Съдържание

[1. Увод (Въведение) 3](#_Toc194482422)

[2. Теоретична част 3](#_Toc194482423)

[2.1 Основи на адаптивния дизайн 3](#_Toc194482424)

[2.1.1 Какво е адаптивен дизайн и защо е важен? 3](#_Toc194482425)

[2.1.2 Разлика между адаптивен и респонсив дизайн 3](#_Toc194482426)

[2.1.3 Примери за адаптивен дизайн в реалния свят 4](#_Toc194482427)

[2.1.4 Заключение 5](#_Toc194482428)

[2.2 Фреймуъркът Bootstrap 5](#_Toc194482429)

[2.2.1 Какво представлява Bootstrap? 5](#_Toc194482430)

[2.3 История и развитие на Bootstrap 5](#_Toc194482431)

[2.3.1 Начало на Bootstrap (2010 – 2011 г.) 5](#_Toc194482432)

[2.3.2 Основни компоненти на Bootstrap. 7](#_Toc194482433)

[2.4 Предимства и недостатъци на Bootstrap 11](#_Toc194482434)

[3. Мрежова система на Bootstrap 14](#_Toc194482435)

[3.1 Как работи мрежовата система? 14](#_Toc194482436)

[3.2 Примери за използване на мрежовата система в уеб дизайн. 14](#_Toc194482437)

[3.2.1 Основен пример – равномерно разпределени колони 14](#_Toc194482438)

[3.2.2 Пример с фиксирана широчина на колони 15](#_Toc194482439)

[3.2.3 Адаптивни (responsive) колони с различни брейкпойнти 15](#_Toc194482440)

[3.2.4 Центриране на съдържание в мрежовата система 16](#_Toc194482441)

[3.2.5 Вертикално подравняване на съдържанието 16](#_Toc194482442)

[3.2.6 Галерия с изображения 17](#_Toc194482443)

[3.2.7 Адаптивност и колони в Bootstrap 17](#_Toc194482444)

[3.3 Мрежова система на Bootstrap 20](#_Toc194482445)

[4. Практическа част 20](#_Toc194482446)

[4.1 Описание на проекта 20](#_Toc194482447)

[4.2 Избор на технологии 20](#_Toc194482448)

[4.3 Структура на уебсайта 20](#_Toc194482449)

[4.4 Проблеми и решения 21](#_Toc194482450)

[4.5 Резултати 21](#_Toc194482451)

[5. Заключение 21](#_Toc194482452)

[6. Литература: 22](#_Toc194482453)

# Увод (Въведение)

* Кратко описание на темата и нейната значимост.
* Основни цели на дипломния проект:
  + Да се изследват основите на адаптивния дизайн и фреймуърка Bootstrap.
  + Да се разработи адаптивен уебсайт за училищен клуб.
* Очаквани резултати:
  + Теоретично разбиране на адаптивния дизайн и неговите принципи.
  + Практическо приложение на наученото чрез разработка на уебсайт.

# Теоретична част

## Основи на адаптивния дизайн

### Какво е адаптивен дизайн и защо е важен?

Адаптивният дизайн е подход в уеб разработката, който позволява създаването на уебсайтове, които се адаптират към различни устройства и размери на екрана. Основната идея е да се осигури оптимално потребителско изживяване, независимо дали сайтът се разглежда на настолен компютър, таблет или мобилен телефон.

С развитието на технологиите и нарастващото използване на мобилни устройства, адаптивният дизайн се превърна в стандарт за съвременните уебсайтове. Според статистиките, над 50% от интернет трафика идва от мобилни устройства, което прави адаптивността ключов фактор за успеха на един уебсайт.

Основната цел на адаптивния дизайн е да предостави съдържание, което е лесно за четене и навигация, без необходимост от мащабиране или хоризонтално превъртане. Това се постига чрез използването на гъвкави оформления, медийни заявки и други техники, които позволяват на уебсайта да се адаптира към различни резолюции и размери на екрана.

### Разлика между адаптивен и респонсив дизайн

Често адаптивният дизайн се бърка с респонсивния дизайн, но между тях има съществени разлики.

