**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Институт цифры**

**ОТЧЕТ**

**о выполнении семестровой работы  
по дисциплине «Языки программирования»  
Погодный веб-сайт**

Студента 2 курса, группы МОА-211

**Киселева Константина Евгеньевича**

Направление 02.03.03 – «Математическое обеспечение и администрирование информационных систем»

Руководитель:

А.И. Зимин

Работа защищена

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_2023г.

Кемерово 2023

Содержание

[Задание 3](#_Toc139087708)

[Критерии оценивания 3](#_Toc139087709)

[Тема проекта 4](#_Toc139087710)

[Архитектура проекта 5](#_Toc139087711)

[Основные файлы 5](#_Toc139087712)

[Основные функции 6](#_Toc139087713)

[Использованные подходы 7](#_Toc139087714)

[Средства разработки 8](#_Toc139087715)

[Ссылки 9](#_Toc139087716)

# Задание

Всего за проект можно получить 40 баллов, которые идут в счет экзамена. Необходимо написать Front-end часть веб-сайта на языке JavaScript без использования каких-либо фреймворков. Готовый проект надо представить и защитить преподавателю лично. Обязательные требования:

* Работа над проектом ведите в гит-репозитории. Ссылку на него необходимо прикрепить в ЭИОС.
* В качестве источника данных возьмите любое HTTP-API на ваш выбор.
* Ваш сайт должен состоять из одного html-файла. Вместе с ним при необходимости могут быть файлы стилей (css) и различные ресурсы (например, картинки).
* Код должен быть разделен на модули и располагаться в отдельных js-файлах. Архитектурно должен быть выделен слой для работы с данными и слой для отображения данных на странице.
* При открытии сайта перед пользователем должна появляться страница приветствия, на которой он только может ввести свое имя и нажать кнопку «Далее». При нажатии на кнопку «Далее» открывается главная страница со всем функционалом вашего сайта. В шапке сайта должно быть сообщение: «Привет, !».
* Данные должны отображаться в удобном для пользователя виде (например, при помощи списков и таблиц).
* Сайт должен быть интерактивным: должны присутствовать разные элементы управления (кнопки, чекбоксы и т.д.) и обработка различных событий (нажатие кнопок мыши, клавиш клавиатуры и т.д.).
* Для пользователя должна быть возможность различных вариантов загрузки данных и манипуляции с ними на странице (сортировка, фильтрация, получение статистик и т.д.).

## Критерии оценивания

Критерии оценки проекта (при условии выполнения всех обязательных требований):

1. Объём программного кода (5 баллов). 5 баллов за 500 и более строк кода, 3 – за 300 и т.д. Учитывается только "разумный код".
2. Применённые технологии (15 баллов). Баллы даются за реализацию и понимание всех пройденных тем (функции, массивы, объекты, классы, модули, асинхронность и т.д.).
3. Архитектурная сложность и чистота кода (20 баллов). Если классовая модель неоправданно бедная, разделение на модули отсутствует или выполнено формально, переменные и классы названы плохо, код оформлен неаккуратно и т.д. – снижаются баллы. Также вы должны понимать и уметь объяснять код вашего проекта.

# Тема проекта

**Основная идея проекта:** создание полезного веб-сайта с помощью которого можно получить информацию о погоде в любой точке мира с использованием JavaScript, HTML, CSS и знаний, полученных в течение семестра.

**Функциональность:** пользователя встречает окно с текстовым полем, в которое он может ввести своё имя, и кнопкой, по нажатии на которую происходит переход на выбор города, а затем на главное окно приложения с приветственным сообщением сверху экрана. На главном окне находятся три кнопки, предоставляющие пользователю информацию, и окно фильтров, которые можно применять к полученным данным, находящимся на экране. Всё время на экране находится заголовок, который содержит в себе логотип сайта.

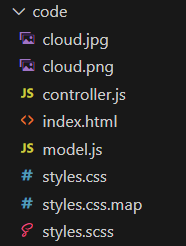
**Используемые API:**

**Yandex HTTP Geocoder** – удобный и простой в освоении API для преобразования названий географических объектов в координаты долготы и широты и наоборот.

**Yandex Weather** – API, дающий возможность быстро и легко узнать данные о погоде в любой точке мира на данный момент, а также получить информацию о прогнозах на будущее.

# Архитектура проекта

## Основные файлы



* **model.js –** модуль транспортного уровня, отвечающий за отправку запросов к API, получение и обработку приходящих данных, перевод погодных характеристик в единый и удобный для визуализации формат.  
  **controller.js** – модуль, соответствующий уровню визуализации, который создаёт все элементы на странице веб-приложения.
* **index.html** – единственный html-документ, выступающий в качестве каркаса приложения и наполняемый с помощью вышеуказанных модулей.
* **styles.css** – единственный файл со стилями, настраивающий комфортный и приятный вид форм, текстовых полей и других элементов на странице, с которыми может взаимодействовать пользователь.
* Для удобства разработки разработки использовался scss файл, который затем компилировался в css.
* Также проект содержит в себе картинку, выступающую в роли логотипа.

## Основные функции

Функции транспортного уровня:

* **fetchWeatherForecast –** используя координаты и количество дней, обращается к API Яндекс.Погоды, возвращает полученный json.
* **fetchCords –** используя название населённого пункта обращается к API Яндекс Геокодера, возвращает координаты.
* **getForecast –** получает в аргументах город и количество дней, прогноз которых необходимо получить. Сначала получает координаты при помощи fetchCords, затем получает прогноз при помощи fetchWeatherForecast, обрабатывает и возвращает его.
* **getCurrentWeather –** работает аналогично предыдущей функции, но получает и возвращает только информацию о погоде на данный момент.

Функции визуализации данных:

* Первоначальная инициализация производится при помощи лямбда-функции.
* **cityChoice** - вызывается для выбора города.
* **mainPage** – основная страница, содержащая в себе 3 кнопки
* **getCurrent**, **getForecast** – слушатели, подключенные к кнопкам. Обращаются к модулю транспортного уровня при помощи динамического импорта

Остальные функции не имеют названия (лямбда-функции) или слишком малы.

# Средства разработки

* Visual Studio Code (VS Code) – текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.
* JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.
* HTML — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере.
* Sass - метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов каскадных таблиц стилей.
* GitHub – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

# Ссылки

Ссылка на гит-репозиторий: <https://github.com/greendolf/python_semestr>