ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО АУДИО-, ВИДЕО-

И ТЕЛЕКОМУНИКАЦИЯ “А.С.ПОПОВ”

**ДИПЛОМЕН ПРОЕКТ**

Разработка на риспонсив уеб дизайн на сайт за електроннен магазин

Ученик: Мартин Даниелов Василев

Клас: 12Е

Специалност: системно програмиране

Ръководител: инж. Боян Мак Михайлов

София, 2023 г.

Съдържание

[Въведение 3](#_Toc133884232)

[Теоретично проучване 4](#_Toc133884234)

[Връзка между HTML и CSS 11](#_Toc133884235)

[Как да свържем CSS към HTML 11](#_Toc133884236)

[Атрибутът ***rel*** 12](#_Toc133884237)

[Атрибутът ***type*** 12](#_Toc133884238)

[Атрибутът ***href*** 12](#_Toc133884239)

[Библиотеката jQuery и риспонсив дизайн 13](#_Toc133884240)

[Практическа реализация 15](#_Toc133884241)

[Избор на риспонсив фреймуърк 15](#_Toc133884242)

[Избор на back-end софтуер за електронна търговия 15](#_Toc133884243)

[Страница за начини на доставка 23](#_Toc133884244)

[Създаване на страница за електронна количка 24](#_Toc133884245)

[Създаване на страница за списък с продукти 24](#_Toc133884246)

[Създаване на детайлизирана страница за продукт 24](#_Toc133884247)

[Заключение 25](#_Toc133884248)

[Използвани източници: 26](#_Toc133884249)

[Кирилица: 26](#_Toc133884250)

[Латиница: 26](#_Toc133884251)

# Въведение

Разработването на уеб сайтове, които изглеждат добре и работят добре на всички устройства, е от съществено значение за постигането на успех в днешната интернет среда. Риспонсив уеб дизайн е техника, която позволява на уеб сайтовете да се адаптират към различни размери на екрана, като запазват функционалността и удобството за потребителите.

Според web.dev: " Отзивчивият уеб дизайн, първоначално дефиниран от [Итън Маркот в A List Apart](http://alistapart.com/article/responsive-web-design/), отговаря на нуждите на потребителите и устройствата, които използват. Оформлението се променя въз основа на размера и възможностите на устройството. Например на телефон потребителите ще виждат съдържание, показано в изглед на една колона; Един таблет може да показва едно и също съдържание в две колони."[[1]](#footnote-1)

# Съществуват различни устройства с различни размери на екрана, като телефони, таблети, компютри и телевизори. За да бъде успешен, вашият уебсайт трябва да се адаптира към всеки от тези размери на екрана, както и към различните начини за взаимодействие с устройствата. Риспонсивният дизайн е създаден, за да включва всички тези фактори и да осигури оптимално потребителско изживяване за всички посетители.

# Теоретично проучване

Разработката на уеб сайтове е важна и динамична сфера, която постоянно се развива. Един от най-важните аспекти на един уеб сайт е дизайнът му. Днес много хора предпочитат да пазаруват онлайн, затова е много важно да имаме електронен магазин с риспонсив дизайн, който да се приспособява към различните устройства и екрани.

Респонсивният уеб дизайн е техника за уеб разработка, която има за цел да осигури оптимално преживяване на потребителя, като сайтът се адаптира автоматично към различните размери на екраните на устройствата, с които се достъпва. Това включва телефони, таблети, настолни компютри, телевизори и други.

От статията "Guidelines for Responsive Web Design" [[2]](#footnote-2)на Smashing Magazine можем да научим, че респонсивният уеб дизайн е изграден въз основа на гъвкава решетка, която може да се променя в зависимост от размера на екрана. Този подход позволява на дизайнера да създаде уеб сайт, който да изглежда добре на всяко устройство, независимо от размера му. В статията също така се подчертава важността на създаването на една основна версия на сайта и използването на CSS медийни заявки, за да се променя изгледът на сайта в зависимост от размера на екрана.

Практиката на респонсивен уеб дизайн се основава на гъвкави решетки, изображения и използване на CSS медийни заявки, което позволява уебсайта да се адаптира автоматично към различните устройства и предпочитания на потребителите. Така, когато потребителят се превключва от лаптоп към iPad или друго устройство, уебсайтът автоматично се приспособява към резолюцията, размера на изображенията и настройките на устройството. Това означава, че не е нужно да се проектира и разработва уебсайт за всяко ново устройство на пазара.

**Софтуер за създаване на риспонсив front-end дизайн**

Софтуерът за създаване на риспонсив front-end дизайн се използва за създаване на уеб дизайн, който да бъде оптимизиран за различни размери на екраните на устройствата. Това означава, че уеб сайтът ще се адаптира автоматично към различните устройства, като мобилни телефони, таблети и десктоп компютри, за да осигури оптимално преживяване на потребителите. Софтуерът за създаване на риспонсив front-end дизайн може да бъде използван от уеб дизайнери, разработчици на софтуер и всички, които желаят да създадат уеб сайт с оптимизиран за мулти-устройство дизайн.

В следната статия[[3]](#footnote-3) се представят десет от най-популярните фреймуърки за създаване на респонсивен уеб дизайн. Респонсивният уеб дизайн е важен, защото позволява на уеб сайтовете да се приспособят към различни устройства и екрани, като таблети, смартфони и настолни компютри. Избирането на правилния фреймуърк може да улесни процеса на създаване на респонсивен дизайн и да помогне на уеб разработчиците да се фокусират върху други аспекти на проекта.

Pure CSS: Безплатен и отворен CSS фреймуърк, който предлага колекция от малки CSS компоненти, които могат да се използват във всякакъв уеб проект. Той е популярен избор сред фронт-енд програмистите поради своя подход на мобилно-първо и минималистичен дизайн.

Skeleton: Още един безплатен и отворен CSS фреймуърк, който предоставя лека и адаптивна рамка, подходяща за разработка на MVP или продуктова разработка. Той е перфектен за създаване на уеб приложения за различни платформи, без да се жертва качеството.

Montage: Безплатен и отворен JavaScript фреймуърк, високо предпочитан от разработчиците. Той позволява на едностраничните приложения да използват HTML фреймуърк с уникални елементи като многократно използваеми компоненти и декларации на данни.

Material UI: Безплатен и отворен React UI фреймуърк, който включва колекция от React елементи, подходящи за Google Simple Design. Той може да се използва за уеб приложения, които изискват значителна функционалност на фронт-енда, поради разнообразните му нива и детайлност.

Bootstrap: Един от най-надеждните фреймуърки за бързо създаване на уеб сайтове. Той предлага отлични структурни елементи, включително CSS решетки, създавайки прост механизъм за навигация за онлайн проектите. Той е полезен за начинаещи фронт-енд разработчици и предприемачи.

