

ВЪВЕДЕНИЕ

Като ученик прекарах последните пет години в създаване на спомени и развитие на образованието ми. Въпреки че имаше и добри, и лоши моменти, реших да ги уловя и съхраня на уникално място - уебсайт. Избрах да създам уебсайт поради многото му предимства пред други алтернативи. Като универсално достъпна платформа чрез интернет, уебсайтът може да бъде достъпен от всяко място и по всяко време от компютър или мобилно устройство.

Уебсайтът предоставя много начини, с които може да се покажат най-ценните ни спомени от училище. Включвайки мултимедийни елементи като изображения, видеоклипове и аудио, уебсайтът не само съживява спомените, но и пренася към дадения момент. Функционалността му позволява да бъде модифициран и персонализиран до последния детайл, за да съживи картината независимо дали е чрез използване на графики или анимации.

Уебсайтът се състои от 5 основни страници, всяка със свой собствен стил. Всяка от страниците има различна функционалност. Началната страница се фокусира върху класните ръководители. Те ще имат последната дума в края на завършването ни. Учениците имаха за задача всеки от тях да обобщи в едно изречение отнасящо се за годините ни в ПГБТ. Албумът съдържа снимки от 2018 до 2023. Делнична програма, която пресмята колко време остава до края на текущия час или до края на междучасието. Показва кой учебен час е сега и следващия час или междучасие. Последната страница се отнася за мен. В нея има кратко описание на проекта и мотивацията ми.

Уебсайтът ще продължи да се поддържа до следващата учебна година. След това ще се трансформира в pdf файл и на всеки ще му бъде предоставено копие.

СЪДЪРЖАНИЕ

ВЪВЕДЕНИЕ.....	0
ГЛАВА 1. ИНТЕРНЕТ И УПОТРЕБАТА НА УЕБСАЙТОВЕ ЗА ЦЕЛИТЕ НА ПОТРЕБИТЕЛЯ	4
1.1. ИНТЕРНЕТ	4
1.2. КАКВО Е УЕБСАЙТ?	4
1.3. НАЧИН НА ДОСТЪП	6
1.4. СТРАТЕГИИ ЗА СЪДЪРЖАНИЕ	8
ГЛАВА 2. ИЗРАБОТВАНЕ НА УЕБСАЙТА	9
2.1. СЪЗДАВАНЕ НА УЕБСАЙТА	9
2.1.1. Описание.....	10
2.1.2. Среда за програмиране.....	11
2.2. СРЕДА ЗА ТЕСТВАНЕ	12
2.3. СТРАНИЦИ	12
2.3.1. Заглавна	13
2.3.1.1. Оформление	14
2.3.1.2. Списък на учители.....	17
2.3.1.3. Ръководители	19
2.3.1.4. Таймер.....	21
2.3.2. Ученици	22
2.3.2.1. Оформление	23
2.3.2.1. Карти.....	24
2.3.3. Албум	25

2.3.3.1. Оформление	26
2.3.3.2. Снимки.....	26
2.3.4. Програма.....	29
2.3.4.1. Оформление	30
2.3.4.2. Дневна програма - система.....	31
2.3.5. ЗаМен	32
2.3.5.1. Оформление	32
2.3.5.2. Текст.....	35
2.4. HEADER	35
2.5. FOOTER	36
2.6. ХОСТИНГ	37
2.7. ДОМЕЙН.....	38
ГЛАВА 3 ЕКСПЕРИМЕНТАЛНА ЧАСТ	40
ЗАКЛЮЧЕНИЕ	40
ПРИЛОЖЕНИЯ.....	40

ГЛАВА 1.

ИНТЕРНЕТ И УПОТРЕБАТА НА УЕБСАЙТОВЕ ЗА ЦЕЛИТЕ НА ПОТРЕБИТЕЛЯ

1.1. Интернет

Интернет е голяма мрежа, която свързва компютри и устройства по целия свят, позволявайки комуникация помежду си. Това става възможно чрез използването на протоколи, които са набор от правила, управляващи как устройствата комуникират едно с друго. Интернет ни дава достъп до огромно количество информация, което прави намирането на това, от което се нуждаем, по-бързо и по-лесно. Интернет улесни комуникацията ни с хора от всички краища на земното кълбо и ще продължава да разширява възможностите си.

1.2. Какво е Уебсайт?

Уебсайтът е съвкупност от уеб страници, достъпни чрез интернет. За да се създаде уебсайт, уеб разработчикът пише код, обикновено в HTML, CSS и JavaScript, който след това се поддържа в уеб сървър.

Уебсайтовете могат да се използват за различни цели. Много уебсайтове са предназначени да предоставят информация по определена тема. Например новинарските уебсайтове предоставят актуална информация за текущи събития, докато образователните уебсайтове предлагат ресурси за учене. Уебсайтовете могат да се използват за улесняване на комуникацията между отделни лица или групи. Социалните медийни платформи, например, позволяват на хората да се свързват помежду си и да споделят информация.

Уебсайтовете често се използват за целите на електронната търговия, което позволява на фирмите да продават продукти или услуги онлайн. Уебсайтовете за онлайн пазаруване, като Amazon и eBay, са примери за уебсайтове за електронна търговия. Уебсайтовете могат да се използват и за развлекателни цели. Онлайн игрите, платформите за стрийминг и социалните медии са примери за уебсайтове, които предоставят забавление. Някои уебсайтове дори комбинират тези различни функции, като платформи за социални медии, които позволяват на потребителите да се свързват помежду си, да споделят информация и да купуват продукти.

Уебсайтовете могат да се използват и за улесняване на сътрудничеството между отделни лица или групи. Например софтуерът за управление на проекти често включва компонент на уебсайт, който позволява на членовете на екипа да споделят файлове и да комуникират помежду си. Освен това много уебсайтове предоставят инструменти за създаване и споделяне на съдържание, като блогове, уикита и форуми. Тези инструменти позволяват на потребителите да си сътрудничат по проекти, да споделят информация и да участват в дискусии с други.

Като цяло уебсайтовете са съществена част от интернет, предоставяйки на потребителите достъп до информация, комуникация, търговия, забавление и сътрудничество. С милиарди уебсайтове онлайн, интернет се превърна в незаменим инструмент за хората по целия свят. Независимо дали търсим информация, свързваме се с приятели или извършваме бизнес, уебсайтовете ни позволяват да навигираме и да взаимодействаме с интернет по безброй начини.

