

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)**

Факультет **Прикладной информатики**

Направление подготовки **45.03.04 Интеллектуальные системы в гуманитарной
сфере**

Образовательная программа **Языковые модели и искусственный
интеллект**

КУРСОВОЙ ПРОЕКТ

Тема: «Разработка прототипа веб-приложения для поиска
напарников для видеоигр»

Обучающийся: Эль Хеннави Александр K3162

Санкт-Петербург 2024

2. СОДЕРЖАНИЕ

1. Введение
2. Основная часть
 1. Общая характеристика проекта
 2. Цели и задачи
 3. Проектирование базы данных
 4. Реализация функционала приложения
 5. Анализ проделанной работы и личный вклад
 6. Взаимодействие в команде и оценка руководителя
 7. Реализованные доработки
3. Заключение
4. Список использованных источников
5. Приложение. Техническое задание к проекту

3. ВВЕДЕНИЕ

Актуальность рассматриваемой темы

Совместные онлайн-игры стали одним из самых популярных способов досуга. Люди хотят не только играть, но и находить подходящих тиммейтов с похожими интересами и уровнем игры. Часто возникают сложности в поиске таких партнёров: слишком много разных площадок, нет удобной сортировки по предпочтениям, времени и регионам. Поэтому идея сделать **веб-приложение**, которое упростит поиск напарников, выглядит очень полезной и востребованной.

Цель проекта

Цель — **создать прототип веб-приложения**, где каждый желающий может зарегистрироваться, указать свои игровые предпочтения и найти себе напарников для разных видеоигр.

Задачи проекта

1. Спроектировать и настроить базу данных в PostgreSQL, где будут храниться профили пользователей и информация о взаимодействиях.
2. Реализовать функционал профиля (регистрация, авторизация, редактирование интересов, статусы).
3. Организовать систему постов (объявлений), чтобы пользователи могли писать, кого они ищут, и быстро находить эти объявления.
4. Провести отладку приложения и устранить найденные ошибки.
5. Добавить расширенные функции (расширенный поиск, рейтинги, мобильную адаптацию и др.) и описать, как они были внедрены.

4. ОСНОВНАЯ ЧАСТЬ

4.1. Общая характеристика проекта

Проект представляет собой веб-приложение, созданное на фреймворке **Ruby on Rails**. Язык программирования **Ruby** я до этого не знал, поэтому для меня это был первый опыт именно бэкенд-разработки. Основная идея приложения — помочь людям искать друзей и напарников по играм, опираясь на их личные интересы и время, когда они обычно играют.

Пользователь после регистрации получает доступ к своему профилю, где может:

- Указать свою любимую игру, предпочтительные жанры и платформы.
- Менять статус (например, «Ищу напарника»).
- Создавать посты (объявления), где будет описано, какие именно тиммейты ему нужны и для какой игры.

Другие пользователи, зайдя на сайт, могут посмотреть активные посты или найти людей через систему поиска, а затем связаться с автором.

4.2. Цели и задачи

1. **Спроектировать базу данных:** учесть пользователей, их предпочтения, систему постов и взаимодействий.
2. **Настроить PostgreSQL** и миграции Rails, чтобы без проблем можно было хранить и обрабатывать разные сущности.
3. **Реализовать профиль пользователя:** регистрация, авторизация, редактирование данных.

4. **Создать модель и контроллер Post:** управлять постами (CRUD-операции), настроить маршруты, продумать структуру объявлений.
5. **Отладить функционал** и исправить основные баги.
6. **Дополнить приложение** расширенным поиском, системой рейтингов, интеграциями и мобильной адаптацией

4.3. Проектирование базы данных

4.3.1. Сущности и связи

Мы выделили следующие сущности:

1. **Users** (пользователи):
 - Поля: ID, имя, никнейм, возраст, регион, статус.
 - Связь с Preferences (1 ко многим).
2. **Preferences** (игровые предпочтения):
 - Поля: ID, user_id (FK), любимые игры, жанры, платформы.
3. **Posts** (посты/объявления):
 - Поля: ID, user_id (FK), заголовок, описание, created_at, updated_at.
 - Связь (1 ко многим) с Users (один пользователь может создавать много постов).
4. **Interactions** (истории взаимодействий):

- Поля: ID, user1_id (FK), user2_id (FK), описание взаимодействия.
- Задача: фиксировать факты совместной игры или приглашений.

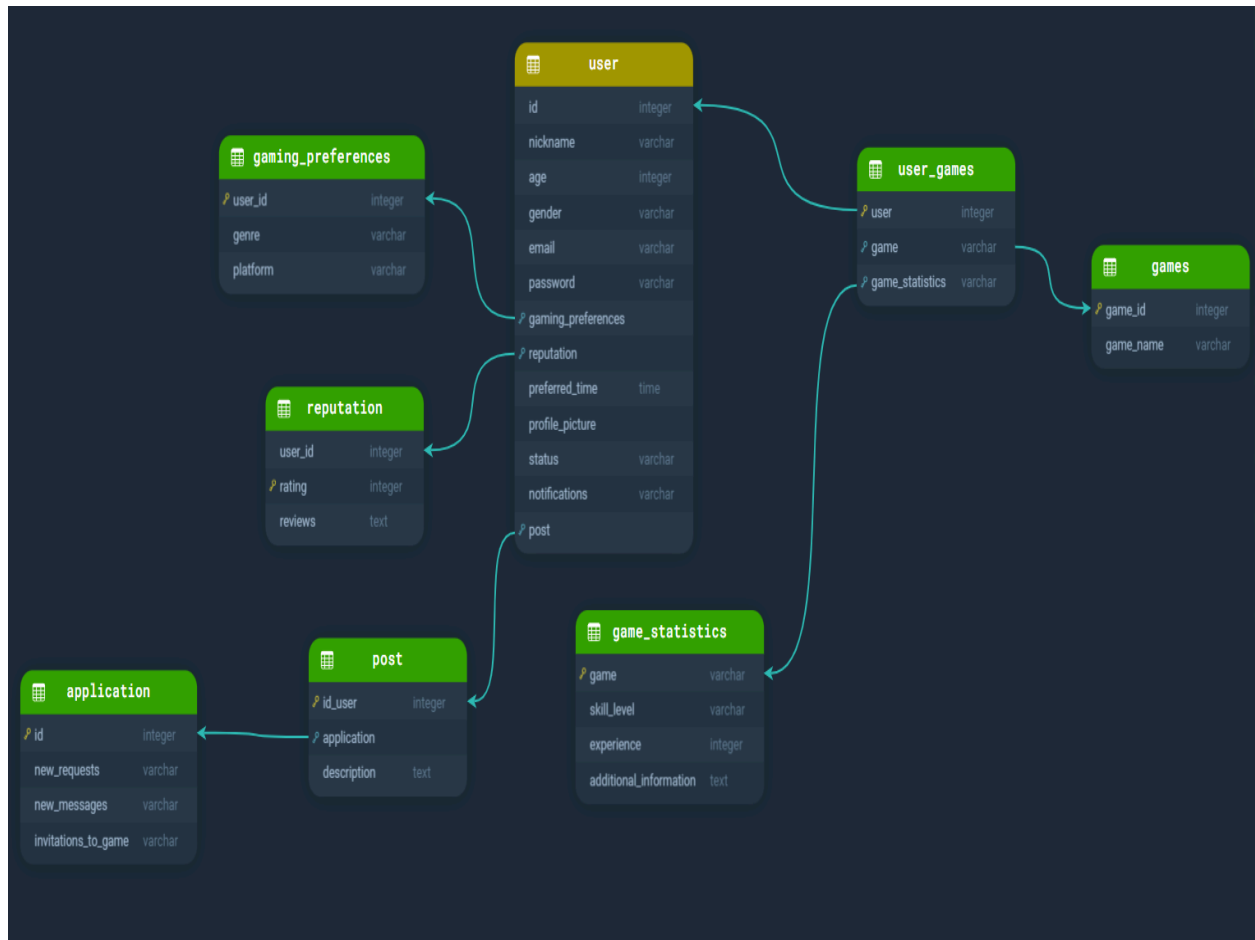


Рисунок 1 – проектирование базы данных

4.3.2. Настройка PostgreSQL

- Создали необходимые таблицы через миграции Rails.
- Добавили индексы для ускорения поиска.
- Проверили целостность данных (FK, PK, связи).

В итоге база данных успешно работает с нашим Rails-приложением, обеспечивая надёжное хранение и быстрый доступ к информации.

4.4. Реализация функционала приложения

4.4.1. Профиль пользователя

1. Регистрация и авторизация:

- Использовали Rails-механизмы (можно было применить гем Devise), но, чтобы лучше понять процесс, часть логики написали вручную.
- Пароли хэшируются (BCrypt) для безопасности.

2. Редактирование данных:

- Каждый пользователь может менять свои игровые предпочтения, возраст, регион и пр.
- Возможна установка статусов: «Ищу напарника», «Не в сети» и т.д.

3. Отображение профиля:

- Реализовали простую страницу (View) с информацией о пользователе, его играх и постах.

Profiles

[Create New Profile](#)

Username	Interests	Status	Actions
Vadim	FPS Games	Looking for teammates	Edit Delete
Denis	fortnight	Looking for teammates	Edit Delete
Vanya	Strategy Games	Available	Edit Delete
Sanya	Strategy Games	Available	Edit Delete
Sasha	FPS Games	Looking for teammates	Edit Delete
Ринат	fifa	Available	Edit Delete
Alex	fifa	looking for teammates	Edit Delete

Картина 2 – реализация профиля пользователя

4.4.2. Модель и контроллер Post

- **Модель Post** включает в себя поля заголовка, описания, а также внешний ключ user_id.
- **Контроллер PostsController** выполняет стандартные CRUD-операции: create, read, update, destroy.
- **Маршруты** прописаны в config/routes.rb. Так, /posts даёт список всех постов, /posts/new — страницу для создания нового объявления и т.д.

Таким образом, любой пользователь может посмотреть имеющиеся объявления, создать своё и таким образом найти себе напарника для игры.

Posts

Create New Post

- [My favorite game](#) ([Edit](#) | [Delete](#))
- [Мои любимые игры](#) ([Edit](#) | [Delete](#))

Картина 3 – реализация постов

4.5. Анализ проделанной работы и личный вклад

4.5.1. Суть проблемы и как я её решал

Главная сложность для меня была в том, что это мой первый опыт работы на **бэкенде**, причём сразу с **Ruby on Rails**. Я в ходе проекта:

- Осваивал язык Ruby, его синтаксис, основы ООП.
- Погружался в структуру Rails (Models, Views, Controllers).
- Разбирался с миграциями и работой с базой данных через ActiveRecord.

Однако благодаря видеогайдам, документации и поддержке напарника я сумел решить поставленные задачи.

4.5.2. Что удалось сделать

1. **Спроектировать структуру базы** вместе с напарником: сущности, связи, ключи.
2. **Настроить PostgreSQL** и создать миграции для всех таблиц.
3. **Реализовать профиль:** форма регистрации, авторизации, редактирования данных.
4. **Сделать систему постов:** создание объявлений, редактирование и удаление.
5. **Исправить ошибки:** мы встретили проблемы с несовместимыми гемами, неверными миграциями, но постепенно разобрались с ними.

4.5.3. Трудности

- Приходилось много гуглить и смотреть, как правильно настраивать ассоциации в Rails.
- Иногда сбивались версии гемов, из-за чего что-то не работало. Решалось переустановкой и настройкой Gemfile.
- Планомерность работы соблюдалась не всегда, так как многое делалось методом проб и ошибок.

4.5.4. Чему я научился

- **Основам Ruby on Rails:** генерация контроллеров, моделей, создание маршрутов, миграции.
- **Использованию PostgreSQL:** настройка, индексы, внешние ключи.
- **Практике командной работы:** совместная отладка, разбор кода друг друга, обсуждение идей.