* **Адаптивен дизайн**: При този подход се създават предварително дефинирани оформления за различни размери на екрана. Например, сайтът може да има отделни версии за настолни компютри, таблети и мобилни устройства. Когато потребителят отвори сайта, устройството му автоматично избира най-подходящото оформление. Това означава, че адаптивният дизайн разчита на фиксирани точки на прекъсване (breakpoints), които определят как ще изглежда съдържанието.
* **Респонсивен дизайн**: Респонсивният дизайн използва гъвкави мрежи (fluid grids) и CSS медийни заявки, за да адаптира съдържанието динамично към всякакъв размер на екрана. Вместо да има предварително дефинирани оформления, сайтът се "приспособява" в реално време към устройството на потребителя.

И двата подхода имат своите предимства и недостатъци. Адаптивният дизайн е по-лесен за изпълнение, когато се работи с ограничен брой устройства, но може да изисква повече ресурси за поддръжка. Респонсивният дизайн, от друга страна, е по-гъвкав, но може да бъде по-сложен за разработка.

### Примери за адаптивен дизайн в реалния свят

Много популярни уебсайтове и платформи използват адаптивен дизайн, за да осигурят безпроблемно потребителско изживяване. Ето няколко примера:

* **Новинарски сайтове**: Сайтове като BBC и CNN използват адаптивен дизайн, за да предоставят съдържание, което е лесно за четене както на големи екрани, така и на мобилни устройства. Например, на настолен компютър потребителят може да види множество колони с новини, докато на мобилен телефон съдържанието се представя в една колона за по-добра четимост.
* **Онлайн магазини**: Платформи като Amazon и eBay използват адаптивен дизайн, за да осигурят удобна навигация и лесно пазаруване на всякакви устройства. Продуктовите страници, количките за пазаруване и формите за плащане са оптимизирани за различни размери на екрана.
* **Образователни платформи**: Сайтове като Khan Academy и Coursera използват адаптивен дизайн, за да предоставят учебни материали, които са достъпни както на настолни компютри, така и на мобилни устройства. Това позволява на потребителите да учат в движение, без значение какво устройство използват.

### Заключение

Адаптивният дизайн е ключов елемент от съвременната уеб разработка, който гарантира, че съдържанието на уебсайтовете е достъпно и удобно за използване на всякакви устройства. Разликата между адаптивния и респонсивния дизайн е важна за разбирането на различните подходи към създаването на уебсайтове. Примерите от реалния свят показват как адаптивният дизайн може да подобри потребителското изживяване и да увеличи ефективността на уебсайтовете.

## Фреймуъркът Bootstrap

### Какво представлява Bootstrap?

Bootstrap е широко разпространен фреймуърк, улесняващ създаването на адаптивни и мобилно-оптимизирани уебсайтове и приложения чрез предварително дефинирани стилове и компоненти като бутони, навигации и формуляри.

## История и развитие на Bootstrap

### Начало на Bootstrap (2010 – 2011 г.)

През 2010 г. Марк Ото и Джейкъб Торнтън , двама инженери от Twitter, започват работа върху вътрешен проект, наречен "Twitter Blueprint" . Целта е да се създаде унифицирана система за проектиране на уеб приложения, която да подобри съвместимостта в дизайна на всички продукти на компанията.

На 19 август 2011 г. Bootstrap става проект с отворен код и е пуснат официално като версия 1.0 в GitHub.

* **Основни версии на Bootstrap**

**Bootstrap 1 (2011 г.)**

🔹 Първата версия на Bootstrap предлага основен стил и UI компоненти като бутони, навигация, таблици и форми.

🔹 Той става популярен заради лесната си употреба и унифициран стил .

**Bootstrap 2 (2012 г.)**

🔹 Най-голямото подобрение в тази версия е добавянето на адаптивен (отзивчив) дизайн .

🔹 Включена е грид система (Grid System ), която помага за по-добро организиране на елементите на страницата.

🔹 Въведени са нови UI компоненти като герои (геройски единици), карусели и модали .

**Bootstrap 3 (2013 г.)**

🔹 Въвежда се концепцията "Mobile First" – дизайнът първо се създава за мобилни устройства, а след това се усложнява за по-големи екрани.

