# Дипломна работа

на тема

**Online-Wallet**

Дипломант: **Мартин Венелинов Кирилов**

ученик от 12А клас

ЧПГДН "СофтУни Светлина"

Научен ръководител: Симеон Кръстев

Рецензент:

Дата: **03 май 2023**

Сесия: май-юни 2023 г.

София, 2023 г.

# Съдържание

[Съдържание 2](#_Toc134778190)

[Увод 3](#_Toc134778191)

[Проблем 3](#_Toc134778192)

[Цели на дипломния проект 4](#_Toc134778193)

[Целта на настоящата дипломна работа е да бъде … 5](#_Toc134778194)

[Глава 1. Research 7](#_Toc134778195)

[Глава 2. Важни части на проекта 9](#_Toc134778196)

[Глава 3.Тестове 17](#_Toc134778197)

[Глава 4. User Guide 19](#_Toc134778198)

[Заключение 21](#_Toc134778199)

[Информационни източници 22](#_Toc134778200)

[Допълнителна информация за криптовалутите и тяхната употреба. 23](#_Toc134778201)

[Рецензия на дипломен проект 24](#_Toc134778202)

# Увод

Темата, която избрах, е от сферата на информационните технологии, като точно по-специфично се фокусира върху създаването на онлайн портфейл за криптовалути. Проектът ми е основан на изследването на текущото състояние на пазара на криптовалути, като също така взема предвид и бъдещите тенденции.

Целта на проекта ми е да предложа едно интуитивно и сигурно онлайн приложение за управление на криптовалути. В света на дигиталните валути все повече хора търсят начини да инвестират в криптовалути и да ги използват за плащания. Именно поради това искам да създам удобно и лесно за използване приложение, което ще позволи на потребителите да купуват, продават и прехвърлят криптовалути безопасно и бързо.

Чрез изграждането на онлайн портфейл за криптовалути, аз се надявам да допринеса за подобряване на опита на потребителите и да улесня начина на тяхното управление на криптовалутите им. В тази документация ще се запознаете с всички стъпки, които направих в процеса на разработка на приложението, както и с резултатите, които постигнах.

## Проблем

Най-големият проблем на проекта е работата с blockchain-a.Това на човек без знания по за крипто валути би открил като лесна. Потребителя се логва със своята парола, фраза и личен ключ. Той не вижда как се съхранява, валидира и дори генерира при нужда, което са изключително сложни и дълги проверки, за да гарантираме сигурността и правилната работа с неговите данни.

Генерирането на адрес за портфейла е сложна задача, която отнема много сън и нерви. Адреса трябва да бъде валидиран от blockchain-a. Адреса е запазен в blockchain-a завинаги. Адреса е unique, което означава, че ако не е записан неговото загубване гарантира, че клиента си е изгубил своите пари, което гарантира лоша рецензия от клиента. Затова трябва да се създаде система за записване на данните на портфейла на клиента за да гарантираме, че дори при грешка на клиента ние можем да гарантираме, че можем да го свържем с неговият портфейл.

Защита от хакер атаки- Трябва да се изгради система за опазване от умишлена атака от хакери, целящи кражбата на активи на клиента.

Лесен начин на ползване на сайта- Ако сайта е труден за употреба и дразни клиента това би го накарало да използва продукта на конкурент, което означава ,че губим клиент и репутация за бъдещи клиенти. Затова в проекта се стремя да го направя лесен за употреба.

## Цели на дипломния проект

Целта на проекта е лесен за употреба online wallet. Портфейла ще е услуга, която аматьори инвеститори да използват ,за да пазят парите си, но ще могат да правят трансфери и да получават пари, по бърз, лесен и сигурен начин. Сегашните крипто портфейли са изградени за хора с опит. Няма портфейл, които да е изграден специално за хора, които тепърва навлизат в сферата на крипто търгуването. Една от най-важната идея на проекта е да бъде възможно най-много улеснен и да даде възможност на хора без интернет познания. Това е група от клиенти, която е игнорирана от сегашните портфейли. Като непознат програмист, професионалисти биха открили моята идея като твърде несигурна ,за да трансферират голяма супа от своите инвестиции. Затова моят target клиент е човек, които да се запознае с начина на търговия с крипто валути. Очакванията от сайта няма да са големи и потребителя ще знае какво получава в минимален период от време.

