|  |  |
| --- | --- |
| RTK_LOGO.jpg | **Правительство Санкт-Петербурга**  **Комитет по образованию**  **Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение «Радиотехнический колледж»** |

|  |
| --- |
|  |

**КУРСОВАЯ РАБОТА**

Тема: Разработка веб-сайта прогноза погоды с помощью API OpenWeatherMap

по МДК 05.02 Разработка кода информационных систем

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность: 09.02.07 Информационные системы и программирование

|  |
| --- |
| Выполнил(а):  студент(ка) 2 курса  гр. № 201 Писукова Анастасия Вадимовна  (Ф.И.О.)  Проверил:  руководитель:  преподаватель  Кондратьев Виталий Сергеевич  (Ф.И.О.)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Курсовая работа защищена

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2021г.

**РЕЦЕНЗИЯ**

**на курсовую работу**

студента группы \_\_\_\_\_\_\_\_ специальности \_\_09.02.07 Информационные системы и программирование\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

Ф.И.О. студента

выполненную по дисциплине \_МДК 05.02 Разработка кода информационных систем \_

на тему: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.** **Актуальность темы:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Соответствие содержания курсовой работы заданию (да, нет)** \_\_\_\_\_\_\_\_;

**3. Наличие ссылок на первоисточники** \_\_\_\_\_\_\_\_\_\_\_\_;

**4. Количество использованных первоисточников** \_\_\_\_\_\_\_\_\_\_\_\_;

**5. Оформление иллюстраций, формул, таблиц**

**соответствует требованиям** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (да, нет);

**6. Положительные стороны работы** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7. Подробный анализ недостатков и ошибок** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**8. Наличие выводов и предложений в курсовой работе** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**9. Курсовая работа допущена к защите, не допущена к защите**

(ненужное зачеркнуть)

Руководитель

курсовой работы

Дата:

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc88347170)

[ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc88347171)

[ГЛАВА 2. ИЗУЧЕНИЕ API OPENWEATHERMAP 11](#_Toc88347172)

[ГЛАВА 3. ОПИСАНИЕ ВОЗМОЖНЫХ ТЕХНИЧЕСКИХ РЕШЕНИЙ. 12](#_Toc88347173)

[ГЛАВА 4. ПРОГРАММИРОВАНИЕ ДИЗАЙНА ВЕБ-САЙТА 14](#_Toc88347174)

[ГЛАВА 5. РАЗРАБОТКА КОДА ВЕБ-САЙТА 16](#_Toc88347175)

[ЗАКЛЮЧЕНИЕ 21](#_Toc88347176)

ВВЕДЕНИЕ

Тема моей курсовой работы: разработка сайта прогноза погоды. Актуальность данной работы очевидна – человек обращается к прогнозу погоды каждый день, и благодаря точному прогнозу в интернете ему удобнее планировать свои дни. При помощи Глобальной сети возможно посмотреть погоду в любой точке мира и практически из любого места. Поэтому, метеорологические сайты всё ещё остаются и продолжат оставаться востребованными среднестатистическими пользователями.

На данный момент, во Всемирной паутине существует множество подобных бесплатных сайтов. Однако, анализ ситуации в сети показал, что сайты, которые вовремя не обновляются и не обеспечивают пользователей актуальной и своевременной информацией, быстро теряют свою аудиторию. Несоответствие содержания сайта требованиям пользователей – довольно распространенная проблема. Также, просматривая типичные интернет-платформы с погодой, можно столкнуться с проблемой навигации по сайту: даже необходимая информация совершенно ни к чему на сайте, если ее невозможно найти на странице среди множества панелей и ссылок.

Таким образом, целью моей работы является разработка сайта с прогнозированием погоды, обеспечивающего удобство, безопасность и интуитивную понятность интерфейса пользователю, а также отображение достоверной информации о погоде.

Можно выделить несколько задач моей работы:

1. Изучение предметной области системы
2. Сравнение аналогов выбранной системы
3. Изучение API OpenWeatherMap
4. Разработка макета сайта
5. Выбор средств создания веб-сайта
6. Определение необходимых моделей для базы данных
7. Разработка веб-сайта

ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Область деятельности – разрабатываемый мной сайт предназначен для обеспечения пользователей информацией о предстоящем состоянии погоды в определенном городе или регионе на заданный период времени.

Название – прогноз погоды «meme meteo»

Целевая аудитория сайта – люди в возрасте от 12 до 80 лет, использующие интернет ресурсы и проживающие в различных странах и регионах. Данной группе пользователей будет доступен такой перечень функций системы, как:

1. Просмотр прогноза на текущий момент
2. Просмотр прогноза на текущий день с интервалом 3 часа
3. Просмотр прогноза на 4 дня после текущего
4. Поиск прогноза погоды по городам

Общие цели информационной системы:

1. Предоставление пользователю точного прогноза погоды
2. Простой, понятный и удобный интерфейс
3. Информация для пользователя в доступном формате
4. Наличие полного функционала типичного сайта с прогнозом погоды

Таким образом, главная задача выбранной мной информационной системы – удобная и понятная демонстрация состояния погоды, ориентированная на среднестатистического пользователя. Наглядность и простоту системы будут обеспечивать интернет-мемы — иронические изображения, спонтанно приобретающие популярность, распространяясь в Глобальной сети Интернет. Изображения такого типа позволят быстро передать основную мысль, наладить коммуникацию и найти общий язык с целевой аудитории системы с помощью юмора.