🔹 Мрежовата система е изцяло обновена и включва 12-колонен модел .

🔹 Bootstrap 3 показва поддръжката на Internet Explorer 7 и 8 .

**Bootstrap 4 (2018 г.)**

🔹 Bootstrap 4 прави големи промени в ядрото на фреймуърка:

Изцяло преминава към Flexbox , което улеснява оформлението на елементите.

Замяна на LESS със Sass , която прави стилизирането по-гъвкаво и лесно.

Въвеждане на нови компоненти като карти (Cards) , които заменят панелите и списъците.

🔹 Подобрен е контролът върху отстоянията (маржини и подложки) чрез класове за помощни интервали .

🔹 Опростена е типографията с нова система за управление на шрифтовете .

**Bootstrap 5 (2021 г.)**

🔹 Премахната е зависима от jQuery , която прави Bootstrap по-лек и бързо.

🔹 Включена е поддръжка на CSS Grid , което дава допълнителни възможности за оформление.

🔹 Опростена е типографията и цветната система, с по-голям фокус върху достъпността.

🔹 Въведена е поддръжка на нови CSS свойства , като CSS променливи, които улесняват персонализацията на теми.

### Основни компоненти на Bootstrap.

* **Респонсивен дизайн (Responsive Design)**

Една от най-важните характеристики на Bootstrap е неговата респонсивност. Той използва система от гъвкави колони и контейнери, която позволява лесно адаптиране на уеб страниците към различни размери на екрани. Благодарение на това, уеб сайтовете, създадени с Bootstrap, изглеждат добре както на мобилни устройства, така и на големи екрани.

* **Гъвкава мрежова система (Grid system):**

Bootstrap използва 12-колонна мрежова система за организация на съдържанието, използваща класове като .col- и .row, която е проектирана да бъде адаптивна към различни размери на екрана, осигурявайки оптимално визуализиране на уебсайтове на мобилни устройства, таблети и настолни компютри.

* **Предварително стилизирани компоненти**

Bootstrap включва множество предварително дефинирани стилове за основните HTML елементи като бутони, форми, таблици и други. Освен това, предоставя готови компоненти като:

* Навигационни ленти (Navbar)
* Модални прозорци (Modals)
* Карти (Cards)
* Акордеони (Accordion)
* Карусели (Carousel)
* JavaScript плъгини

Освен CSS и HTML, Bootstrap включва набор от JavaScript компоненти, които подобряват функционалността на уеб страниците. Те позволяват добавяне на интерактивност, без да се налага използването на допълнителни библиотеки.

Някои от най-използваните JavaScript компоненти включват:

* **Tooltip** – изскачащи подсказки при задържане на курсора върху даден елемент
* **Collapse** – скриване и показване на съдържание
* **Dropdown** – падащи менюта
* **Popover** – допълнителни информационни пр
* **Какво е помощната програма (Utility) в Bootstrap?**

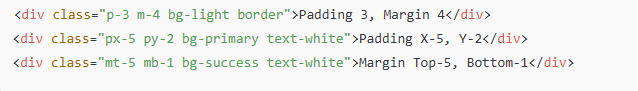
В Bootstrap , утилитите (помощни) класове са малки, готови стилове , които могат да се прилагат директно към HTML елементите, без да се пише допълнителен CSS код. Те помагат за бързо оформяне и стилизиране на уеб елементи на уеб елементи елементите, като например управление на разстояния, подравняване, цветове, размери и други.

* **Основни видове помощни класове в Bootstrap**

**Разстояния (Spacing – Margin и Padding)**

Bootstrap предоставя m- (margin) и p- (padding) класове за бързо задаване на класове за бързо задаване на разстояния около елементите.