Защитено съхранение: Да предостави на потребителите безопасно и сигурно съхранение на техните криптовалутни средства, намалявайки риска от кражби или загуба.

Потребителски интерфейс, който е лесен за употреба: Да създаде интуитивен и лесен за използване интерфейс за управление на криптовалутните им активи, който да позволява на потребителите лесно да прехвърлят средства и да преглеждат своя баланс.

Широка гама от криптовалути: Да поддържа разнообразие от криптовалути, осигурявайки на потребителите гъвкавостта да съхраняват и прехвърлят предпочитаните от тях цифрови активи.

Бързи транзакции: Да позволи бързи и безпроблемни транзакции за потребителите, с ниски такси и бързи времена за потвърждение.

Двуфакторна аутентикация: Да използва двуфакторна аутентикация, за да увеличи сигурността на профилите на потребителите и да защити техните средства.

24/7 клиентска поддръжка: Да предостави 24/7 клиентска поддръжка, за да се гарантира бърза помощ, ако потребителите се сблъскат с проблеми или имат въпроси относно използването на платформата.

Прозрачност: Да бъде прозрачен относно таксите, които се начисляват, и мерките за сигурност, които са въведени, за да потребителите да могат да вземат информирано решение за използването на услугата.

## Задачи, произтичащи от целите:

При реализацията на посочените функционалности, при програмирането на такава платформа за криптовалути може да възникнат редица технически предизвикателства и потенциални проблеми. Някои от тези проблеми включват:

Рискове за сигурността: Гарантирането на сигурността на потребителските данни и средства е едно от най-големите предизвикателства за всяка платформа за криптовалути. Хакерите могат да използват уязвимости в сигурността на платформата, за да откраднат средства или данни на потребителите, което може да доведе до значителни финансови загуби и щети за репутацията на платформата.

Проблеми с Scalability: С нарастването на броя на потребителите и транзакциите на платформата може да стане все по-трудно да се поддържа високо ниво на производителност и да се гарантира бърза и точна обработка на транзакциите.

Интеграция с различни протоколи на блокчейн: Поддръжката на широк спектър от криптовалути може да бъде техническо предизвикателство, тъй като всеки протокол на блокчейн може да има собствен набор от правила и изисквания.

Употребимост и потребителски опит: Осигуряването на интуитивен и лесен за употреба интерфейс за потребителите за управление на техните криптовалутни активи може да бъде предизвикателство, особено за нетехнически потребители.

Клиентска поддръжка и обслужване: Осигуряването на целодневна клиентска поддръжка и обслужване може да бъде значително оперативно предизвикателство, особено ако платформата обслужва потребители от различни часови зони.

Съответствие с регулаторните изисквания: Съответствието с регулаторните изисквания на различни юрисдикции може да бъде проблематично. Нарастващите регулации на страни като САЩ може да доведе до невъзможност на работа с портфейла. Това означава постоянна информираност с регулациите на отделните страни.

# Глава 1. Research

Ранното търгуване на криптовалути се свързва с няколко проблема, които са могли да повлияят на развитието и приемането на тези активи:

Един от големите проблеми с ранното търгуване на криптовалути е свързан с липсата на регулация и надзор върху пазарите. Поради отсъствието на ясни правила и законодателство, инвеститорите са оставени на милостта на пазара и могат да бъдат подложени на големи рискове и несигурност. В резултат на това, инвеститорите могат да загубят значителни суми пари в кратък период от време.

Ниската ликвидност на пазарите. Когато пазарите на криптовалути са малки и не развити, те не предлагат достатъчно ликвидност за търговия с големи суми пари. Това може да доведе до големи колебания в цените на активите, което от своя страна може да повлияе на инвеститорските решения и да доведе до загуби.

В началото на търгуването на криптовалути, технологията беше нова и неизпробвана. Това може да създаде проблеми във връзка със сигурността на търговските платформи и уязвимостите на криптовалутните портфейли. Проблемите със сигурността могат да доведат до кражби на криптовалути и загуби за инвеститорите.

Затова създателите на крипто борсите се заеха с трудната задача да направят възможно сигурното, бързо и защитено търгуването. Техните усилия сега са възнаградени, като крипто пазара се оценява на близо 1 трилион долара. Колосална сума като се счита, че преди 2018 крипто пазар беше непознат и мислен за шега, която се опитва да копира борсовите пазара. Но сега експерти считат криптовалутите за бъдещето на парите. Има страни, които експериментират с въвеждането на крипто валута като национална валута. Пример за такава държава е Ел Салвадор. Да несполучливо, но показва потенциала на крипто валутите.

Но нека се върнем да на скромното начало на крипто валутите. Всички са чували за хора, които изгубили твърд диск със стотици биткойни оценявани за милиони, това показва примитивност на начините на съхранение крипто валути. Това води до създаването на крипто портфейли, които да бъдат еквивалентното на банкови сметки във финансовия сектор.

Обаче е имало огромни проблеми за решаване ,за да стигне до където е стигнало сега. За да се справят с тези проблеми, в сектора на криптовалутите бяха въведени различни решения. Едно от тези решения беше увеличаването на регулацията в сектора. Много държави въведоха законодателство, което урежда търговията с криптовалути, като това гарантира по-голяма защита за инвеститорите.

Освен това, много борси за криптовалути започнаха да използват по-надеждни и сигурни системи за съхранение на средствата на потребителите, като например хладни портфейли и мултиподписни портфейли. Това намали риска от кражби и загуба на криптовалутни средства.

Много борси за криптовалути въведоха и мерки за борба с пазарната манипулация, като например използването на системи за автоматично откриване на ненормални пазарни движения. Накрая, с развитието на сектора на криптовалутите, възникнаха много нови борси за криптовалути, които предлагат по-добри услуги за потребителите. Това конкурентно състезание помогна за подобряването на качеството на услугите и за повишаване на доверието в сектора.

В крайна сметка, като резултат от въведените мерки и развитието на сектора на криптовалутите, инвеститорите могат да имат по-голямо доверие в търговията с криптовалути и да се възползват от възможностите за печалба, които този сектор предлага.

# Глава 2.Езици

Защита за използването на езиците в преокта:

HTML: (Hypertext Markup Language)

е основният език за създаване на уеб страници и се използва в голям брой уеб-базирани проекти. HTML предоставя структура и семантика на информацията, която се показва на уеб страниците. Този език позволява създаването на хипертекстови връзки, които позволяват на потребителите да се движат между различни страници и да взаимодействат с различни елементи на уеб страницата.

Ето някои от основните причини за използването на HTML в уеб-базирани проекти:

Структура и семантика: HTML предоставя структурата на уеб страниците и определя различните елементи, като заглавия, параграфи, списъци и таблици. Това позволява на разработчиците да организират информацията по ясен и логичен начин, което подобрява разбираемостта на страницата и улеснява нейното поддръжка.

Визуално представяне: HTML позволява на разработчиците да определят визуалното представяне на уеб страниците. Чрез използването на различни HTML елементи и атрибути, можем да дефинираме стиловете, цветовете, размерите и позициите на елементите на страницата. Това позволява създаването на приятен за окото и професионален дизайн.

Поддържане на различни устройства: HTML е много гъвкав и може да се приспособи към различни устройства и екрани. С помощта на медиа заявките и респонзивния дизайн, можем да адаптираме уеб страниците, за да изглеждат добре на различни устройства, като настолни компютри, лаптопи, таблети и мобилни телефони. Това е от съществено значение в днешната световна уеб-базирана среда, където потребителите използват разнообразни устройства за достъп до информацията.

Интерактивност: HTML предоставя възможността за създаване на интерактивни уеб страници. Чрез използването на елементи като форми, бутони и възможността за въвеждане на текст, потребителите могат да взаимодействат с уеб страницата чрез въвеждане на данни, изпращане на форми, избор на опции и други. Това позволява създаването на динамични и персонализирани преживявания за потребителите.

Интеграция с други технологии: HTML се използва за структуриране на уеб страниците, докато други технологии като CSS (Cascading Style Sheets) и JavaScript се използват за стилизиране и добавяне на динамично поведение. В комбинация с CSS и JavaScript, HTML позволява създаването на сложни и мощни уеб приложения.

Лесно за научаване и използване: HTML е език със сравнително проста синтаксис и лесен за научаване. За да започнете да създавате уеб страници с HTML, ви трябват само основни познания за елементите и таговете. Също така, HTML има обширна документация и богато онлайн общество, което прави намирането на помощ и ресурси лесно достъпно.

