



F1 MAP DIARY – 2025

Уеб приложение базирано на HTML, JavaScript и CSS
Дисциплина : Практическо обучение и работа с Java script

Изготвил : Христина Христова

Специалност : ГПСТ

Курс : 4-ти

ФКН : 5GG0800015

ДАТА : 21.05.2025

Съдържание

Въведение 2

.html 2

.css 3

.js 5

Въведение

Проектът F1 Map Diary 2025 представлява информативно интерактивно уеб приложение, което визуализира световните Формула 1 писти, състезатели, отбори и резултати от състезания чрез съвременен потребителски интерфейс и карта, реализирана с помощта на библиотеката Leaflet.js. Приложението е изградено изцяло чрез HTML, CSS и JavaScript, без използване на сървърна страна или база данни, като всички данни са дефинирани в самия JavaScript код. Използваните данни са взети от официалния сайт на Формула 1, който присъства и в уеб страницата като връзка.

.html

HTML файлът дефинира основната структура на интерфейса. Той започва с включване на необходимите ресурси в `<head>`, включително Leaflet стиловете и основния CSS файл на приложението.

В `<body>` е изграден визуалният скелет на страницата, разделен на няколко ключови секции. В горната част на екрана се намира хедърът. Той съдържа логото на F1, заглавие на приложението, година на сезона и навигационни линкове, както и автоматично пренасочване към официалния сайт на Формула 1. Навигационното меню е проектирано така, че да бъде видимо при по-големи екрани, с цел осигуряване на отзивчив дизайн.

Следва основната част на съдържанието, която се състои от три паралелно разположени панели: панел за състезатели, панел за карта и панел за писти. Панелът за състезатели показва списък с водещи състезатели за сезона, генериран динамично чрез JavaScript. Централно разположена е интерактивната карта, в която се зареждат всички писти, базирани на тяхната географска ширина и дължина.

Третият панел съдържа списък с писти, който включва име, локация и търсачка за филтриране. Освен това, в панела на пистите се добавя и панел, който служи за връзка между js и html файла и показва най-близката писта до която се намира потребителя. Следва структурата на панела с отбори във формата на секция от потребителския интерфейс. Контейнерът с клас `teams-panel` съдържа заглавие „F1 TEAMS 2025“ и празен елемент `teams-grid`, в който JavaScript по-късно динамично зарежда всички отбори. Този панел е част от

основната структура на страницата и се използва за визуално подреждане на отборите в мрежа.

След картата и панелите, HTML-ът дефинира още една важна секция – тази за класиране и резултати от състезанията. Визуално тя е разделена на два блока: единият показва класирането на отборите за сезона, а другият – списък с последните състезания, заедно с победител, отбор, време и дата.

В края на документа се включва `script.js`, който осигурява цялата функционалност на приложението.

.CSS

CSS файлът отговаря за цялостното визуално оформление на проекта. Той започва с дефиниране на два шрифта – Formula1 Regular и Formula1 Bold, които се зареждат директно от официалния сайт на Формула 1. Те придават на страницата визуална идентичност, сходна с тази на реалния F1 бранд. В корена на документа са дефинирани CSS променливи, които се използват навсякъде в стиловете – като основни цветове за черно, червено, сиво и светло сиво. Цялото тяло на страницата използва черен фон и светъл текст, придавайки модерен и контрастен външен вид. Класът `.app`, който обгражда цялото приложение, е дефиниран като флекс контейнер с вертикална подредба. Това гарантира, че съдържанието се подрежда отгоре надолу и винаги запълва поне височината на екрана. Хедърът е стилизиран така, че съдържа логото и заглавието вляво, а навигацията и индикаторът за сезона вдясно. Елементите са подравнени чрез Flexbox, така че да изглеждат балансирано на различни резолюции. (`justify-content: space-between;` е CSS свойство, което се използва във Flexbox или Grid контекст, за да разпредели свободното пространство между елементите по основната ос (обикновено хоризонтално)). Навигационните линкове променят цвета си при задържане с мишка, за по-интуитивно взаимодействие. (Flexbox е система за едноизмерно подреждане на елементи – или по ред, или по колона. Основната идея е да се създаде контейнер (flex контейнер), който управлява подредбата на своите елементи (flex елементи) автоматично, независимо от техния брой или размер.)

Основните панели са обособени с рамки, заоблени ъгли и вътрешни отстояния. Картата, изградена с Leaflet, е разположена в отделен контейнер с фиксирана височина, за да заема

централно място. Маркерите на картата използват персонализирана икона, дефинирана в JavaScript, а стиловете на рорир прозорците са променени така, че да следват цветовата палитра на сайта. Панелите със състезателите и пистите използват карти и списъчни елементи със стилове за заобляне, сенки и плавни преходи при задържане. Панелът, показващ пистите е обособен с търсачка, която за да се центрира перфектно, се използва `transform: translateY(-50%)`, което премества иконата нагоре с половината ѝ височина, компенсирайки позицията `top: 50%`. При клик върху даден състезател се отваря модален прозорец с разширена информация – снимка, статистика и биография. Модалните прозорци са напълно стилизирани със затъмнен фон и центрирано съдържание, поддържащи визуално последователна естетика. Всеки отделен състезател е визуализиран чрез `.team-driver-card`. Целият текст и съдържание са центрирани хоризонтално. Снимката на състезателя вътре в картата се стилизира така, че да заема цялата ширина на контейнера, с фиксирана височина от 150 пиксела. Изображението използва `object-fit: cover`, което гарантира, че няма да се разтегне или изкриви – при нужда се изрязва, за да запази пропорции.