**Пример:**

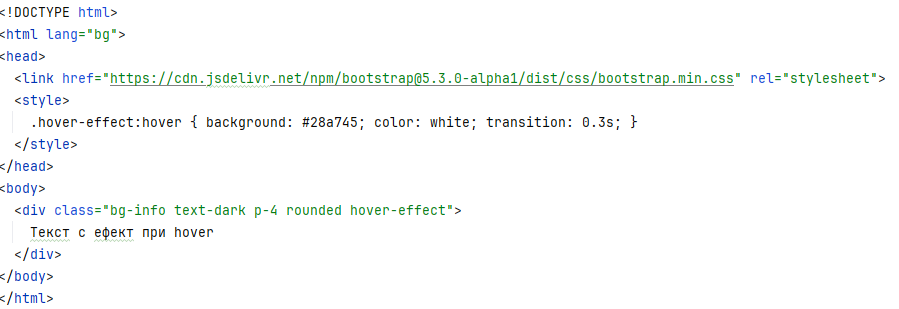
****

**Класове за цветове (Background & Text Colors)**

Bootstrap предоставя предварително дефинирани цветови класове за фонове и текст.

* **Фон (Background)**
* bg-primary (син), bg-secondary (сив), bg-success (зелен), bg-danger (червен), bg-warning (жълт), bg-info (светлосин), bg-light, bg-dark, bg-white, bg-transparent.
* **Текст (Text Colors)**
* text-primary, text-success, text-danger, text-warning, text-info, text-dark, text-light, text-muted, text-white.

**Пример:**

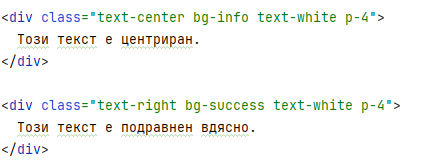
****

**Типове подравняване на текст**

Bootstrap предлага класове за **подравняване на текста**. Това е важно за контрола на оформлението на съдържанието.

* **text-left**: Подравнява текста вляво.
* **text-center**: Подравнява текста в център.
* **text-right**: Подравнява текста вдясно.

**Пример**:

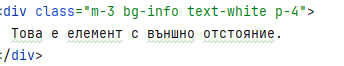


**Размери на отстояния (Spacing)**

Bootstrap има помощни класове за контролиране на **външни отстояния** (margin) и **вътрешни отстояния** (padding). Това са много важни класове за подредба на елементите на страницата.

* **Пример за външно отстояние (margin):**
* **m-\***: За задаване на външно отстояние (margin).
  + m-0: Няма отстояние.
  + m-1, m-2, ..., m-5: Увеличават отстоянието.

**Пример:**

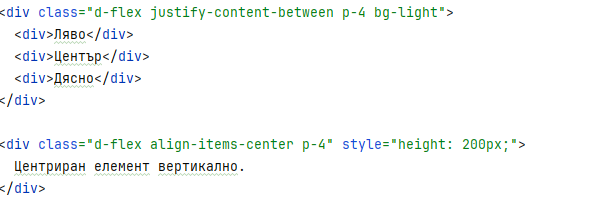
****

**Flexbox помощни класове**

Bootstrap включва и много класове, свързани с **Flexbox**, за по-гъвкаво подравняване и подредба на елементите.

* **d-flex**: Активира Flexbox.
* **justify-content-\***: Подравнява елементите по основната ос (ляво/дясно).
* Пример: justify-content-center, justify-content-between, justify-content-end.
* **align-items-\***: Подравнява елементите по кръстосаната ос (горе/долу).
* Пример: align-items-center, align-items-start, align-items-end.

**Пример:**

****

## Предимства и недостатъци на Bootstrap

* **Предимства на Bootstrap**
  1. **Бързо разработване**

**Bootstrap** осигурява готови компоненти, стилове и шаблони, които значително ускоряват процеса на разработка. Това е особено полезно за нови проекти или за разработчици, които искат да избегнат писането на много CSS код.

Пример: Готови бутони, навигационни панели, форми, таблици и други елементи могат да бъдат внедрени със само няколко реда код.редимства на Bootstrap.

* 1. **Отзивчив дизайн (Responsive design)**

**Bootstrap** предлага вградени грид системи и media queries, които правят сайтът отзивчив, т.е. адаптиран за различни устройства (мобилни телефони, таблети, десктопи и т.н.).

**Пример: Използването на класове като col-, container, и row прави дизайна** адаптивен към всякакви размери на екрана.

* **Предварително дефинирани компоненти**

Bootstrap предоставя много графични компоненти като бутони, навигационни барове, модални прозорци, инструменти за карусели, табове, предупреждения, и много други, които лесно могат да се добавят към уеб приложението.

