



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

## Отчет по лабораторной работе №7

Студент Вардумян Арсен Тигранович  
*фамилия, имя, отчество*  
Группа ИУ5-51Б

Студент 18.12.2021 Вардумян А.Т.  
*подпись, дата фамилия, и.о.*  
Преподаватель 18.12.2021 Гапанюк Ю.Е.  
*подпись, дата фамилия, и.о.*

## Описание задания:

1. Используя открытое апи, сверстать страницу погоды. Можно захардкодить Москву как изначальный город.
2. Нужно реализовать поиск
3. Вот наше предложение <https://openweathermap.org/current> в качестве апи, можете найти более удобное для себя, мы не против
4. Приложение должно быть адаптивным
5. Дизайн нужно придумать самому

## Текст программы:

“geo.js”:

```
export function getGeoLocation() {  
  return fetch('https://extreme-ip-lookup.com/json/')  
    .then(res => res.json())  
    .then(res => {  
      if (res.status === 'success'){  
        return res.city;  
      } else {  
        return 'London';  
      }  
    });  
}
```

“index.js”:

```
import {getWeatherByCity, getForecastByCity} from './weather.js'  
import {getSvg} from './svg_switcher.js'  
import {getGeoLocation} from './geo.js'  
import {min, max} from './minMax.js'  
  
const button = document.querySelector('.button');  
const input = document.querySelector('form__field');  
const days = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'];  
const days_short = {  
  'Sun.': 'Sunday',  
  'Mon.': 'Monday',  
  'Tue.': 'Tuesday',  
  'Wed.': 'Wednesday',  
  'Thu.': 'Thursday',  
  'Fri.': 'Friday',  
  'Sat.': 'Saturday',  
};  
  
window.addEventListener('load', loadDataFromApi);  
button.addEventListener('click', loadDataFromApi);  
document.addEventListener('keypress', loadDataFromApi);
```

```

window.addEventListener('resize', (e) => {
  const weekDay = document.querySelectorAll('.week-day');
  weekDay.forEach((el) => {
    if (window.innerWidth < 850) {
      el.innerHTML = el.innerHTML.slice(0, 3) + '!';
    } else if (el.innerHTML[3] === '.') {
      el.innerHTML = days_short[el.innerHTML];
    }
  });
});

async function loadDataFromApi(e) {
  if (input.value === '' && e.type !== 'load') return;

  if (e.type === 'load' || e.key === 'Enter' || e.type === 'click') {
    let data;
    const cityName = (e.type === 'load') ? await getGeoLocation() : input.value;

    data = getWeatherByCity(cityName);
    data.then(data => {
      document.querySelector('#temperature').innerHTML = Math.round(data.main.temp);
      document.querySelector('#celsius').innerHTML = '°C';
      document.querySelector('#city-name').innerHTML = data.name;

      const time = new Date((data.dt + data.timezone - 3*60*60) * 1000);

      let timeField = days[time.getDay()] + ' '
        + ((time.getHours() < 10) ? '0' : '') + time.getHours()
        + ':' + ((time.getMinutes() < 10) ? '0' : '') + time.getMinutes();

      document.querySelector('#time').innerHTML = timeField;
      document.querySelector('#description').innerHTML = data.weather[0].main;

      document.querySelector('#more-info :nth-child(1)').innerHTML = 'Cloudiness: ' + data.clouds.all + ' %';
      document.querySelector('#more-info :nth-child(2)').innerHTML = 'Humidity: ' + data.main.humidity + ' %';
      document.querySelector('#more-info :nth-child(3)').innerHTML = 'Wind: ' + data.wind.speed + ' m/sec';

      let img = document.createElement('img');
      img.src = getSvg(data.weather[0].icon);
      document.querySelector('.weather-info :nth-child(1)').innerHTML = '';
      document.querySelector('.weather-info :nth-child(1)').append(img);
    }).catch(err => alert(err));

    data = getForecastByCity(cityName, 'forecast');
    data.then(data => {
      const forecast = document.querySelector('.forecast');
      forecast.innerHTML = '';

      const arr = data.list;
      for (let i = 0; i < arr.length; i+=8) {
        let div = document.createElement('div');
        let div1 = document.createElement('div');

