Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

«Балтийский федеральный университет имени Иммануила Канта»

Кафедра цифровых технологий

**Итоговая работа**

на тему: «Разработка гео-сервиса по информированию о мероприятиях с использованием нейронных сетей»

Выполнил: Мусин Вадим Маратович

Руководитель: доцент ОНК «Институт высоких технологий»

Ткаченко Сергей Николаевич

Калининград

2025

**Оглавление**

Введение …………………………………………………… стр.2

Глава 1. Анализ гео-сервисов и технологий рекомендаций ………… стр.3

1.1. Цифровые геосервисы для информирования о мероприятиях ……… стр.3

1.2. Методы визуализации событий на карте ……………………………… стр.3

1.3. Структура платформы событий и фильтрация данных ………… стр.4

1.4. Потенциал использования нейросетей для рекомендаций мероприятий … стр.4

Глава 2. Разработка проекта «MeowMap» …………………………… стр.5

2.1. Архитектура и компоненты геосервиса ……………………………… стр.5

2.2. Инструменты и технологии разработки …………………………… стр.6

2.3. Парсинг данных о мероприятиях из ВКонтакте …………………… стр.7

2.4. Автообновление данных через GitHub Actions …………………… стр.7

2.5. Интерактивная карта на Leaflet и JSON API …………………… стр.8

2.6. Идеи по интеграции AI-доработок ………………………………… стр.8

Заключение …………………………………………………… стр.10

Список источников ……………………………………………… стр.11

Приложения …………………………………………………… стр.13

**Введение**

В современную эпоху цифровых сервисов жители городов испытывают потребность оперативно получать информацию о предстоящих концертах, выставках, фестивалях и других культурно-развлекательных мероприятиях. Данные об этих событиях зачастую разрознены и публикуются на различных интернетпорталах и в социальных сетях, что затрудняет планирование досуга и приводит к упущенным возможностям. Актуальной задачей становится создание геоинформационных сервисов, которые агрегируют информацию о событиях и предоставляют её пользователю в удобной форме.

Цель настоящей выпускной квалификационной работы – разработать интеллектуальный геосервис «MeowAfisha» для города Калининграда. Этот сервис должен объединять карту города с метками событий и механизмы их отбора в соответствии с интересами пользователя. Для достижения цели необходимо:

* проанализировать существующие платформыафиши (Afisha7, KudaGo, Яндекс.Афиша и др.) и способы визуализации данных о событиях;
* выбрать оптимальные технологии и API для реализации сервиса;
* спроектировать архитектуру системы, включающую модуль автоматизированного сбора данных, хранилище и пользовательский интерфейс;
* реализовать прототип вебприложения с использованием JavaScript и библиотеки OpenLayers для отображения карты;
* исследовать возможности применения алгоритмов машинного обучения для персонализированных рекомендаций мероприятий.

Научная новизна работы заключается в интеграции методов искусственного интеллекта (рекомендательных моделей) в локальный городской афишный сервис. Практическая значимость проекта состоит в том, что созданный прототип может быть использован жителями Калининграда для планирования отдыха, а заложенные архитектурные и алгоритмические решения легко масштабируются на другие регионы.

**Глава 1. Анализ геосервисов и технологий рекомендаций**

**1.1. Цифровые геосервисы для информирования о мероприятиях**

В современном информационном пространстве существует множество онлайнплатформ, предоставляющих сведения о предстоящих событиях. Такие агрегаторы объединяют данные о концертах, выставках, фестивалях и встречах на единой платформе и, как правило, используют геолокацию пользователя для показа ближайших мероприятий.

Сервисы типа Afisha7, KudaGo и Яндекс.Афиша собирают информацию из различных источников — от официального портала PRO.Культура.рф до API социальных сетей. Например, сервис «Афиша7» интегрирует данные нескольких API и предоставляет инструменты для добавления событий организаторами. Платформа KudaGo формирует открытую базу, ежемесячно пополняемую тысячами событий, и предоставляет бесплатный API для разработчиков.