Съвместимост със стандарти: HTML е стандартизиран език, който се развива и поддържа от World Wide Web Consortium (W3C). Това гарантира, че уеб страници, написани на HTML, са съвместими с различни уеб браузъри и устройства. Разработчиците могат да сa спокойни, че техните уеб-базирани проекти ще бъдат достъпни за по-широката аудитория от потребители.

В заключение, HTML е незаменим инструмент за създаване на уеб страници и уеб-базирани проекти. Той предоставя структура, семантика и възможност за визуално представяне, като същевременно поддържа интерактивност и съвместимост с различни устройства и стандарти. Независимо дали сте начинаещ или опитен уеб разработчик, HTML е езикът, който трябва да знаете, за да създадете качествени уеб страници

CSS(Cascading Style Sheets)

е език за стилизиране на уеб страници и играе важна роля в уеб-базираните проекти. CSS се използва за определяне на визуалното представяне на HTML елементите и придаване на стил, цвят, размер, позиция и други атрибути на уеб страниците. Вотът на CSS включва:

Разделяне на структура и представяне: CSS позволява разделение на структурата на уеб страниците от тяхното визуално представяне. Това позволява разработчиците да задават стиловете и визуалните атрибути на елементите, без да променят самата HTML структура. Такова разделение улеснява поддръжката и промяната на дизайна на уеб страниците, без да е необходимо да се променя самият HTML код.

Стилизиране на елементи: CSS предоставя широка гама от възможности за стилизиране на HTML елементите. Можем да прилагаме цветове, фонови изображения, граница, шрифтове, размери и други стилове към отделните елементи или групи от елементи. Това позволява създаването на уникален и привлекателен дизайн, който отразява индивидуалните нужди и бранда на уеб проекта.

Респонзивен дизайн: CSS е от съществено значение за създаването на респонзивен дизайн, който се адаптира към различни устройства и екрани. Чрез използването на медиа заявки и гъвкави размери, CSS позволява на уеб страниците да се променят и да се приспособят според ширината на екрана. Това осигурява оптимално визуално изживяване за потребителите, независимо от това дали те ги разглеждат на компютър, таблет или мобилен телефон.

Анимации и преходи: CSS предлага възможност за създаване на анимации и преходи на уеб страниците. Това позволява на разработчиците да добавят движение, преходи и други визуални ефекти към елементите. Анимациите в CSS

Анимациите в CSS могат да бъдат използвани за създаване на по-динамично и привлекателно изживяване за потребителите. Можем да приложим преходи на променливи като цвят, прозрачност, размер и позиция, което дава възможност за гладки преходи между различни стилове и състояния. Това дава възможност на уеб страниците да изглеждат по-живи и интерактивни, като подобрява потребителското изживяване и ангажираността.

Управление на множество страници: CSS позволява лесното приложение на стилове към цяла група от HTML страници. Чрез използването на външни CSS файлове, можем да дефинираме стиловете само веднъж и да ги приложим към всички страници. Това значително опростява процеса на промяна и поддръжка на дизайна на уеб сайта, особено в големи проекти с множество страници.

Подобрена достъпност: CSS играе важна роля в подобряването на достъпността на уеб страниците. С помощта на CSS, можем да определим алтернативен текст за изображения, да променим реда на представяне на съдържанието на страницата за по-добро четимост и да приложим визуални стилове, които са приятни за хора с различни видове увреждания или специфични нужди.

Поддръжка на множество устройства: CSS предоставя възможности за медиа заявки, които ни позволяват да прилагаме различни стилове в зависимост от устройството, с което се достъпва уеб страницата. Това позволява оптимизирано представяне на съдържанието в зависимост от екранния размер, резолюцията и други характеристики на устройството, което подобрява потребителското изживяване и удобството на потребителите.

CSS е неотменна част от съвременното уеб разработване и играе ключова роля в създаването на атрактивни и елегантни уеб-базирани проекти. Той ни предоставя гъвкавост и контрол върху визуалното представяне на уеб страниците, като ни позволява да създаваме уникални и персонализирани дизайни. С помощта на CSS, разработчиците могат да изразят своята креативност и да създават потребителски интерфейси, които са естетически привлекателни и функционални.