```

```

    let div2 = document.createElement('div');
    let div3 = document.createElement('div');
    let img = document.createElement('img');
    let br = document.createElement('br');

    br.classList.add('mini');
    img.src = getSvg(max(arr.slice(i, i + 7)).weather[0].icon);

    const time = new Date((arr[i].dt + data.city.timezone - 3*60*60) * 1000 );
    div1.append(days[time.getDay()]);
    div1.classList.add('week-day');
    div2.append(img);

    div3.append(Math.round(max(arr.slice(i, i + 7)).main.temp) + '°' + ' ');
    div3.append(br);
    div3.append(Math.round(min(arr.slice(i, i + 7)).main.temp) + '°');

    div.append(div1, div2, div3);
    forecast.append(div);
  }
}).catch(err => {});
}
}

```

“minMax.js”:

```

export const min = (arr) => {
  let min = arr[0];
  arr.forEach(el => {
    if (el.main.temp < min.main.temp) {
      min = el;
    }
  });
  return min;
};

export const max = (arr) => {
  let max = arr[0];
  arr.forEach(el => {
    if (el.main.temp > max.main.temp) {
      max = el;
    }
  });
  return max;
};

```

“svg\_switcher.js”:

```

export function getSvg(str) {
  switch (str.slice(0, 2)) {
    case '01':
      return '/lab7/img/wi-day-sunny.svg';
  }
}

```

```

    case '02':
      return '/lab7/img/wi-day-cloudy.svg';
    case '03':
      return '/lab7/img/wi-cloud.svg';
    case '04':
      return '/lab7/img/wi-cloudy.svg';
    case '09':
      return '/lab7/img/wi-rain.svg';
    case '10':
      return '/lab7/img/wi-rain-mix.svg';
      break;
    case '11':
      return '/lab7/img/wi-thunderstorm.svg';
      break;
    case '13':
      return '/lab7/img/wi-snow.svg';
      break;
    case '50':
      return '/lab7/img/wi-fog.svg';
      break;
    default:
      return '/lab7/img/wi-day-sunny.svg';
  }
}

```

“weather.js”:

```

async function getDataFromApi(params, option = 'weather') {
  const urlDefaultParams = {
    lang: 'en',
    units: 'metric',
    appid: '94bf8e9c966cf8e0a6a55e75b258f840',
  };

  const urlParams = Object.assign(params, urlDefaultParams);

  const url = new URL('https://api.openweathermap.org/data/2.5/${option}');
  Object.entries(urlParams).forEach(([key, value]) => {
    url.searchParams.append(key, value);
  });

  return fetch(url)
    .then(data => data.json())
    .then(data => {
      if(data.cod === '404') {
        throw Error(data.message);
      } else {
        return data;
      }
    })
  });
}

async function getWeatherByCity(city) {

```

```

    return getDataFromApi({q: city});
  }

  async function getForecastByCity(city) {
    return getDataFromApi({q: city}, 'forecast');
  }

  export {
    getWeatherByCity,
    getForecastByCity,
  };

```

“index.html”:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="css/style.css" />
    <title>Lab7</title>
  </head>

  <body>
    <nav>
      <div class="logo-container"></div>
      <div class="submit">
        <div class="form__group field">
          <input
            type="input"
            class="form__field"
            placeholder="City"
            name="city"
            id="city"
            required
          />
          <label for="city" class="form__label">City</label>
        </div>

        <div><a class="button">Search</a></div>
      </div>
    </nav>

    <div id="app">
      <div class="weather-container">
        <div class="weather">
          <div class="weather-info">
            <span></span>
            <span id="temperature"></span>
            <span id="celsius"></span>
            <span>
              <ul id="more-info">

```

```

        <li></li>
        <li></li>
        <li></li>
    </ul>
</span>
</div>
<div class="sys">
    <div id="city-name" style="font-size: 30px"></div>
    <div id="time"></div>
    <div id="description"></div>
</div>
</div>
<div class="forecast"></div>
</div>

<script type="module" src="js/index.js"></script>
</body>
</html>

```

“style.css”:

```

*{
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

:root {
    /* main colors */
    --backgroundColor: #181818;
    --lightText: #FFFFFF;
    --white: #fff;
}

a{
    text-decoration: none;
    color: var(--lightText);
}

html {
    width: 100%;
    height: 100%;
    margin: 0;
    overflow-x: hidden;
    overflow-y: hidden;
}

body {
    width: 100%;
    min-height: 100%;
    margin: 0;
}

```

```
overflow-x: hidden;
overflow-y: hidden;
font-family: 'Graphik';
font-style: normal;
font-weight: normal;
letter-spacing: 0.2px;
color: var(--lightText);
background-color: var(--backgroundColor);
}

nav {
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-items: center;
  height: 130px;
  padding: 0 30px;
}

.weather-container {
  margin: 30px 20% 0;
}

.submit {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  align-items: baseline;
}

#app {
  display: flex;
  flex-direction: column;
  justify-content: center;
  height: calc(100vh - 130px);
}

/*button css*/
a.button {
  display: inline-block;
  padding: 0.5em 3em;
  border: 0.16em solid #FFFFFF;
  margin: 0 0.3em 0.3em 0;
  box-sizing: border-box;
  text-decoration: none;
  text-transform: uppercase;
  font-weight: 400;
  color: #FFFFFF;
  text-align: center;
  transition: all 0.15s;
}

a.button:hover {
  color: #DDDDDD;
}
```



```
border-color:#DDDDDD;
cursor: pointer;
}
a.button:active {
color:#BBBBBB;
border-color:#BBBBBB;
}
@media all and (max-width:30em) {
a.button {
display:block;
margin:0.4em auto;
}
}

/*input style*/
.form__group {
position: relative;
padding: 15px 0 0;
margin-top: 10px;
width: 50%;
}

.form__field {
font-family: inherit;
width: 100%;
border: 0;
border-bottom: 2px solid var(--white);
outline: 0;
font-size: 1.3rem;
color: var(--white);
padding: 7px 0;
background: transparent;
transition: border-color 0.2s;
}

.form__field::placeholder {
color: transparent;
}

.form__field::placeholder-shown ~ .form__label {
font-size: 1.3rem;
cursor: text;
top: 20px;
}

.form__label {
position: absolute;
top: 0;
display: block;
transition: 0.2s;
font-size: 1rem;
}
```

```
.form__field:focus {
  padding-bottom: 6px;
  border-width: 3px;
  border-image-slice: 1;
}

.form__field:focus ~ .form__label {
  position: absolute;
  top: 0;
  display: block;
  transition: 0.2s;
  font-size: 1rem;
  font-weight: 700;
}

.form__field:required, .form__field:invalid{
  box-shadow: none;
}

.logo-container > img {
  margin-right: 10px;
  height: 70px;
  width: 70px;
}

.weather-info{
  display: flex;
  flex-direction: row;
}

.weather-info > span {
  margin-right: 10px;
}

.weather-info > span > img {
  height: 50px;
  width: 50px;
}

ul {
  display: inline-block;
  list-style-type: none;
  font-size: 15px;
}

.weather {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}

.sys > div {
  text-align: right;
```

```
}

.forecast > div {
  height: 200px;
  width: 18%;
}

.forecast {
  margin-top: 100px;
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}

.forecast > div {
  display: flex;
  flex-direction: column;
  text-align: center;
}

.mini {
  display: none;
}

span#temperature{
  font-size: 50px
}

.weather-info > span {
  font-size: 30px
}

@media screen and (max-width:700px){
  a.button {
    display: block;
    width: 100px;
    padding: 0.5em 1em;
  }

  ul {
    display: none;
  }

  .mini {
    display: block;
  }

  span#temperature {
    font-size: 40px;
  }

  .weather-info > span{
    font-size: 25px
  }
}
```

```

}

.weather-container {
  margin: 30px 15% 0;
}
}

@media screen and (max-width:375px){
  .submit, .weather {
    flex-direction: column;
    align-items: flex-end;
  }

  nav {
    padding: 0 7%;
  }

  .form__group.field{
    width: 60%;
    margin: 30px 0 15px;
  }
}

```

## Экранные формы с примерами выполнения программы:



