

Московский авиационный институт
(национальный исследовательский университет)

Институт компьютерных наук и прикладной математики
Кафедра вычислительной математики и программирования

Курсовой проект по курсу «Базы данных»

Студент: А. А. Федоров
Преподаватель: А. В. Малахов
Группа: М8О-309Б-22
Дата: 13.12.2024
Оценка:
Подпись:

Москва, 2024

Курсовой проект: сервис для поиска знакомств

1 Описание проекта

Сервис для поиска знакомств представляет собой веб-приложение, которое позволяет пользователям находить новых знакомых на основе их предпочтений, местоположения и интересов. Основная цель проекта — создание интуитивно понятного интерфейса, который предоставляет удобный функционал для поиска и взаимодействия с потенциальными друзьями и партнёрами, а также надёжной серверной части и базы данных.

2 Основной функционал

- Регистрация и аутентификация пользователей
- Просмотр и редактирование пользовательских профилей
- Загрузка и управление (смена главной, удаление) фотографиями профиля
- Просмотр всех пользователей
- Система лайков и мэтчей (совпадений) на основе взаимных лайков
- Обмен сообщениями в режиме реального времени (используется WebSocket) при мэтче
- Подписка на премиум функционал

3 Архитектура

- Frontend: Для создания пользовательского интерфейса используется фреймворк шаблонизатор Thymeleaf, который генерирует HTML файлы на основе данных, полученных с бэкенда. Также используются чистые HTML, CSS, JS.
- Backend: Для обработки запросов и логики взаимодействия с базой данных используется Spring Boot, мощный DI фреймворк на Java. Он предоставляет эндпоинты для работы с пользователями, лайками, чатами, фотографиями, подписками и пр.
- Database: Для хранения данных используется PostgreSQL

4 Используемые технологии

При разработке приложения были использованы следующие технологии:

- **Spring Boot** для создания серверной части приложения.
- **Thymeleaf** для динамической генерации HTML-страниц.

- **Spring Security** для реализации механизмов аутентификации и авторизации.
- **Spring JDBC** для подключения, формирования и отправки запросов к базе данных
- **Spring Messaging** и **Spring WebSocket** для мгновенной передачи сообщений между пользователями через веб-сокеты
- **PostgreSQL** в качестве системы управления базами данных.
- **Docker** для контейнеризации и упрощения развертывания приложения.
- **Lombok** для упрощения написания кода.
- **Maven** для управления зависимостями.

5 Связи между таблицами базы данных

Для реализации функционала приложения была спроектирована реляционная база данных с таблицами, описанными ниже:

Таблица	Описание
users	Основная таблица, содержащая информацию о пользователях, включая имя, фамилию, пол, дату рождения и контактные данные.
bios	Таблица, хранящая биографии пользователей, включая информацию о том, что они ищут и описание себя.
user_photos	Таблица для хранения фотографий пользователей, включая имя файла, дату загрузки и информацию о том, является ли фото основным.
likes	Таблица, содержащая информацию о лайках пользователей, где хранятся идентификаторы отправителя и получателя, а также временная метка.
matches	Таблица, которая хранит мэтчи между пользователями, а именно идентификаторы двух пользователей и дату подтверждения совпадения.
messages	Таблица для хранения сообщений между пользователями, а именно идентификаторы отправителя и получателя, текст сообщения и временную метку.
subscriptions	Таблица, хранящая информацию о подписках пользователей на премиум функционал, в т.ч. даты начала и окончания подписки.
admins	Таблица, содержащая данные администраторов системы, включая имя, email и хеш пароля.

