

**Министерство образования и науки Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**УНИВЕРСИТЕТ ИТМО**  
Факультет инфокоммуникационных технологий

Образовательная программа: 09.03.02

Направление подготовки (специальность): Мобильные сетевые  
технологии

## **О Т Ч Е Т**

о курсовой работе

Тема задания: реализация многостраничного веб-приложения (МРА) на  
HTML/CSS/JavaScript

Обучающийся: Новиков Глеб Витальевич, К33402

Руководитель: Добряков Д. И., преподаватель

Оценка за курсовую работу \_\_\_\_\_

Дата \_\_\_\_\_

Санкт-Петербург

2022

## ЧАСТЬ 1. СРЕДСТВА РАЗРАБОТКИ И ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

### 1. Средства разработки

Проект представляет собой многостраничное веб-приложение, которое основано на работах, выполняемых в рамках дисциплины “Фронт-энд разработка”. Приложение написано на HTML/CSS/JavaScript с использованием поставщика картографических данных — Mapbox.

### 2. Функциональные требования

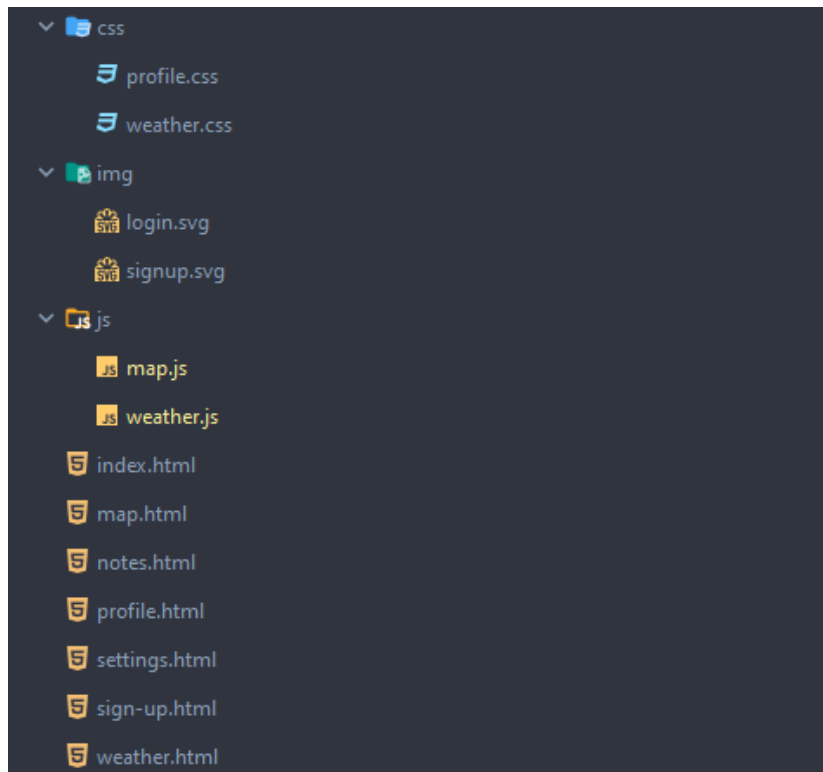
1. Разработка многостраничного веб-приложения (МРА) на HTML/CSS/JavaScript
2. Использование фреймворка Bootstrap
3. Взаимодействие со сторонними сервисами по REST API
4. В проекте должно быть, как минимум, 7 страниц

## ЧАСТЬ 2. ОПИСАНИЕ ПРОЕКТА

### 1. Описание проекта

Веб-приложение представляет собой прототип платформы для аналитики агро-хозяйства. С помощью платформы можно изучить 3D рельефа территории, на котором находятся поля. Также получить для них погодную сводку и вносить заметки по планированию и ведению хозяйственной деятельности.

