

# SAVEBLACKSEA



**НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ**  
**ФАКУЛТЕТ ЗА БАЗОВО ОБРАЗОВАНИЕ**  
**Програма „ИНФОРМАЦИОННИ ТЕХНОЛОГИИ“**

**Курсов проект**  
**CITB558 Web-програмиране с HTML и JavaScript**

**SaveBlackSea.org**

**Разработил:**

Симона Цветомирова Енева  
F86955

**Ръководител:**

ас. Д-р Делян Керемедчиев

## Описание на проекта

---

Проектът представлява уебсайт, чиято основна цел е да служи за визитка на неправителствена организация, целяща опазването на водите на Черно море и отточните му реки.

Основната страница на проекта и най-вече header частта служат за “корица” на идеята на проекта. В нея се съдържа името и синтезирано описание на целите ѝ. Целта на header-а в проекта ми е да впечатли с набор от CSS анимация и да насочи цялото внимание на потребителя към най-основното послание в страницата. Навигацията е застъпена над него. Header-а е последван от main отдел, разбит на section-и, където идейно се разделят: по-подробно описание на организацията, информация за прогреса, който е постигнат до тук (събрани подписи) и важни за движението лица. В края на страницата е поставен и footer.

Втората уеб-страница от проекта е предназначена за регистрация, тя има несложната структура от nav и main секция, в която е разположено и цялото съдържание на страницата. Основен компонент е формата за регистрация.

## Реализация

---

Проектът е разработен с HTML5, CSS3, JavaScript. В директорията на index.html и signup.html страницата има още три директории - /img (за изображенията), /js (за скриптовете), /styles (за CSS файловете). В HTML частта на проекта се следва правилна семантика а описване на уеб страници, но основната и цел е да представи съдържанието му. Основно внимание е отделено на стилизацията и CSS частта на проекта. В нея са използвани различни техники за контролиране на елементите на HTML: keyframes за CSS анимация, media queries за допълнително

стилизиране при различни екрани, pseudo classes и pseudo elements, за допълнително стилизиране на елементите при интеракция (например при hover върху някой елемент, или при въвеждане от клавиатура). За постигане на отзивчив дизайн е използван flex модела. Чрез media queries той променя посоката си (напр. От flex-direction: row към flex-direction: column) и осигурява плавна транзиция между устройствата. Keyframes са контролери за CSS анимация, на които се задава стойности за различни стилизиращи атрибути при стойности от 0% до 100%, при създаване на keyframe се определя и неговото име, след което то се задава на атрибут animation: заедно с време за изпълнение.

В .js файловете на проекта са разработени две допълнителни функционалности - запълване на индикатора за прогрес и движение на компоненти спрямо мишката на потребителя (при десктоп) и една за проверка на входа в страницата за регистрация. Използвани са концепции въведени с ES6. При първата се симулира държание на сървър, който дава информация за прогреса. Подобна функционалност е зададена с callback функция и hardcoded стойност от 15%, която се увеличава до 100% при натискане на update бутона. Вградената функция setInterval определя времето за изпълнение на тялото на функцията, чрез задаване на втори параметър:

```
setInterval( (аргумент) => {  
    ...ТЯЛО...  
}, време);
```

Втората функционалност следи координатите на мишката по X и Y и предава тези координати като аргументи на CSS атрибути, за динамично обновяване на стилизацията на елементите. Чрез querySelectorAll(".square") се достъпва NodeList от всички елементи с клас square, които след това се достъпват с forEach, за да бъдат обновени.

За проверка на входа се използват регулярни изрази и динамично се зареждат съответно съобщение за грешка или

потвърждение. Една от проверките (за приети правила за ползване) провокира появата на прозорец със съобщение за попълване на нужната информация.

С този проект целя да демонстрирам умения за работа с DOM и задълбочено разбиране на CSS принципи без използване на допълнителни инструменти (фреймуърци).

## ИЗТОЧНИЦИ

---

1. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
2. <https://css-tricks.com/blobs/>
3. <https://css-tricks.com/animate-a-container-on-mouse-over-using-perspective-and-transform/>
4. <https://developer.mozilla.org/en-US/>
5. <https://www.w3schools.com/default.asp>
6. <https://www.pexels.com/>
7. <https://undraw.co/>