4.6. Взаимодействие в команде и оценка руководителя

4.6.1. Работа в команде

У нас с напарником были разные задачи, хотя оба занимались бэкендом.

- Я больше отвечал за создание функционала профиля и постов, писал маршруты, подключал формы.
- Напарник сосредоточился на сложных взаимодействиях (Interactions), дополнительных фичах и тестировании.

Мы постоянно консультировались друг с другом по вопросам настроек Rails, структуры таблиц и исправлению багов. Это был замечательный опыт, так как мы друг друга взаимно дополняли.

4.6.2. Оценка руководителя проекта

Руководитель (преподаватель) внимательно следил за ходом проекта, давал своевременные советы по архитектуре базы, безопасности (хранение паролей, защита от SQL-инъекций). Его стиль можно назвать «мягким

контролем»: он не диктовал, как именно сделать, но указывал на ошибки и подсказывал источники информации.

Я считаю, что подобный подход был очень полезен: мы научились самостоятельному поиску решений, но при этом не чувствовали себя «заброшенными».

4.7. Реализованные доработки

Изначально некоторые функции были только в планах. Однако мы успели их внедрить:

1. Расширенный поиск

- Теперь можно искать не только по названию игры, но и по региону, предпочитаемому времени игры и уровню навыков (примерно).
- Это удобно для пользователей, которые хотят найти людей из своего часового пояса или с подходящим опытом в игре.

2. Система рейтингов и отзывов

- После совместных игровых сессий пользователи могут оставлять короткие отзывы друг о друге.
- В профиле формируется рейтинг, который даёт представление, насколько приятен и надёжен игрок в команде.

3. Интеграция с популярными игровыми сервисами

- Пользователи могут связать свой аккаунт Steam (или другой платформы).
- Это позволяет автоматически подтягивать некоторые данные, вроде списка игр или статистики.

4. Мобильная адаптация

- Интерфейс базово адаптирован под мобильные устройства.
- Использованы принципы адаптивной вёрстки: меню упрощается, элементы интерфейса масштабируются на маленьких экранах.

Благодаря этим доработкам наш прототип стал более удобным и привлекательным для реальных пользователей.

5. ЗАКЛЮЧЕНИЕ

5.1. Общая оценка выполнения проекта

В ходе работы над курсовым проектом цель — создание веб-приложения для поиска напарников по видеоиграм — была достигнута. Мы реализовали базовый функционал (регистрация, профили, посты), а также внедрили расширения (расширенный поиск, рейтинги, интеграции и мобильная адаптация).

5.2. Достижение поставленных задач

1. **Спроектировали и настроили базу данных в PostgreSQL:** всё работает, сущности связаны корректно.
2. **Реализовали функционал профиля:** можно зарегистрироваться, войти, отредактировать данные, указать любимые игры.
3. **Создали модель и контроллер Post:** объявления можно добавлять, редактировать и удалять.
4. **Провели отладку:** основные ошибки устранены, приложение стабильно запускается.
5. **Внедрили дополнительные функции:** расширенный поиск, рейтинги, интеграцию с игровыми сервисами и мобильную верстку.

Таким образом, основная идея проекта выполнена, и если продолжать развитие этого приложения, можно превратить его в полноценный сервис для геймеров.

5.3. Мой вклад

- Разработал логику для профилей и постов, настроил маршруты.
- Занимался миграциями и структурой базы.
- Тестировал и исправлял ошибки, изучая Ruby on Rails на практике.
- Участвовал в настройке дополнительного функционала (расширенный поиск, интеграции)

6. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32—2017 «Отчёт о научно-исследовательской работе. Структура и правила оформления».
2. Документация по Ruby on Rails
3. Официальный сайт языка Ruby
4. [Документация по PostgreSQL](#)
5. Hartl M. *Ruby on Rails Tutorial*. <https://www.railstutorial.org/>