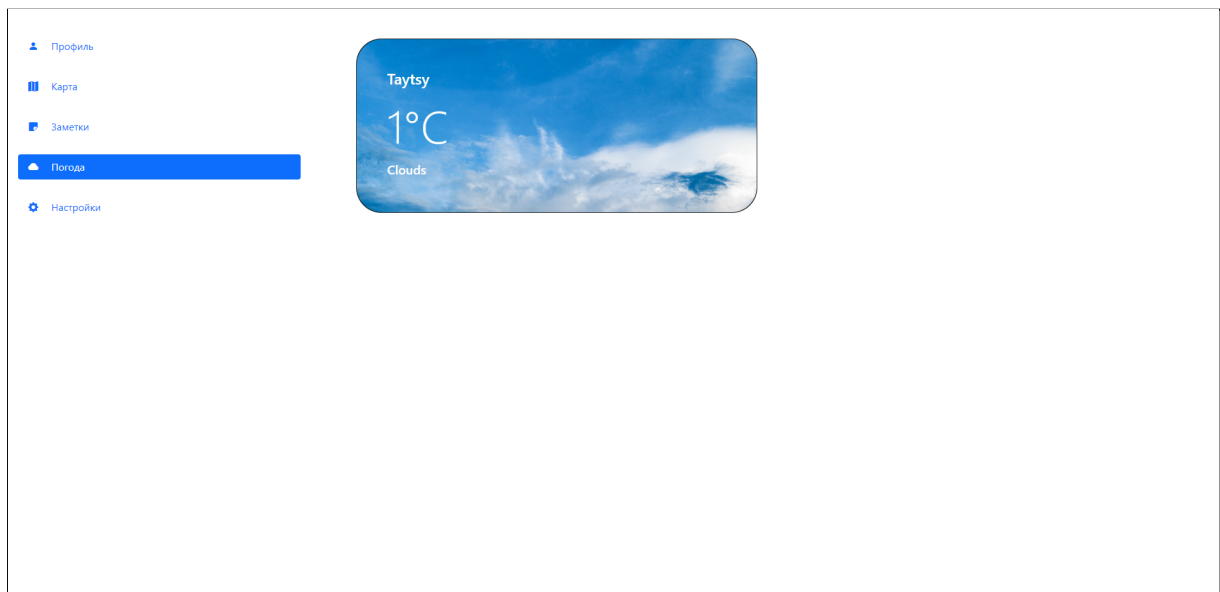
### 2. Структура проекта



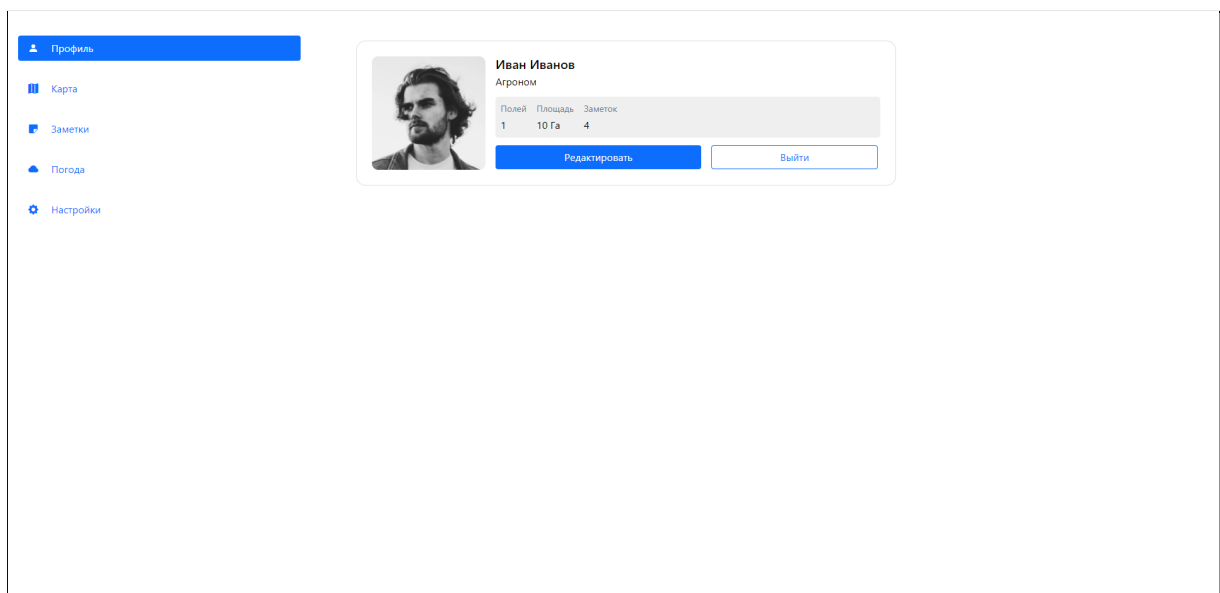
### 3. Основные экраны



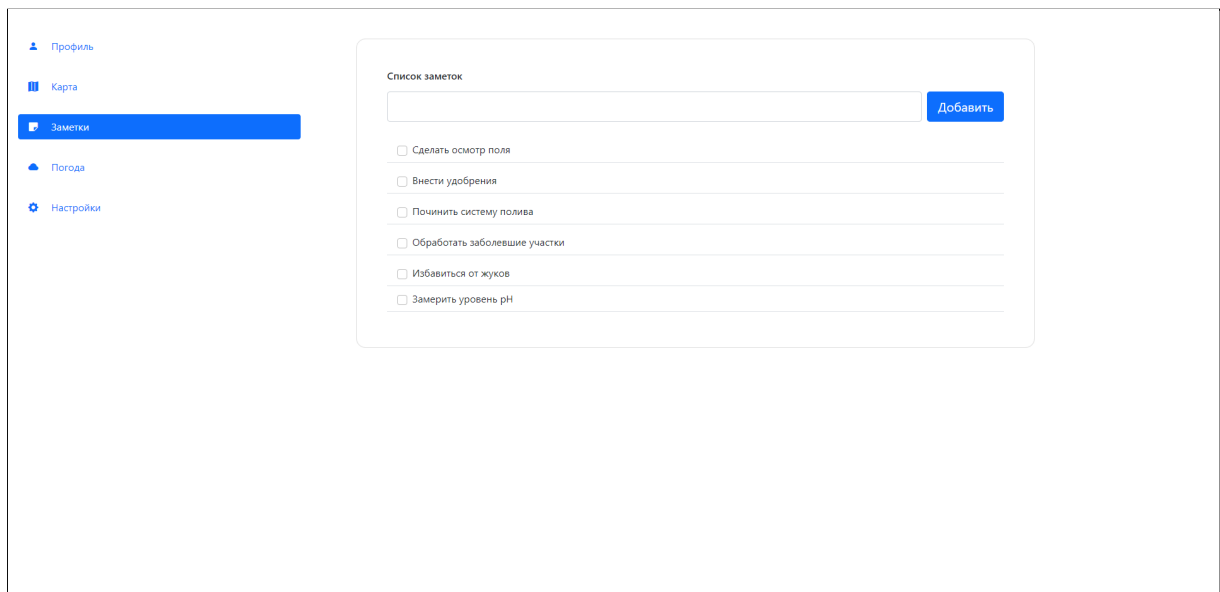
Экран с 3D картой рельефа



Экран с погодой



Экран профиля пользователя



Экран заметок

## ЧАСТЬ 3. ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

### 1. Bootstrap

```

<div class="w-75">
  <form class="px-4 pt-5" action="">
    <h3>Уведомления</h3>

    <div class="form-check mt-3">
      <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault" />
      <label class="form-check-label" for="flexCheckDefault">0 прогнозируемом граде</label>
    </div>
    <div class="form-check mt-3">
      <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault2" />
      <label class="form-check-label" for="flexCheckDefault2">0 засухе на полях</label>
    </div>
    <div class="form-check mt-3">
      <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault3" />
      <label class="form-check-label" for="flexCheckDefault3">0 заболеваниях растений</label>
    </div>
    <div class="form-check mt-3">
      <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault4" />
      <label class="form-check-label" for="flexCheckDefault4">0 минеральном истощении</label>
    </div>
    <div class="form-check mt-3">
      <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault5" />
      <label class="form-check-label" for="flexCheckDefault5">0 загрязнении воздуха</label>
    </div>
  </form>

  <button type="button" class="btn btn-primary ms-3 mt-5">Сохранить</button>
</div>