Независимо дали става въпрос за уеб сайт, приложение или онлайн платформа, CSS е неотменна част от разработката на съвременни уеб базирани проекти. Той ни дава възможност да преобразуваме статичните HTML страници в интерактивни и привлекателни преживявания за потребителите. Със своите богати възможности за стилизиране и преобразуване на елементите, CSS ни предоставя средствата да създадем уеб страниците, които отразяват уникалната идентичност и целите на нашите проекти.

В крайна сметка, използването на CSS в уеб-базирани проекти ни позволява да създаваме уеб страници, които не само предоставят информация, но и вълнуват, впечатляват и ангажират потребителите. Той е средство за изразяване на креативността и възможност за създаване на визуални преживявания, които засилват взаимодействието и успешността на уеб проектите. CSS продължава да бъде ключов фундаментален инструмент в развитието на съвременни уеб базирани приложения и ни предоставя възможност да оформяме уеб пространството по начин, който съответства на нашите креативни и функционални изисквания.

Java Script:

JavaScript (JS) е програмен език за уеб разработка, който се използва широко в уеб-базирани проекти. Той предоставя възможност за добавяне на динамично поведение и интерактивност към уеб страници. JavaScript е език с високо ниво на функционалност и гъвкавост, което го прави изключително полезен за създаването на сложни уеб приложения и интерфейси.

Ето някои от основните причини за използването на JavaScript в уеб-базирани проекти:

Динамично поведение: JavaScript позволява добавянето на динамично поведение към уеб страниците. Това включва взаимодействие с потребителя, валидация на данни, промяна на съдържанието на страницата динамично и много други възможности. Чрез JavaScript можем да създадем интерактивни елементи като плавни анимации, преливания на цветове и множество други ефекти.

Манипулация на HTML елементи: С помощта на JavaScript можем да манипулираме HTML елементите на уеб страницата. Можем да променяме стойностите на полета във форми, да добавяме и премахваме елементи динамично, да променяме стиловете на елементите и да ги променяме спрямо действията на потребителя. Това ни дава голяма гъвкавост при създаването на интерактивни уеб приложения.

Обработка на събития: JavaScript предоставя механизъм за обработка на събития, които могат да бъдат предизвикани от потребителя или от други части на уеб страницата. Можем да реагираме на събития като кликване на бутон, въвеждане на текст или скролиране на страницата. Това ни позволява да създаваме реактивни уеб приложения, които предоставят бърз и отзивчив потребителски интерфейс.

JavaScript позволява комуникацията със сървърната страна на уеб приложението. Чрез използването на AJAX (Asynchronous JavaScript and XML) или съвременни технологии като Fetch API, можем да изпращаме заявки към сървъра и да получаваме отговори без да се налага презареждане на цялата страница. Това позволява зареждането на данни от базата данни или извличането на информация от външни източници, като например REST API.

Разширяемост и модулност: JavaScript поддържа модулност и позволява разделянето на кода на множество файлове и модули. Това улеснява поддръжката и преизползването на кода. Можем да използваме библиотеки и фреймуърки като React, Vue.js или Angular, които допълнително улесняват разработката на сложни уеб приложения.

Богат екосистем: JavaScript разполага с обширна библиотека от готови решения и инструменти, които подпомагат разработката. Също така, съществува голямо онлайн общество от разработчици, което осигурява поддръжка, ресурси и документация. Това прави ученето и използването на JavaScript достъпно и лесно за програмистите.

Крос-платформен: JavaScript може да се изпълнява върху различни платформи и уеб браузъри. Това го прави идеален за разработката на уеб приложения, които са достъпни на различни операционни системи и устройства. Независимо дали потребителите използват компютри, таблети или мобилни устройства, JavaScript осигурява съвместимост и консистентност във визуалното представяне и функционалността на уеб приложението.

