



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

ОТЧЕТ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ

по дисциплине

«Технология разработки программных приложений»

Выполнил:

Студент группы ИКБО-11-22

Гришин А. В.

Проверил:

Преподаватель

Копанева А. А.

Москва 2024 г.

1. Тема проекта

WEB-разработка. Разработка бэклога игр.

2. Состав команды

- Гришин Андрей Валерьевич
- Голованёв Никита Алексеевич
- Андрусенко Лада Дмитриевна

3. Формулировка требований к проекту

Заголовок		Бэклог игр
Заказчик (actor)	Как	Пользователь
Примечание	Я хочу	Иметь приложение для оценки игр.
Цель	Чтобы	Отмечать пройденные игры, желаемые игры. Ставить оценки пройденным играм.

4. Интерфейс продукта

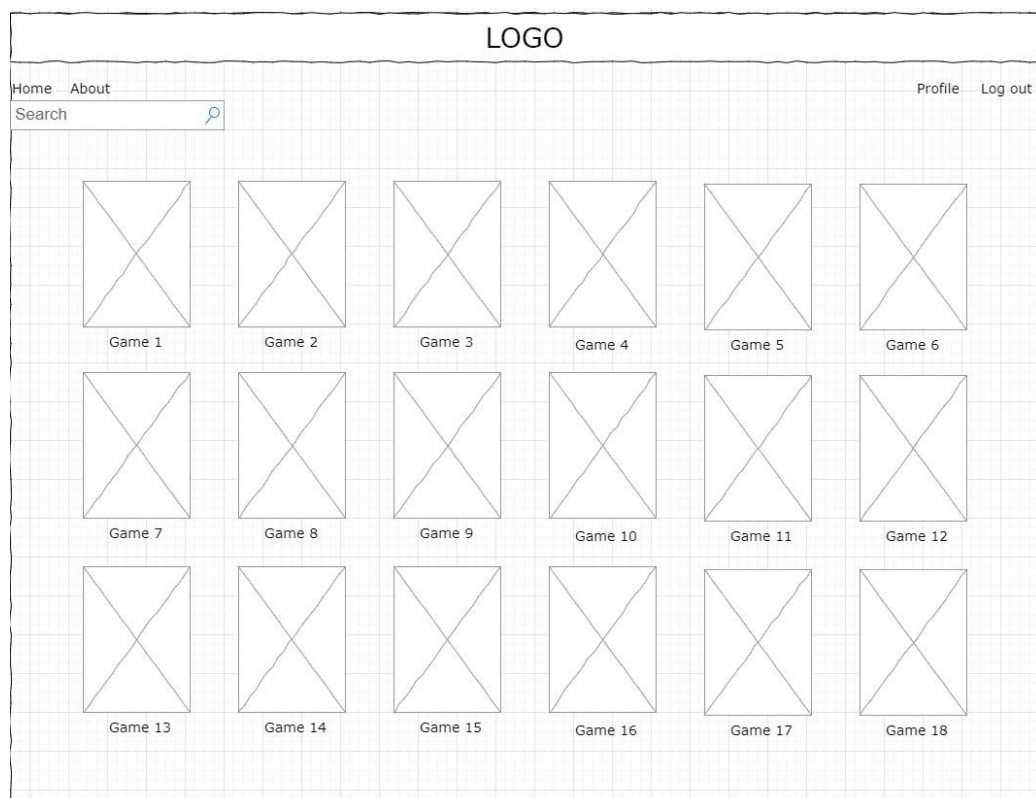


Рисунок 1 – Макет главного экрана