Социальные сети — ещё один богатый источник данных. ВКонтакте и Facebook позволяют получать анонсы мероприятий через официальный API. Метод *wall.get* API VK возвращает список постов сообщества и применяется для автоматизированного извлечения информации о событиях. Анализ существующих платформ показывает тенденцию к консолидации разрозненных данных о мероприятиях и созданию удобных интерфейсов для поиска по месту, времени и тематике.

**1.2. Методы визуализации событий на карте**

Ключевой элемент геосервисов — наглядное отображение событий на электронной карте. Пользователю важно быстро понять, какие мероприятия проходят рядом, поэтому интерактивные карты используются практически во всех современных афишах. Библиотека OpenLayers является популярным opensource инструментом для создания интерактивных карт. Она позволяет отображать плиточные слои, векторные данные и маркеры из различных источников и отличается легковесностью и гибкостью. В качестве подложек часто применяются карты OpenStreetMap, предоставляющие свободные и актуальные геоданные.

При визуализации событий применяются приёмы кластеризации маркеров, всплывающие окна с подробной информацией, фильтры по дате и категории. Выбор библиотеки OpenLayers обусловлен её расширяемостью и совместимостью с различными источниками данных.

**1.3. Структура платформы событий и фильтрация данных**

Типичная архитектура сервиса афиши включает несколько уровней: модуль сбора данных, базу событий, API/JSONинтерфейс для доступа к данным и фронтенд для отображения информации. Модуль сбора регулярно получает посты со стен сообществ ВКонтакте через метод *wall.get*, извлекает из них дату, название и место проведения события, геокодирует адреса и сохраняет их в структурированном виде (JSON).

Слой хранения представляет собой файл events.json, содержащий упорядоченный список мероприятий. Для фильтрации данных по дате или тематике фронтенд загружает весь JSON и выбирает записи, соответствующие выбранным пользователем критериям. При необходимости можно расширить архитектуру полноценным серверным API, предоставляющим выборку событий по параметрам.

**1.4. Потенциал использования нейросетей для рекомендаций**

Рекомендательные системы помогают пользователям ориентироваться в большом объёме информации. В области событийных сервисов существуют специфические трудности: событие ещё не произошло, поэтому отсутствуют отзывы и оценки (проблема нового объекта). Исследователи отмечают, что для повышения качества рекомендаций необходимо учитывать географическую дистанцию, время до начала события и предпочтения пользователя.

Для сервиса MeowAfisha перспективно внедрять гибридные модели, комбинирующие контентную фильтрацию, анализ RSVPданных и социальные связи. Например, API ВКонтакте позволяет получать списки участников мероприятий, что может служить дополнительным фактором при формировании персонализированной подборки. Другим направлением является использование нейросетевых моделей обработки естественного языка для автоматической классификации анонсов и выявления скрытых тем.

**Глава 2. Разработка проекта «MeowAfisha»**

**2.1. Архитектура и компоненты геосервиса**

Прототип MeowAfisha реализует модульную архитектуру и состоит из следующих компонентов:

* **Интерфейс**: одностраничное вебприложение, созданное на HTML и JavaScript, с картой и панелью выбора даты. Карта реализована с использованием OpenLayers, позволяющей загружать тайлы OpenStreetMap и отображать маркеры событий.
* **Данные событий**: файл events.json в формате JSON, содержащий актуальный список мероприятий с полями «title», «date» (YYYYMMDD), «lat», «lon», «location».
* **Скрипт парсинга**: Pythonскрипт scrape\_meow.py, который обращается к VK API, выгружает посты сообщества «Meow Афиша | Калининград», извлекает дату и место из текста, геокодирует адреса с помощью ArcGIS Geocoding API через библиотеку geopy и формирует/ обновляет events.json.
* **Автообновление**: рабочий процесс GitHub Actions (update.yml), который запускает парсер каждые 6 часов, чтобы автоматически собирать свежие анонсы и коммитить обновления в репозиторий. Развёртывание статического сайта осуществляется на GitHub Pages, что обеспечивает бесплатный хостинг и простоту эксплуатации.

Отказ от традиционного бэкенда позволяет уменьшить стоимость поддержки и упростить инфраструктуру. Для масштаба города Калининграда (десятки событий в неделю) подход с использованием статических файлов и клиентского рендеринга достаточен.

**2.2. Инструменты и технологии разработки**

При реализации MeowMap использовались проверенные opensource инструменты, что обеспечило гибкость и минимальные затраты. Основные технологии:

* **Источник данных — VK:** сообщество «Meow Records» во ВКонтакте используется как основная площадка для анонсов. Доступ к постам осуществляется через метод *wall.get* API VK с параметрами domain, count, offset и access\_token, что позволяет получать структурированный JSON и избегать парсинга HTML.
* **Язык Python:** для парсинга выбран Python благодаря богатой экосистеме. Скрипт использует requests для HTTPзапросов, re для поиска шаблонов в тексте, pandas для структурирования данных и geopy для геокодирования адресов.
* **Геокодирование:** сервис ArcGIS Geocoding API предоставляет конвертацию адресов в координаты. В бесплатном режиме его достаточно для нескольких сотен запросов в день. Альтернативой являются Nominatim (OSM) или Яндекс.Геокодер, но ArcGIS выбран за простоту и точность.
* **Фронтенд:** JavaScript и OpenLayers. Карта и маркеры реализованы на клиенте без лишних зависимостей. Пользователь выбирает дату, и скрипт фильтрует список событий, отображая их на карте. Дизайн минималистичен: карта занимает большую часть экрана, а сверху расположена панель с выбором даты и кнопкой открытия списка ближайших мероприятий.
* **CI/CD:** GitHub Actions используется для автоматического обновления events.json. Workflow выполняет проверку изменений, запускает парсер и в случае обновления коммитит новые данные, тем самым поддерживая актуальность сервиса без участия администратора.

**2.3. Парсинг данных о мероприятиях из ВКонтакте**

Парсер scrape\_meow.py выполняет последовательные шаги:

* **Получение постов:** через API VK выгружается несколько тысяч последних записей стены сообщества, используя параметры count и offset. Между запросами делается пауза, чтобы не превысить лимиты API.
* **Извлечение структуры:** текст постов анализируется регулярными выражениями. Алгоритм ищет шаблон даты (ДД.ММ) и место (название площадки после спецсимвола). Если дата или место отсутствуют, пост пропускается. Найденные даты дополняются текущим годом и приводятся к формату YYYYMMDD.
* **Геокодирование:** найденные адреса передаются в ArcGIS через библиотеку geopy, которая возвращает координаты. При отсутствии координат событие удаляется.
* **Формирование JSON:** список событий сортируется по дате и сохраняется в файл events.json. Каждая запись содержит название, дату, широту, долготу и человекочитаемый адрес.

Такой подход обеспечивает независимость от изменений в верстке сайта ВКонтакте и сокращает время обработки.

**2.4. Автообновление данных через GitHub Actions**

Для поддержания актуальности данных используется GitHub Actions. Workflow в файле update.yml настроен на запуск каждые шесть часов. Он выполняет следующие шаги:

* checkout: клонирует репозиторий;
* setuppython: устанавливает зависимости;
* run: запускает парсер с передачей токена VK через переменные окружения;
* commit & push: в случае изменений коммитит обновлённый events.json обратно.

Этот механизм обеспечивает бесперебойное получение новых анонсов без вмешательства разработчика. Частоту обновлений можно адаптировать, например, увеличивать частоту в выходные дни, когда событий больше.

**2.5. Интерактивная карта на OpenLayers и JSON API**

Фронтенд сервиса реализован максимально просто. При открытии страницы загружаются карта OpenLayers и файл events.json. Затем выполняется:

* вычисление минимальной и максимальной даты из списка событий для ограничения элемента <input type="date">;
* выбор текущей даты или ближайшей даты с событиями;
* функция updateMap(selectedDate) очищает существующие маркеры, фильтрует список событий и добавляет на карту новые маркеры. Каждая точка снабжается всплывающим окном с названием, адресом и датой мероприятия;
* количество событий отображается рядом с выбором даты.

Использование OpenLayers обеспечивает поддержку всех основных браузеров и мобильных устройств. Интерфейс минималистичен: карта занимает основное пространство, что делает работу сервиса интуитивно понятной.

**2.6. Идеи по интеграции AI-доработок**

Хотя текущая версия MeowMap предоставляет базовый функционал, перспективы развития связаны с внедрением искусственного интеллекта. Возможные направления:

* **Персонализация:** создание рекомендательной системы, которая будет предлагать мероприятия, учитывая историю взаимодействия пользователя и его предпочтения. Возможно применение факторизационных моделей и нейронных сетей.
* **Автоматическая классификация:** использование моделей обработки естественного языка (NLP) для определения типа события по тексту анонса, фильтрации нерелевантных записей и модерации контента.
* **Анализ популярности:** на основе данных о взаимодействии пользователей (лайки, комментарии, количество участников) предсказывать посещаемость и ранжировать события по ожидаемой популярности.

Реализация этих идей повысит ценность сервиса и приблизит его к классу умных городских информационных систем.

**Заключение**

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель — разработан прототип геоинформационного сервиса для информирования жителей Калининграда о культурных мероприятиях. Были решены все ключевые задачи:

* проведён анализ существующих сервисов и агрегаторов событий, выявлены их достоинства и недостатки;
* изучены методы визуализации данных на карте, обоснован выбор библиотеки OpenLayers и свободных картографических данных OpenStreetMap для фронтенда;
* спроектирована структура платформы, включающая сбор данных из VK, хранение и автоматическую актуализацию;
* создан Pythonскрипт для парсинга постов сообщества ВКонтакте, извлечения информации о событиях и геокодирования адресов;
* реализован механизм непрерывной интеграции на GitHub Actions, обеспечивающий автоматическое обновление базы событий каждые 6 часов;
* построен минималистичный интерфейс с интерактивной картой и фильтром даты.

Полученный прототип демонстрирует жизнеспособность выбранных технологий и может быть использован как жителями, так и гостями города. Работа заложила основу для дальнейшего развития сервиса: расширения покрытия (добавление других городов и источников данных), внедрения пользовательских аккаунтов, интеграции систем персональных рекомендаций и создания мобильного приложения. Комбинация геоинформационных технологий с методами искусственного интеллекта обладает значительным потенциалом, позволяя сделать поиск мероприятий удобным и персонализированным. Проект MeowAfisha в перспективе может стать частью интеллектуальной городской инфраструктуры, помогая людям узнавать о интересных событиях буквально на кончиках пальцев.