Semantic UI: Безплатен и отворен CSS фреймуърк, базиран на LESS и jQuery, който включва правила за дизайн от трети страни. Поддържа широка гама от елементи на потребителския интерфейс като зареждачи, навигационни елементи, бутони и работи добре със сложни елементи като изскачащи прозорци. Предоставя значително повече възможности за персонализация в сравнение с Bootstrap.

Foundation: Безплатен и отворен фреймуърк с UI компоненти, който помага за създаването на уеб приложения, мобилни приложения, уеб сайтове, шаблони и теми за електронна поща. Това е основна, лека, но лесна за приложение архитектура, от която могат да се възползват стартиращи компании и нови разработчици.

Simple Grid: Безплатен и отворен CSS фреймуърк, който създава уеб проекти с основен дизайн и архитектура и потребителски интерфейс, удобен за потребителите. Simple Grid е полезен за разработчици, търсещи основна CSS мрежа за всяко уеб развитие.

Gumby: Безплатен и отворен CSS фреймуърк, базиран на SaaS, който притежава различни възможности като персонализируема мрежа, адаптивен уеб дизайн, превключватели, много добре структуриран потребителски интерфейс и много други.

Cascade: Безплатен и отворен CSS фреймуърк, който генерира както семантични, така и несемантични мрежови шаблони. Той опростява процеса на разработка, като приема общ подход и има компоненти за навигация, дизайн на таблица, основни шаблони и други възможности.

В заключение, всеки от тези уеб фреймуърци има своите особености и предимства. Използването на тези добре познати уеб разработвателски структури се препоръчва, тъй като може да опрости процеса на разработка и да доведе до по-добър краен продукт. Изборът на подходящия фреймуърк зависи от конкретните изисквания и нужди на проекта, както и от уменията и предпочитанията на програмиста. Въпреки това, всички тези фреймуърки предоставят готови решения, които могат да бъдат използвани за бързо и ефективно разработване на уеб приложения, което ги прави подходящи за начинаещи програмисти, стартиращи компании и професионални разработчици.

**Методи за създаване на риспонсив CSS**

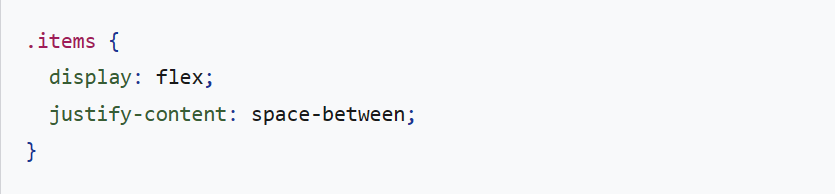
Създаването на риспонсивен CSS е важна част от уеб дизайна, тъй като много потребители достъпват уеб сайтове от мобилни устройства. Различните методи за създаване на риспонсивен CSS включват използване на медийни заявки - [Медийните заявки](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries)[[4]](#footnote-4) ни позволяват да изпълним серия от тестове (например дали екранът на потребителя е по-голям от определена ширина или определена резолюция) и да приложим CSS избирателно, за да оформим страницата по подходящ начин за нуждите на потребителя.

Например, следните медийни заявки тестват, за да видят дали текущата уеб страница се показва като екранен носител (следователно не е отпечатан документ) и изгледът е поне широк. CSS за селектора ще се прилага само ако тези две неща са верни.*80rem.container*



Можете да промените цялото оформление или части от него, за да отговарят на различните размери на екрана, като добавите множество медийни заявки в стиловия лист. Тези медийни заявки определят точките на прекъсване, където се извършват промените в оформлението. Често използван подход е да започнете с просто оформление с една колона за мобилни устройства и да добавите още колони за по-широки екрани. Това се нарича мобилен първи дизайн. Най-добрите практики при дефинирането на точките на прекъсване на медийните заявки насърчават използването на относителни единици, а не на абсолютни размери на устройствата.

Друг метод за създаване на риспонсив css влючва използването на Flexbox. Този метод на оформление е идеален, когато имате набор от елементи с различни размери и бихте искали те да се поберат удобно в ред или редове, като по-малките елементи заемат по-малко място, а по-големите получават повече пространство.



Използването на Css grid [[5]](#footnote-5) e още един метод за създаване на риспонсив css. CSS Grid е съвременен метод за организиране на елементи в уеб дизайна, който осигурява гъвкава и мощна система за позициониране и разпределение на съдържанието на уеб страницата. Чрез използването на CSS Grid можете да определите редове и колони, които се използват за подреждане на съдържанието на страницата.

CSS Grid позволява на дизайнерите да задават съотношения между елементите и да определят какво да се случва, когато съдържанието на страницата се променя или когато размерът на прозореца на браузъра се променя. Можете да задавате размери на колони и редове, да ги комбинирате, да задавате различни ширини и височини, както и да задавате празни пространства между елементите.

CSS Grid се състои от контейнер, който задава външните граници на мрежата, и от елементите, които трябва да се разположат в нея. Можете да използвате CSS свойства за позициониране, за да определите как трябва да се разположат елементите в мрежата.

CSS Grid е гъвкав и мощен метод за управление на разположението на елементите в уеб дизайна. Предоставя голям контрол и много опции за позициониране и разпределяне на елементите на уеб страницата, което го прави изключително полезен инструмент за дизайнерите и разработчиците на уеб сайтове.

## Връзка между HTML и CSS

Връзката между HTML и CSS е много важна за уеб дизайна. HTML езикът се използва за създаване на структурата на уеб страницата - определяне на съдържанието и разположението на елементите на страницата. С други думи, HTML се използва за създаване на съдържанието на уеб страницата.

CSS се използва за стилизиране на този HTML код и управление на външния вид на уеб страницата - цветове, шрифтове, размери, маргини, позициониране и т.н. С други думи, CSS се използва за оформяне на съдържанието на уеб страницата.

За да направим стайлинга, който имплементираме със CSS, отразен в HTML, трябва да намерим начин да свържем CSS с HTML. Можем да направим свързването, като напишем вграден CSS, вътрешен CSS или външен CSS.

Най-добрата практика е да се държи CSS отделно от HTML, така че тази статия[[6]](#footnote-6)[[7]](#footnote-7) се фокусира върху това как можем да свържем този външен CSS към HTML.

## Как да свържем CSS към HTML

За да свържем CSS към HTML, трябва да използваме маркера за връзка с някои подходящи атрибути.

Тагът за връзка е самозатварящ се таг, който трябва да поставим в главната част на нашия HTML файл.