Таблица 1: Описание таблиц в базе данных

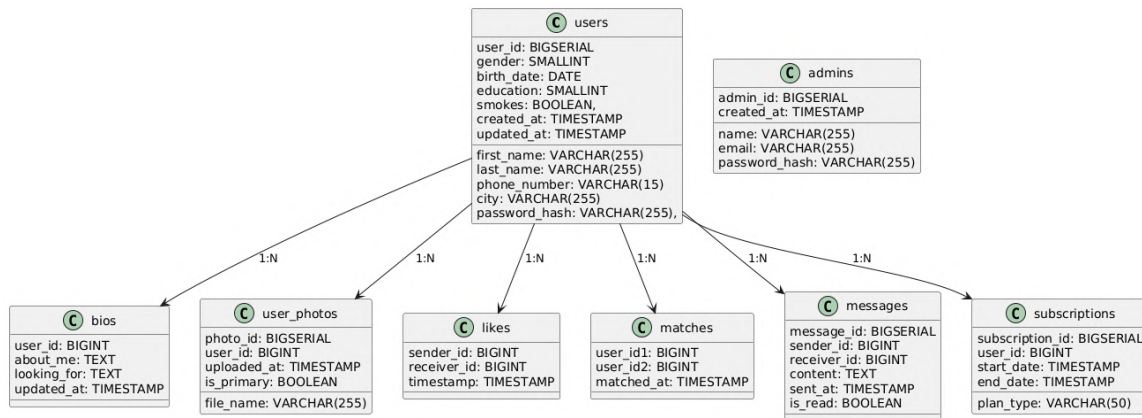
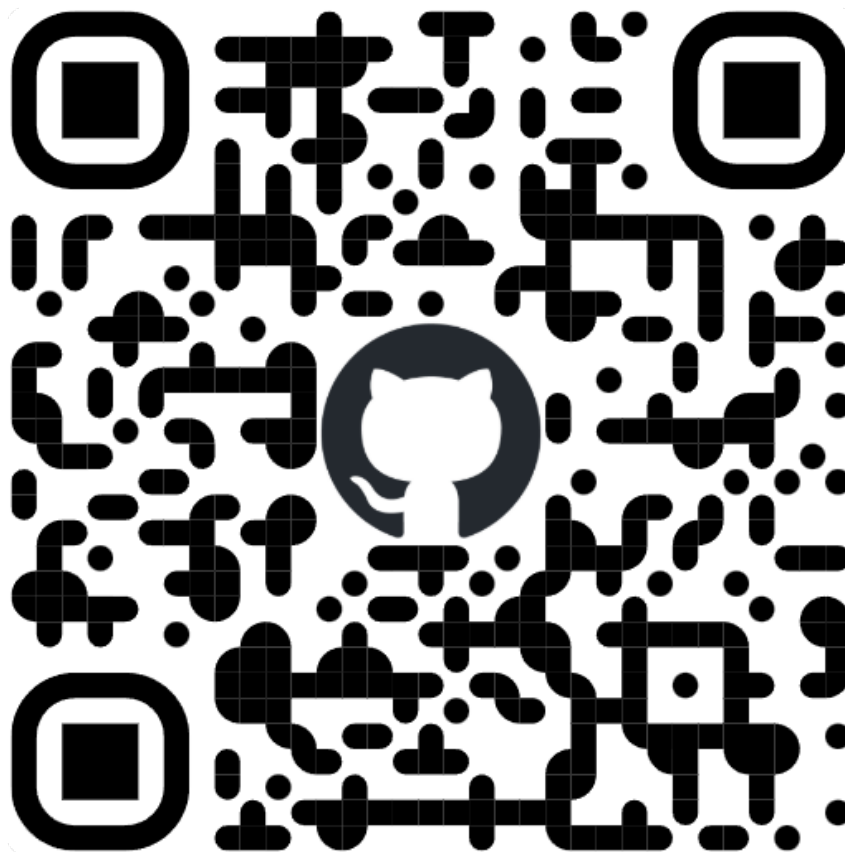


