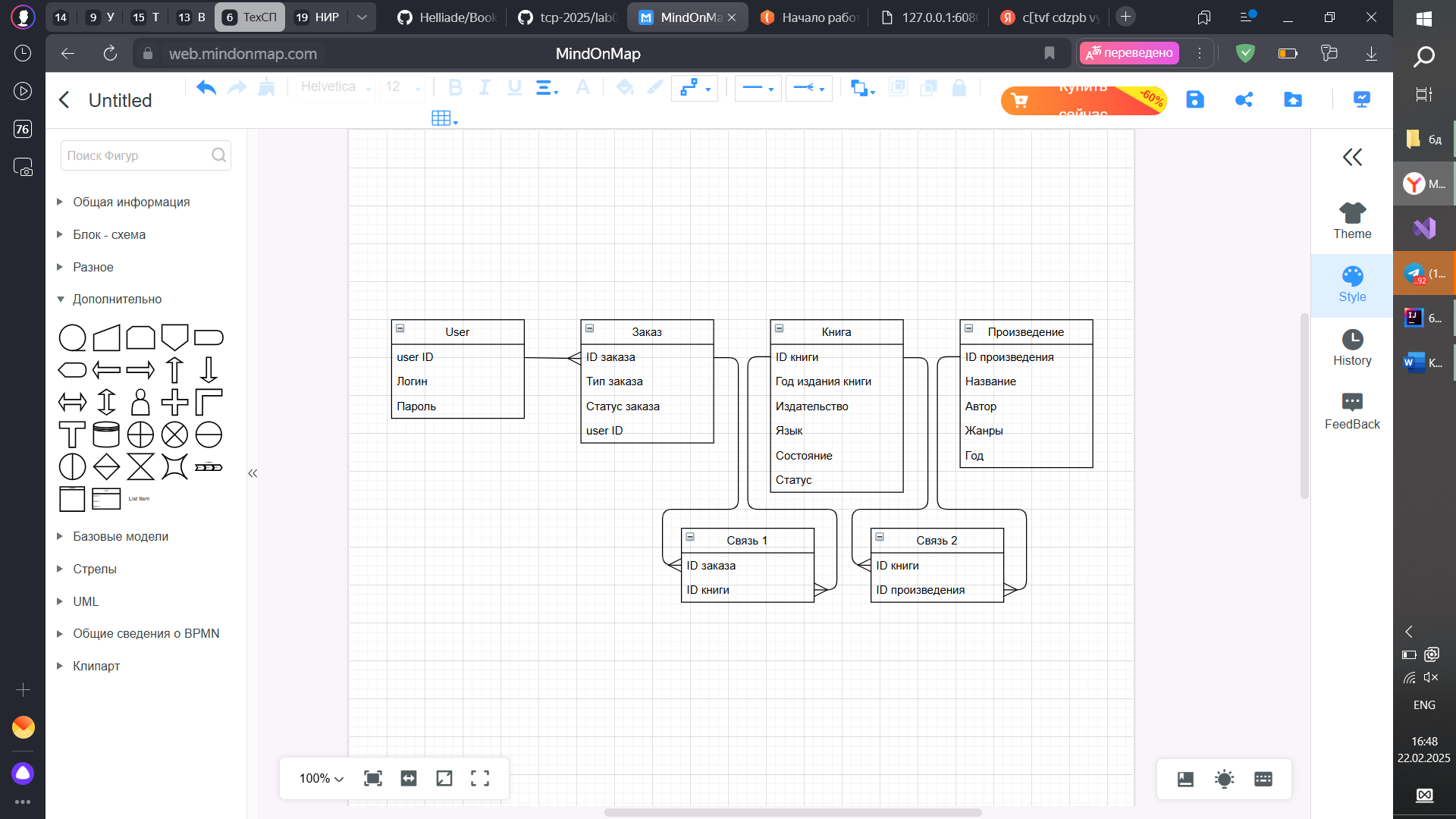
Наше приложение будет представлять собой сервис обмена книгами.

Заходя в веб-приложение, клиенту будет предложено пройти аутентификацию, то есть ввести логин и пароль, которые будут храниться в базе данных (сервер принимает логин и пароль и сверяет их в базе данных). Пользователь после успешной аутентификации попадает на основное окно программы, при условии, что введенные данные корректны и клиент зарегистрирован (иначе ему будет предложено зарегистрироваться).

Пользователь может добавлять в свой профиль заказы на отправку книг и создавать корзину с книгами, которые доступны для получения от других пользователей. Важно отметить, что обмен книгами осуществляется через сервис, что означает, что для получения книги пользователь должен предоставить сервису свою книгу. То есть функционал работы следующий: пользователь может, например, отдать три книги и получить две; однако невозможно отдать две книги и взять три.