По-долу е показан пример за свързване на css към html файл:



За да свържем CSS с HTML, можем да използвате тага за връзка, който има атрибути, като: “rel”, “type” и “href”.

### Атрибутът ***rel***

***rel*** е релацията между външния файл и текущия файл. За CSS използвате например stylesheetrel="stylesheet"

### Атрибутът ***type***

***type*** е типът на документа, който свързваме към HTML. За CSS, това е например text/csstype="text/css"

### Атрибутът ***href***

***href*** означава "хипертекстова препратка". Можем да го използваме, за да укажем местоположението на CSS файла и името на файла. Това е връзка, върху която може да се кликне, така че можем също да я задържим и да щракнем върху нея, за да видим CSS файла .CTRL

Свързване ... [[8]](#footnote-8)

https://www.freecodecamp.org/news/how-to-link-css-to-html/

## Библиотеката jQuery и риспонсив дизайн

https://jquery.com/

Официалната дефиниция на jQuery откриваме в техния сайт:

„jQuery е бърза, малка и богата на функции JavaScript библиотека. Той прави неща като обхождане и манипулиране на HTML документи, обработка на събития, анимация и Ajax много по-лесни с лесен за използване API, който работи в множество браузъри. С комбинация от гъвкавост и разширяемост, jQuery промени начина, по който милиони хора пишат JavaScript.“[[9]](#footnote-9)

“jQuery[[10]](#footnote-10) може да се използва за намиране на конкретен HTML елемент в HTML документа с определен ID, клас или атрибут и по-късно можем да използваме jQuery, за да променим един или повече атрибути на същия елемент като цвят, видимост и т.н. JQuery може да се използва и за да направи уеб страница интерактивна, като отговори на събитие като кликване с мишката.

JQuery е безплатен софтуер с отворен код, който попада под разрешителния лиценз MIT. Към април 2021 г. jQuery се използва от 77,8% от първите 10 милиона най-популярни уебсайтове.

jQuery е разработен със следните принципи:

* **Разделяне на JavaScript и HTML,** което насърчава разработчиците напълно да отделят JavaScript кода от HTML маркирането.
* **Краткостта и яснотата** насърчават функции като **верижни** функции и стенографски имена на функции.
* **Елиминира несъвместимостите между браузърите**, така че разработчиците не трябва да се притесняват за съвместимостта на браузъра, докато пишат код, използвайки библиотеката jQuery.
* **Разширяемост**, което означава, че нови събития, елементи и методи могат лесно да се добавят в библиотеката на jQuery и след това да се използват повторно като плъгин.”

<https://demos.jquerymobile.com/1.4.5/rwd/>

# Практическа реализация

https://colorlib.com/wp/free-bootstrap-ecommerce-website-templates/

## Избор на риспонсив фреймуърк

Изборът на риспонсивен фреймуърк е важен елемент при проектирането на уеб сайтове, тъй като той позволява лесното създаване на уеб страници, които се адаптират към различни устройства и размери на екрана. Затова, след като проучих различните риспонсивни фреймуърци, аз реших да избера Bootstrap поради следните причини:

Лесна за употреба: Bootstrap е известен със своята лесна за употреба и интуитивна документация. Това прави възможно за всеки разработчик, независимо от нивото на опит, да създаде красиви и функционални уеб страници.

Голям брой компоненти: Bootstrap предлага голям брой готови компоненти, като например менюта, формуляри, бутони и много други, които могат да бъдат използвани за бързо създаване на уеб страници.

Актуална технология: Bootstrap е базиран на HTML, CSS и JavaScript, които са актуални технологии и са добре поддържани от браузърите. Това гарантира, че уеб сайтът ще работи добре и ще бъде съвместим с различни браузъри и устройства.

Разширяемост: Bootstrap е разширяем и може да бъде персонализиран с помощта на CSS и JavaScript. Това позволява на разработчиците да създават уникални дизайни, които отговарят на конкретните нужди на проекта.

Сумирайки, избрах Bootstrap като риспонсивен фреймуърк поради неговата лесна употреба, голям брой компоненти, актуална технология и възможност за разширяемост.

