МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ   
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (МАИ)

Кафедра №806 «Вычислительная математика и программирование»

**ОТЧЕТ ПО IT-ПРОЕКТУ**

**«Ассистент по досугу»**

Студент Гузова Виктория

Студент Никитина Анастасия

Студент Павлова Юлия

Группа М8О-107БВ-24

Руководитель Самир Ахмед

Москва, 2025

**Введение**

**Актуальность задачи**

В современных городских условиях жители регулярно сталкиваются с проблемой организации досуга. Несмотря на обилие культурных и развлекательных мест, процесс выбора подходящего варианта остается сложным и трудоемким. Пользователи вынуждены тратить значительное время на поиск информации в разрозненных источниках: социальных сетях, специализированных сайтах и через личные рекомендации. Эти подходы имеют существенные недостатки, так как информация часто бывает неполной или устаревшей, рекомендации не учитывают индивидуальные предпочтения, отсутствует единая система оценки мест на основе реального опыта посетителей.

Разработка централизованной платформы для поиска и рекомендации мест отдыха позволяет решить эти проблемы, предоставляя пользователям персонализированный и эффективный инструмент для планирования досуга.

**Цель проекта**

Создание веб-приложения "Ассистент по досугу", которое предоставляет пользователям:

- Удобный каталог мест отдыха с актуальной информацией

- Персонализированные рекомендации на основе предпочтений

- Инструменты для сохранения и оценки посещенных мест

- Возможность делиться опытом через комментарии и рейтинги

**Задачи проекта**

Для достижения поставленной цели решаются следующие задачи:

1. Разработка системы добавления и каталогизации мест досуга

2. Реализация функционала оценки и комментирования посещенных мест

3. Создание персонального раздела для учета истории посещений

4. Внедрение фильтрации по категориям и сортировки по рейтингу

5. Разработка интуитивного пользовательского интерфейса

6. Обеспечение постраничного вывода информации

7. Предоставление возможности пользователю получить краткую сводку о положительных и отрицательных комментариях о месте

**Обзор структуры отчета**

Отчет систематизирует процесс разработки решения и состоит из следующих разделов:

- Описание задачи: детализация функциональных требований

- Архитектура системы: обзор компонентов и технологий

- Реализация: ключевые аспекты программной реализации

- Технические решения: значимые инженерные подходы

- Проблемы и решения: преодоление сложностей разработки

- Перспективы: направления дальнейшего развития

Последовательное изложение позволяет проследить эволюцию проекта от концепции до рабочего прототипа.

# Описание задачи

**Постановка задачи**

Система "Ассистент по досугу" предназначена для решения проблемы эффективного поиска мест отдыха. Приложение должно:

1. Предоставлять централизованный каталог мест досуга

2. Позволять пользователям добавлять новые места с описанием

3. Обеспечивать механизм оценки и комментирования посещенных мест

4. Формировать персонализированную историю посещений

5. Предлагать фильтрацию по категориям и сортировку по рейтингу

*Входные данные:*

- Запросы пользователей через веб-интерфейс

- Данные о местах (название, описание, категория, изображение)

- Пользовательские оценки и комментарии

*Выходные данные:*

- Структурированный список мест с фильтрацией

- Детальная информация о конкретном месте

- Персональная история посещений

- Рейтинги на основе пользовательских оценок

**Функциональные характеристики**

Система реализует следующие ключевые функции:

Система реализует комплекс функций, разделенных на базовый и расширенный функционал.

1. Добавление новых мест: Пользователи могут создавать карточки мест с обязательными полями: название, описание, категория и изображение.
2. Постраничный вывод мест: Система отображает места порциями по 5 объектов на странице с возможностью сортировки по рейтингу.
3. История посещений: В персональном разделе пользователя автоматически сохраняются все посещенные места.
4. Детализация информации о месте: Для каждой локации система предоставляет полную информацию, включающую полное описание, фотографию, средний рейтинг.
5. Комментирование посещенных мест: Пользователи могут оставлять текстовые отзывы о посещенных местах, что помогает другим пользователям принимать информированные решения.
6. Категоризация мест: Система поддерживает 11 фиксированных категорий (“Прогулка”, “Театр”, “Музей”, “Ресторан”, “Кафе”, “Парк аттракционов”, “Бассейн”, “Детский досуг”, “Коворкинг”, “Кино”, “Развлекательные центры”) с возможностью фильтрации по ним.
7. Система оценок: Пользователи выставляют рейтинг от 1 до 10 баллов, на основе которых система автоматически рассчитывает среднюю оценку для каждого места.
8. Система генерирует краткие сводки о положительных и отрицательных аспектах места на основе пользовательских комментариев. Для реализации используется интеграция с YandexGPT API, которая анализирует отзывы и выделяет ключевые темы.