В секцията за класиране са използвани прогрес барове, които визуализират точките на отборите като съотношение в хоризонтален мащаб. Всеки отбор е представен чрез лого, име, брой точки и визуален индикатор. Състезанията са представени чрез екрани с детайли като кръг, дата, победител и времетраене, оформени със засенчване и интерактивен ефект при преминаване с курсора.

В стила е добавена и функционалност за плавно преминаване през страницата с `scroll-behavior: smooth`, кореспондиращо с `html` файла. Освен това, в този файл се задава стил и на функционалност свързана с определяне на най-близката писта, спрямо локацията на потребителя в същия стил като на самото приложение.

Важно е да се отбележе, че в самия код има изграден стил, както за устройства с голям екран (лаптоп и таблет) , така и за мобилна версия (разбира се при инициализиране на локален хост). Всичко това става чрез реализиране на отговарящ стил, веднъж реализиран за устройства с голям екран, а в края на кода за мобилна версия.

.js

В началото на файла са дефинирани четири основни масива от данни: `circuits`, `drivers`, `teams` и `raceResults`. Масивът `circuits` съдържа подробна информация за всяка писта – като име, локация, координати, дължина, първо състезание, брой обиколки и изображение. Данните са структурирани в обекти и в някои случаи включват непълна информация, която при нужда се обработва чрез условна логика. Масивът `drivers` представя пилотите, като за всеки от тях се описват ключови атрибути като име, отбор, стартов номер, държава, снимка, брой подиуми, точки и кратка биография. Отборите се намират в масива `teams`, като притежават име и лого. Последният масив, `raceResults`, съдържа детайли за проведените състезания – включително кръг, име, дата, победител, отбор, брой обиколки и време.

След дефиницията на данните започва инициализацията на Leaflet картата. Картата се зарежда в контейнера с ID `map`, със стартова гледна точка върху целия свят, и използва сателитен слой, предоставен от ArcGIS. За визуално разграничаване се създава персонализирана иконка за маркерите, която заменя стандартната. С помощта на `forEach` функция се обхожда масивът `circuits` и за всяка писта се добавя маркер върху картата. При клик върху маркер се отваря роруп с подробна информация: име на пистата, местоположение, дължина, първо състезание, брой обиколки и изображение. Ако има налично изображение, то се визуализира.

След визуализирането на пистите се пристъпва към генериране на списъка с пилоти. Секцията `.drivers-list` се запълва динамично с HTML елементи, всеки от които съдържа стартовия номер на пилота, неговото име и отбор. Когато потребителят кликне върху някой от тези елементи, се отваря модален прозорец, който показва разширена информация за съответния пилот, включително снимка, биография, точкова статистика и други детайли. Модалът може да бъде затворен както чрез бутон, така и чрез клик върху фона.

Следващата логика е посветена на филтрирането и визуализирането на списъка с писти. В полето за търсене потребителят може да въведе текст, а функцията `renderCircuits` ще филтрира пистите според въведеното – като проверява дали търсеният низ се съдържа в името или локацията на пистата. Всеки резултат се визуализира като интерактивен елемент, който при клик центрира картата върху съответната писта с помощта на метода `map.setView`.

Модалните прозорци се управляват централизирано – всеки бутон за затваряне, както и клик върху фона, води до премахване на класа `hidden` от модалния елемент. Това се постига чрез обхождане на всички `.modal` елементи и добавяне на слушатели за събитие.