## Избор на back-end софтуер за електронна търговия

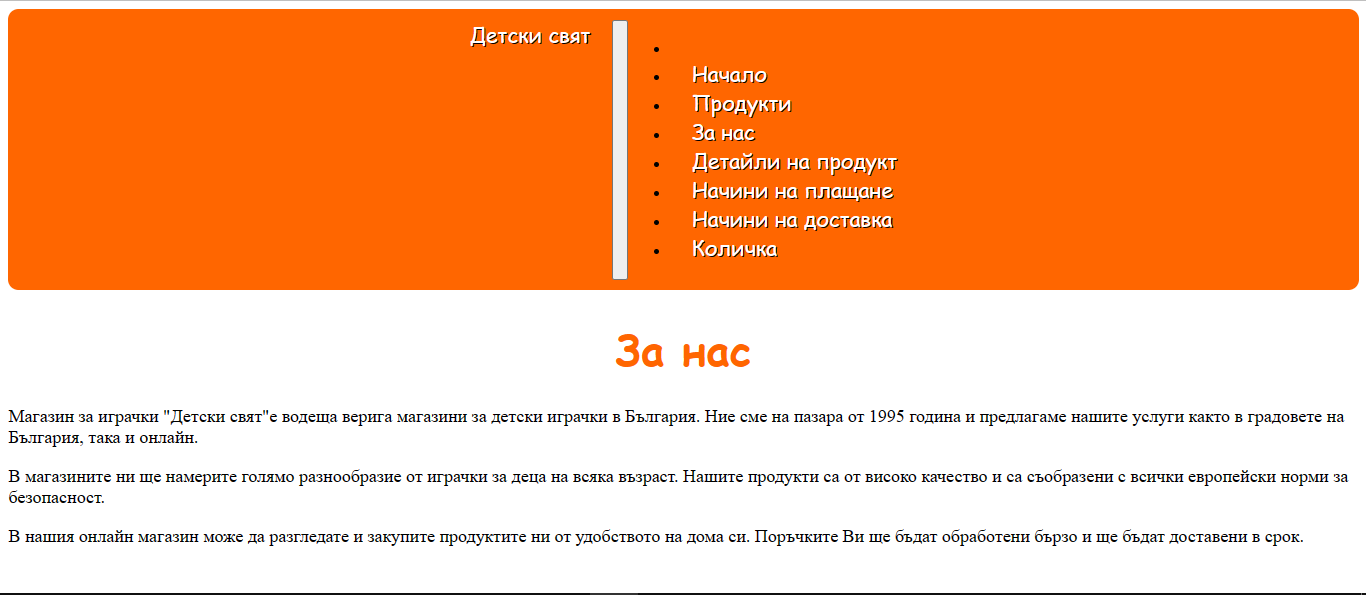
https://www.opencart.com/

https://www.infinityfree.net/

**Страница “за нас”**

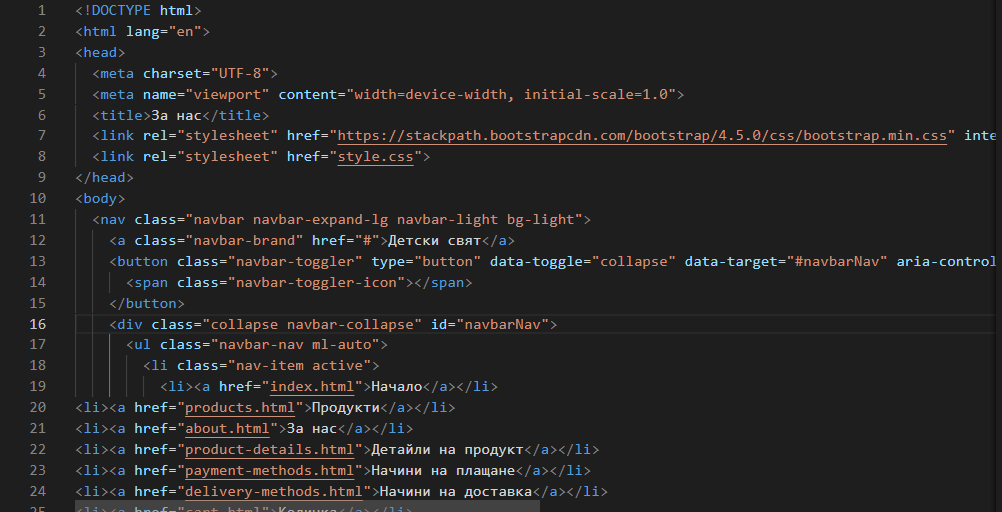
Страницата “за нас” съдържа кратка примерна информация за онлайн магазин “Детски свят”.

Визуално страницата “За нас” можем да видим на фигура 1:



Фигура 1

По-долу е показан кода на страницата “За нас” с подробно описание:



Този HTML код представлява страницата "За нас" на един онлайн магазин за детски играчки:

<!DOCTYPE html>

<html lang="en">

Това е декларацията на типа на документа (Doctype). В този случай, документът е написан на HTML и използва езика на английски.

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>За нас</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous">

<link rel="stylesheet" href="style.css">

</head>

Това е елементът <head>, който съдържа метаданни за документа. Тук има дефинирани метатагове, които определят знаковата таблица на документа (UTF-8) и изгледа му (viewport). Също така има таг <title>, който определя заглавието на страницата и два тага <link>, които свързват документа с външни CSS файлове.

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">Детски свят</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<li><a href="index.html">Начало</a></li>

<li><a href="products.html">Продукти</a></li>

<li><a href="about.html">За нас</a></li>

<li><a href="product-details.html">Детайли на продукт</a></li>

<li><a href="payment-methods.html">Начини на плащане</a></li>

<li><a href="delivery-methods.html">Начини на доставка</a></li>

<li><a href="cart.html">Количка</a></li>

</li>

</ul>

</div>

</nav>

Това е елементът <body>, който съдържа цялата информация за съдържанието на страницата. В този случай, той започва с елемент <nav>, който съдържа менюто на сайта. Това е Bootstrap компонент, който е настроен да показва навигационно меню. Навигационното меню е наредено като списък от елементи, като всеки елемент е етикет за връзка (anchor tag), който води към различни страници на уеб сайта.

Този компонент включва основна HTML структура, която включва:

DOCTYPE - указващ типа на документа

html елемент - който определя началото и края на HTML документа

head елемент - който съдържа метаданни, като заглавие на страницата, символен низ за символна таблица и връзки към външни ресурси, като CSS и JavaScript файлове.

body елемент - който съдържа основното съдържание на страницата.

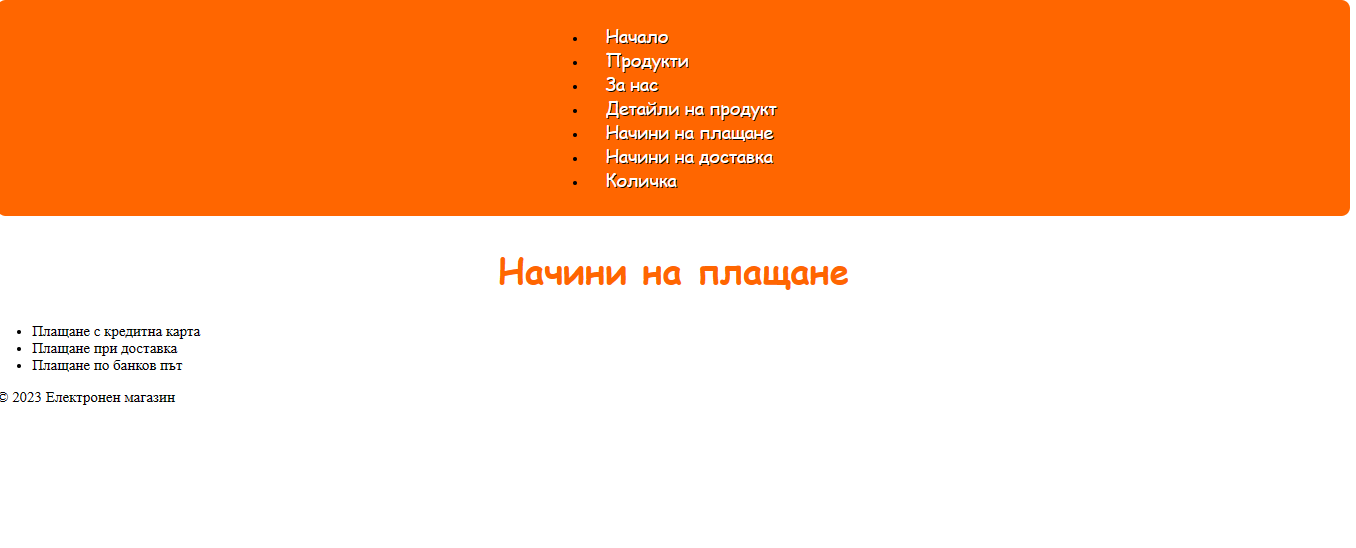
Освен това, този код използва Bootstrap библиотека за стилизиране на навигационното меню. Той включва връзки към три различни CDN файла за Bootstrap CSS, jQuery и Bootstrap JavaScript. Тези връзки са настроени така, че да вземат файловете от CDN-а, когато страницата се зареди, вместо да ги изтеглят от локалния сървър. Това може да подобри бързодействието на страницата, като се използва кеширане на браузъра.

Накрая, този код включва елементи на страницата, като заглавие на страницата и основното съдържание на страницата, като описание на магазина за детски играчки и услугите, които той предлага.

**Страница за начини на плащане**

Страницата “Начини на плащане” съдържа информация за 3 начина на плащане:1-Плащане с кредитна карта; 2-Плащане при доставка; 3-Плащане по банков път .

Визуално страницата “Начини на плащане” можем да видим на фигура 1:



Фигура 2

По-долу е показан кода на страницата “Начини на плащане” с описание:

Тази линия на кода указва, че това е HTML документ с език на комуникация "en" (английски език):

html

<html lang="en">

Следващата част от кода е <head> елемент, който съдържа информацията, свързана с документа. В този случай имаме два елемента: <meta> и <title>.



html

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Начини на плащане</title>

<link rel="stylesheet" href="style.css">

</head>

<meta> елементите представляват метаданни, които не се показват на страницата, а се използват за определяне на различни настройки за документа. charset атрибута задава символната таблица, използвана в документа. name и content атрибутите се използват за определяне на метатегове, свързани с описанието на документа и мащабирането на страницата.<title> елементът определя заглавието на документа, което се показва в горната част на прозореца на браузъра.<link> елементът свързва документа с CSS файл, който съдържа стила на страницата.

След <head> елемента следва <body> елемента, който съдържа всичко, което се показва на страницата.



html

<body>

<header>

<nav>

<ul>

<li><a href="index.html">Начало</a></li>

<li><a href="products.html">Продукти</a></li>

<li><a href="about.html">За нас</a></li>

<li><a href="product-details.html">Детайли на продукт</a></li>

<li><a href="payment-methods.html">Начини на плащане</a></li>

<li><a href="delivery-methods.html">Начини на доставка</a></li>

<li><a href="cart.html">Количка</a></li>

</ul>

</nav>

</header>

<main>

<section>

<h1>Начини на плащане</h1>

<ul>

<li>Плащане с кредитна карта</li>

<li>Плащане при доставка</li>

<li>Плащане по банков път</li>

</ul>

</section>

</main>

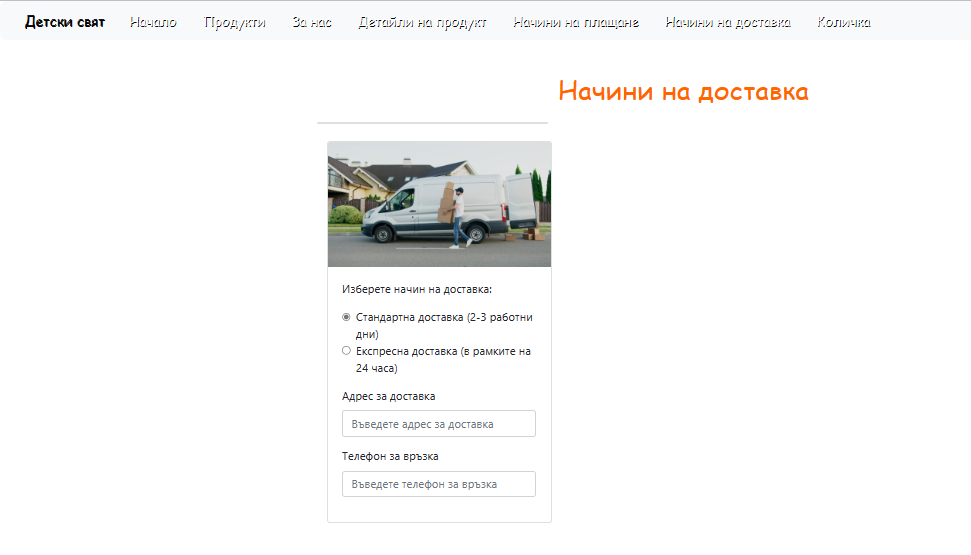
<footer>

<p>© 2023 Електронен магазин

## Страница за начини на доставка

Страницата “Начини на доставка” съдържа информация за начините за доставка, които са експресна и стандартна и опциите: адрес за доставка и телефон за връзка.

Визуално страницата “Начини на доставка” можем да видим на фигура 3:



Фигура 3

По-долу е показан кода на страницата “Начини на доставка” с описанието му:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta http-equiv="X-UA-Compatible" content="ie=edge" />

    <link

      rel="stylesheet"

      href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

      integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"

      crossorigin="anonymous"

    />

    <link rel="stylesheet" href="style.css" />

    <title>Начини на доставка</title>

  </head>

  <body>

    <nav class="navbar navbar-expand-lg navbar-light bg-light">

      <a class="navbar-brand" href="#">Детски свят</a>

      <button

        class="navbar-toggler"

        type="button"

        data-toggle="collapse"

        data-target="#navbarNav"

        aria-controls="navbarNav"

        aria-expanded="false"

        aria-label="Toggle navigation"

      >

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarNav">

        <ul class="navbar-nav">

          <li class="nav-item">

            <li><a href="index.html">Начало</a></li>

<li><a href="products.html">Продукти</a></li>

<li><a href="about.html">За нас</a></li>

<li><a href="product-details.html">Детайли на продукт</a></li>

<li><a href="payment-methods.html">Начини на плащане</a></li>

<li><a href="delivery-methods.html">Начини на доставка</a></li>

<li><a href="cart.html">Количка</a></li>

</li>

        </ul>

      </div>

    </nav>

    <div class="container my-5">

      <h1 class="text-center mb-4">Начини на доставка</h1>

      <div class="row">

        <div class="col-lg-4 col-md-6 col-sm-12">

          <div class="card mb-4">

            </div>

          </div>

        </div>

        <div class="col-lg-4 col-md-6 col-sm-12">

          <div class="card mb-4">

            <img

              class="card-img-top"

              src="https://images.pexels.com/photos/5025669/pexels-photo-5025669.jpeg?auto=compress&cs=tinysrgb&w=1260&h=650&dpr=1"

              alt="Начин на доставка"

            />

            <div class="card-body">

              <p class="card-text">Изберете начин на доставка:</p>

              <form>

                <div class="form-check">

                  <input class="form-check-input" type="radio" name="deliveryMethod" id="standardDelivery" value="standard" checked>

                  <label class="form-check-label" for="standardDelivery">

                      Стандартна доставка (2-3 работни дни)

                    </label>

                  </div>

                  <div class="form-check">

                    <input class="form-check-input" type="radio" name="deliveryMethod" id="expressDelivery" value="express">

                    <label class="form-check-label" for="expressDelivery">

                      Експресна доставка (в рамките на 24 часа)

                    </label>

                  </div>

                  <div class="form-group mt-3">

                    <label for="address">Адрес за доставка</label>

                    <input type="text" class="form-control" id="address" placeholder="Въведете адрес за доставка">

                  </div>

                  <div class="form-group">

                    <label for="phoneNumber">Телефон за връзка</label>

                    <input type="text" class="form-control" id="phoneNumber" placeholder="Въведете телефон за връзка">

                  </div>

                </form>

              </div>

Този код е HTML код за уеб страница, която показва начините на доставка за детски свят. На първите редове се задават основни метаданни за уеб страницата, включително заглавието на страницата и езикът на уеб страницата. След това следва HTML код за навигационно меню, което позволява на потребителите да навигират към различни страници на уеб сайта.