Пример: За да създадеш бутон с клас btn btn-primary, ти не трябва да пишеш CSS от нулат

* 1. **Поддръжка за различни браузъри**

Bootstrap предлага поддръжка за всички основни браузъри (Chrome, Firefox, Safari, Edge), което намалява необходимостта от преглеждане на сайта в различни браузъри.

Пример: Ти можеш да бъдеш сигурен, че сайтът ти ще изглежда добре и в Internet Explorer, и в най-новите версии на Chrome.

* 1. **Лесна интеграция с други библиотеки и технологии**

Bootstrap е лесен за интегриране с други технологии и библиотеки, като например jQuery, Angular, React, Vue.js и други.

Пример: Можеш да използваш Bootstrap в проект с React, без да имаш проблеми с интеграцията.

* **Недостатъци на Bootstrap**

1. **Тежест и излишък от код**

**Bootstrap** идва със значително количество CSS и JavaScript код, който може да не се използва в целия проект. Това води до по-големи размери на файловете, което може да доведе до по-бавно зареждане на страниците.

Пример: Ако използваш само малка част от компонентите на Bootstrap, ще зареждаш много ненужен код, което може да забави сайта.

1. **Елементи, които изглеждат сходно („Bootstrap външен вид“)**

За начинаещите или за хората, които не се запознават с персонализирането на стилове, сайтът може да изглежда твърде стандартно или "Bootstrap-изиран".

Пример: Ако не промениш стандартните стилове на бутоните, те ще изглеждат като типичните бутони на Bootstrap, което може да не отговаря на уникалния стил на твоя бранд.

1. **Ограничена гъвкавост**

Въпреки че Bootstrap предлага много възможности за конфигуриране, когато искаш много персонализирани дизайни, може да се окаже, че фреймуъркът не е достатъчно гъвкав. За специфични дизайни или приложения, може да се наложи да пишеш много персонализиран CSS.

Пример: Ако имаш специфичен изисквания за дизайн, като нестандартни анимации или аранжиране на елементите, може да бъде трудно да го направиш само с Bootstrap.

1. **Често обновление на версията**

Bootstrap често пуска нови версии, които може да изискват обновления на съществуващия код. Това може да доведе до несъвместимости и допълнителни усилия за адаптация на по-стария код към новата версия.

Пример: Ако започнеш проект с Bootstrap 4, преминаването към Bootstrap 5 може да изисква сериозни промени в CSS класовете и структурите на проектите.

1. **Проблеми със специфичността на CSS**

Въпреки че Bootstrap е добре структурирана система, когато започваш да персонализираш стиловете си, може да срещнеш проблеми със специфичността на CSS класовете.

Пример: Ако не можеш да пренапишеш някой от стандартните стилове на Bootstrap, може да се наложи да използваш **!import**, което често е нежелано в добрите практики.

# Мрежова система на Bootstrap

## Как работи мрежовата система?

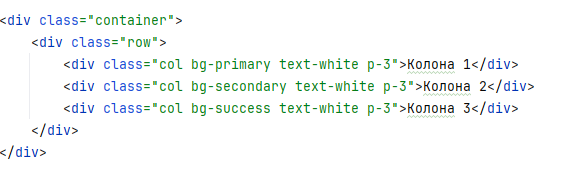
* Мрежовата система на Bootstrap (Grid System) работи на базата на Flexbox и използва контейнери, редове и колони, за да създаде адаптивен (responsive) дизайн. Тя позволява съдържанието на уеб страницата да се разпределя равномерно и динамично, така че да изглежда добре на различни размери на екрана.

## Примери за използване на мрежовата система в уеб дизайн.

* Мрежовата система на Bootstrap позволява лесно изграждане на адаптивни уеб сайтове. В тази статия ще разгледаме различни подробни примери, които ще ти помогнат да разбереш как да прилагаш мрежовата система в реални уеб проекти.

### Основен пример – равномерно разпределени колони

* В този пример имаме три равномерни колони, които автоматично ще се адаптират към ширината на екрана.