Технические характеристики

Типы документов и данных:

- Пользовательские профили (учетные данные, история действий)

- Карточки мест (текстовые описания, изображения, категории)

- Оценочные данные (рейтинги, комментарии)

**Архитектура системы**

**Общая схема**

Система включает четыре ключевых этапа обработки данных, обеспечивающих полный цикл работы с пользовательскими запросами:

1. Прием и валидация запросов: Веб-интерфейс принимает пользовательские запросы через формы и кнопки, валидирует входные данные

2. Обработка запроса: Сервер ищет нужные данные и выполняет необходимые расчеты

3. Работа с данными: Взаимодействие с базой данных для извлечения и сохранения информации

4. Формирование ответа: Подготовка структурированных данных для отображения в интерфейсе

**Описание модулей**

Модуль интерфейса (Frontend):

- HTML/CSS: Внешний вид страниц и элементов

- JavaScript: Реакция на действия пользователя (клики, выбор фильтров)

- Fetch API: Отправка запросов на сервер и получение ответов

Модуль обработки (Backend):

- Django: Принимает запросы и управляет данными

Основные функции:

Поиск мест по категориям

Расчет средних оценок

- YandexGPT API: Анализирует отзывы о месте и выводит краткую информацию о положительных и отрицательных моментах

Модуль данных (Database):

- PostgreSQL: Хранит всю информацию (места, пользователей, оценки)

**Инструменты**

Backend:

- Django: Основной фреймворк для реализации серверной логики

- PostgreSQL 14: Реляционная СУБД для хранения структурированных данных

- Requests: Библиотека для выполнения внешних HTTP-запросов

- YandexGPT API: Суммаризация пользовательских отзывов и генерация аналитических сводок

Frontend:

- HTML5: Базовая разметка интерфейса

- CSS3: Стилизация элементов

- JavaScript: Клиентская логика без сторонних фреймворков

- Fetch API: Технология для асинхронного взаимодействия с сервером

Данный набор инструментов обеспечивает гибкость и масштабируемость системы, позволяя адаптировать ее под различные сценарии использования и будущие расширения функционала.

**База данных**

Для хранения данных разработана реляционная структура в PostgreSQL 14:

Основные сущности:

- Категории (PlacesCategory): id, name

- Пользователи (Users): id, username, first\_name, last\_name, password\_hash.

- Места (Places): id, name, image, description, short\_description, category\_id

- Посещения (VisitedPlaces): id, mark, feedback, visited\_at, place\_id, users\_id

- Оценки(Review): id, mark, comment, places\_id

Связи:

- Один-ко-многим между PlacesCategory и Places

- Один-ко-многим между Users и VisitedPlaces

- Один-ко-многим между Places и VisitedPlaces

- Один-ко-многим между Places и Review

Хранение данных реализовано на выделенном сервере PostgreSQL, что обеспечивает масштабируемость для обработки тысяч записей, отказоустойчивость через механизм репликации, высокую производительность при интенсивной нагрузке

Такая архитектура гарантирует стабильную работу системы даже при одновременном использовании сотнями пользователей в таких сценариях, как массовые культурные мероприятия или туристические сезоны.

**Заключение**

**Основные результаты**

Разработанная система решает актуальную проблему выбора мест для досуга, предоставляя пользователям централизованный каталог с объективными оценками. Техническая реализация на Django и PostgreSQL обеспечивает стабильную работу и масштабируемость. Дальнейшее развитие платформы позволит создать полноценную экосистему для планирования отдыха, учитывающую индивидуальные предпочтения каждого пользователя.

**Перспективы развития**

Система имеет значительный потенциал для расширения функциональности:

1. Рекомендательная система

- Внедрение алгоритмов машинного обучения для персонализированных рекомендаций, а также для более естественного общения с чатом

- Учет истории посещений и оценок пользователя для корректного подбора мест

2. Интеграция с геосервисами

- Показ ближайших мест отдыха

3. Отметки “Хочу сходить”

- Возможность сохранения понравившихся мест в личный список планов