След това има контейнер с клас "container", който включва елементи за показване на начините на доставка. Има контейнер с клас "row", който съдържа две колони - една за изображения и друга за информация за доставка.

Има форма за избор на начин на доставка, която съдържа два радиобутони за избор на стандартна или експресна доставка, като се изисква въвеждане на адрес за доставка и телефонен номер за връзка.

В HTML кода се използват стилове от Bootstrap framework, за да се приложи визуално оформление и да се улесни разполагането на елементите на уеб страницата в зависимост от размера на екрана на устройството, от което се преглежда.

След това следват основните елементи на уеб страницата - <body> тага, който съдържа основната информация, която потребителят ще вижда на страницата, както и <nav> тага, който представлява навигационното меню на страницата.

В тялото на <body> тага се съдържат всички елементи на страницата - заглавия, секции, текст, форми и др.

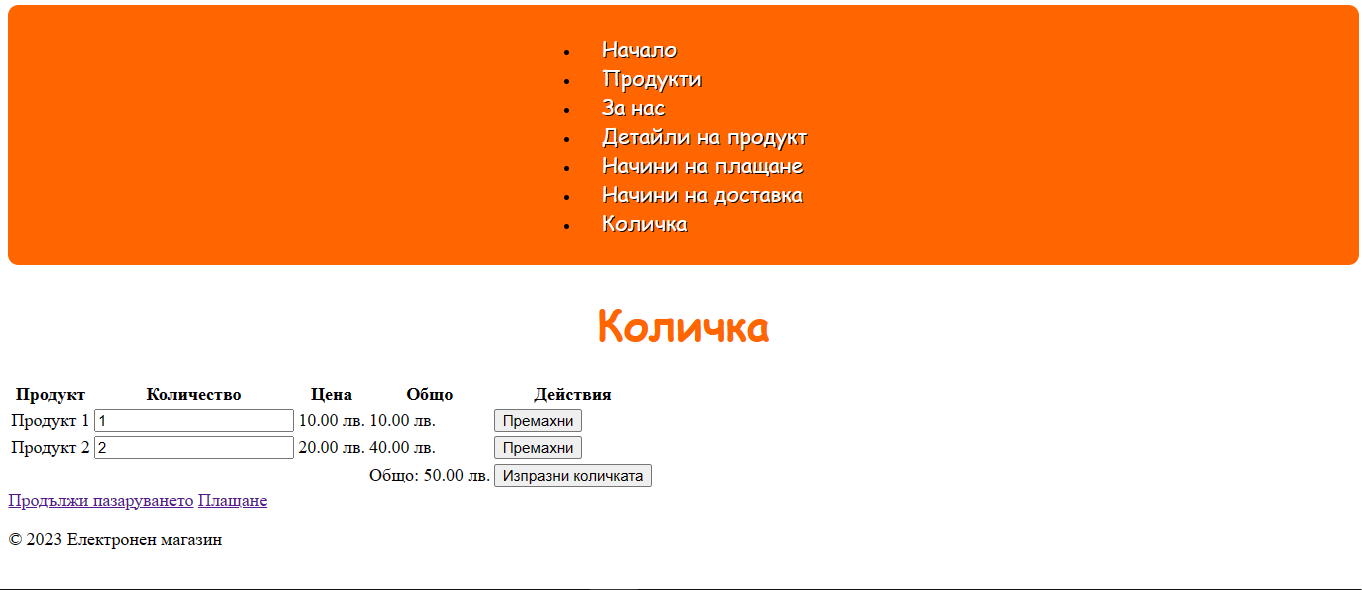
В този код, след <nav> тага има <div> елемент с клас container, който служи за създаване на контейнер за съдържанието на страницата, след което следва <h1> заглавие за начините на доставка. След това има <div> елемент с клас row, който съдържа два <div> елемента с клас col-lg-4 col-md-6 col-sm-12. Тези елементи служат за създаване на секции на страницата, където могат да се включват изображения, текст и други елементи.

В този код, във втория <div> елемент има <div> елемент с клас card, който представлява карта за показване на информация. Този елемент съдържа изображение на начина на доставка и форма за избор на начин на доставка, адрес за доставка и телефон за връзка. Вътре в формата има два <div> елемента с клас form-check, които съдържат радио бутони за избор на начин на доставка. След това има два <div> елемента с клас form-group, които съдържат текстови полета за въвеждане на адрес за доставка и телефон за връзка.

## Страница за електронна количка

Страницата “Количка” съдържа пример за “продукти”, “количество”, “общо”и действия: “премахни” и “изпразни количката”

Визуално страницата “Количка” може да се види на фигура 4 по-долу:



Фигура 4

## Кода и обяснението му на страницата “Количка” са представени по-долу:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Количка</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <header>

    <nav>

      <ul>

        <li><a href="index.html">Начало</a></li>

<li><a href="products.html">Продукти</a></li>

<li><a href="about.html">За нас</a></li>

<li><a href="product-details.html">Детайли на продукт</a></li>

<li><a href="payment-methods.html">Начини на плащане</a></li>

<li><a href="delivery-methods.html">Начини на доставка</a></li>

<li><a href="cart.html">Количка</a></li>

</ul>

    </nav>

  </header>

<main>

    <h1>Количка</h1>

    <table>

      <thead>

        <tr>

          <th>Продукт</th>

          <th>Количество</th>

          <th>Цена</th>

          <th>Общо</th>

          <th>Действия</th>

        </tr>

      </thead>

      <tbody>

        <tr>

          <td>Продукт 1</td>

          <td><input type="number" value="1" min="1"></td>

          <td>10.00 лв.</td>

          <td>10.00 лв.</td>

          <td><button>Премахни</button></td>

        </tr>

        <tr>

          <td>Продукт 2</td>

          <td><input type="number" value="2" min="1"></td>

          <td>20.00 лв.</td>

          <td>40.00 лв.</td>

          <td><button>Премахни</button></td>

        </tr>

      </tbody>

      <tfoot>

        <tr>

          <td colspan="3"></td>

          <td>Общо: 50.00 лв.</td>

          <td><button>Изпразни количката</button></td>

        </tr>

      </tfoot>

    </table>

    <a href="#">Продължи пазаруването</a>

    <a href="#">Плащане</a>

  </main>

  <footer>

    <p>© 2023 Електронен магазин

  </footer>

</body>

</html>

Това е HTML код за страницата "Количка" на електронен магазин “Детски свят”. Кода съдържа основната структура на HTML документ, включително елементи за заглавие, метаданни, основен контейнер за съдържанието на страницата, включващ заглавие, таблица с продукти в количката и бутони за действия, футър с авторско право и допълнителни връзки. Кода включва и връзки към CSS файл, които се използват за стилизиране на страницата.