Рис. 1: Связи между таблицами

6 GitHub проекта



7 Иллюстрации использования приложения

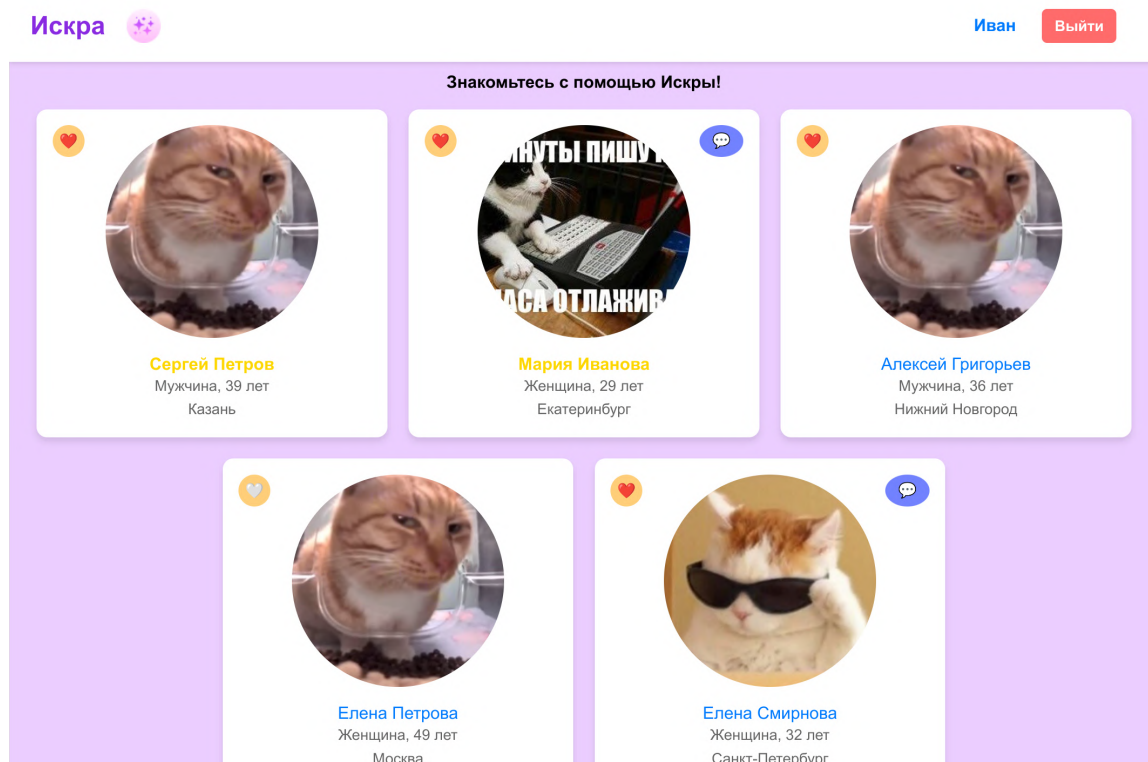


Рис. 2: Главная страница для вошедшего пользователя