Чтобы отразить отношения между пользователями и прецедентами, т.е. функциями, выполняемыми системой, я построила UML-диаграмму прецедентов (или Use Case диаграмму), позволяющую описать систему на концептуальном уровне. Для ее построения я воспользовалась онлайн сервисом [plantuml.com](http://www.plantuml.com/plantuml/uml/LLB1Ji904BtlLqmuoQ5-00V2pNZmHFo0JLbX8nPIbeCn9WKY5uW7xuPVQ504mTPVcFaZPxP2IyZkPjw-z-RDqjR8wTYD7_ggR-uSk074jjjpqB6n2PqTHAg1AypnG4lSyhd6bAQuaRe1e4TmEpAnKaCTtkkkWHlZp02UL0rNb61E4_pXVKe9uAy0-84PxrjC6P2IroSjQk9qSMMSrzT5pXSxbk_FUBu0gIZnoAnGUbdRw3Zi5U8D9RnUX0FSCHND0LEEdz6CNe4c4eg1ZBlCJpRQHg9_vuil6PIMNirq_YtOOKypmBM0uhLZrvJcVEQCo63Vd51SvOLmLuxrR_p68ebKJY3HrYmwZbi3hxcEXcFNjkwnhfwLAe9145msIpjeW8t2_hXZLFbo1QVw7bNUEUwfPQSgKSc_6WHDE1gV06cjMYRgy2Vp1m00). Данная диаграмма вариантов использования представлена на рисунке 1.1.

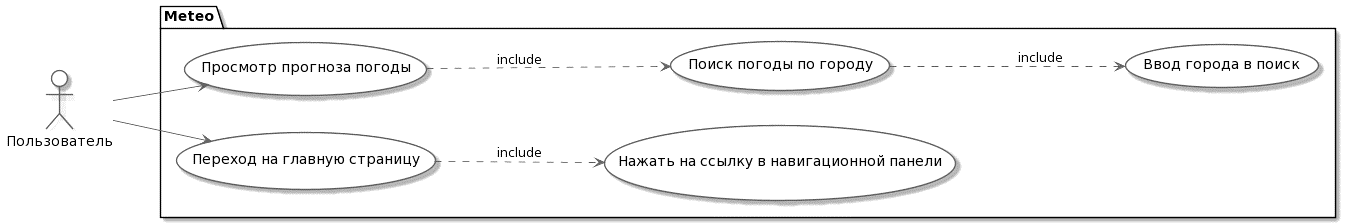
**

Рисунок 4 – UML-диаграмма прецедентов системы

Для того, чтобы лучше понимать работу системы и конкретизировать последовательность действий при любом сценарии, необходимо составить описание сценариев использования прецедентов проектируемой информационной системы:

1. Название прецедента: Просмотр прогноза погоды в городе Москва

Действующее лицо: Пользователь

Цель: найти информацию о погоде в Москве на текущий период времени

Предусловия:

- Зайти на сайт

Главная последовательность действий:

- Ввести в поисковую строку в навигационной панели название города «Москва»

- Нажать на кнопку «Найти»

Ожидаемый результат: Переадресация на страницу с прогнозом погоды в городе Москва

1. Название прецедента: Просмотр прогноза погоды в несуществующем городе

Действующее лицо: Пользователь

Цель: Проверка работы страницы ошибки

Предусловия:

- Зайти на сайт

Главная последовательность действий:

- Ввести в поисковую строку в навигационной панели «абвгд»

- Нажать на кнопку «Найти»

Ожидаемый результат: Переадресация на страницу ошибки

1. Название прецедента: Переход на главную страницу

Действующее лицо: Пользователь

Цель: Переход на главную страницу сайта

Предусловия:

- Зайти на сайт

Главная последовательность действий:

- Нажать на ссылку «Главная» на навигационной панели

Ожидаемый результат: Переадресация на главную страницу

1.1. Сравнение аналогов выбранной системы

На сегодняшний день существует огромное количество интернет сайтов на русском языке, отображающих метеорологические прогнозы по всей Земле. Для изучения аналогов по выбранной теме, я отобрала 3 сайта с прогнозом погоды - [gismeteo.ru](https://www.gismeteo.ru/weather-sankt-peterburg-4079/) (см. Рисунок 1.1), [yandex.ru/pogoda](https://yandex.ru/pogoda/saint-petersburg?lat=59.938951&lon=30.315635) (см. Рисунок 1.2), [weather.com](https://weather.com/ru-RU/weather/today/l/Saint+Petersburg+Saint+Petersburg?canonicalCityId=04f8b509ffc5eed7a43bd68b017401916c37182eddbcb14a104bfbafd73278c0) (см. Рисунок 1.3).