1.3. Начин на достъп

Когато потребител иска да получи достъп до уебсайт, той въвежда URL адреса на уебсайта (Uniform Resource Locator) в своя уеб браузър. Браузърът изпраща заявка до сървъра на уебсайта за уеб страницата, която включва HTML код, изображения и друго съдържание, което съставлява страницата.

URL адресът на уебсайта всъщност е четим от човека начин за представяне на IP адрес, който е уникален идентификатор за устройство в интернет. Всяко устройство в интернет има IP адрес, който се използва за маршрутизиране на пакети с данни между устройствата. Въпреки това, IP адресите са трудни за запомняне от хората, така че DNS (система за имена на домейни) е създадена, за да съпостави четливи имена на домейни към IP адреси.

DNS е разпределена система, която се състои от хиляди сървъри по целия свят. Когато потребител въведе URL адреса на уебсайт в браузъра си, браузърът изпраща заявка до DNS резолвер, който обикновено се предоставя от ISP (доставчик на интернет услуги) на потребителя. След това резолверът изпраща заявка до основен DNS сървър, който връща IP адреса на домейн сървъра от първо ниво за името на домейна на уебсайта (като .com или .org).

Резолверът изпраща заявка до домейн сървър от най-високо ниво, който връща името на домейна на уебсайта. Сървър за имена е отговорен за запис на IP адресите на устройствата, свързани с името на домейна на уебсайта.

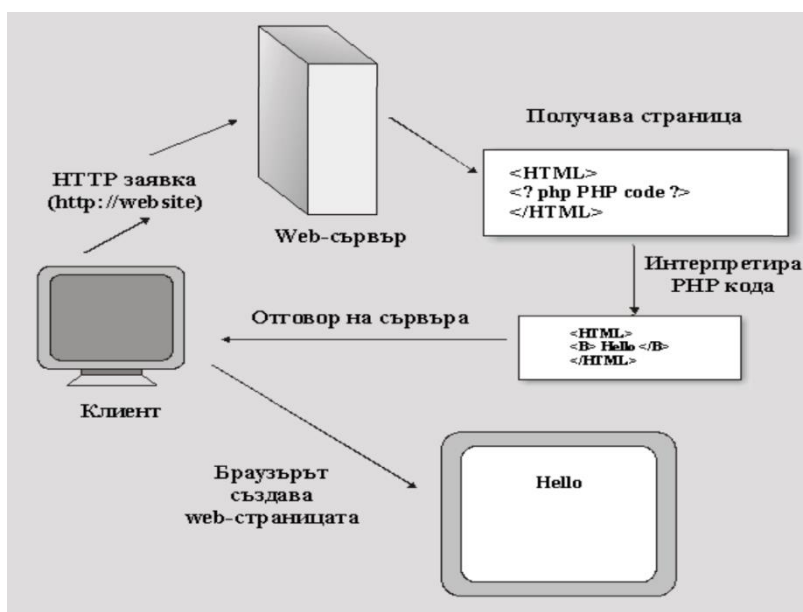
След като браузърът получи IP адреса на уебсайта, той изпраща заявка до сървъра за уеб страницата. Тази заявка включва информация за вида на използвания браузър, исканата страница и всички допълнителни параметри.

Сървърът получава заявката и започва да я обработва. Първо, сървърът може да провери дали исканата страница е в неговия кеш. Кеширането е процес на съхраняване на често достъпни данни в локално пространство за съхранение за по-бърз достъп. Ако исканата страница е в кеша, сървърът може да я върне веднага, без да се налага достъп до базата данни или други ресурси.

Ако страницата не е в кеша, сървърът извлича необходимите данни и генерира HTTP (Hypertext Transfer Protocol) отговор. Отговорът включва HTML код, изображения и друго съдържание, което съставлява уеб страницата, както и всякаква допълнителна информация, като например бисквитки.

Сървърът изпраща HTTP отговор обратно към браузъра на потребителя, който го получава и започва изобразяването на страницата. Браузърът използва HTML кода, за да покаже текста и изображенията на екрана.

След като страницата приключи изобразяването, потребителят може да взаимодейства със страницата, като кликва върху връзки, попълва формуляри или да извършва други действия. Когато потребителят щракне върху връзка, неговият браузър изпраща нова заявка до сървъра, който отговаря с подходящото съдържание за новата страница. Този процес продължава, докато потребителят напусне уебсайта или затвори своя браузър. По време на целия този процес сървърът на уебсайта непрекъснато комуникира с браузъра на потребителя, като обменя данни и изобразява съдържанието на страницата в реално време.



Фигура 1. HTTP заявки

1.4. Стратегии за съдържание

Стратегията за съдържание има голямо значение за създаването на уебсайта. В началото трябваше да направя проучване за това какво ще бъде най-подходящо да се показва на самия уебсайт, така че да бъде в контекст с

предстоящите събития. Това ми помага да определя кои теми ще бъдат най-подходящи. Търсех начини да създам комбинация от най-доброто съдържание и най-добрия дизайн съчетани в едно. Това в крайна сметка се оказва от ключово значение за темата на проекта.

ГЛАВА 2. ИЗРАБОТВАНЕ НА УЕБСАЙТА

2.1. Създаване на уебсайта

Уебсайтът е създаден посредством комбинация от софтуерни инструменти и програми за създаване. Те включват предимно текстови редактори, системи за актуализации, системи за управление на съдържанието,

скриптові езици от страна на сървъра и от страна на клиента, уеб сървъри, инструменти за тестване и отстраняване на грешки и инструменти за внедряване.

2.1.1. Описание

Сайтът съдържа 5 главни страници и 5 подстраници, всяка която изпълнява различна роля. На всяка една от страниците има Header, който служи за навигация към различните страници. При натискане на текста, който всъщност е бутон, с името на страницата посетителят на уебсайта ще се пренесе в съответната страница, върху която е натиснал. Header е форматиран с шрифт "Open Sans" и "Lobster", за да може да изглежда добре и лесен за навигация. В центъра на Header е изписано името на нашият клас, от ляво е текста(бутон) с навигация към началната страница, от дясно се намират и останалите страници "Ученици", "Албум", "Програма", "За мен". При посещение на всяка от страниците ще се активира текста и ще промени цвета си в златен цвят. Ще остане активен(текста(бутона)), докато не посети друга страница.

2.1.2. Среда за програмиране



Visual Studio Code е популярен, безплатен редактор на код, разработен от Microsoft. Едно от основните предимства на Visual Studio Code е неговата гъвкавост. Той поддържа широк набор от програмни езици, които използват различни технологии. Той също така има голяма и активна общност от потребители, които допринасят за неговото развитие и създават различни разширения, които подобряват неговата функционалност. VS Code има удобен за потребителя интерфейс с персонализирани теми, икони и оформления. Освен това има вградена поддръжка за контрол на версиите на Git и отстраняване на грешки. Освен това той предоставя функции като довършване на код, подчертаване на синтаксиса и сгъване на кода, които ни помагат да пишем код по-ефективно и с по-малко грешки.

Друга полезна функция на Visual Studio Code е неговият интегриран терминал, който позволява да изпълняваме инструменти и скриптове от командния ред, без да напускаме редактора. Това спестява време и улеснява превключването между кодиране и изпълнение на команди.

Visual Studio Code също поддържа различни разширения и плъгини, които могат да се използват за повече функционалност.

2.2. Среда за тестване

Live server е разширение за Visual Studio Code, което ни позволи да стартираме локален уеб сървър и да визуализираме уеб страниците си в браузъра с функция за презареждане в реално време. Това означава, че всички промени, които правим в нашия код, ще бъдат автоматично отразени в браузъра, без да се налага да презареждаме страницата ръчно. Това също ни позволи да синхронизираме уеб страниците на множество устройства и браузъри. Това означава, че можем да визуализираме уеб страници си на различни устройства едновременно, което е полезно за тестване. Live Server също има възможности за маркиране на грешки, което улеснява идентифицирането и коригирането на грешки в кода. Докато правихме промени в кода, използвахме и различни браузъри като Google Chrome, Mozilla Firefox и други, за да тестваме възможностите на уебсайта. Различните браузъри имат свои собствени уникални характеристики, които могат да повлияят на това как изглежда и се държи една уеб страница. Ето защо е важно да тествате уеб страници в множество браузъри, за да сме сигурни, че са съвместими и достъпни за всички потребители.

2.3. Страници

Страниците са 5 и всяка от тях изпълнява различна важна роля. Страниците са форматирана според идеята на създаването им и според

стилизирането в цветовата гама на уебсайта, като всяка страница съдържа Header и Footer.

2.3.1. Заглавна

Заглавната страница или началната страница е мястото където посетителят на уебсайта се натъква, след като въведе уеб адреса на уебсайта – 12vktptgbt.ml. Началната ни страница е и нашата главна страница. Точно затова, когато посетителя потърси и въведе домейн адреса, той ще отвори началната страница – index.html. Тази страница е пълно копие на заглавната ни страница, която е с файлово име Начало.html.

Заглавната страница се дели на няколко части. Първата от които е място в самото начало на страницата където са разположени няколко заглавия и кратък написан текст, който да изрази нашата благодарност към учителите. Класът, за който се отнася този уебсайт е изписан на едно от заглавията. Под него е випускът и след това специалността изучавана по време на престоя ни.

Спускаме се по-надолу и откриваме списък с автоматично сменящи се имена. Това е списък на поканените учители, които единодушно с класа решихме да поканим на нашия бал или нашето финално завършване или разделяне. Списъкът е интерактивен и всеки може да го прегледа от всяко мобилно устройство. Има и стрелки, които служат за навигация на сменяне на имената.

По нататък в страницата откриваме златно блестящ и лъскащ текст разположен в центъра на секцията. Тази секция е по-специална от другите, защото има две снимки. Те представляват нашите класни ръководители и

техните снимки. Зад тях е създаден ефект на златен дъжд с помощта на инструменти и видео от платформата YouTube.com. Снимките имат и друга функционалност а тя е създаване на две нови секции или контейнери. Новите секции или контейнери се появяват с красива анимация при натискане на която и да е от снимките на двата ни класни ръководители. Новите контейнери съдържат снимка и текст под снимката, който цели да покаже финалната реч на нашите класни ръководители.

В края намираме златен текст изказващ датата на абитуриентския ни бал. Под него се намира таймер, който отброява дните, часовете, минутите и секундите оставащи до двадесет и пети май, когато всъщност е нашият бал. Зад текстовите елементи се намира canvas елемент, който създава ефект на празничност и дълбока потайност с графиките си. Фона е черен като най-специалното в този елемент са падащите звезди (пиксели) от програмата, която създадохме.

2.3.1.1. Оформление

В края намираме златен текст изказващ датата на абитуриентския ни бал. Под него се намира таймер, който отброява дните, часовете, минутите и секундите оставащи до двадесет и пети май, когато всъщност е нашият бал. Зад текстовите елементи се намира canvas елемент, който създава ефект на празничност и дълбока потайност с графиките си. Фона е черен като най-специалното в този елемент са падащите звезди (пиксели) от програмата, която създадохме.

Първата секция има сив цвят на фона (grey-80) и включва две заглавия и абзац. Първото заглавие е оформено с персонализиран шрифт, наречен „Lobster“ и е в бяло. Той показва текста "12В клас Випуск 2023" с прекъсване на ред (
) между "клас" и "Випуск". Второто заглавие също е оформено с шрифта „Lobster“, но има по-голям размер (50px) и горен отстъп от 40px (margin: 40px;). Той показва текста „Компютърна техника и технологии 2018-2023“ с прекъсване на реда между „техника и технологии“ и „2018-2023“. Параграфът под заглавията е центриран и включва няколко реда текст, който благодари на учителите и отправя покана към тях за бала.

Втората секция има по-светло сив цвят на фона (grey-75) и включва заглавие и параграф. Заглавието е с размер и цвят на шрифта по подразбиране и показва текста "Поканени учители". Параграфът е с по-голям размер на шрифта и показва текста "Ако не сте в този списък, не се сърдете моля!".

Третата секция има ефект на въртележка с пет елемента, които се плъзгат хоризонтално. Той включва лента за навигация с пет точки под въртележката, за да посочи текущата позиция. Всеки елемент е div със сив цвят на фона (grey-75), центрирано съдържание и заглавие, което показва името на учителя.

Четвърта секция има фоново видео и раздел със съдържание с две препоръки от учители. Фоновото видео е настроено да се възпроизвежда автоматично. Разделът със съдържание има заглавие в персонализиран стил на шрифта „Класни ръководители“, което се превежда на „Класни учители“ на английски. Фонът на видеото се постига чрез вграждане на видеоклип в YouTube в елемент на iframe. Вградената рамка е стилизирана така, че да заема

цялата ширина и височина на своя контейнер, и е настроена на автоматично възпроизвеждане, повтаряне и заглушаване. Съдържа и два елемента от карти, които представляват учителите. Картите се създават с помощта на елемента `div` с клас `list-item`. Във всяка карта има елемент `h3`, представляващ името на учителя, и елемент `p`, съдържащ цитат от учителя. Картите са стилизирани с шрифта `Lobster`, включва рамка и `zoom` ефект на снимката.