В приложении реализована функция поиска произведения по названию, автору, жанру или году написания/публикации, а также поиска книг, соответствующих искомым произведениям, по гиду издания, издательству, языку и состоянию. Доступна история обменов для каждого пользователя, что позволяет отслеживать все взаимодействия в системе.

**2.Разработать логическую схему базы данных (использование, как минимум, 3-х сущностей и 1-ой связи many-to-many);**

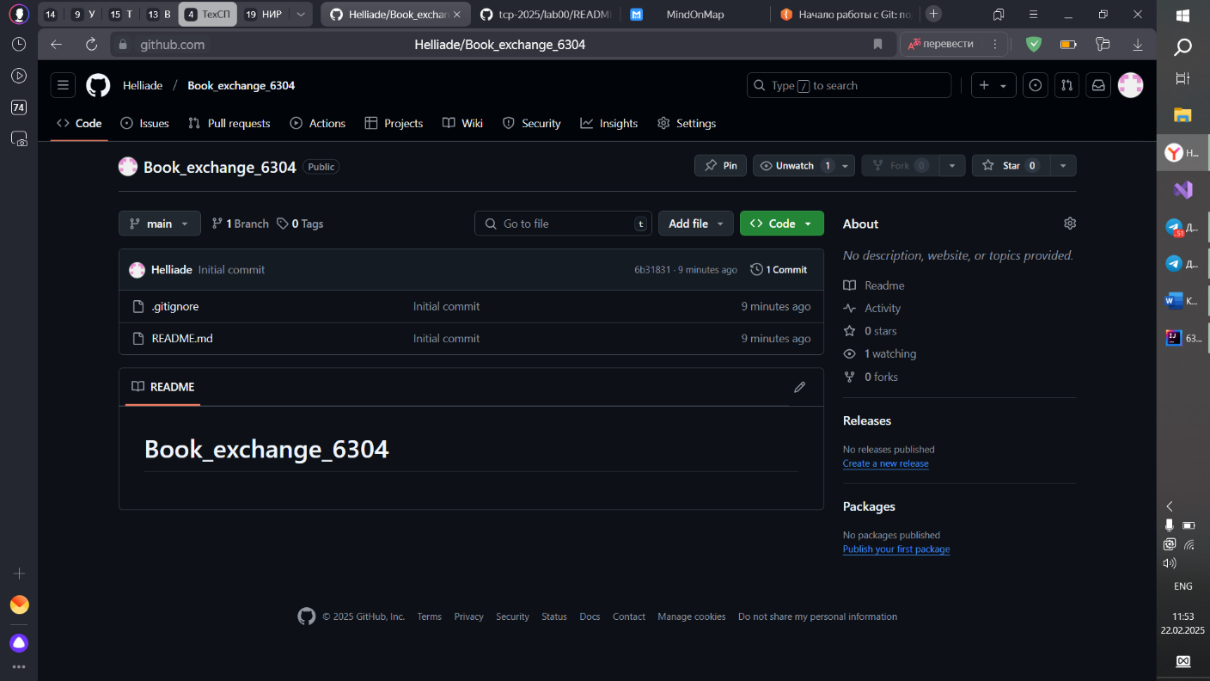


**3.Определить структуру API (REST, методы, URL, параметры, форматы запросов и ответов).**

Запросы:

1. POST /api/v1/users - Регистрация пользователя
2. GET /api/v1/users/{id} - Вход в систему
3. GET /api/v1/book/{id} - Просмотр информации о конкретной книге
4. GET /api/v1/book - Просмотр информации о всех книгах
5. GET /api/v1/work/{id} - Просмотр информации о конкретном произведении
6. GET /api/v1/work - Просмотр информации о всех произведениях
7. GET /api/v1/work?author=Tolstoy&genre=classic – Пример поиска по произведениям
8. PATCH /api/v1/basket/{book} - Добавление/удаление продуктов в/из корзины
9. PATCH /api/v1/basket/{count} - Указание количества выбранного товара
10. POST /api/v1/book/{info} – Создание книги
11. POST /api/v1/basket - Сохранение заказа
12. GET /api/v1/basket - Просмотр истории заказов

**Создание Git-репозитория**:



* **Добавить .gitignore для исключения ненужных файлов (виртуальное окружение, логи, артефакты сборки и т. д.).**

# Compiled class file

\*.class

# Log file

\*.log

# BlueJ files

\*.ctxt

# Mobile Tools for Java (J2ME)

.mtj.tmp/

# Package Files #

\*.jar

\*.war

\*.nar

\*.ear

\*.zip

\*.tar.gz

\*.rar

# virtual machine crash logs, see http://www.java.com/en/download/help/error\_hotspot.xml

hs\_err\_pid\*

replay\_pid\*

**Подготовка окружения**:

* + Установить Java;
  + Настроить виртуальное окружение и зависимости.