```

Веб-приложение версталось с использованием фреймворка Bootstrap,

чтобы быстро реализовать адаптивный дизайн без необходимости прописывать стиль каждого элемента в CSS.

## 2. OpenWeatherMap API

```
lat = 59.693088;
lon = 30.200033;

async function getWeather() {
  url = `https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&appid=${apiKey}`;
  const response = await fetch(url, {init: {method: "GET"}});
  return await response.json();
}

function setWeather() {
  const place = document.getElementById( 'weather-place');
  const temp = document.getElementById( 'weather-temp');
  const status = document.getElementById( 'weather-status');
  getWeather().then((json) => {
    place.innerText = json.name;
    temp.innerText = parseInt( s: json.main.temp - 273.15) + '°C';
    status.innerText = json.weather[0].main;
    console.log(json.name)
  });
}

setWeather();
```

Информация о погоде на полях выгружается из открытого сервиса OpenWeatherMap по REST API. После чего данные отображаются в интерфейсе пользователя.

## 3. Mapbox

```
const map = new mapboxgl.Map({
  container: 'map',
  zoom: 13.1,
  center: [30.200033, 59.693088],
  pitch: 60,
  bearing: 90,
  style: 'mapbox://styles/mapbox-map-design/ckhgnf2tz0dt119ny6azh975y'
});

map.on('load', () => {
  map.addSource('mapbox-dem', {
    'type': 'raster-dem',
    'url': 'mapbox://mapbox.mapbox-terrain-dem-v1',
    'tileSize': 512,
    'maxZoom': 14
  });

  // add the DEM source as a terrain layer with exaggerated height
  map.setTerrain({ 'source': 'mapbox-dem', 'exaggeration': 1.5 });

  // add a sky layer that will show when the map is highly pitched
  map.addLayer({
    'id': 'sky',
    'type': 'sky',
    'paint': {
      'sky-type': 'atmosphere',
      'sky-atmosphere-sun': [0.0, 0.0],
      'sky-atmosphere-sun-intensity': 15
    }
  });
});
```

3D карта рельефа создана с помощью картографического сервиса Mapbox API. Для этого необходимо было изменить подложку карты на Terrain. Также карта была повернута по оси Z (для просмотра рельефа в изометрии) и установлены стили для неба.

```
// Add a new layer to visualize the polygon.
map.addLayer({
  'id': 'maine',
  'type': 'fill',
  'source': 'maine', // reference the data source
  'layout': {},
  'paint': {
    'fill-color': '#25d907',
    'fill-opacity': 0.5
  }
});

// Add a black outline around the polygon.
map.addLayer({
  'id': 'outline',
  'type': 'line',
  'source': 'maine',
  'layout': {},
  'paint': {
    'line-color': '#FFFFFF',
    'line-width': 3
  }
});
```

Для разметки полей на карте используется отрисовка с помощью создания дополнительных слоёв в Mapbox. Данные о поле представляют собой полигон в формате GeoJSON.

## ЧАСТЬ 4. ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы, мной были освоены фреймворк Bootstrap для стилизации элементов. Также научился работать с REST API из JavaScript. Кроме того, был изучен и применен на практике картографический сервис Mapbox.

## ЧАСТЬ 5. СПИСОК ЛИТЕРАТУРЫ

1. Документация Bootstrap: <https://getbootstrap.com/>
2. Документация Mapbox API: <https://www.mapbox.com/mapbox-gl-js>
3. Документация OpenWeatherMap API: <https://openweathermap.org/api>
4. Спецификация формата GeoJSON: <https://geojson.org/>