## 

## Страница за списък с продукти

Страницата “Продукти” съдържа кратък списък с 3 примерни продукта: детска кола с дистанционно, детски панталон и яке с кратко описание и опция добави в количка.

## По-долу е показан визуалния изглед на страницата “Продукти” съответно на фиг. 5 и 6:

## 

Фигура 5

## 

Фигура 6

## Отдолу е показан кода на страницата с пояснения:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Всички продукти - Детски свят</title>

  <link rel="stylesheet" href="style.css">

  <style>

    /\* Responsive styles \*/

    @media only screen and (max-width: 600px) {

      body {

        font-size: 16px;

      }

      .products-list {

        display: flex;

        flex-direction: column;

      }

      .products-list li {

        margin-bottom: 2rem;

      }

      img {

        max-width: 100%;

        height: auto;

      }

    }

  </style>

</head>

<body>

  <header>

    <nav>

      <ul>

        <li><a href="index.html">Начало</a></li>

        <li><a href="products.html">Продукти</a></li>

        <li><a href="about.html">За нас</a></li>

        <li><a href="product-details.html">Детайли на продукт</a></li>

        <li><a href="payment-methods.html">Начини на плащане</a></li>

        <li><a href="delivery-methods.html">Начини на доставка</a></li>

        <li><a href="cart.html">Количка</a></li>

        </ul>

    </nav>

  </header>

<main>

    <section>

      <h1>Списък с продукти</h1>

      <ul class="products-list">

        <li>

          <h2>Детска кола с дистанционно управление</h2>

          <img src="https://cdncloudcart.com/11561/products/images/43837/rastar-kola-bmw-z4-s-distancionno-upravlenie-1-24-39700-image\_5bda600e2c731\_150x150.jpeg" alt="Детска кола с дистанционно управление">

          <p>Описание на продукта: Прекрасен подарък за вашия малчуган.</p>

          <p>Цена: 15 лв.</p>

          <a href="#">Добави в количката</a>

        </li>

        <li>

          <h2>Детски панталон</h2>

          <img src="https://matiniki.com/UserFiles/products/F5BC7A12-431A-E60E-0974-DC6E529E9205.jpg?q=150&cache&block&color=ffffff&w=150&h=150" width="150" height="150" alt="Детски панталон">

          <p>Описание на продукта: елегантен детски панталон за вашия мъник.</p>

          <p>Цена: 10 лв.</p>

          <a href="#">Добави в количката</a>

        </li>

        <li>

          <h2>Детско яке</h2>

          <img src="https://www.sportihobi.bg/sites/default/files/styles/uc\_product\_cats/public/977297459/1292756-911-1.jpg?itok=ewJ90NxQ" width="150" height="150" alt="Детско яке">

          <p>Описание на продукта: Топло и меко яке за вашето дете за зимния сезон.</p>

          <p>Цена: 80 лв.</p>

        </li>

        </section>

Кодът представлява HTML страница, която съдържа информация за списък с продукти на един онлайн магазин.

## В първите редове на кода са дефинирани основни метаданни за HTML страницата, като засичане на езика и мащабиране на страницата. Заглавието на страницата се определя в следващия ред.

## В следващите редове се свързва външен CSS файл, който съдържа допълнителни стилове за оформяне на страницата. В самия HTML код също има допълнителни стилове, които се използват за мащабиране на страницата в зависимост от ширината на екрана (responsive design).

## След заглавието и стиловете на страницата, има секция "header", която включва меню с линкове към различните страници на сайта.

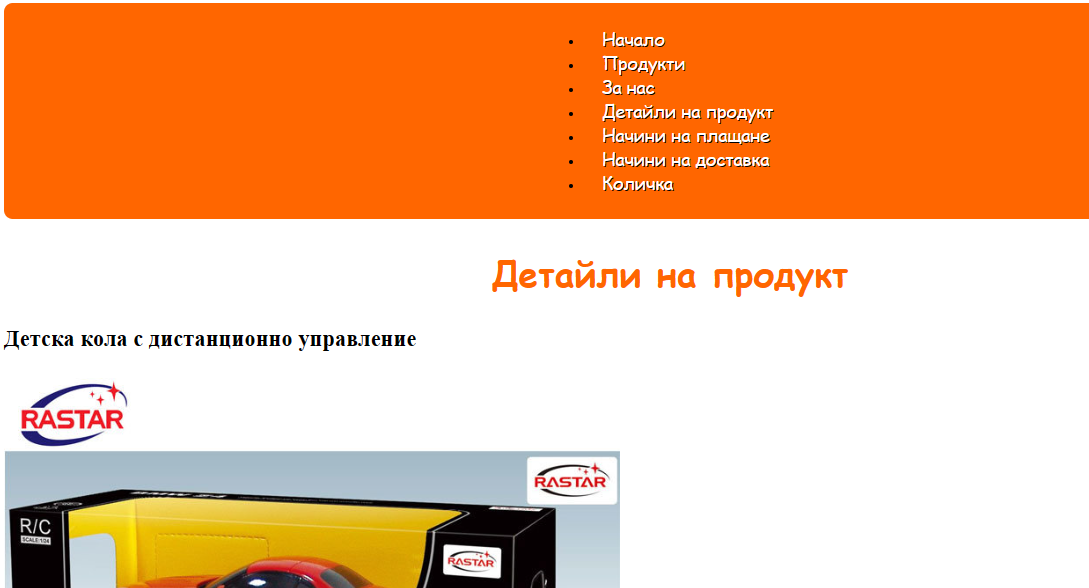
## .Главната секция на страницата ("main") съдържа списък с продукти, като всеки продукт е представен в "li" елемент. В тях се съдържа заглавие на продукта, снимка, описание, цена и бутон за добавяне на продукта в количката.