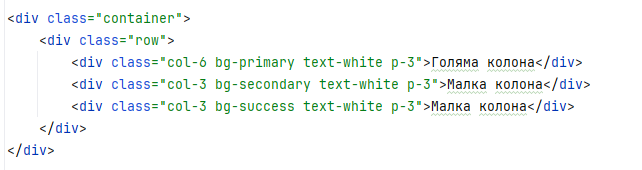


**Как работи този код?**

* .col автоматично разпределя пространството поравно между всички колони.
* Цветове: Всяка колона е оцветена, за да се вижда разположението ѝ.

### Пример с фиксирана широчина на колони

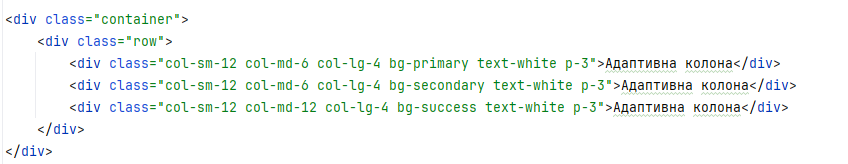
* В този пример създаваме **две големи и една малка колона**.

****

* Обяснение
* .col-6 заема половината от реда.
* .col-3 заема четвърт от реда.
* Общата сума е 12, така че всички колони се побират в реда.

### Адаптивни (responsive) колони с различни брейкпойнти

* Тук ще използваме брейкпойнти, за да направим оформлението гъвкаво за различни устройства.



**Какво прави този код?**

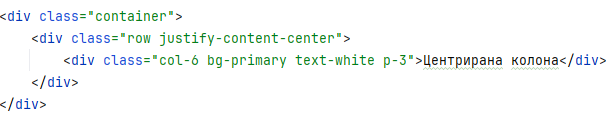
На малки екрани (sm) – всички колони са по 100% и се подреждат една под друга.

На средни екрани (md) – първите две колони са по 50%, а третата остава 100%.

На големи екрани (lg) – всички колони стават по 1/3 (33%) от ширината.

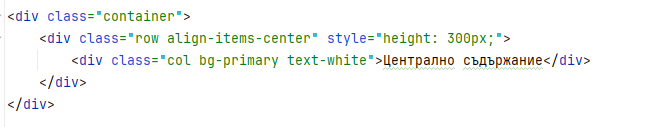
### Центриране на съдържание в мрежовата система

* Можем да центрираме колоните в реда с justify-content-center.



### Вертикално подравняване на съдържанието

Ако искаме съдържанието на колоните да бъде центрирано вертикално, използваме align-items-center.



* Какво прави този код?
* align-items-center подравнява колоната по средата на реда.
* Добавяме height: 300px;, за да има видим ефект.

### Галерия с изображения

Често се използва мрежовата система за изграждане на галерии.Галерия с изображения.



### Адаптивност и колони в Bootstrap

* **Какво представлява адаптивността на Bootstrap?**

Адаптивността означава, че даден уебсайт се приспособява автоматично към различни размери на екрани – настолни компютри, лаптопи, таблети и смартфони.

Bootstrap използва брейкпойнти (breakpoints), за да контролира как се показват колоните в зависимост от ширината на устройството.

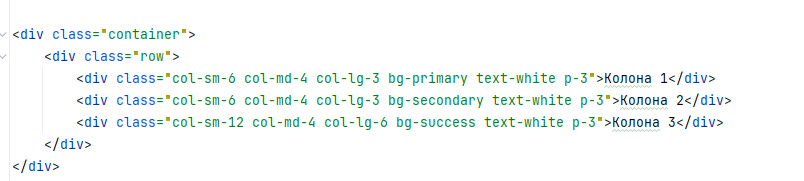
****

**Как работи това?**

Ако използваш col-sm-6, колоната ще заема 50% от ширината на екрана, но само когато екранът е по-голям от 576px.

Ако използваш col-lg-4, колоната ще заема 1/3 от ширината на екрана, но само когато екранът е по-голям от 992px.