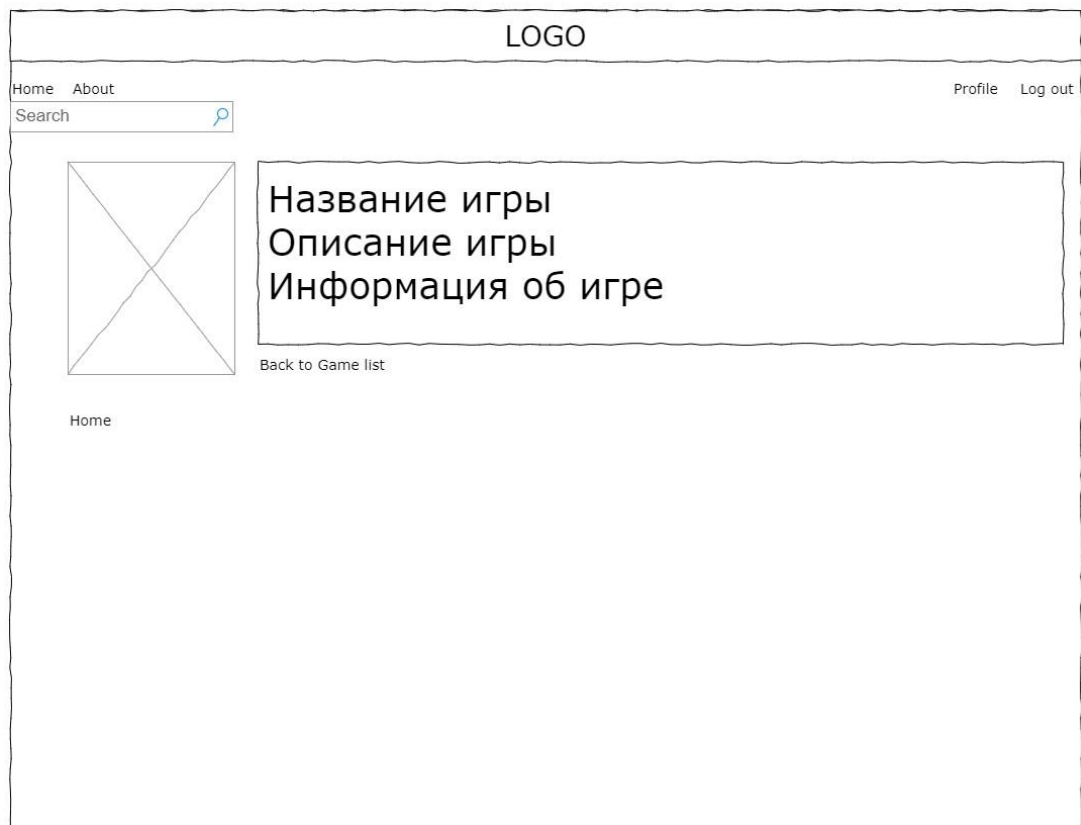


Рисунок 2 – Макет страницы игры

5. Репозиторий продукта

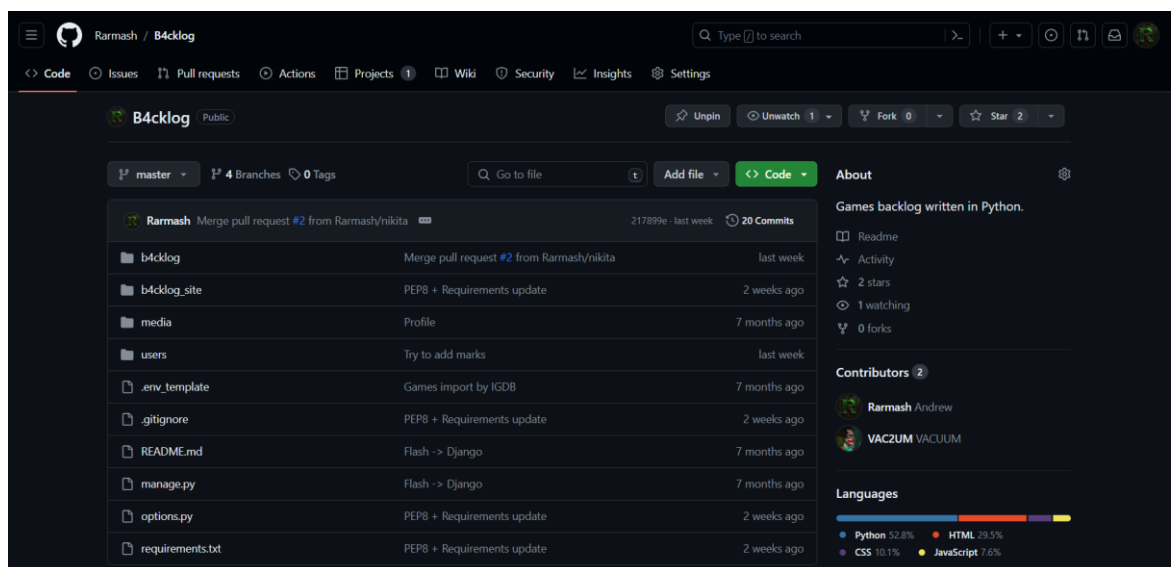


Рисунок 3 – Репозиторий продукта

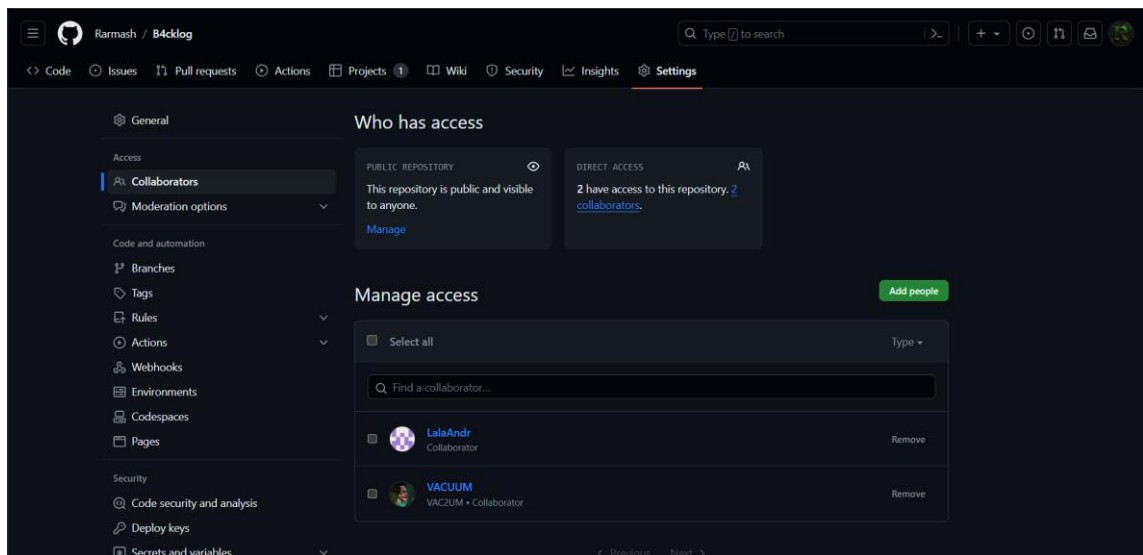


Рисунок 4 – Коллабораторы проекта

6. Программный код проекта

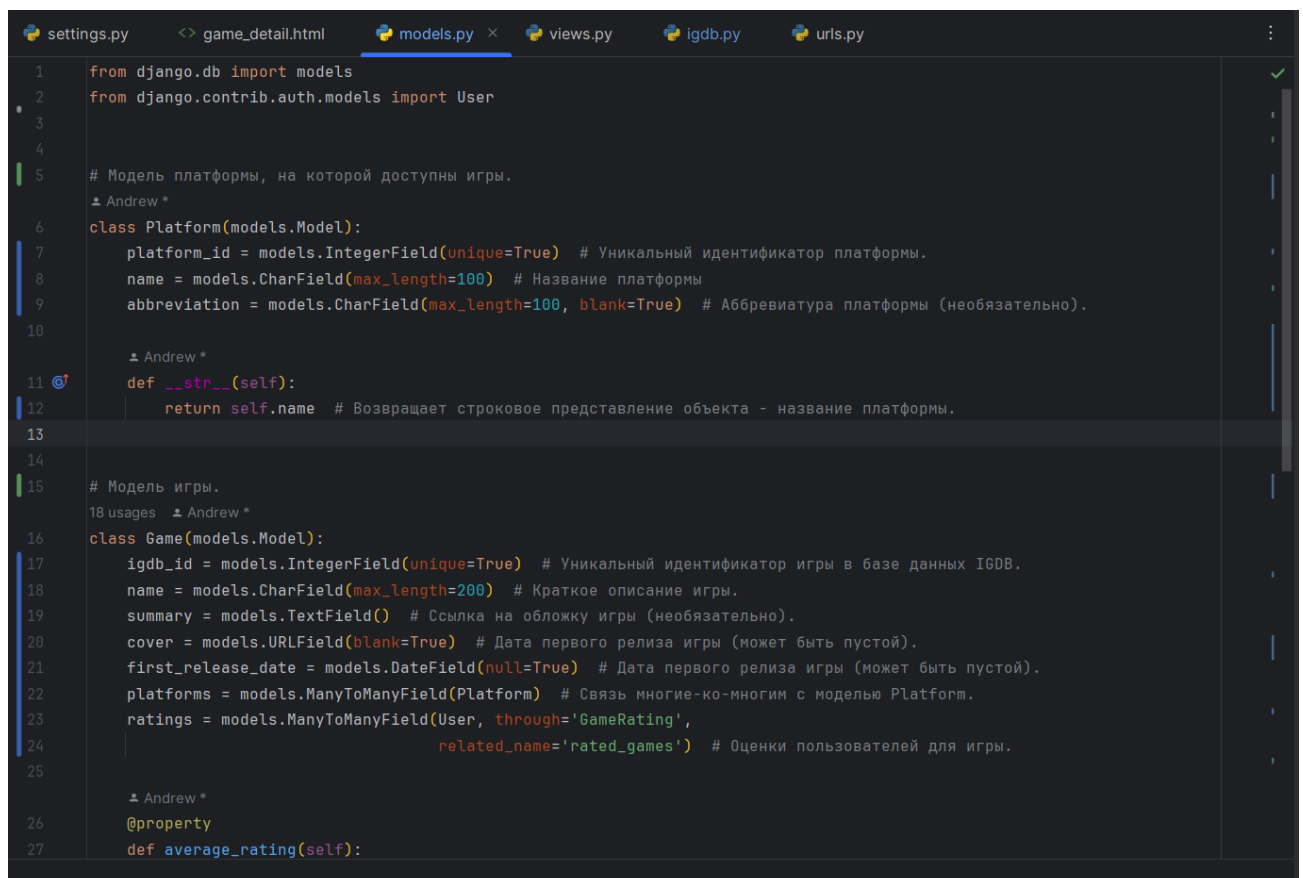


Рисунок 5 – реализация моделей Game и Platform

```
settings.py  <> game_detail.html  b4cklog\models.py  users\models.py  views.py  igdb.py  urls.py

1  from django.db import models # Импорт модуля models из библиотеки Django для работы с базой данных.
2  from django.contrib.auth.models import User # Импорт модели пользователя Django для аутентификации.
3  from b4cklog.models import Game # Импорт модели игры из приложения b4cklog.