Следващите две секции 5 и 6 съдържат `div` с класа `"sheet"`, който се използва за оформяне на съдържанието на раздела, включително изображение, заглавие и параграф от текст. Първоначално секциите са скрити с помощта на вградените стилове `"opacity: 0; display: none;"`. Идентификаторите `„section-5“` и `„section-6“` се използват за идентифициране на тези секции в JavaScript кода. Причината, поради която елементът `div` се използва тук, е да осигури контейнер за съдържанието на раздела, което позволява по-лесно стилизиране и оформление. Елементът `div` е контейнерен елемент на ниво блок, който може да съдържа други елементи, като изображения, заглавия и параграфи. Чрез използването на елемента `div` с класа `"sheet"`, съдържанието на всяка секция се съдържа в един контейнерен елемент, което опростява стила на съдържанието и улеснява прилагането на последователно оформление към всички секции.

Секция 5 и 6 има JavaScript към тях, който дефинира четири променливи: `„Cvetkova“`, `„Qneva“`, `„imageContainer“` и `„imageContainer2“`. `„Cvetkova“` и `„Qneva“` са препратки към HTML елементи с ID съответно `„cvetkova“` и `„qneva“`. Кодът на JavaScript добавя `event listener` към елементите `„Cvetkova“` и `„Qneva“`, така че когато се щракне върху тях, функцията `„toggleSection()“` се извиква с `„imageContainer“` или `„imageContainer2“` като аргумент. Функцията

"toggleSection()" приема един аргумент, който е препратка към една от двете секции, дефинирани в HTML кода. Функцията първо проверява стойността на свойството CSS "display" на секцията. Ако е „none“, функцията задава свойството „display“ на „block“, задава непрозрачността на 0 и добавя ефект на преход, който постепенно избледнява в секцията за половин секунда. Накрая непрозрачността се задава на 1 след кратко забавяне. Ако свойството "display" вече е зададено на "block", функцията задава непрозрачността на 0, изчаква половин секунда и след това задава свойството "display" на "none", за да скрие секцията.

Секция 7 включва div с елемент canvas, който има ID „motion“. Над него има таймер за обратно отброяване, който показва дни, часове, минути и секунди до 25 май 2023 г. Обратното броене се актуализира всяка секунда от функция на JavaScript, която задава стойностите на съответните HTML елементи. JavaScript кодът инициализира променлива countDownDate до датата и часа на 25 май 2023 г. в полунощ. След това дефинира функция, която се изпълнява всяка секунда с помощта на setInterval. Функцията изчислява времевата разлика между текущия час и датата на обратното броене и изчислява броя на оставащите дни, часове, минути и секунди. След това задава вътрешния HTML на съответните HTML елементи на изчислените стойности. Това води до актуализиране на таймера за обратно броене всяка секунда.

2.3.1.2. Списък на учители

Така сме представили списък с имена, които се сменят периодично през 5 секунди.

Списъкът се дефинира с помощта на контейнерен елемент с клас `carousel-inner`. Целта на този контейнерен елемент е да побере всички слайдове или елементи, които ще бъдат показани във списъка.

Комбинацията от HTML и JavaScript код създава въртележка или плъзгач, който показва различни слайдове с информация за инженерите. HTML кодът създава контейнер за въртележката с всеки слайд, дефиниран като `div` елемент с уникално име на клас. Всеки слайд съдържа заглавен елемент с името на инженера.

Кодът на JavaScript инициализира въртележката и дефинира функции, които обработват автоматичното плъзгане на списъка, ръчна навигация чрез щракване върху бутоните със стрелки и пауза на автоматичното плъзгане, когато потребителят задържи курсора на мишката върху списъка. JavaScript кодът започва с избиране на контейнера и всички слайдове с помощта на методите `querySelector` и `querySelectorAll`. Той също така определя интервала за автоматично плъзгане и променлива за проследяване на текущия индекс на слайда.

Функцията `showSlide()` се извиква, за да покаже слайда с посочения индекс чрез добавяне на име на клас `active` към елемента `div` на слайда и премахване на името на класа от всички останали елементи `div` на слайд.

Функцията `changeSlide()` се извиква, за да промени индекса на текущия слайд с указаната стойност (1 или -1) и гарантира, че индексът остава в обхвата на наличните слайдове. След това извиква `showSlide()`, за да покаже новия слайд.

Функцията `automaticSlide()` увеличава индекса на слайда с 1 и извиква `changeSlide()`, за да покаже новия слайд.

Методът `setInterval()` се използва за автоматично извикване на `automaticSlide()` на посочения интервал и съхраняване на получения идентификатор в променлива, за да може по-късно да изчисти интервала.

Два детектора на събития се добавят към списъка, за да поставят на пауза автоматичното плъзгане, когато потребителят задържи курсора на мишката върху въртележката, и да възобновят автоматичното плъзгане, когато потребителят напусне зоната на въртележката.

И накрая, два детектора на събития се добавят към бутоните със стрелки наляво и надясно, за да позволят на потребителя да навигира ръчно в списъка. Тези слушатели изчистват интервала на автоматично плъзгане и извикват функцията `changeSlide()` с подходящата стойност за показване на предишния или следващия слайд.

2.3.1.3. Ръководители

Класните ни ръководители имат специално място на нашия уебсайт. Мястото, на което се намират представлява падащ златен дъжд. HTML кода използван за тази цел вгражда видеоклип в YouTube със специфични настройки в уеб страница. Използваме елемент `div` с класа `"background-video"`, за да създадем раздел за видеото. Вътре в този `div` поставяме друг `div` с класа `"embed-responsive embed-responsive-1"`, който се използва за адаптивно вграждане на видео. След това вмъкваме `iframe` елемент във втория `div`, който зарежда видеоклипа в YouTube с посочените настройки. Видеото е настроено

на автоматично пускане, повтаряне и заглушаване. Освен това видео информацията и контролите са скрити, за да осигурят безпроблемно гледане. Видеото започва да се възпроизвежда винаги от самото начало. За да накараме видеото да заема цялото пространство, прилагаме CSS стила "position: absolute;top: 0;left: 0;width: 100%;height: 100%;" към елемента `iframe`.

Снимките са разположени централно на страницата и имат способността при натискане да отворят 2 нови секции. Те представляват снимка и текст на класните ни ръководители, написан от тях. Това се състои от две секции. Всеки раздел се дефинира с помощта на маркера за раздел с уникален идентификатор (`id`), който може да се използва за насочване към него в JavaScript кода.

Във всеки раздел има `div` елемент с клас `sheet`, който се използва за създаване на контейнер за съдържанието на раздела. Този клас прилага някои основни стилове към контейнера, като например центрирането му върху страницата и добавянето на подложка.

В рамките на `div sheet` има елемент `h1`, който се използва за показване на заглавие за раздела. Към заглавието има приложени няколко CSS класа, които контролират външния му вид. Тези класове включват `align-center`, който центрира заглавието хоризонтално, `font-Calibri`, който определя персонализиран шрифт за текста, и `text-white`, който задава цвета на текста по подразбиране за страницата.

След това има елемент `img`, който показва изображение във всяка секция. Класът `image` се прилага към елемента, заедно с някои други класове, които контролират размера, формата и позицията на изображението. Те

включват `align-center-xs`, който центрира изображението хоризонтално на малки екрани, `image-round`, който закръгля ъглите на изображението и `radius-25`, който задава радиуса на границата на изображението на 25 пиксели.

Към текста на класните ни ръководители са приложени няколко CSS класа, които контролират външния му вид. Тези класове включват `align-justify`, който оправдава текста, `text-body-alt-color`, който задава цвета на текста на алтернативен цвят и `text-default-lg`, `text-default-md`, `text-default-sm` и `text-default-xl`, които контролират размера на текста при различни размери на екрана.

2.3.1.4. Таймер

Последната секция представлява таймер разделен на дни, часове, минути и секунди, които отброяват времето оставащо до абитуриентския ни бал. Те са разположени централно и еднакво подравнени. Секцията е съставена от няколко различни елемента, които се използват за показване на информация за събитието и таймера за обратно отброяване.

Класът `"align-center"` се използва за центриране на съдържанието на секцията. Класовете `"border-6"` и `"border-palette-3-base"` добавят граница с ширина 6 пиксела към секцията с цветова палитра от `palette-3`. Класът `"clearfix"` се използва за изчистване на всякакви плаващи елементи и установяване на нов контекст за форматиране на блок, който е полезен за съдържане на елементите в секцията. Класът `"image"` се използва за показване на изображение като фон на раздела, а класът `"section-5"` се използва, за да даде на раздела уникален идентификатор.

В рамките на раздела има елемент `div`, който съдържа елемент `canvas` с идентификатор „motion“. Елементът `canvas` се използва за показване на анимация на фона на секцията.

Използваме елемент `div` с клас "list-1", който съдържа няколко `div` с клас "repeater-item". Всеки от тези `div` съдържа таймер за обратно отброяване, който показва колко дни, часове, минути и секунди остават до събитието. Таймерът за обратно отброяване се генерира динамично с помощта на JavaScript.

JavaScript кодът използва функцията `setInterval`, за да актуализира таймера за обратно отброяване всяка секунда. Той изчислява разликата във времето между текущия час и датата на обратно броене и след това изчислява колко дни, часове, минути и секунди остават до събитието. След това тези стойности се използват за актуализиране на HTML елементите в секцията със съответните идентификатори. Таймерът за обратно отброяване се създава чрез актуализиране на свойството `innerHTML` на всеки елемент със съответния идентификатор.

2.3.2. Ученици

Целта на тази страница е да представи всеки ученик от класа с негова снимка и изречение или цитат, с което да опише престоя си в ПГБТ. Те имаха право да изберат своя снимка или да изкачат до бала, от който ще се сдобием с хубави снимки. Цитатът или изречението, което трябва да напишат трябва да е нещо, което те са усетили през престоя си в гимназията. Имат право да напишат всичко, което сметнат за правилно.

Снимката и цитатът всъщност представляват карта, която ученикът може да персонализира. Лицето на картата е самата снимка на ученика с името

му в долния ляв ъгъл. При натискане на картата или снимката ще се създаде анимация на завъртане и гърба на картата ще се появи. Те също така имаха право персонализират картата си.

Вариантите бяха:

- Да се добави друга снимка по избор на ученика.
- Да се добави background снимка на гърба на картата.
- Да се добави бутон, който да включва звук, избран от ученика.
- Текста или цитата са напълно избираеми от всеки ученик.

Каквито идеи бяха предложени, такива бяха имплементирани. Никой не беше разочарован!

2.3.2.1. Оформление

Първият елемент на раздела има два вложени елемента div, единият от които съдържа елемент h1 с текст „Ученици“. Този раздел има класовете "align-center", "clearfix", "palette-2-dark-2" и "section-1".

Втората секция има два елемента div. Първият div съдържа класовете "list-1" и "repeater-1", които се повтарят два пъти в div "flip-card". Всеки div „flip-card“ съдържа два вложени div: един с клас „flip-card-front“, който показва името на ученика, а другият с клас „flip-card-back“, който показва гърба на картата с цитат, две икони на социални медии и съответните им връзки.

Има и няколко други div елемента с различни класове и стилове, като "sheet", "valign-middle", "image-round", "container-style", "radius-15" и "u-

shading". Накрая има елемент за връзка, който препраща към външни таблици със стилове. Отнася се за шрифта "Pacifico" от Google Fonts.

2.3.2.1. Карти

Картите както споменахме по-горе са основен елемент в тази страница. Те представляват лице с снимка на ученика и текст с името на ученика, който се намира долу в ляво. И гръб с текст или цитат и свободно избираеми елементи по избор.

Допълнителните елементи са:

- Снимка
- Видеоклип
- Текст
- Звук
- Анимация
- Допълнително форматиране на картата (по избор)
- Линк към социална медия (Facebook, Instagram, Github...)

Създадохме флип картата с предна и задна страна, всяка със собствен набор от CSS класове. Предната страна съдържа изображение и заглавие, а задната страна съдържа цитат и две връзки към профили в социални медии (Facebook и Instagram).

JavaScript кодът включва две функции за отваряне на профилите във Facebook и Instagram в съответните им мобилни приложения, ако има такива. Ако приложението не е налично, връзката ще се отвори в браузъра с нов раздел.

Кодът jQuery съдържа няколко event listeners за различни елементи. Когато потребител щракне върху обърнатата карта, класът „активен“ се включва и изключва, за да създаде анимацията за обръщане. Когато потребителят щракне върху бутон или връзка, event listener спира по-нататъшното разпространение на събитието. Освен това има два event listener за бутон с класа "myButton", който възпроизвежда и поставя на пауза аудио файл съответно при натискане на мишката.