**Список источников**

* Борщева А. Как вытянуть данные через официальный API ВК // Хабр, 2022. — URL: <https://habr.com/ru/articles/662858/> (дата обращения: 01.07.2025).
* AlekseyBr. «Афиша7» – интегрировать 4 API, чтобы разродиться собственным // Хабр, 13.01.2022. — URL: <https://habr.com/ru/articles/645429/> (дата обращения: 01.07.2025).
* VK Developers. Официальная документация VK API — URL: <https://dev.vk.com/method> (дата обращения: 28.07.2025).
* API KudaGo – открытая база событий и мест. Документация для разработчиков, 2017. — URL: <https://docs.kudago.com/api/> (дата обращения: 28.07.2025).
* Портал «PRO.Культура.рф» – единая база культурных мероприятий России. Документация API 2.4, Минкультуры РФ, 2019. — URL: <https://pro.culture.ru/api> (дата обращения: 01.07.2025).
* Esri. ArcGIS Geocoding Service – REST API Documentation, 2025. — URL: <https://developers.arcgis.com/rest/geocode/> (дата обращения: 20.07.2025).
* OpenLayers – JavaScript библиотека для интерактивных карт. Официальный сайт, 2025. — URL: <https://openlayers.org/> (дата обращения: 20.07.2025).
* OpenStreetMap – открытые географические данные мира. — URL: [https://www.openstreetmap.org](https://www.openstreetmap.org/) (дата обращения: 20.07.2025).
* Jain A. Answer to “How to list event details in a calendar card display and filter events” // StackExchange: User Experience, 2015. — URL: <https://ux.stackexchange.com/questions/87668/how-to-list-event-details-in-a-calendar-card-display-and-filter-events> (дата обращения: 01.07.2025).
* NVIDIA. What Is a Recommendation System? // AI Glossary, NVIDIA, 2024. — URL: <https://www.nvidia.com/en-us/glossary/recommendation-system/> (дата обращения: 01.07.2025).
* de Macedo A.Q., Marinho L.B. Event Recommendation in Eventbased Social Networks // Proc. of Social Personalization Workshop, CEURWS Vol. 1210, 2014. — С. 5–13.
* Kaggle. Event Recommendation Engine Challenge – Competition, 2013. — URL: <https://www.kaggle.com/c/event-recommendation-engine-challenge/> (дата обращения: 01.07.2025).
* Perumalla P. Event Recommendation Engine – project report, 2020. — URL: <https://pruthviperumalla.github.io/Event-Recommender-System/> (дата обращения: 01.07.2025).

**Приложения**

**Приложение А — Скриншоты интерфейса пользователя**

В приложении А приведены примеры интерфейса MeowAfisha: главный экран с картой Калининграда и отмеченными событиями; всплывающее окно с информацией о мероприятии.

 

**Приложение Б — Фрагменты программного кода**

Ниже приведён пример скрипта инициализации карты в файле index.html:

<div id="controls">

<label for="event-date">Дата:</label>

<input type="date" id="event-date">

<span id="count"></span>

</div>

<div id="map"></div>

<script src="https://cdn.jsdelivr.net/npm/ol@10.6.0/ol.js"></script>

<script>

const JSON\_URL = "events.json";

const map = new ol.Map({

target: "map",

layers: [ new ol.layer.Tile({ source: new ol.source.OSM() }) ],

view: new ol.View({ center: ol.proj.fromLonLat([20.45, 54.71]), zoom: 11 })

});

let markers = [];

function clearMarkers() { markers.forEach(m => map.removeLayer(m)); markers = []; }

function addEventMarker(ev) {

const marker = new ol.Feature({ geometry: new ol.geom.Point(ol.proj.fromLonLat([ev.lon, ev.lat])) });

marker.setStyle(new ol.style.Style({ image: new ol.style.Circle({ radius: 6, fill: new ol.style.Fill({ color: "#ff5722" }) }) }));

const vector = new ol.layer.Vector({ source: new ol.source.Vector({ features: [marker] }) });

marker.set("popup", `<b>${ev.title}</b><br>${ev.location}<br>${ev.date}`);

markers.push(vector);

map.addLayer(vector);

}

function updateMap(selectedDate) {

clearMarkers();

const todays = events.filter(e => e.date === selectedDate);

todays.forEach(addEventMarker);

document.getElementById("count").textContent = `– мероприятий: ${todays.length}`;

}

let events = [];

fetch(JSON\_URL).then(r => r.json()).then(data => {

events = data.sort((a,b) => a.date.localeCompare(b.date));

const dateInput = document.getElementById("event-date");

dateInput.min = events[0].date;

dateInput.max = events[events.length-1].date;

const today = new Date().toISOString().slice(0,10);

const firstDate = events.find(e => e.date >= today) ? today : events[0].date;

dateInput.value = firstDate;

updateMap(firstDate);

dateInput.addEventListener("change", e => updateMap(e.target.value));

});

</script>

Этот код инициирует карту OpenLayers, загружает тайлы OSM и отображает события из файла events.json. При выборе даты список маркеров обновляется, а каждый маркер имеет всплывающее окно с подробностями.