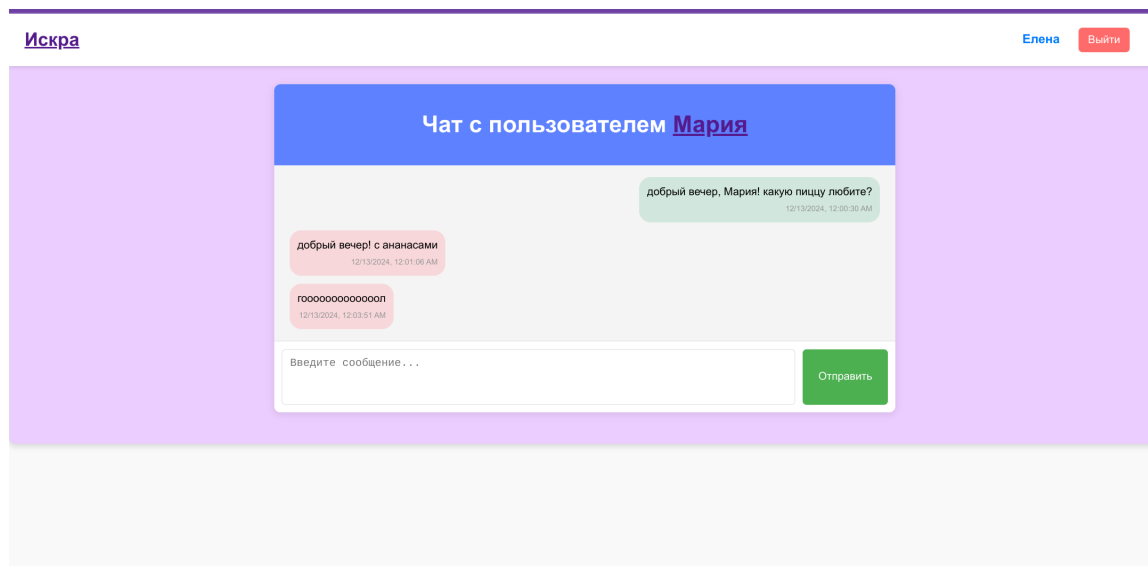


Рис. 3: Чат между двумя пользователями

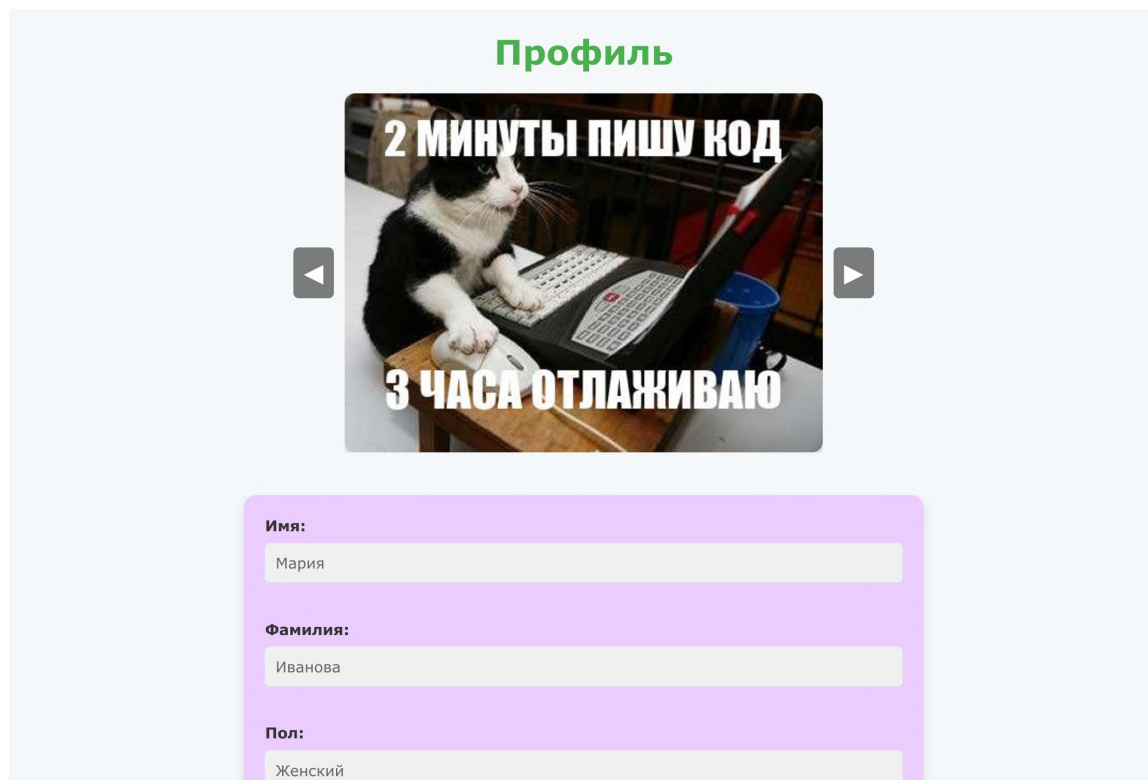


Рис. 4: Вид профиля для всех, кроме владельца