Отделен сегмент от кода се занимава с визуализацията на отборите и тяхното класиране. Този сегмент от JavaScript логиката се грижи за визуализирането на информацията за отборите в две отделни секции на потребителския интерфейс: мрежа с отбори и класиране според натрупаните точки. В началото на процеса се избира контейнерът, в който ще се визуализират отборите, като се таргетира HTML елемент с клас `.teams-grid`. В този контейнер се поставя HTML съдържание, съставено чрез обхождане на масива с отбори. За всеки отбор се създава малка карта, която съдържа логото и името му. Картите се подреждат в мрежа, благодарение на CSS Grid, като целта е да се осигури ясна и компактна визуализация на всички участващи отбори. След това се пристъпва към изграждането на секцията с класирането. Първо се избира контейнерът за класирането чрез `.standings-grid`, след което отборите се сортират по точки в низходящ ред. Това осигурява коректно подреждане, където лидерът се появява най-отгоре. Сортирането се извършва чрез създаване на копие на оригиналния масив, така че данните да останат непроменени. За всеки отбор в сортирания списък се генерира HTML блок, който включва текущата му позиция в класирането, логото и името. В допълнение, всяка позиция съдържа хоризонтална лента (прогрес бар), която визуализира съотношението на точките спрямо лидера. Широчината на тази лента се изчислява като процент от точките на отбора с най-висок резултат. Например, ако водещият отбор има 200 точки, а друг – 100, неговата лента ще заема 50% от пълната ширина. Това осигурява бърза визуална представа за относителната разлика в представянето между участниците. Целият този процес е напълно динамичен и не изисква ръчно писане на HTML код. Всичко се генерира чрез JavaScript, на база предварително дефинираните данни за отборите. След това скриптът преминава към състезателните резултати. В самото начало се избира HTML контейнерът, който ще съдържа визуално резултатите. Това е елемент с клас `.results-grid`, предназначен да подреди отделните състезания в логическа и визуално разграничена последователност. След това се използва масивът с резултати, наречен `raceResults`, който съдържа информация за всяко състезание, включително номер на кръга, име, дата, победител, кола, обиколки и време. С помощта на метод за обхождане на масива, за всяко състезание се генерира HTML блок, който представя данните във визуално

подредена карта. Всяка такава карта съдържа две основни части. Първата е горната информационна секция, където са показани кръгът от шампионата, името на състезанието и датата, на която е проведено. Това служи като заглавие и контекст за събитието. Втората част е разделена на две колони – в лявата се визуализира информацията за победителя, включително неговото име и отборната кола, а в дясната се показва статистика за броя на обиколките и финалното време на състезанието.

Що се отнася до навигацията, в горната част на сайта има меню, което включва връзки към основните секции: писти, пилоти, отбори и класиране. В основата си логиката започва с избиране на всички връзки (елементи `<a>`) вътре в контейнера с клас `.header-nav`. За всяка от тези връзки се добавя слушател за събитие при клик. Когато потребителят натисне някой от тези линкове, браузърът по подразбиране би се опитал директно да скочи към съответния елемент на страницата, чийто идентификатор е зададен в атрибута `href`. Това поведение се предотвратява умишлено чрез извикване на `preventDefault`, с което се цели да се избегне резкия скок и вместо това да се активира плавно преминаване. След това от атрибута `href` на натиснатия линк се извлича стойността на целевата секция, като се премахва символът „#“. Например, ако връзката сочи към `#teams`, резултатът ще бъде просто `teams`. Въз основа на това извлечено име се търси DOM елемент със съответен клас – като например `.teams-panel` или `.teams-section`. Това търсене осигурява гъвкавост, защото структурата на HTML може да използва различни наименования за панелите и секциите, без кодът да зависи от конкретен ID. Ако бъде открит такъв елемент в DOM дървото, се извиква методът `scrollIntoView`, който скролира страницата до съответната секция. Специфицира се поведение „smooth“, което казва на брауъра да анимира прехода, и се задава блок „start“, което означава, че скролираната секция трябва да се подравни в горната част на изгледа. Така потребителят получава по-плавен, модерен и интуитивен начин за придвижване между основните части на сайта, без резки движения и без презареждане или загуба на контекст.

В последната част от кода се добавя функционалност, която взема локацията на потребителя и на база координатите му намира най-близката писта от представените. Функцията `findNearestCircuitFromGPS()` има за цел да определи коя е най-близката F1 писта до текущото местоположение на потребителя, използвайки GPS координатите, които браузърът му предоставя. Тази функция първо проверява дали устройството поддържа геолокация чрез

обекта `navigator.geolocation`. Ако това е възможно, тя извиква метода `getCurrentPosition()`, който предоставя географските координати на потребителя – ширина и дължина. След като се получи позицията, функцията преминава през списък от обекти (`circuits`), които съдържат координатите и информацията за всяка налична писта. За всяка от тях се изчислява разстоянието до потребителя с помощта на формулата на Haversine – математически метод за прецизно измерване на дистанция между две точки върху повърхността на Земята, като се отчита нейната сферична форма. Така се намира пистата с най-малко изчислено разстояние, която се определя като „най-близка“. След като тази писта бъде установена, се визуализира върху картата с помощта на Leaflet. Потребителят вижда изскачащ прозорец (popup), който показва коя е най-близката писта и на какво разстояние се намира. Освен това, на картата се добавя иконка за местоположението на потребителя, а камерата на картата плавно се придвижва първо до текущата му позиция и след това до пистата. Ако в HTML има списък с писти (елементи с клас `.circuit-item`), съответният елемент за най-близката писта се оцветява и автоматично се скролира в изгледа. Функцията разчита на предварително дефинирани обекти с информация за пистите, на правилно инициализирана Leaflet карта (`map`), както и на потребителското разрешение за достъп до местоположението.