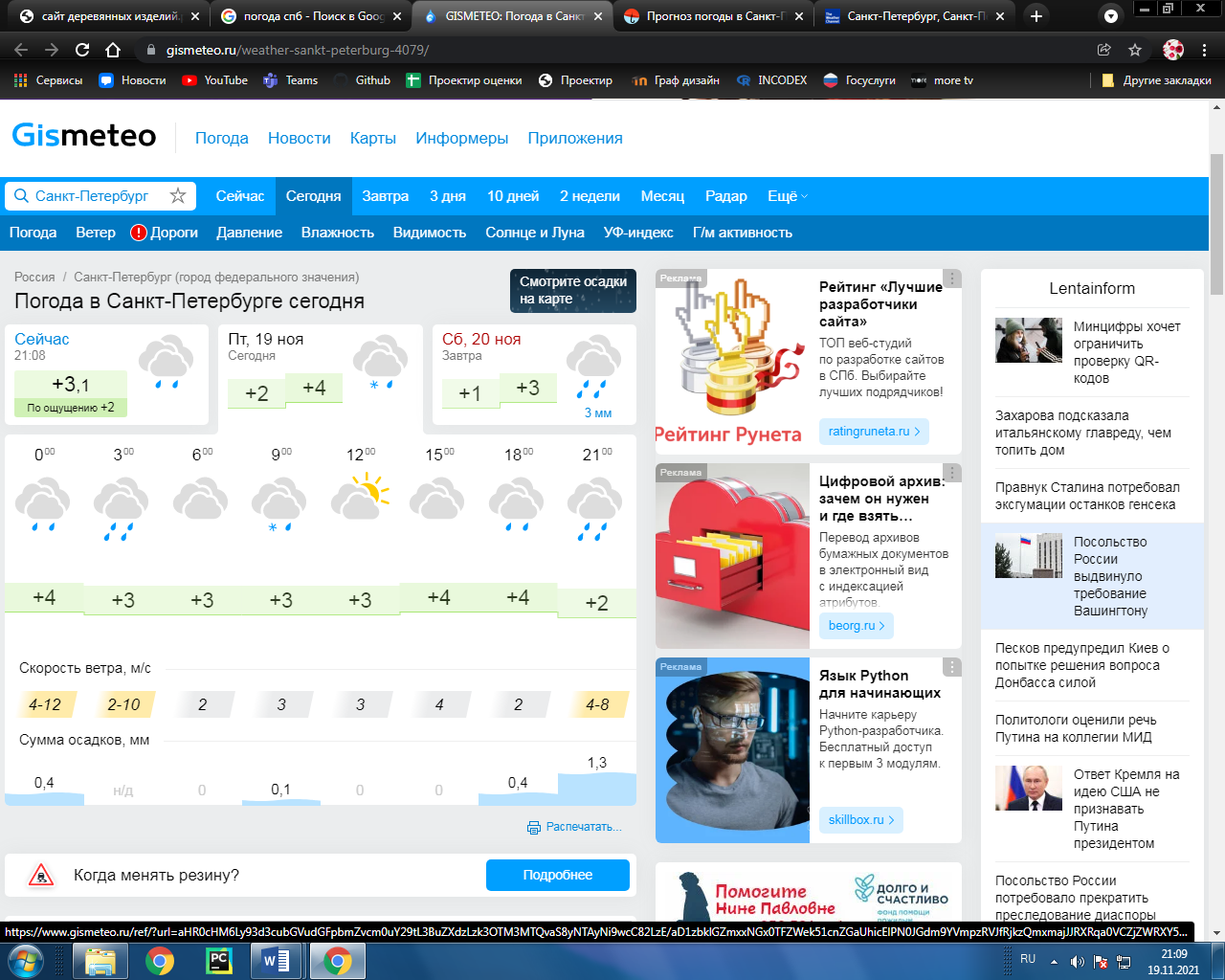


Рис. 1.1 – веб-сайт Gismeteo



Рис. 1.2 – веб-сайт Яндекс погоды

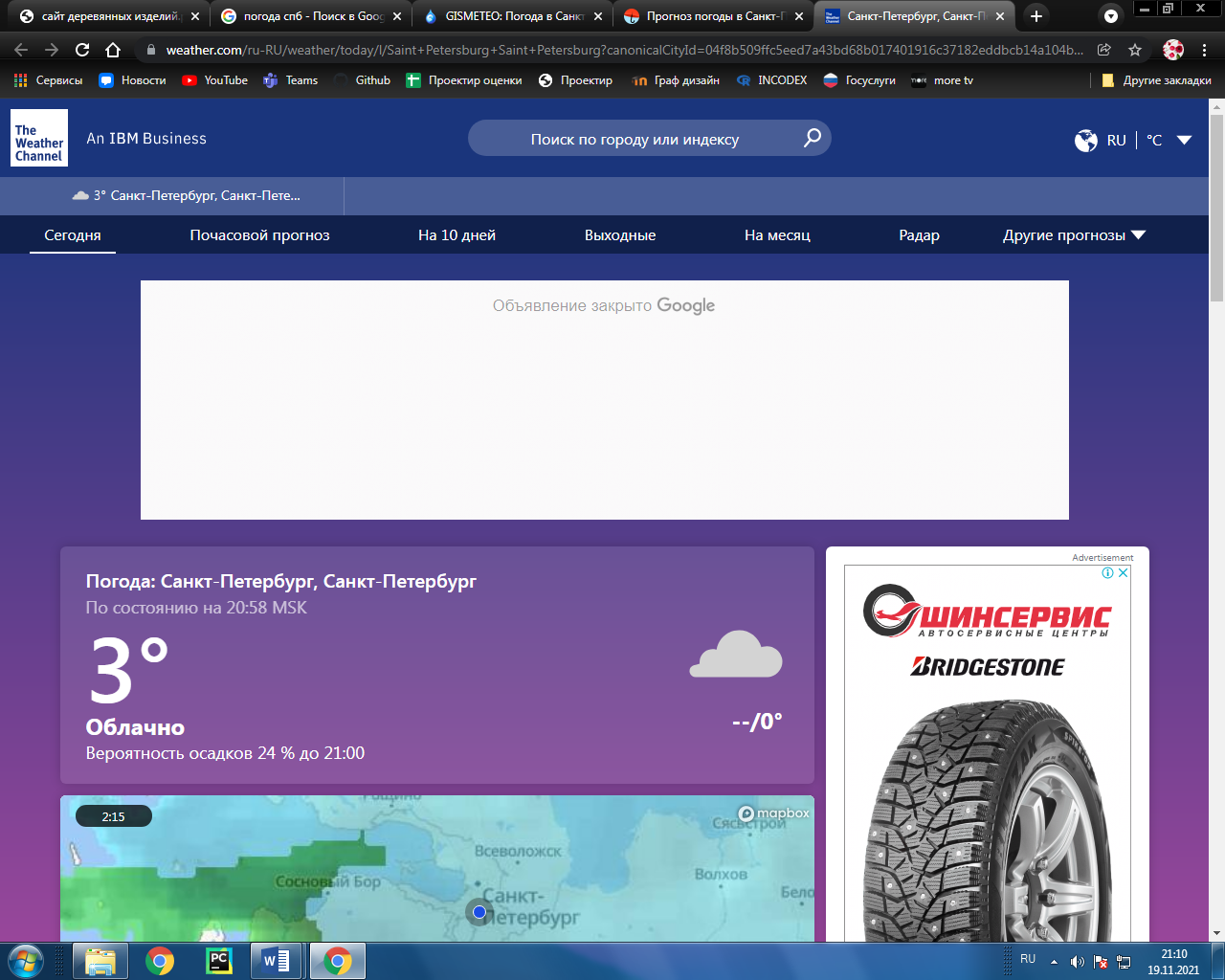


Рис. 1.3 – веб-сайт The Weather Channel

Сравнительный анализ представленных аналогов описан в таблице 1.

Таблица 1 – Сравнение аналогов системы

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии/Описание соответствия системы критериям | [gismeteo.ru](https://www.gismeteo.ru/weather-sankt-peterburg-4079/) | [yandex.ru/pogoda](https://yandex.ru/pogoda/saint-petersburg?lat=59.938951&lon=30.315635) | [weather.com](https://weather.com/ru-RU/weather/today/l/Saint+Petersburg+Saint+Petersburg?canonicalCityId=04f8b509ffc5eed7a43bd68b017401916c37182eddbcb14a104bfbafd73278c0) |
| Удобство использования | Из-за большого количества ссылок теряется основная информация | Удобный | Удобный |
| Функционал | Прогноз на данный момент, сегодня, завтра, 3 дня, 10 дней, 2 недели, месяц, радар, карта осадков и пр. функции | Прогноз на данный момент, сегодня, 10 дней, месяц, карта осадков, световой день, прямой эфир погоды | Прогноз на данный момент, сегодня, почасовой, 10 дней, месяц, выходные, радар, карта осадков, специальные прогнозы |

Продолжение Таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Дизайн | Достаточно простой, без особенностей | Удобный и наглядный | Минималистичный дизайн, удачное сочетание цветов |
| Доступность информации | Большое количество информации на одной странице, мелкие значки | Легко найти нужную информацию | Информация в доступном виде |
| Количество действий при поиске информации | Не более 3х | Не более 2х | Не более 3х |
| Скорость отклика | Быстрая | Быстрая | Быстрая |
| Визуальное отображение (картинки или анимация) | Картинки и анимированная карта осадков | Наличие всех визуальных составляющих (картинки, анимация, прямой эфир состояния погоды) | Картинки и анимированная карта осадков |

После изучения всех представленных в таблице 1 аналогов, очевидно полагать, что они все имеют полный функционал, характерный для системы прогноза погоды. Однако, большее внимание стоит уделить веб-сайту Яндекс погоды, который удовлетворяет большему количеству заданных критериев, то есть имеет самый простой, но в то же время удобный интерфейс, и хорошую визуальную составляющую. На сильные стороны упомянутого сайта я и буду ориентироваться при создании дизайна и разработке кода своей системы.

1.2. Общая структура системы

Структура сайта довольно проста, так как его функционал невелик (см. Рисунок). Навигация включает в себя всего две основные составляющие – главную страницу с ссылкой в навигационной панели и страницу с прогнозом погоды, на которую возможно перейти только после ввода запроса в поиск. Также, при вводе некорректных данных в поисковую строку, происходит переход на страницу ошибки, которая формально будет считаться третьей страницей системы.

Главная

Прогноз погоды

Ошибка

Рисунок – Структура сайта прогноза погоды «meme meteo»

ГЛАВА 2. ИЗУЧЕНИЕ API OPENWEATHERMAP

API (Application Programming Interface или интерфейс программирования приложений) — это набор классов, функций, процедур и стандартов в виде интерфейса для создания новых приложений, благодаря которому одна программа имеет возможность взаимодействовать с другой. Это позволяет разработчикам расширять функциональность своего продукта.

OpenWeatherMap — это онлайн сервис, предоставляющий API для доступа к данным о текущей погоде, прогнозам, для web-сервисов и мобильных приложений. Архивные данные доступны только на коммерческой основе, однако существует функционально ограниченная бесплатная версия, которую я использовала для разработки выбранной системы. В качестве источника данных, OWM используют профессиональные и частные погодные станции. На сегодня таких станций более 40 тысяч. Сервис предоставляет бесплатный API ко всем данным о погоде, к их истории, прогнозам и всему многообразию погодных карт.

Для работы с API необходимо было зарегистрироваться на сайте OpenWeatherMap и получить ключ доступа. После регистрации и получения API ключа, можно приступать к разработке системы.

Для разработки своего сайта с погодой я использовала бесплатный метод 5 Day / 3 Hour Forecast, который выводит информацию о погоде на 5 дней начиная с текущего момента с интервалом в 3 часа. Данные из API выводятся в формате JSON.

ГЛАВА 3. ОПИСАНИЕ ВОЗМОЖНЫХ ТЕХНИЧЕСКИХ РЕШЕНИЙ.

Тщательно подойдя к выбору технологий для разработки моего проекта, я кратко описала преимущества выбранных платформ.