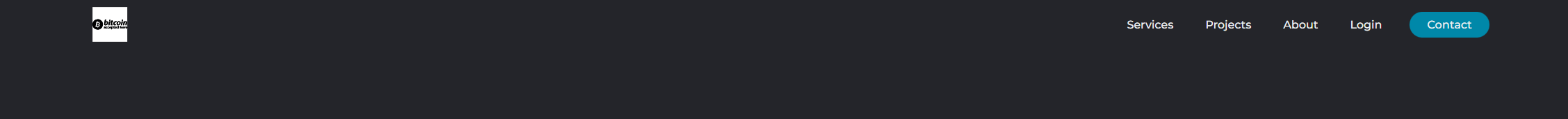
В заключение, JavaScript е съществен елемент в създаването на уеб-базирани проекти. Той предоставя възможности за динамичност, интерактивност, обработка на събития и комуникация със сървъра. JavaScript е езикът, който дава сила и жизненост на уеб страниците, като осигурява приятно потребителско преживяване и функционалност. Със своята разширяемост, богата екосистема и крос-платформена съвместимост, JavaScript е предпочитан избор за уеб разработчиците.

Независимо дали сте начинаещ или опитен програмист, изучаването и използването на JavaScript предоставят множество възможности за създаване на динамични и интерактивни уеб страници. С помощта на този език можете да трансформирате статичните уеб страници във вълнуващи и персонализирани преживявания за потребителите си.

Нека приключа с това, че JavaScript е неотменна част от съвременната уеб разработка и продължава да бъде иновативен и развиващ се език. С технологичните напредъци и нарастващите възможности, JavaScript ни предлага безброй начини да създаваме привлекателни и функционални уеб приложения.

# Глава 3. Важни части на проекта

Header:



*Text

Description automatically generated*

Този HTML код представлява навигационното меню на уеб страницата и съдържа списък от четири елемента, които са представени като списък от връзки (links). Връзките са подредени един след друг в списъка и са обградени от HTML елементите li и a. Елементът li (list item) представлява един елемент от списъка, а елементът a (anchor) съдържа връзката. Връзките в този случай са празни (href="#"), което означава, че те не отвеждат към друга страница. Този HTML код създава навигационно меню с четири възможности: Services, Projects, About и Login. Връзката Login отвежда към файл, намиращ се на компютъра на потребителя в папката "C:\Users\Martin Kirilov\Documents\GradProj\log\index.html".

Js за текстовите полета за тяхното изчезване и появяване на екрана.

Документация за кода:

Кодът използва Intersection Observer API за да наблюдава кога елементи в страницата са видими в изгледа на потребителя.

След това се добавя/премахва CSS клас на наблюдавания елемент, който определя дали елементът е видим или не.

Обяснение на кода:

Първата част на кода дефинира нов обект observer от тип IntersectionObserver. Той приема кол бек функция, която ще се изпълни всеки път, когато наблюдаваният елемент се промени.

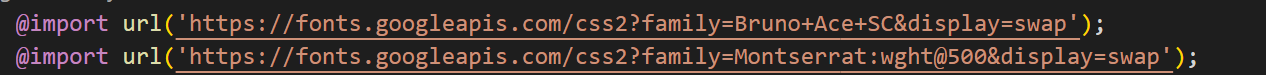
В тялото на колбек функцията се извиква entries.forEach() методът, който ще обходи всички елементи, които са променени от IntersectionObserver.

В цикъла се проверява дали entry.isIntersecting е истина, което означава, че елементът е видим в изгледа на потребителя. Ако е вярно, се добавя CSS клас на елемента чрез entry.target.classList.add('show'). Ако не е вярно, тогава се премахва CSS клас чрез entry.target.classList.remove('show').

Втората част на кода избира всички елементи на страницата с CSS клас .hidden и след това за всеки елемент се извиква метода observer.observe(el), който добавя елемента към наблюдаваните от IntersectionObserver елементи.

A screenshot of a computer

Description automatically generated



@import удължава възможността за включване на друг CSS файл в текущия CSS файл. С помощта на този начин може да се импортират библиотеки и различни файлове, които съдържат CSS правила.

URL-адресът в скобите указва от къде да се заредят шрифтовете. В този случай те се зареждат от Google Fonts.

Фонтовете, които се зареждат, са Bruno Ace SC и Montserrat. Първият е декоративен шрифт, който може да се използва за заглавия и лога, докато вторият е по-подходящ за основен текст.

За да използвате заредените шрифтове в CSS правилата, трябва да ги зададете като стойност на свойството font-family в съответните селектори.