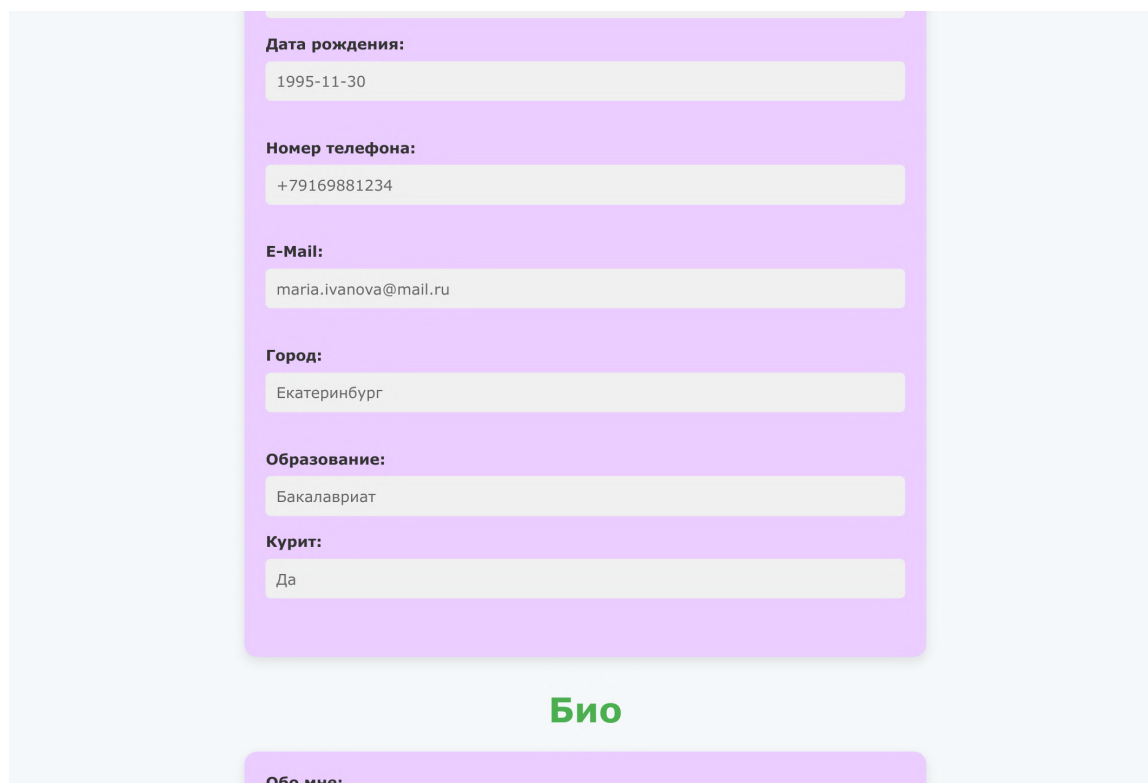


Рис. 5: Вид профиля для всех, кроме владельца (2)

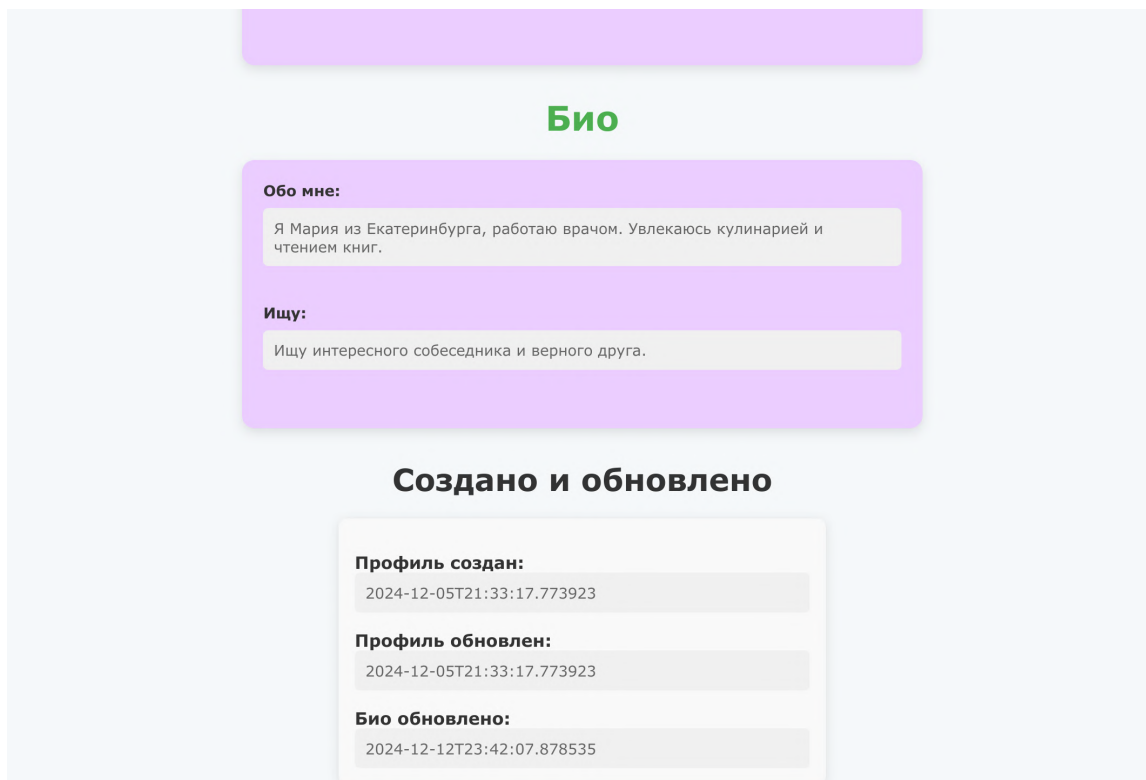


Рис. 6: Вид профиля для всех, кроме владельца (3)