Язык программирования Python:

1. Простота использования, простой синтаксис
2. Легкость изучения (доступные учебные пособия) и читабельность кода
3. Гибкость, т.е. способность смешиваться с другими языками
4. Работа с фреймворком Django (простой, безопасный и быстрый)
5. Универсальность (подходит для различных проектов)

Фреймворк Django:

1. Простой и быстрый (создание сайта из компонентов)
2. Функциональность (административная панель, формы, миграции и др.)
3. Безопасность
4. Работа с крупными проектами и большой загруженностью

База данных SQLite:

1. Высокая скорость и производительность
2. Простота использования
3. Безопасность и простота установки
4. Бесплатное обслуживание и открытый доступ

HTML, CSS и JS фреймворк Bootstrap:

1. Высокая скорость создания сайта
2. Корректная работа сайта во всех поддерживаемых браузерах и ОС
3. Готовые компоненты и шаблоны
4. Элементарные навыки верстки
5. Однородность дизайна
6. Легкость изучения (огромное количество обучающих материалов)

Среда разработки PyCharm:

1. Помощь при написании кода (функции автодополнения и анализа кода, подсветка и исправление ошибок)
2. Встроенные инструменты (инструмент запуска тестов, терминал, инструменты для работы с базами данных)
3. Поддержка различных веб-фреймворков

API OpenWeatherMap:

1. Поддерживается бесплатная функционально ограниченная версия
2. Источник данных - официальные метеорологические службы
3. Используются форматы JSON, XML и HTML
4. Данные о погоде в более чем 200 тысячах городов
5. Погода до 30 дней вперед и прогноз за прошедшие периоды

ГЛАВА 4. ПРОГРАММИРОВАНИЕ ДИЗАЙНА ВЕБ-САЙТА

Разработка интерфейса является непростой и трудоемкой задачей, так как именно он считается одним из самых важных элементов для веб-сайта. При разработке оформления, подборе шрифтов и цветовой схемы нужно руководствоваться не только принципами эргономики, чтобы возможности человека удовлетворялись и поддерживались вариантами дизайна, но и учитывать предпроектный анализ.

Для своей системы я выбрала ограниченную цветовую палитру и минималистичный стиль. Также, была определена геометрия веб-сайта: ссылка на главную страницу, логотип и поисковая строка должны быть расположены стандартным образом на навигационной панели, что обеспечивает удобство взаимодействия пользователя с информацией. С помощью бесплатной онлайн платформы [textdrom.com](https://textdrom.com/3D-text.html) я разработала логотип сайта, изображенный на рисунке 4.1.



Рисунок 4.1– Логотип сайта

Сами страницы подразделены на логические и тематические блоки в зависимости от их наполнения. Например, первый блок на странице с прогнозом погоды должен включать в себя состояние погоды на текущий момент времени в выбранном городе, а также дополнительные показатели, такие как скорость ветра, влажность и прочее. Интересным аспектом в дизайнерском решении системы является использование интернет-мемов, отображающих состояние погоды в шуточной форме, тем самым обеспечивая интуитивную понятность сайту.

Для быстрой вёрстки HTML-кода веб-приложения был выбран HTML, CSS и JS фреймворк Bootstrap. Благодаря наличию у него большого количества готовых шаблонов, можно создавать сайт с красивым, приятным, профессиональным дизайном за достаточно короткое время.

В результате, дизайн веб-сайта получился следующим (см. Рисунок 4.2).



Рисунок 4.2– Дизайнерское решение будущего сайта

ГЛАВА 5. РАЗРАБОТКА КОДА ВЕБ-САЙТА

Для начала, создадим проект Meteo и приложение main в среде разработки PyCharm. Создадим элементы сайта в html формате – навигационную панель, главную страницу, страницу с погодой, страницу ошибки. Оформим дизайн страниц согласно созданным ранее прототипам интерфейса и выбранным дизайнерским решениям. Создадим папку static и загрузим в нее изображения, которые будут использоваться на сайте.

Выполним проектирование базы данных системы, необходимое для структурирования элементов и составных частей при ее разработке. На этапе проектирования системы было принято решение использовать графическое отображение погоды с целью визуализации ее состояния. Вследствие этого, база данных будет включать в себя единственную модель Pictures, которая в свою очередь должна хранить ID объекта, название состояния погоды и картинку, которая в дальнейшем будет отображаться на странице в зависимости от состояния погоды на текущий момент. Для наглядности, в сервисе [Draw io](https://app.diagrams.net/) мной была создана ER диаграмма спроектированной БД, представленная на рисунке 6.1.

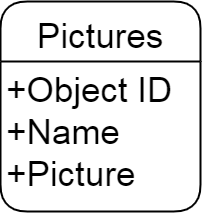


Рисунок 6.1 - ER диаграмма базы данных

Исходя из спроектированной базы данных, создадим модель Pictures (см. Рисунок 6.2).

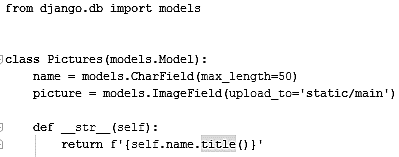


Рисунок 6.1 - Модель Pictures

Для получения прогноза погоды через API я буду использовать метод 5 Day / 3 Hour Forecast, описанный в Главе 2. Необходимо будет реализовать 3 функции системы: получение погоды в данный момент времени (см. Рисунок 6.2), получение погоды на текущий день с интервалом в три часа (см. Рисунок 6.3), получение погоды на 4 дня после текущего (см. Рисунок 6.4). Во всех этих функциях мы будем получать данные в формате JSON с помощью ссылки и входящего в него API токена (см. Рисунок 6.5). Все упомянутые функции возвращают словарь с полученными значениями, такими как погода в градусах Цельсия, скорость ветра, влажность, состояние погоды и другое.