2.3.3. Албум

Както вече споменахме по-рано в дипломния проект снимките и видеосъдържанието се намират точно в нашия албум. Той се състои от 1 главна страница и 5 подстраници, които всяка от тях изпълнява различна роля, но с еднаква цел. Тук ще разгледаме как сме оформили съдържанието, конвертирали, сортирали и накрая показали снимките и видеосъдържанието.

Главната страница на албума служи като навигация за отделните години, в които сме разположили нашите снимки. На нея намираме навигационни бутони с текст годината, която искаме да посетим.

Нашите снимки и видеоклипове се нуждаят от специална обработка, за която ще използваме набор от трикове и инструменти. След обработката на снимките, те ще се покажат на страницата с помощта на библиотека наречена Nanogallery2. Тази библиотека е най-добрият вариант за нашия дипломен проект, тъй като предоставя възможности да се персонализира.

2.3.3.1. Оформление

Първата секция има клас "black clearfix section-1" и id "section-1". Той съдържа div елемент с клас "clearfix sheet sheet-1" и два дъщерни елемента. Първият дъщерен елемент е h1 заглавен елемент с класове "align-center custom-font font-lobster text text-1". Заглавният текст е "Албум". Вторият дъщерен елемент е p параграфен елемент с класове "align-center custom-font font-merriweather text text-2". Текстът на параграфа е „Албумът съдържа снимки в периода от 2018-2023 г.“.

Втората секция има класове "align-center black clearfix section-2" и id "sec-1a96". Той също така съдържа div елемент с клас "clearfix sheet sheet-1" и пет дъщерни елемента. Всеки дъщерен елемент е h2 заглавен елемент с класове "custom-font font-lobster text text-default" и дъщерен елемент anchor. Първият елемент на котва има клас "active-none border-none btn button-link button-style hover-none none text-active-custom-color- 1 text-hover-grey-40 text-white btn-1" и href атрибут на "Албум8.html". Анкорният текст е "8 клас". Останалите четири елемента на котва имат подобни класове и href атрибути, но различен текст на котва и имена на класове (btn-2, btn-3, btn-4, btn-5), представляващи различни нива на клас (9-то до 12 клас).

2.3.3.2. Снимки

Снимките са важна част от нашия уебсайт, тъй като представляват уловени спомени от конкретен момент. Те са общо около 500 и са събрани от всякакви източници като: Facebook, Instagram, галерии и други. Снимките са разположени в библиотека за прожектиране на снимки. Събрани са от

началото на престоя ни заедно с класа до абитуриентския бал. Те са групирани по класове, пример: 8 клас – 2018 до 2019 година. Те са официални и не толкова официални. Тъй като снимките са много и се налага да се направят няколко необходими оптимизации, които са важни за финалния продукт. Разглеждаме ги в няколко стъпки, които следвахме докато създавахме албума.

Първата стъпка е да разделим снимките по дати и събития. Много е важно тази стъпка да се случва с необходимото време и внимание, тъй като е много лесно да объркаме времето и мястото на конкретната снимка, ако тя няма правилно зададена дата и час на създаване.

Втора стъпка е да изберем снимките, които искаме да конвертираме с цел оптимизация. От собствения си файлов формат ще се трансформират в WebP файлов формат. В сравнение с други формати на изображения, като JPEG и PNG, WebP предлага по-добро компресиране и по-малки размери на файловете, което помага за времето за зареждане на уеб страницата и намалява разхода на интернет трафик.

WebP също е силно съвместим с различни уеб браузъри, включително популярни като Google Chrome, Microsoft Edge и Mozilla Firefox. Това означава, че повечето потребители ще могат да преглеждат WebP изображения без проблеми.

Трета стъпка е да компресиране WebP файловете. WebP изображенията се компресират, за да се намали размерът на файла и да се улесни зареждането им. Компресирането на този вид формат не е задължително. Но в нашия случай, тъй като имаме голямо количество снимки, които трябва също да могат

да се заредят и от мобилни устройства с мобилни данни, ние трябва да намалим размера им възможно най-много. Размерът на файла ще се намали с около 20%, за да не доведе до известна загуба на качество на изображението.

Четвърта стъпка е сортирането на снимките. То се осъществява чрез програма, която ние създадохме и има за цел да преименува всички снимки от 001 до NNN според датата на създаването им.

Пета стъпка е да качим нашите снимки в HTML страницата ни. Това се осъществява с нашата среда за програмиране Visual Studio Code, като прибавяме всяка снимка на отделен ред и според номерацията ѝ в `<div>` тага, който е всъщност нашата галерия.

Създадохме код на HTML, който показва списък с връзки към албума ни със снимки за различните изминали училищни класове (8-ми до 12-ти клас). Всяка връзка е представена от таг `<a>` в таг `<h2>` и има персонализиран клас и стил, приложен към него, за да създаде външен вид, подобен на бутон.

Когато щракнем върху една от тези връзки, ще се пренасочим към нова HTML страница, която показва албума със снимки за съответния клас.

Фотоалбумите са създадени с помощта на библиотека на JavaScript, наречена Nanogallery2, която предоставя начин за показване на изображения в различни стилове, като миниатюри, зидани решетки и слайдшоу. Кодът, който създадохме дефинира `div` елемент с клас `"galeriq"` и ID `"galeriq8"` за 8 клас. Елементът `div` съдържа множество закрепени елементи, всеки от които представлява снимка или видеоклип в албума. Атрибутът `href` на всеки `anchor`

елемент сочи към изходния файл на снимката или видеото намиращи се в папка на сайта.

В края на кода библиотеката Nanogallery2 се инициализира чрез извикване на функцията `nanogallery2` на елемента `div` с клас `"galeriq"`. Функцията приема JavaScript обект като параметър, който указва различни опции за конфигуриране на галерията, като размер на миниатюра, ефекти на преход и инструменти за преглед. Свойството `"thumbnailWidth"` задава ширината на миниатюрните изображения на 300 пиксела, докато свойството `"thumbnailHoverEffect2"` дефинира ефект на задържане, който мащабира миниатюрното изображение и добавя по-светла рамка. Свойството `"viewerToolbar"` указва кои бутони трябва да се показват в лентата с инструменти на визуализатора, докато свойството `"viewerTools"` определя позицията на инструментите за преглед (като бутони за мащабиране и завъртане) в прозореца на визуализатора.

2.3.4. Програма

Тази страница представлява делничната програма на всеки ученик от 12 В клас. Тя се разглежда със снимки и система за пресмятане на текущите часове. Между часовете се намират междучасията, които също се пресмятат от системата. На първия елемент се показва текущия час или междучасие. Втория елемент представлява времетраенето до края на съответния елемент (час или междучасие). Третия елемент е с две думи – какво следва. Часовете се пресмятат и изреждат автоматично от програмата или системата, която създадохме. Времетраенето се показва в минути.