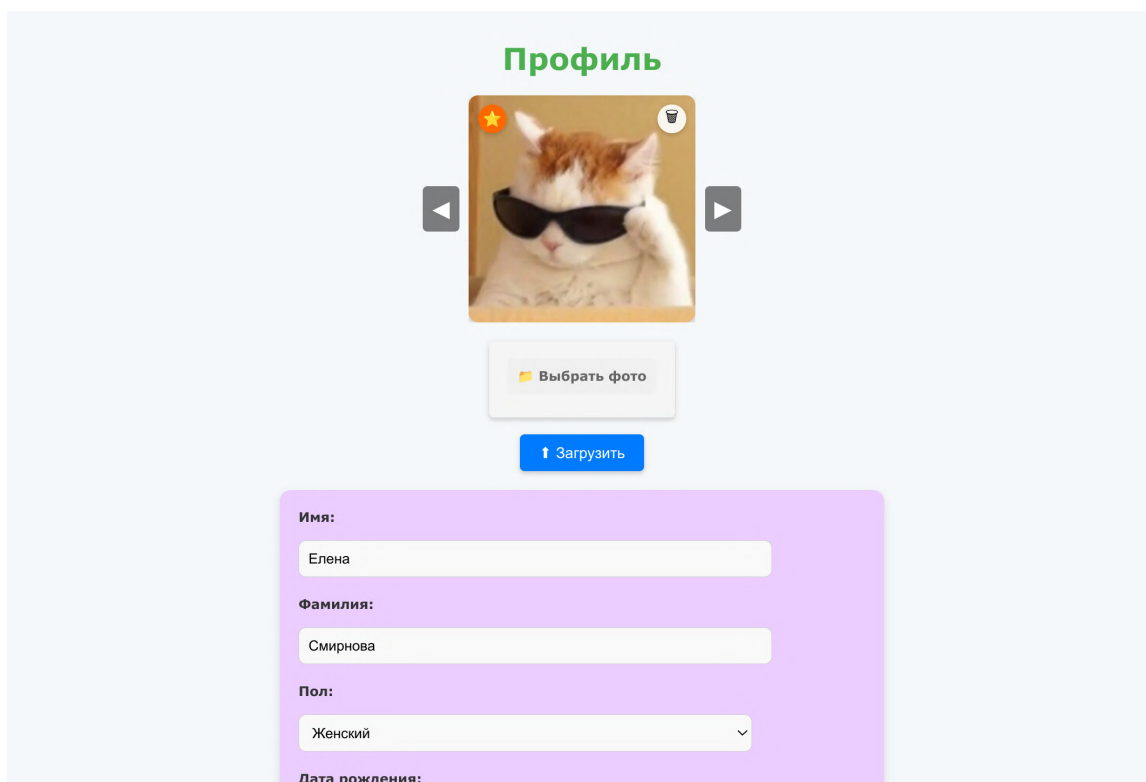


Рис. 7: Вид профиля для владельца страницы

Дата рождения:

03/21/1992

Номер телефона:

+79169876543

E-Mail:

elena.smirnova@mail.ru

Город:

Санкт-Петербург

Образование:

Среднее

Курит:

Да

Сохранить профиль

Био

Обо мне:

Привет, я Елена, работаю дизайнером в Санкт-Петербурге. Увлекаюсь искусством и йогой.

Рис. 8: Вид профиля для владельца страницы (2)

Обо мне:

Привет, я Елена, работаю дизайнером в Санкт-Петербурге. Увлекаюсь искусством и йогой.

Ищу:

Пупу

Сохранить био

Создано и обновлено

Профиль создан:	2024-12-05T21:19:52.228074
Профиль обновлен:	2024-12-13T07:04:57.543113
Био обновлено:	2024-12-13T07:03:16.418651

Рис. 9: Вид профиля для владельца страницы (3)

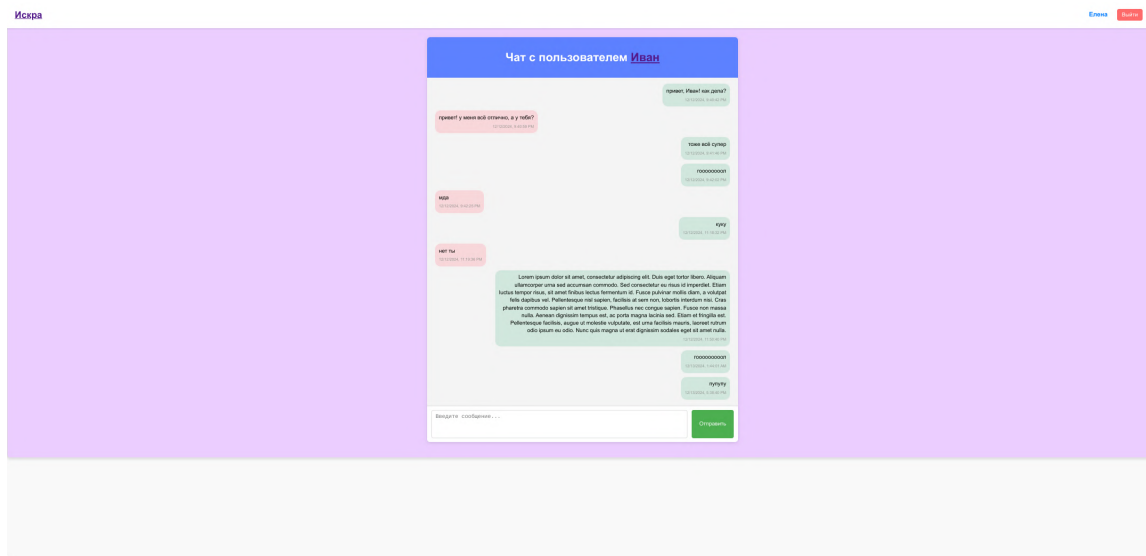


Рис. 10: Чат между двумя пользователями (больше текста)