Рисунок 6.2 – Функция получения погоды на текущий момент



Рисунок 6.3 – Функция получения погоды на текущий день с интервалом в 3 часа



Рисунок 6.4 – Функция получения погоды на 4 после текущего дня

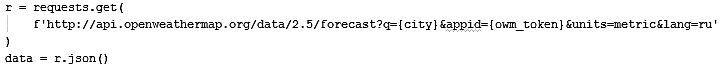


Рисунок 6.5 – Ссылка для получения данных через API

Для того, чтобы отображать картинки из созданной ранее модели в базе данных на страницу в зависимости от состояния погоды, я создам словарь с самыми распространенными состояниями погоды, такими как облачность, ясно, дождь и прочее. Каждому из этих значений будет присвоено ID картинки из базы данных. Таким образом, по ключу из такого словаря на страницу будет посылаться соответствующая картинка (см. Рисунок 6.6).

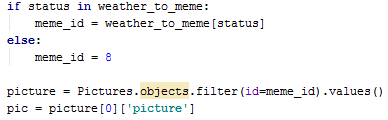


Рисунок 6.6 – Получение картинки из базы данных

Далее пропишем представления страниц приложения main, таких как главная страница и страница с ошибкой (см. Рисунок 6.7).

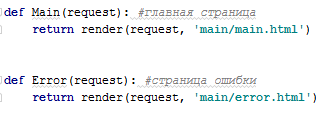


Рисунок 6.7 – Представление страниц во views.py

В функции представления страницы прогноза погоды вызовем все функции, описанные выше, и отправим полученные данные на страницу (см. Рисунок 6.8). Также, пропишем обработчик ошибок, чтобы в случае неверного запроса происходил переход на страницу ошибки. В html коде создадим запросы на получение необходимых данных.

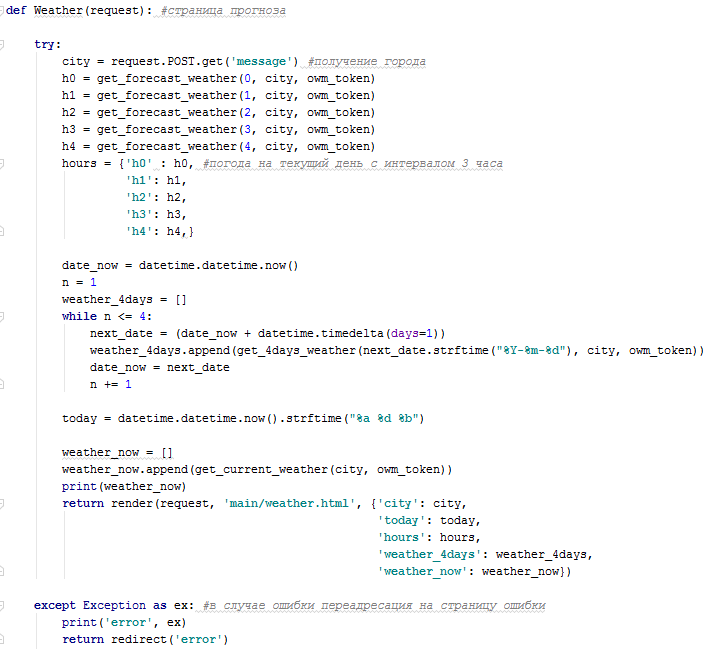


Рисунок 6.8 – Представление страницы прогноза погоды

Пропишем ссылки страниц приложения main, указанные на рисунке 6.9.

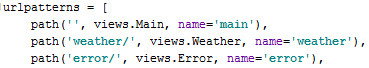


Рисунок 6.9 – Ссылки страниц

В результате проделанной работы, у меня получился полноценный сайт, осуществляющий поиск прогноза погоды в заданном городе. Скриншоты готового сайта изображены на рисунках 7, 8 и 9. После программной реализации проекта, система полностью готова к переходу на этап тестирования и дальнейшей ее публикации в сети.

