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 6301-010302D

Рамазанова Милана Иванова Анна

Самара 2025

**Задание на лабораторную работу**

* Определение концепции приложения.
* Выбрать стек технологий.
* Создание схем базы данных и API.
* Настройка репозитория Git, добавление .gitignore.

**Схема взаимодействия компонентов (клиент, сервер, БД).**

1. Клиент (Frontend):

* Веб-приложение
* Страница входа
* Страница регистрации
* Страница сеансов и психологов
* Информация о помощнике-браслете

2. Сервер (Backend):

* API
* Обработка запросов от клиента
* Логика аутентификации и авторизации
* Взаимодействие с базой данных

3. База данных:

* Хранение информации о пользователях, сеансах, психологах и уведомлениях

**Концепция приложения.**

1. Регистрация нового пользователя:

* Клиент отправляет POST-запрос на сервер с данными регистрации
* Сервер проверяет данные и создает нового пользователя в базе данных
* Сервер отправляет ответ клиенту

2. Вход зарегистрированного пользователя:

* Клиент отправляется на страницу регистрации, заполняет поля и отправляет данные на сервер.
* Сервер проверяет данные и возвращает токен аутентификации (например, JWT) при успешной проверке.
* Клиент далее использует логин и пароль, которые ввел в регистрации.

3. Получение списка сеансов и психологов:

* После входа клиент отправляет GET-запрос на сервер для получения информации о сеансах и доступных психологах.
* Сервер извлекает данные из базы данных и возвращает их клиенту.

4. Создание нового сеанса:

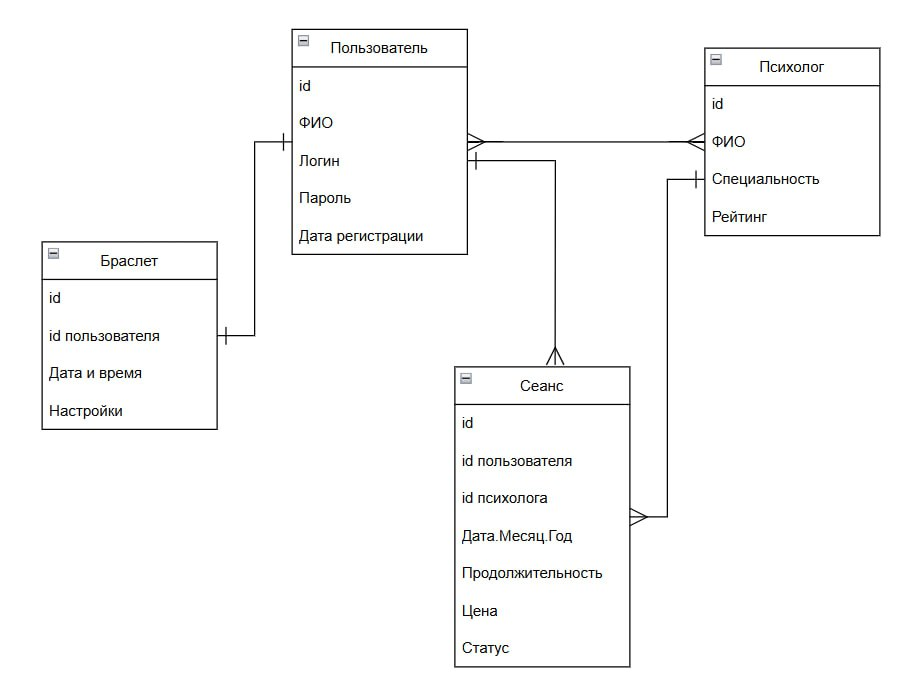
* Клиент отправляет POST-запрос на сервер с данными о новом сеансе (выбранный психолог, дата и время).
* Сервер добавляет новый сеанс в базу данных и возвращает подтверждение клиенту.

5. Уведомления о сеансах:

* Сервер проверяет предстоящие сеансы и отправляет уведомления пользователям.
* Так же уведомления отправляются браслетом-помощником, который так же напоминает похвалить себя в определенные часы.

**Создание схем базы данных и API.**

Схема базы данных:



Структура API:

1. Регистрация пользователя

Метод: POST

URL: /api/users/register

Формат запроса

{

"email": "user@example.com",

"password": "securepassword"

}

Формат ответа:

{

"message": "User registered successfully"

}

{

"error": "Invalid input"

}

2. Вход пользователя

Метод: POST

URL: /api/users/login

Формат запроса:

{

"email": "user@example.com",

"password": "securepassword"

}

Формат ответа:

{

"token": "jwt\_token\_here"

}

{

"error": "Invalid credentials"

}

3. Получение сеансов пользователя

Метод: GET

URL: /api/users/{userId}/sessions

Формат ответа:

[

{

"id": 1,

"date\_time": "2020-10-01T10:00:00Z",

"psychologist\_id": 2

},

...

]

4. Получение списка психологов

Метод: GET

URL: /api/psychologists

Формат ответа:

[

{

"id": 1,

"name": "Dr. Smith",

"specialization": "Cognitive Behavioral Therapy"

},

...

]

5. Получение информации о браслете

Метод: GET

URL: /api/users/{userId}/bracelet

{

"reminders": [

{

"message": "Time for your session!",

"time": "2025-10-01T09:00:00Z"

},

{

"message": "Remember to praise yourself!",

"time": "2024-10-01T20:00:00Z"

}

]

}

**Cтек технологий.**

1. PostgreSQL

2. Visual Studio Code

3. Git, GitHub

4. OpenAPI/Swagger