**Пример с различни брейкпойнти**

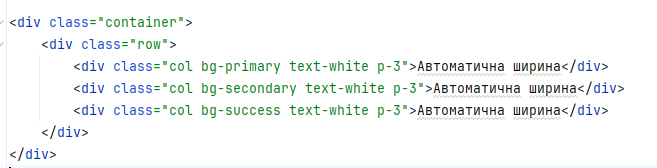
****

**Как се променя оформлението?**

* На малки екрани (sm) → първите две колони заемат 50%, третата заема 100%.
* На средни екрани (md) → всички колони заемат 1/3 от реда.
* На големи екрани (lg) → първите две колони заемат 25%, третата заема 50%.
  + - **Видове колони в Bootstrap**

В Bootstrap колоните могат да бъдат:

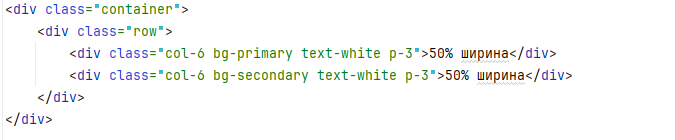
* **Автоматични** (.col)
* **С фиксирана ширина** (.col-4, .col-6)
* **Адаптивни спрямо брейкпойнти** (.col-sm-6, .col-md-4)

****

**Как работи?**

**Колоните автоматично се разпределят поравно независимо от размера на екрана.**

**Пример за фиксирани колони**

****

**Как работи?**

* Първата колона заема 50% от ширината.
* Втората също заема 50% от ширината.
  + Какво представлява Bootstrap.
  + История и развитие на Bootstrap.
  + Основни компоненти на Bootstrap:
    - Мрежова система (Grid System).
    - Компоненти (бутон, навигация, карусел и др.).
    - Утилити класове за оформление.
  + Предимства и недостатъци на Bootstrap.

## Мрежова система на Bootstrap

* + Как работи мрежовата система.
  + Примери за използване на мрежовата система в уеб дизайн.
  + Адаптивност и колони в Bootstrap.

# Практическа част

## Описание на проекта

* + Кратко представяне на разработения уебсайт за училищния клуб.
  + Основни цели на уебсайта:
    - Да представя информация за клуба.
    - Да бъде адаптивен и достъпен на различни устройства.
    - Да включва функционалности като форма за обратна връзка и навигация.
  + Изисквания към дизайна и функционалността.

## Избор на технологии

* + Обосновка за използването на HTML, CSS, JavaScript и Bootstrap.
  + Как Bootstrap улеснява създаването на адаптивен дизайн.

## Структура на уебсайта

* + Описание на основните секции на уебсайта:
    - **Начална страница**: Представяне на клуба.
    - **Секция "За нас"**: Информация за клуба и неговите цели.
    - **Секция "Контакти"**: Контактна информация и форма за обратна връзка.
  + Обяснение на използваните компоненти на Bootstrap.

## Проблеми и решения

* + Проблеми, срещнати по време на разработката (напр. адаптивност на различни устройства).
  + Как са решени тези проблеми.

## Резултати

* + Постигнати резултати от практическата част.
  + Как уебсайтът отговаря на изискванията за адаптивност и функционалност.

# Заключение

* Обобщение на постигнатите цели и резултати.
* Изводи относно значимостта на адаптивния дизайн и използването на Bootstrap.
* Предложения за бъдещо развитие на проекта:
  + Добавяне на допълнителни функционалности.
  + Интеграция с база данни за съхранение на информация.
  + Разширяване на дизайна с други библиотеки или технологии.

# Литература:

1. Колисниченко, Денис, Адаптивен уеб дизайн с Bootstrap, Асеневци, С 2019
2. D.K Academy, Python – практическо програмиране. Асеневци, С 2022
3. Донълдсън, Тоби. Бързо ръководство:Програмиране с Python.АлексСофт, С 2017
4. Есканази, Аврам. Софтуерни техологии. КЛМН, С 2006
5. https://bg.savtec.org/articles/coding/the-basics-of-rest-and-restful-api-development.html
6. https://urocibg.eu/
7. https://bg.wikipedia.org/wiki/MySQL
8. https://html.w3schools.bg
9. <https://www.inventum.bg/web-design/css/>
10. W3Schools. Bootstrap Tutorial. <https://www.w3schools.com/bootstrap/>
11. Smith, J. Responsive Web Design. 2020.

кйк