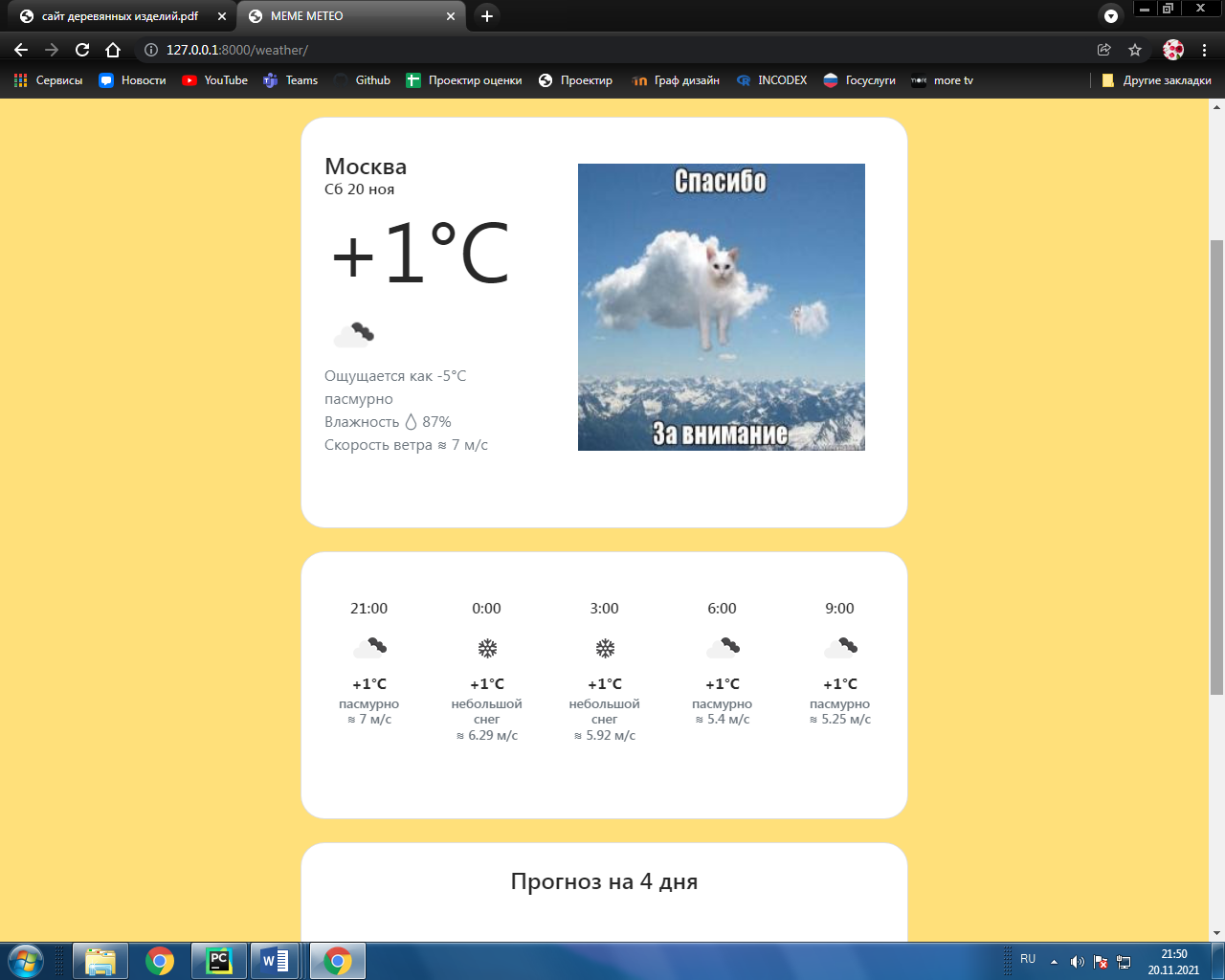


Рисунок 7 – Страница прогноза погоды

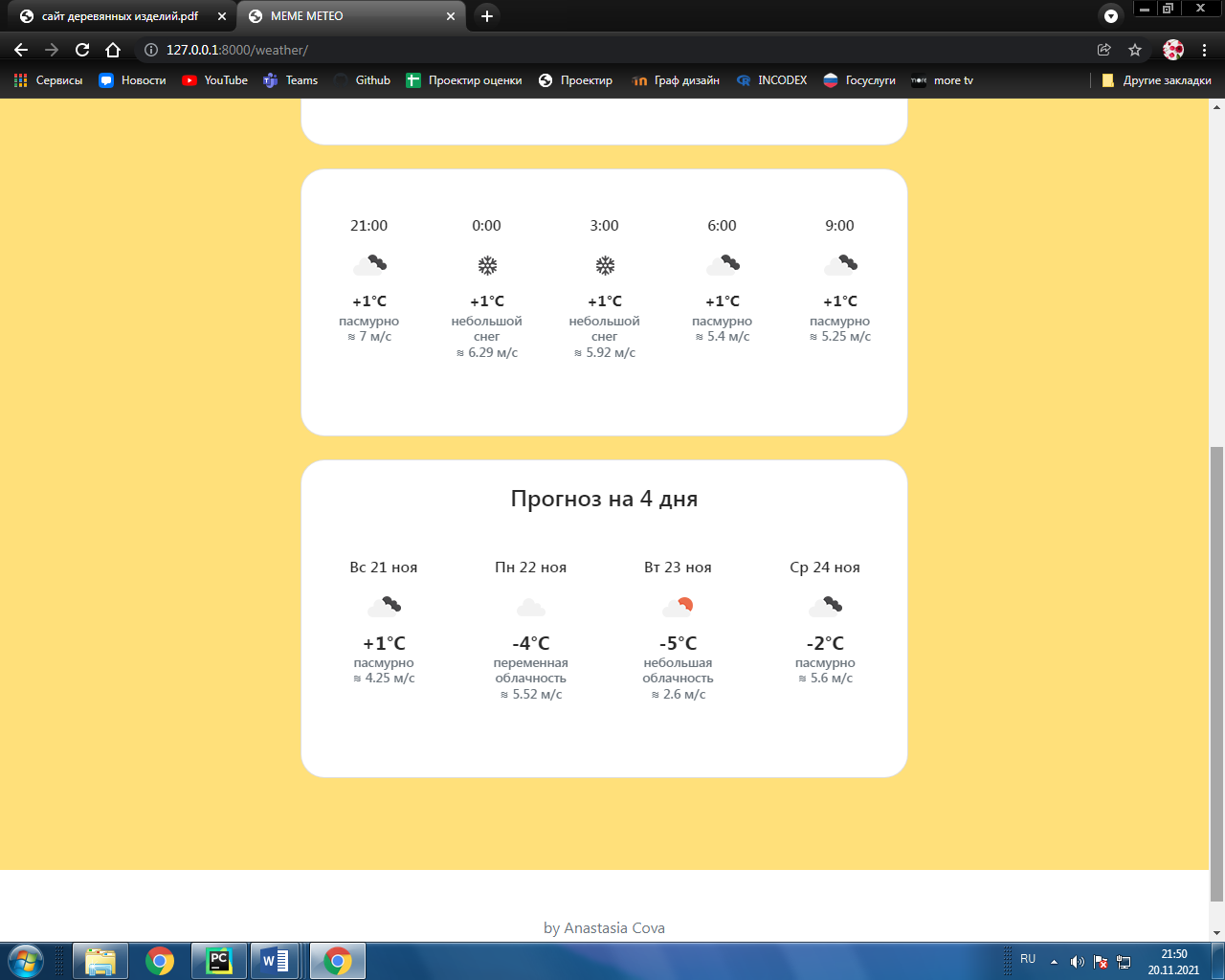


Рисунок 8 – Страница прогноза погоды

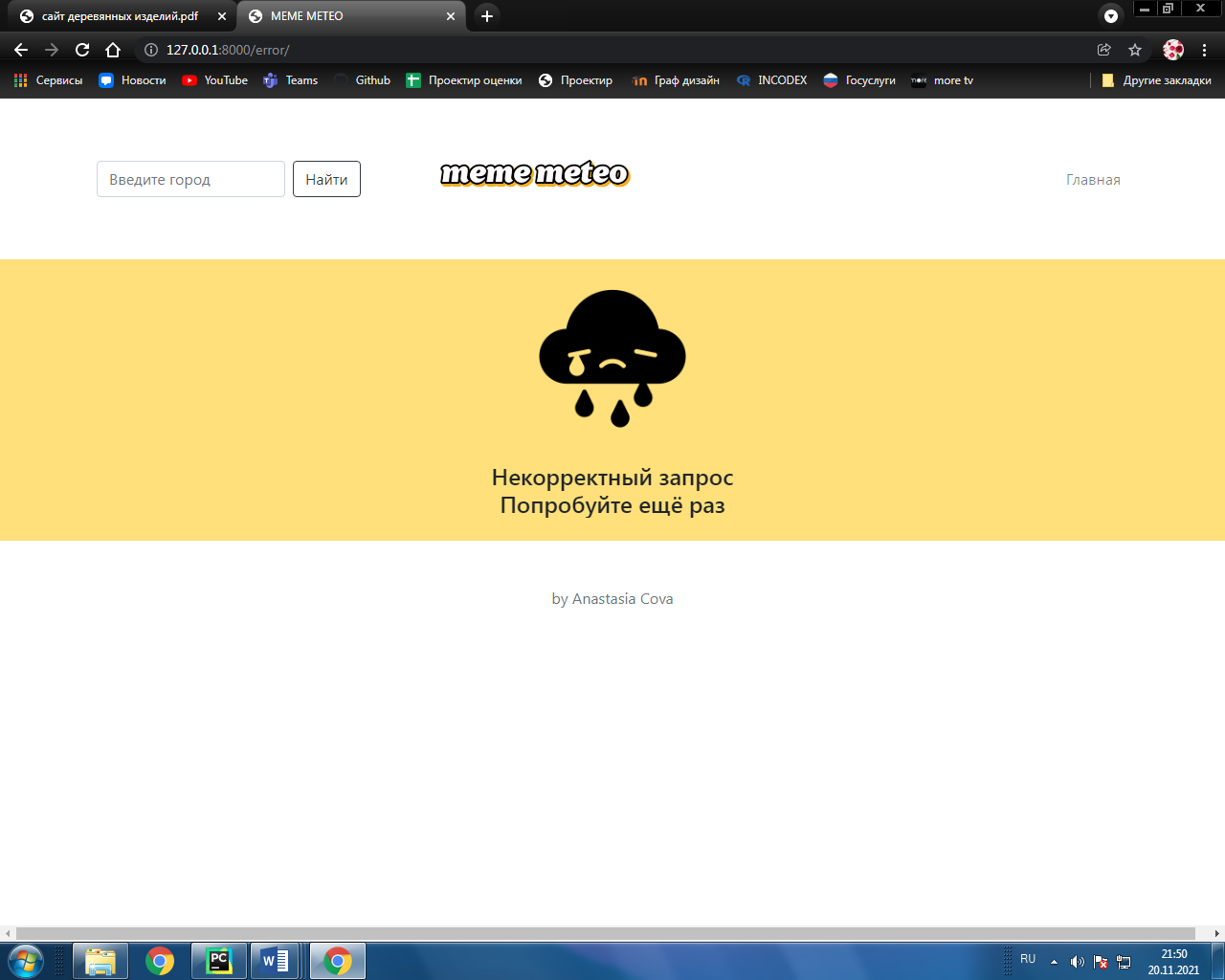


Рисунок 9 – Страница ошибки

ЗАКЛЮЧЕНИЕ

Уровень развития информационных технологий в наши дни делает веб-сайты одной из ключевых составляющих для решения многих задач. Существует огромное множество сайтов с прогнозом погоды, использование которых помогает сэкономить время, решает задачу предоставления информации о погоде пользователю.

В ходе выполнения мной курсовой работы, на основе предпроектного исследования была выявлена актуальность работы, а также особенности создания систем данного вида сайтов. Опираясь на результаты этих исследований, была построена структура веб-сайта и впоследствии разработана информационная система прогноза погоды «meme meteo».

Использование выбранных технических решений, важнейшими из которых является язык программирования Python и фреймворк Django, помогло достичь всех поставленных целей. Опираясь на приобретенные в процессе обучения знания, мне удалось воплотить основной функционал системы и совместить простоту и удобство ее интерфейса.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Справочная документация по языку Python [Электронный ресурс], - URL: <https://www.python.org/doc/>
2. Официальный сайт фреймворка Django [Электронный ресурс], - URL: https://www.djangoproject.com
3. Информация о поддерживаемых типах баз данных фреймворком Django [Электронный ресурс], - URL: https://docs.djangoproject.com/en/3.1/ref/databases/#databases
4. Документация PlantUML по построению WBS диаграмм [Электронный ресурс], - URL: https://plantuml.com/wbs-diagram
5. Документация SQLite базы данных [Электронный ресурс], - URL: https://www.sqlite.org/docs.html
6. Документация Bootstrap [Электронный ресурс], - URL: <https://bootstrap-4.ru/>
7. Документация PyCharm [Электронный ресурс], - URL: <https://www.jetbrains.com/ru-ru/pycharm/learn/>
8. Документация API OpenWeatherMap [Электронный ресурс], - URL: <https://openweathermap.org/>