## В последния ред на кода се затваря секцията "section", която съдържа списъка с продукти.

## Детайлизирана страница за продукт

## В страницата “Детайли на продукт” е показан примерен продукт: “Детска кола с дистанционно управление” с кратко описание, цена, количество и опция: “добави в количка”

На фигури 7 и 8 по-долу съответно е показан визуалния изглед на страницата:



Фигура 7

# 

Фигура 8

# Отдолу е показан кода на страницата с пояснения:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Детайли на продукт - Детски свят</title>

  <link rel="stylesheet" href="style.css">

  <style>

    /\* риспонсив дизайни \*/

    @media only screen and (max-width: 600px) {

      body {

        font-size: 16px;

      }

      img {

        max-width: 100%;

        height: auto;

      }

    }

  </style>

</head>

<body>

  <header>

    <nav>

      <ul>

        <li><a href="index.html">Начало</a></li>

        <li><a href="products.html">Продукти</a></li>

        <li><a href="about.html">За нас</a></li>

        <li><a href="product-details.html">Детайли на продукт</a></li>

        <li><a href="payment-methods.html">Начини на плащане</a></li>

        <li><a href="delivery-methods.html">Начини на доставка</a></li>

        <li><a href="cart.html">Количка</a></li>

        </ul>

    </nav>

  </header>

<main>

    <section>

      <h1>Детайли на продукт</h1>

      <div class="product-details">

        <h2>Детска кола с дистанционно управление</h2>

        <img src="https://cdncloudcart.com/11561/products/images/43837/rastar-kola-bmw-z4-s-distancionno-upravlenie-1-24-39700-image\_5bda600e2c731\_800x800.jpeg" alt="Детска кола с дистанционно управление">

        <p>Описание на продукта: Прекрасен подарък за вашия малчуган.</p>

        <p>Цена: 15 лв.</p>

        <form>

          <label for="quantity">Количество:</label>

          <input type="number" id="quantity" name="quantity" min="1" max="10" value="1">

          <button type="submit">Добави в количката</button>

        </form>

      </div>

    </section>

  </main>

</body>

</html>

Във файла има няколко елемента, които образуват основната структура на уеб страницата. За начало, имаме декларацията <!DOCTYPE html>, която указва, че този файл е написан в HTML5 формат.

След това, имаме основния HTML елемент, който обгражда целия код. Вътре в него има head елемент, който съдържа различни мета данни за страницата, като език, който се използва (lang="en"), описание на страницата, заглавие на страницата и линк към външен CSS файл (style.css), който съдържа стилове за оформление на уеб страницата.

След това имаме body елемент, който съдържа цялата видима част на уеб страницата. В него има header елемент, който съдържа меню с няколко връзки към други страници на сайта. Има и main елемент, който обгражда цялото съдържание на страницата.

В main елемента има section елемент, който съдържа основното съдържание на страницата - детайли за продукт. В section елемента има заглавие h1 и div елемент, който съдържа картинка, описание на продукта, цена и форма за добавяне на продукта в количката. Формата съдържа label елемент, който описва въведеното количество на продукта и input елемент, който позволява на потребителя да въведе количество и бутон за потвърждение на избора.

Накрая, имаме style блок с медиа заявка, която описва различни CSS стилове за елементите в страницата при определен размер на екрана. В този случай, при екран с максимална ширина 600 пиксела, се задава по-голям шрифт за body елемента и максимална ширина за картинката.

# Заключение

# *Заключението на дипломния проект на тема "разработка на риспонсив уеб дизайн на сайт за електронен магазин" е, че проектът е успешно завършен и отговаря на поставените цели и изисквания. В рамките на проекта бяха изградени и стилизирани различни уеб страници за електронен магазин, като основният акцент беше върху създаването на риспонсивен уеб дизайн, който да осигури оптимално преживяване за потребителите на уеб сайта от всякакви устройства.*

# *Проектът включва разработка на уеб дизайн, използвайки HTML и CSS, за да се постигне ефективна навигация и визуална привлекателност на уеб страниците. Бяха приложени най-добрите практики за риспонсивен дизайн, като са използвани медийни заявки, за да се гарантира оптимално изглеждане и функционалност на уеб сайта от всякакви устройства.*

# *Проектът е свързан с изграждането на електронен магазин, който има за цел да улесни онлайн пазаруването на потребителите и да ги убеди да закупят продуктите предлагани от магазина. С цел подобряване на потребителското изживяване, са включени различни функционалности като добавяне на продукти към количка за пазаруване и други.*

# Използвани източници:

## Кирилица:

Колисниченко, Денис. HTML5 & CSS3. Практическо програмиране за начинаещи. София, изд.Асеневци, 2011 г.

## Латиница:

jQuery, достъпно на: <https://jquery.com/>, последно посетено на: 27.02.2023 г.

Responsive Design, в: developer.mozilla.org , достъпно на: <https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design>, последно посетено на: 27.2023 г.

Wadhwa, Danish. Frameworks for responsive web design: Everything you need to know. Достъпно на: <https://www.webdew.com/blog/framework-for-responsive-web-design>, последно посетено на: 27.2023 г.

# 

1. Lepage Pete, Rachel Andrew: [Responsive web design basics](https://web.dev/responsive-web-design-basics/), дата на достъпване: 30.04.2023 [↑](#footnote-ref-1)
2. Friedman Vitaly; [Responsive Web Design: What It Is And How To Use It — Smashing Magazine](https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/); дата на достъпване - 30. 04. 2023г. [↑](#footnote-ref-2)
3. [Top 10 Frameworks For Responsive Web Design In 2022 (classicinformatics.com)](https://www.classicinformatics.com/blog/top-10-frameworks-for-responsive-web-design); дата на достъпване - 30.04.2023 [↑](#footnote-ref-3)
4. [Responsive design - Learn web development | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design) дата на достъпване – 30.04.2023г. [↑](#footnote-ref-4)
5. House Chris: [A Complete Guide to CSS Grid | CSS-Tricks - CSS-Tricks](https://css-tricks.com/snippets/css/complete-guide-grid/) дата на достъпване – 30.04.2023г. [↑](#footnote-ref-5)
6. Kolade Chris: <https://www.freecodecamp.org/news/how-to-link-css-to-html/>; дата на достъпване - 30.04.2023г. [↑](#footnote-ref-6)
7. [↑](#footnote-ref-7)
8. Колисниченко, Денис. HTML5 & CSS3. Практическо програмиране за начинаещи. София, изд.Асеневци, 2011 г., стр. 115 - 116 [↑](#footnote-ref-8)
9. Jquery, достъпно на: <https://jquery.com/>, последно посетено на: 27.02.2023 г. [↑](#footnote-ref-9)
10. [jQuery Tutorial (tutorialspoint.com)](https://www.tutorialspoint.com/jquery/index.htm) дата на достъпване – 30.04.2023г. [↑](#footnote-ref-10)