2.3.4.1. Оформление

Тази страница има за цел да представи програмата по красив и достъпен начин, затова създадохме няколко приложения.

Страницата е подредена както по предназначение, така и по дизайн. Всяка важна част е отделена от друга, за да може да се открият.

Първата секция има клас "section-1". Той съдържа заглавен таг „h1“ с персонализиран шрифт и атрибути на класа за стилизиране. Заглавният текст е „Текущи часове“. Освен това има етикет за изображение "img" с персонализиран клас и атрибути за стилизиране. Източникът на изображението е "images/image1.png" и има персонализирани атрибути за данни за ширина и височина.

Вторият раздел има клас "section-2" и идентификатор "carousel_7680". Той съдържа пет "h1" тагове с персонализирани шрифтове и класови атрибути за стилизиране. Първият таг "h1" има текста "Текущ клас" и няма персонализирани атрибути на класа. Вторият таг "h1" има ID "chasNow" и персонализиран шрифт и атрибути на класа. Текстът в този етикет ще бъде динамично актуализиран с текущия час от деня. Третият и четвъртият тагове "h1" имат съответно текста "Remains" и ID "chasRemain". Текстът в тага "h1" с ID "chasRemain" също ще бъде динамично актуализиран с оставащото време до края на текущия клас. Петият таг "h1" има текста "Следващ клас" и ID "chasNext". Текстът в този етикет ще бъде динамично актуализиран с името на следващия клас.

Третият раздел има клас "section-3" и идентификатор "carousel_3a15". Той съдържа таг "div" с персонализиран клас и атрибути за стилизиране. В рамките на този таг „div“ има таг „carousel“ с потребителски клас и атрибути на данни за стилизиране. Този таг "carousel" е галерия, която показва поредица от изображения и има плъзгащ ефект. Той има пет маркера „li“ с персонализираны атрибути за клас и данни, които представляват всеки слайд в галерията. Всеки таг „li“ има различен източник на изображение и персонализиран таг за заглавие „h3“ и таг за абзац „p“, които ще се показват като надпис за всеки слайд.

2.3.4.2. Дневна програма - система

Дневната програма представлява часовете от програмата в училище, като към тях прибавяме и междучасията. За целта трябва да създадем програма, която да извършва тези операции.

HTML кодът, който създадохме дефинира раздел с идентификатор "carousel_7680" и няколко заглавия с различни стилове и идентификатори. Тези заглавия ще се използват за показване на текущото събитие, оставащото време до следващото събитие и името на следващото събитие.

Кодът на JavaScript дефинира функция с име "checkTime", която се извиква всяка секунда с помощта на метода setInterval. Тази функция използва библиотеката luxon за работа с дати и часове. Първо инициализира някои променливи и след това използва команда за превключване, за да зададе графика на събитията за текущия ден от седмицата.

След това функцията преминава през събитията в графика и проверява дали текущият час попада в началния и крайния час на всяко събитие. Ако това стане, функцията актуализира заглавията, за да покаже информация за текущото събитие и оставащото време до края на събитието.

Ако текущият час не попада в нито едно събитие в графика, функцията проверява дали е преди или след края на учебния ден (който се предполага, че е 14:00). Ако е преди края на учебния ден, функцията изчислява оставащото време до началото на първото събитие за деня и показва тази информация. Ако е след края на учебния ден, функцията показва съобщение, че училището е приключило.

И накрая, функцията задава стойността на променливата "nextState" на името на следващото събитие (ако е намерено такова) и актуализира заглавието с ID на "chasNext", за да покаже тази информация.

2.3.5. ЗаМен

Тази страница представлява кратко описание на самия уебсайт и думите на автора, тоест Аз. Те са разделени по две заглавия, на които автора отговаря с добре оформен и четлив текст, за да може читателя да вникне в това, което искаме да кажем. Едно от заглавието е „Каква цел има този уебсайт?“, което всъщност е въпрос, с който описваме предназначението на уебсайта.

2.3.5.1. Оформление

Тази страница има само една секция, в която се изобразяват всички елементи на страницата.

Тагът „section“ с класове "align-center black clearfix section-1 css-selector" и идентификатор "sec-fb19" създава тази секция на уеб страницата. Клас "section" задава няколко стила за секцията, включително центриране на съдържанието хоризонтално, задаване на черен цвят на фона и изчистване на всички плаващи елементи в секцията. Атрибутът id се използва за идентифициране на тази конкретна секция в кода и може да се използва за CSS и JavaScript цели.

В рамките на раздела имаме елемент div с атрибут на клас "clearfix sheet-1". Този елемент div се използва като контейнер за съдържанието в секцията. Атрибутът клас задава стилове, за да изчисти всички плаващи елементи в div и да осигури външен вид, подобен на лист.

Елементът на изображението е с класове "image image-round radius-26 image-1 tracking-in-contract-bck" и src атрибут на "images/Screenshot_230.png" се използва за показване на изображение на уеб страницата. Атрибутът клас задава стилове, за да направи изображението кръгло, задава радиуса на границата на 26 пиксела и прилага към изображението анимация tracking-in-contract-bck.

Двата елемента h1 с класове "custom-font font-lobster text text-default title text-1 text-focus-in" и "custom-font font-ubuntu text text-default text-2", съответно се използват за показване на текст в определен стил на шрифт. Първият елемент h1 има персонализиран шрифт „Lobster“, докато вторият елемент h1 има персонализиран шрифт „Ubuntu“. Атрибутите на класа задават стилове за цвят, размер и подравняване на текста.

Двата р елемента с идентификатори съответно "men" и "cel" се използват за показване на текст на уеб страницата. Идентификаторите се използват за насочване към специфичните р елементи в кода на JavaScript по-късно. Атрибутът class и за двата р елемента задава стилове за подравняване на текст, семейство шрифтове и полета. Освен това елементът „мъже“ има персонализирана подложка от 500 пиксела, зададена в долната част на елемента.

Скриптовият елемент <script> в края на кода съдържа JavaScript код, който използва библиотеката TypeIt. Тази библиотека предоставя лесен начин за създаване на анимации за писане на уеб страници. Първият текст с TypeIt се създава с помощта на селектора "#men" и задава скоростта на 20 знака в секунда, задава курсора на false и задава опцията "waitUntilVisible" на true. Тази опция гарантира, че анимацията не започва, докато елементът не се види на екрана.