4  from PIL import Image # Импорт модуля Image из библиотеки Pillow для работы с изображениями.
5
6
7  # Модель профиля пользователя.
8  10 usages  Andrew *
9  class Profile(models.Model):
10     user = models.OneToOneField(User, on_delete=models.CASCADE) # Связь "один к одному" с моделью User.
11     image = models.ImageField(default='default.jpg', upload_to='profile_pics') # Изображение профиля пользователя.
12     # Игры, которые пользователь хочет поиграть.
13     backlog_want_to_play = models.ManyToManyField(Game, related_name='backlog_want_to_play', blank=True)
14     # Игры, которые пользователь в настоящее время играет.
15     backlog_playing = models.ManyToManyField(Game, related_name='backlog_playing', blank=True)
16     # Игры, которые пользователь уже играл.
17     backlog_played = models.ManyToManyField(Game, related_name='backlog_played', blank=True)
18     # Игры, которые пользователь завершил.
19     backlog_completed = models.ManyToManyField(Game, related_name='backlog_completed', blank=True)
20     # Игры, которые пользователь завершил на 100%.
21     backlog_completed_100 = models.ManyToManyField(Game, related_name='backlog_completed_100', blank=True)
22
23     Andrew *
24     def __str__(self):
25         return f'{self.user.username} Profile' # Возвращает строковое представление объекта - профиль пользователя.
26
27     Andrew *
28     def save(self, *args, **kwargs):
29         super().save(*args, **kwargs) # Сохранение профиля пользователя.
30
31         img = Image.open(self.image.path) # Открытие изображения профиля.
```

Рисунок 6 — реализация модели Profile