

«Национальный исследовательский университет  
«Высшая школа экономики»  
Лицей

Индивидуальная выпускная работа

**ИТ-ПРОЕКТ: CurioSpace**

<https://github.com/XRay876/IVR/tree/master>

*Выполнил: Семенов Тимофей Игоревич*

*Группа: 11ИЗ*

Москва – 2023

## Оглавление

1. Введение .....	2
2. Проблемное поле .....	2
3. Образ продукта .....	3
4. Функциональность продукта .....	3
5. Backend часть.....	4
6. Средства разработки.....	5
7. Этапы разработки .....	6
8. Рефлексия .....	6
9. Заключение.....	7

## **1. Введение**

В качестве ИВР я хотел сделать что-то полезное и одновременно затрагивающее повседневные занятия и увлечения людей. Более того, я уже давно занимаюсь программированием и веб-разработкой. За последний год я уже успел написать 2 небольших, но полезных сайта, один из которых используется среди сотрудников компании Росатом. Именно поэтому мой выбор пал на IT-проект: я решил сделать мессенджер, который позволит находить и общаться с людьми со схожими интересами, искать новые знакомства и делиться полезными материалами.

## **2. Проблемное поле**

В наше время многим школьникам и взрослым не хватает возможностей легко и быстро найти людей по интересам, одновременно с этим не хочется далеко отходить от своего компьютера. Существует несколько оснований, по которым можно утверждать существование данной проблемы:

1. Отсутствие специализированных ресурсов: на данный момент в интернете сложно найти, хороший сервис, в который эффективно помог бы людям находить друг друга с учетом их интересов. Нередко общие социальные сети или форумы могут быть слишком общими для поиска более узконаправленных интересов. Пример: чаты в Телеграм, ВК и других соц. сетях - нужно долго искать подходящие чаты, суть мессенджеров не направлена на обучение и поиск людей по схожим интересам
2. Недостаток времени и удобства: в современном мире люди, особенно школьники и взрослые, сталкиваются с ограниченным количеством времени. Они могут предпочесть оставаться за компьютером из-за удобства и ограниченного времени, нежели тратить его на физические встречи или длительные поиски единомышленников.
3. Предпочтение находиться в интернете: в наше технологичное время люди предпочитают общаться в интернете нежели искать физические встречи.

4. Алгоритмы социальных сетей: Существующие алгоритмы социальных сетей могут ограничивать видимость нишевых сообществ или интересных групп, что затрудняет людям нахождение друг друга в онлайне.

Именно решением этой проблемы занимается мой продукт, который более специализированный и удобный для поиска людей по интересам.

### **3. Образ продукта**

Мой продукт представляет из себя адаптивный вебсайт, позволяющий людям находить нужный чат на любом устройстве для общения по общему интересу или занятию, обеспечивая более глубокие и качественные взаимодействия. Во-первых, мой продукт находится в интернете, поэтому им могут пользоваться более 60 процентов населения всего мира у кого есть интернет (по данным: <https://news.rambler.ru/internet/45705599-tsifra-dnya-skolko-chelovek-v-mire-polzuyutsya-internetom/>). Во-вторых, он сразу же заполняет пробел в отсутствии специализированного ресурса, потому что этот сайт прямонаправленный на решение поставленной проблемы. Также у него эффективный поиск, который позволяет экономить время и быстро найти релевантные чаты без необходимости длительных поисков и фильтрации большого количества ненужной информации.

Таким образом, продукт предоставляет эффективное решение для соединения людей с общими интересами, обходя препятствия, связанные с отсутствием специализированных ресурсов, ограниченным временем, предпочтением онлайн-коммуникаций и ограничениями алгоритмов социальных сетей.

### **4. Функциональность продукта**

Итоговый продукт удалось реализовать с большинством заявленными выше характеристиками. Так мне удалось полностью реализовать сценарий аутентификации пользователя, где юзер создает себе аккаунт с собственным логином, почтой и паролем; присутствует функция входа в аккаунт, где пользователь указывает свои логин и пароль; существует личный кабинет пользователя, в котором отображается фото профиля, имя и возраст

пользователя, информация о хобби, интересах, городе, цели общения, а также информация о друзьях; есть возможность редактировать профиль: изменить информацию о себе, поменять логин при входе, сменить изображение профиля и прочее; сценарий добавления пользователя в друзья, а также “попап” окно с отображением всех друзей пользователя и входящие заявки в друзья; возможность посмотреть профиль друга или другого пользователя; присутствует сценарий с отображением всех возможных чатов с названиями по интересам; возможность найти подходящий чат в поиске, лишь указав интересующее вас поле для общения; также есть вкладка с чатами, где юзер уже писал сообщения и может продолжить переписку в этой группе; есть возможность отправлять не только текстовые сообщения, но и документы, фотографии и прочие файлы с любым расширением; также для полного удобства есть кнопка, чтобы выйти из аккаунта и можно было создать либо новый профиль, либо войти в другой; дополнительный сценарий это возможность использовать сайт на любом устройстве, то есть создание адаптивного сайта, который можно открыть и с вертикальным разрешением на телефоне и с горизонтальным расширением на компьютере; также еще одним хорошим дополнением в работе было создание темной и светлой тем сайта.

## **5. Backend часть**

Backend составляющая сайта была реализована по средствам ORM Django. База данных использовалась SQLite3. В ней присутствуют основные Django модели, включая модель `auth_user` для регистрации и хранения информации о пользователях с полями `id`, `password`, `last_login`, `is_superuser`, `username`, `last_name`, `email`, `is_staff`, `is_active`, `data_joined`, `first_name`; также я создал дополнительную модель `Profile` со связью `OneToOne` с таблицей `auth_user` по `id`, для того, чтобы добавить дополнительные поля к информации о пользователе, а именно: `avatar` (со ссылкой на фото), `bio` (информация “о себе”), `user_id`, `age` (информация о возрасте пользователя), `hobbies` (информация о хобби пользователя), `city` (информация о городе проживания пользователя), `created` (дата создания профиля), `updated` (дата обновления профиля) и поле `slug`

(для хранения уникальной ссылки на пользователя). Для создания чата, участников чата и сообщений я создал 3 дополнительные модели: chat, chat\_members, message – соответственно. В модели chat присутствуют поля: id, title (название чата), created\_at (дата создания чата), updated\_at (дата изменения чата), owner\_id (для хранения id владельца чата, в моем проекте пока не успел использовать – создал для будущих доработок). В модели chat\_members есть поля id, chat\_id (id чата в котором находится юзер) и user\_id (id юзера). В модели message присутствуют поля: id, message (текст сообщения), created\_at (дата создания сообщения), updated\_at (дата изменения сообщения), chat\_id (хранить id чата), user\_id (хранить id пользователя), document (со ссылкой на документ), image (со ссылкой на картинку). Также была создана дополнительно модель для хранения связей между пользователями Relationships, где два поля связаны ForeignKey с моделью Profile.

## **6. Средства разработки**

Для создания бэкенд части, как уже упоминалось выше, я использовал ORM Django. Как уже было понятно основным фреймворком в моей работе был именно Django, потому что я уже работал с этим фреймворком и имею хорошие знания языка программирования Python. Для работы над фронт частью я использовал HTML5, CSS и JavaScript, также в дополнения к ним я использовал библиотеку JS JQuery и Ajax, Google иконки и подключенный Google шрифт. На сайте присутствует возможность поменять тему (светлая и темная), что также написано с использованием JS и CSS. Для создания адаптивного сайта пришлось помучаться с CSS и функцией @media. Для реализации чатов и сообщений я выбрал наиболее удобную для моего фреймворка библиотеку, а именно Django Channels и подходящий для нее WebSocket. Подключение и использование базы данных не сильно усложнило задачу (кроме проблем, которые описаны в части “Рефлексия”), ведь я использовал стандартную Django БД, а именно SQLite3. Также в планах выгрузить сайт на хостинг с использованием технологии Docker, потому что один раз, но уже приходилось с ней работать.

## **7. Этапы разработки**

План разработки был выстроен следующим образом:

- 1) Написать основу frontend части сайта: это получилось сделать за первую неделю после одобрения заявки
- 2) Сделать backend часть: появилась огромная проблема со сложным использованием и подключением Django Channels для работы с чатами и сообщениями. В результате огромное количество времени и сил ушло на изучение и создание рабочего backend.
- 3) Подключить БД
- 4) Доделать дизайн и внешнюю составляющую сайта: сильно пришлось повозиться с созданием адаптивного сайта, чтобы все объекты нормального выглядели как на компьютере, так и на телефоне
- 5) Протестировать сайт на локальном сервере
- 6) Выгрузить сайта на сервер: появилась проблема, потому что при создании мессенджера не многие хостинги могут поддерживать выгрузку чатов и онлайн-общения
- 7) Протестировать функциональность продукта
- 8) Отдебажить сайт
- 9) Выгрузить сайт на постоянное использование
- 10) Думать над улучшением и продвижением сайта

На этапе написания отчета все до 8 пункта задачи были проведены в том же порядке, как и предполагалось.

## **8. Рефлексия**

Я столкнулся с рядом проблем:

- 1) Проблема с реализацией поиска: В предыдущих своих проектах я использовал MySQL, но в этот раз выбор пал на БД SQLite3. В процессе работы, как оказалось, она не поддерживает поиск без учета регистра, поэтому чат на данном этапе немного не доработан и требует поиск с учетом регистра.

- 2) Сложность с подключением Django Channels и отсутствие хороших гайдов и документов
- 3) Сложность с использованием веб сокета

На будущее:

- 1) Улучшить поиск
- 2) Добавить личные сообщения
- 3) Реализовать функцию сортировки чатов по сложности, уровню заинтересованности
- 4) Добавить рейтинг пользователя, чтобы избежать общения с нежелательными пользователями
- 5) Добавить функцию создания собственных чатов, чтобы БД пополнялась новыми увлечениями и интересами

Приобретённый опыт:

- 1) Навыки работы с Django Channels и создание сайта-мессенджера
- 2) Работа с WebSocket
- 3) Создание адаптивной верстки (с такой работой пришлось столкнуться в первый раз, но мне понравилось)
- 4) Использование SQLite3 в качестве БД

Возможные риски:

- 1) Невозможность спроектировать удобный пользовательский интерфейс: в результате, все удалось и мне понравилось работать над адаптивной версткой сайта
- 2) Неправильная оценка стека технологий, которые мне необходимо изучить, как следствие – нехватка времени на изучение необходимых языков программирования, фреймворков и т. п.: и правда, с Django Channels и сокетами пришлось помучаться

## **9. Заключение**

Работа над проектом получилась непростая, но увлекательная. Я научился работать с новыми библиотеками, улучшил знания языков программирования, а



также изучил понятие веб сокетов. Я создал уникальный продукт, у которого практически не существует качественных аналогов.