Вторият текст с ефект на писател от TypeIt се създава с помощта на селектора "#cel". Подобно на първия случай, той задава скоростта на 20 и курсора на false, но също така извиква метода "move" с нулев параметър и обект със свойството "to", зададено на "END". Това премества курсора в края на съдържанието на елемента. След това извиква метода "break" два пъти, за да създаде две нови линии, въвежда низа "text..." и накрая извиква метода "go", за да започне анимацията за въвеждане.

2.3.5.2. Текст

Текста в тази страница се изобразява посредством инструмент, който трябваше да създадем и да персонализираме с помощта на библиотеката TypeIt.

```
<script src="https://unpkg.com/typeit@8.7.1/dist/index.umd.js"></script>
```

С горе показания код вкарахме библиотеката, от която се нуждаем в този случай. Библиотека има много възможности и приложения и ние можем да се възползваме от тях.

Създадохме код на HTML, който съдържа няколко елемента h1, всеки с уникален атрибут id. Елементите h1 съдържат текст, който ще бъде актуализиран от JavaScript. Има също таг на скрипт, който зарежда библиотеката TypeIt от CDN.

Кодът на JavaScript създава два екземпляра на класа TypeIt. Първата инстанция е насочена към елемент с id "men" и въвежда низ със скорост от 20 знака в секунда. Вторият екземпляр е насочен към елемент с идентификатор "cel", въвежда низ, премества курсора до края на текста, вмъква два нови реда, въвежда последен низ и след това завършва.

2.4. Header

Header включва две навигационни менюта, едно отдясно и едно отляво, които са скрити или показани в зависимост от размера на екрана.

Дясното меню за навигация се съдържа в таг <nav> и има класа „menu-one-level menu-open-right offcanvas menu-1“. Той е подравнен вдясно и включва

икона на хамбургер (3 ленти една под друга), която превключва менюто при щракване. Когато менюто е отворено, връзките се показват вертикално.

Лявото меню за навигация също се съдържа в таг `<nav>` и има класа „menu-dropdown offcanvas menu-2“. Той е подравнен вляво и включва падаща икона, която превключва менюто при щракване. Когато менюто е отворено, връзките се показват хоризонтално.

Заглавието на уебсайта се съдържа в таг `<h3>` с клас „font font-lobster text text-default text-white text-1“. Логото е текстът "12В ПГБТ", който е стилизиран с шрифт "Lobster".

2.5. Footer

Долният колонтитул на уебсайта е съществен компонент от дизайна на сайта. Показва се в долната част на всяка страница и предоставя информация, която е важна за посетителите на сайта. Долният колонтитул на нашия уебсайт е кодиран в HTML и съдържа няколко елемента, които персонализираме според нашите предпочитания.

Нашият долен колонтитул има две основни секции. Първият раздел е прост параграф, който е центриран и подравнен с останалото съдържание на страницата. Този параграф съдържа малък текст, който гласи „Правено от сърце“. Това е нашият начин да изразим благодарност към нашите посетители и да им покажем, че държим на качеството на нашата работа.

Вторият раздел на долния ни колонтитул съдържа връзка към нашата страница "За мен". Тази страница предоставя допълнителна информация за

лицето или хората зад уебсайта. Връзката е стилизирана като бутон и има текст „За мен“. Това позволява на посетителите да научат повече за мен – създателя на уебсайта.

Добавихме и допълнителен стил към долния колонтитул, за да го направим по-привлекателен визуално. Например използвахме класа "grey-75", за да дадем на долния колонтитул сив цвят на фона, който контрастира добре с останалата част от дизайна на страницата. Използвахме и класа „text-variant“, за да придадем на текста малко по-различен цвят от останалата част от текста на страницата, което му помага да се открие.

2.6. Хостинг

GitHub Pages е безплатна услуга, която ни позволи да хостваме уебсайта директно от хранилище на GitHub. Всичко, което трябваше да направим, е да създадем хранилище с кода на уебсайта си и да активираме GitHub Pages в настройките на вашето хранилище. След това уебсайтът ви ще бъде достъпен на URL като `username.github.io/repository-name`.

Едно от страхотните неща при хостването на уебсайта ви на GitHub Pages е, че идва с вграден контрол на версиите. Това означава, че можете лесно да правите промени в кода на уебсайта си и да прилагате тези промени само с няколко кликания. Можете също така да си сътрудничите с други хора на уебсайта си, като използвате системата за изтегляне на заявки на GitHub.

Важно е да се отбележи, че GitHub Pages е специално проектиран за хостване на статични уебсайтове. Това означава, че ако вашият уебсайт

изисква обработка от страна на сървъра или база данни, ще трябва да потърсите други опции за хостинг.

По отношение на разликите между динамични и статични уебсайтове, статичният уебсайт е колекция от уеб страници, които са фиксирани и не се променят, освен ако разработчикът не ги актуализира ръчно. Те обикновено се състоят от HTML, CSS и JavaScript файлове. От друга страна, динамичният уебсайт използва технологии от страната на сървъра като PHP, Python или Ruby, за да генерира уеб страници в движение в отговор на потребителски заявки. Динамичните уебсайтове могат да извличат данни от бази данни, да взаимодействат с потребителите и да създават персонализирано съдържание. Въпреки това, те обикновено са по-сложни за разработване и изискват по-усъвършенствана хостинг настройка.

2.7. Домейн

Когато използваме Github Pages за хостване на уебсайт, имаме опцията да използваме потребителски домейн вместо URL адреса на Github Pages по подразбиране. В нашия случай използвахме Freenom, за да регистрираме нашия потребителски домейн, който е 12vkttpgibt.ml.

За да настроим функцията за потребителски домейн, първо трябваше да създадем CNAME файл в нашето хранилище на Github Pages. Този файл съдържа нашето потребителско име на домейн и указва на Github Pages да обслужва нашия уебсайт от този домейн.

След това трябваше да конфигурираме DNS настройките за нашия потребителски домейн във Freenom. Добавихме CNAME запис, който насочва

нашия персонализиран домейн към URL адреса на Github страници за нашия уебсайт.

След като тези стъпки бяха изпълнени, изчакахме DNS промените да се потвърдят, което отнеха около 24 часа. След като промените бяха потвърдени, нашият уебсайт беше достъпен чрез нашия потребителски домейн, 12vkttpgbt.ml.

ГЛАВА 3

ЭКСПЕРИМЕНТАЛНА ЧАСТ

.....

ЗАКЛЮЧЕНИЕ

.....

ПРИЛОЖЕНИЯ

.....