# **ТЗ: Информационная система для поиска и взаимодействия с репетиторами**

## **1. Общее описание**

Веб-приложение (Next.js) для учеников и настольное приложение (Avalonia UI) для репетиторов.

Серверная часть — ASP.NET Core Web API.

## **2. Роли пользователей**

1. **Ученик (через сайт)**

* Регистрируется и авторизуется
* Ищет репетиторов по фильтрам (предмет, город, формат занятий и т.д.)
* Смотрит профиль репетитора
* Оставляет заявку на занятие
* Видит историю своих заявок
* Может оставить отзыв

1. **Репетитор (через десктоп-приложение)**

* Регистрируется и авторизуется
* Заполняет профиль (предметы, цена, формат занятий, расписание)
* Получает и обрабатывает заявки
* Принимает/отклоняет заявки
* Управляет расписанием
* Просматривает отзывы

## **3. Функциональные требования**

**Клиент (Next.js):**

* Регистрация / Авторизация (JWT)
* Поиск репетиторов по предмету, цене, рейтингу, формату (онлайн/оффлайн)
* Просмотр карточки репетитора (о себе, опыт, предметы, цена, отзывы)
* Форма отправки заявки на занятие
* Просмотр отправленных заявок и их статуса (в ожидании / подтверждено / отклонено)
* Система отзывов

**Настольное приложение (Avalonia UI):**

* Регистрация / Авторизация
* Профиль с возможностью редактирования:
  + ФИО
  + Описание
  + Предметы
  + Цена
  + Формат занятий
  + Расписание (можно упростить)
* Список входящих заявок:
  + Подтверждение / отклонение
* Просмотр расписания (занятые/свободные слоты)
* Просмотр отзывов

**API (ASP.NET Core):**

* Auth: регистрация, логин, refresh token
* Users: профили учеников и репетиторов
* Subjects: справочник предметов
* Tutors: фильтрация и выдача списка репетиторов
* Requests: заявки от учеников
* Schedule: расписание репетитора
* Reviews: отзывы от учеников

## **4. Сущности БД (минимум)**

1. User (Id, Name, Email, PasswordHash, Role)
2. TutorProfile (UserId, Bio, PricePerHour, Format, City)
3. Subject (Id, Name)
4. TutorSubjects (TutorId, SubjectId)
5. LessonRequest (Id, StudentId, TutorId, SubjectId, Message, Status, CreatedAt)
6. ScheduleSlot (Id, TutorId, StartTime, EndTime, IsBooked)
7. Review (Id, StudentId, TutorId, Rating, Text, CreatedAt)

**5. Авторизация:**

* JWT токены
* Роли: Student, Tutor
* Защищенные эндпоинты (например, только репетитор может редактировать расписание)

**6. Возможности расширения в будущем:**

* Онлайн-оплата (Stripe)
* Уведомления (email / push)
* Чат между учеником и репетитором
* Интеграция с календарями (Google/Outlook)
* Админка (для модерации)