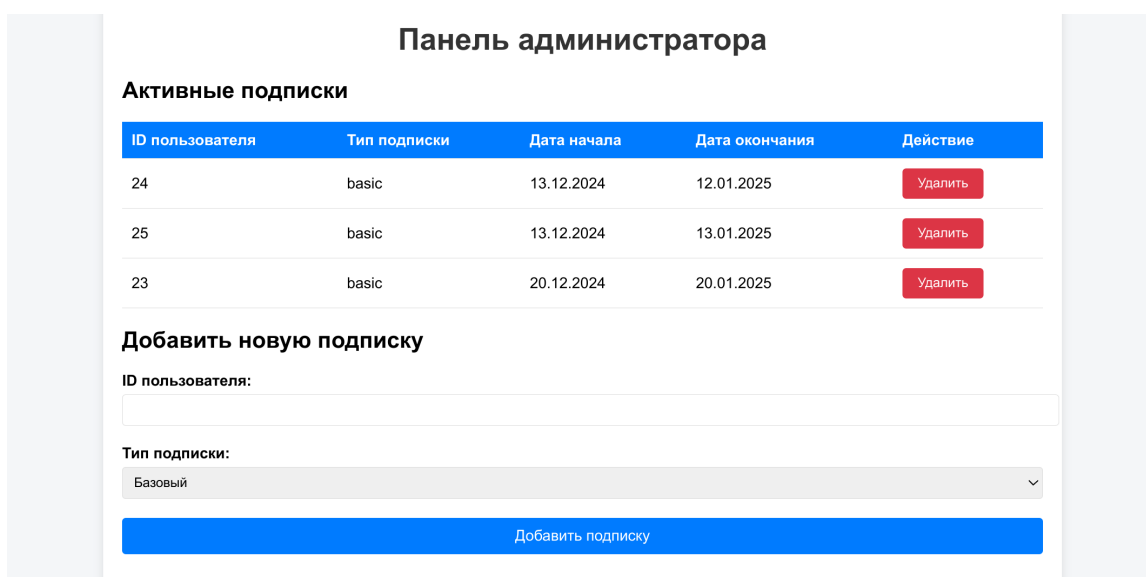


Рис. 11: Админ-панель для добавления и удаления премиум-подписок пользователей

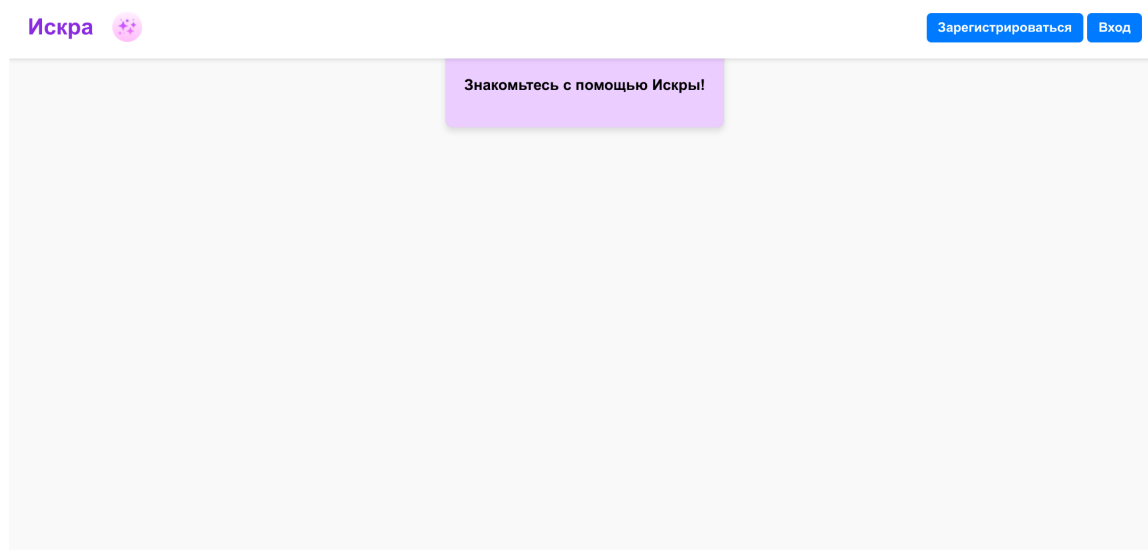


Рис. 12: Главная страница для новошедших

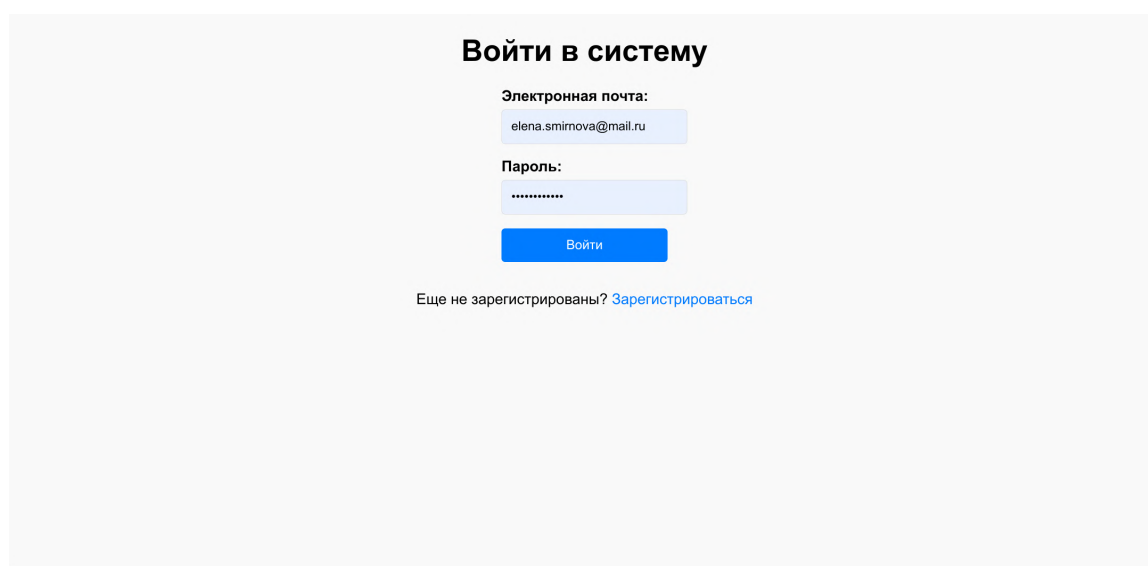


Рис. 13: Страница для входа

Регистрация

Имя:

Фамилия:

Пол:
 ▼

Дата рождения:
 □

Телефон:

Email:

Город:

Образование:
 ▼

Курите ли вы?
☐

Пароль:

Обо мне:

Ищу:

[Зарегистрироваться](#)

Рис. 14: Страница для регистрации

8 Выводы

В рамках этого проекта я закрепил знания по базам данных, бэкенду и фронтенду. Хотелось бы отметить, что запросы были несложными (и по сути, если бы не практика их написания, лучше использовать Spring Data JPA) и на них времени уходило немного. Из нового - попробовал написать триггеры руками (использовал для мэтчей), прикольно, но более серьезно буду знакомиться уже на работе. Однако много времени ушло на интерфейс, несмотря на использование более простых решений типа шаблонизатора Thymeleaf, это не особо неприятно (хотя в перспективе полезно для фуллстек-разработки).

Раньше я считал шаблонизаторы не очень хорошим и полезным инструментом (т.к. в какой-то степени бэк проникает во фронт визуально, но фактически всё разделено фреймворком + выглядит странно + непонятно насколько менее эффективно). Но в условиях ограниченного времени они здорово помогают, когда можно передать данные без дополнительных запросов к серверу. В продакшене правда я все ещё не думаю, что это нужно, т.к. зоны ответственности разделены чаще всего.

Подводя итог, хотел бы сказать, что проект мне понравился, считаю его полезным. Планирую его доработать, т.к. в рамках 2 месяцев (из которых большая часть времени уходит не на проект, а на другие предметы и дела) сложно сделать хороший продукт в одиночку, где-нибудь да выскакивают костыли. Возможно ещё добавлю несложную ML модель, интересно разобраться в этой сфере, т.к. по рассказам друзей современные рекомендации в подобных приложениях работают плохо, возможно что-то смогу улучшить или как минимум лучше разобраться для себя.