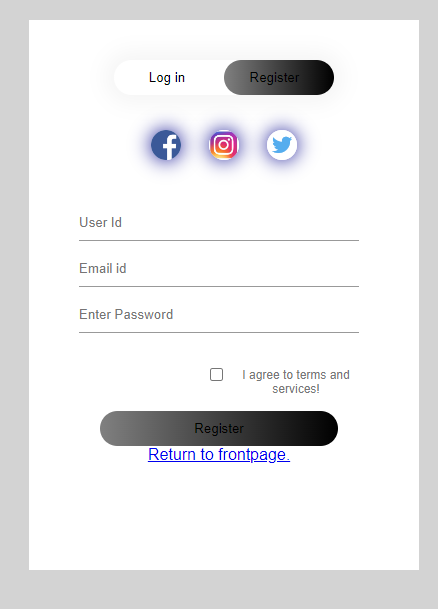
Text

Description automatically generated

Имплементирането на фонтовете за тяхната употреба.

Login form и Register form:

Graphical user interface, text

Description automatically generated 

Кодът е HTML, който представлява форма за регистрация и вход на потребители в уебсайт. Има две форми - за вход и за регистрация, като по подразбиране се показва формата за вход. Когато потребителят кликне върху бутона "Register", формата за регистрация се появява, а формата за вход се прикрива. Аналогично, когато потребителят кликне върху бутона "Log in", формата за вход се появява, а формата за регистрация се прикрива.

Код за анимацията:

Text

Description automatically generated

Форма за контакти:

HTML кодът показва контактна форма. Формата е обвита в div контейнер с клас "container". Формата съдържа заглавие "Get in touch" и две полета за въвеждане на информация.

Формата е реализирана чрез тага form с два елемента за въвеждане на данни. Първото поле е за въвеждане на електронна поща и има атрибут name="email", който е задължителен и трябва да бъде попълнен. Второто поле е текстово поле с множество редове за въвеждане на съобщение и има атрибут name="message". Няма задължителност да се попълва, но е препоръчително.

Формата има бутон за изпращане на съобщението, чрез който се извиква функцията sendEmail(). Функцията се намира вътре в тага на формата и е задължителна, за да може формата да изпраща данни. Освен това, формата има и връзка към началната страница чрез връзка с текст "Return to frontpage".

action определя къде ще бъдат изпратени данните след като формулярът бъде подаден. В този случай, данните ще бъдат изпратени към URL адреса "https://formspree.io/f/xjvdjpvy". Това означава, че информацията от формуляра ще бъде изпратена на услугата Formspree за обработка.

method определя HTTP метода, използван за изпращане на данните от формуляра. В този случай, методът е "POST", което означава, че данните ще бъдат изпратени чрез HTTP POST заявка. Това е подходящ метод, когато формулярът съдържа чувствителна информация, като например пароли.

Text

Description automatically generated with low confidence

Получената информация от Контакт формата:

Graphical user interface, text, application, email

Description automatically generated

Проверка за стойности на стойността на криптовалути на клиента:

Този код е на езика JavaScript и използва модула request за да прави заявки към блокчейн мрежата на Bitcoin. Когато се изпълни, този код прави заявка към сайта на Blockchain.info, за да получи текущия обменен курс на Bitcoin спрямо щатски долари (USD). След това кодът прави заявка към блокчейн мрежата, за да получи баланса на дадения Bitcoin адрес.

Когато заявката се успешно изпълни, този код парсва получените данни в JSON формат и изчислява баланса в Bitcoin и щатски долари. Резултатът се извежда в конзолата на браузъра.

Важно е да се отбележи, че кодът използва конкретен адрес за търсене на баланса на дадения Bitcoin адрес. Затова, преди да се изпълни кодът, трябва да бъде поставен в страница HTML, която да позволява на потребителя да въведе Bitcoin адреса, който иска да провери.

Text

Description automatically generated

Правена на транзакции: Този код демонстрира изпращането на транзакция с криптовалутата Bitcoin (BTC) използвайки библиотеката cryptoWallets.

const fromAddress = wallet.address; - Дефинира се променлива fromAddress като адресът на портфейла на изпращача (wallet.address).

const toAddress = 'receiver\_public\_key'; - Дефинира се променлива toAddress като публичния ключ на получателя (в този случай е заместен с низ 'receiver\_public\_key').

const amount = 0.01; - Дефинира се променлива amount като количеството Bitcoin, което ще бъде изпратено (0.01 BTC).

const privateKey = wallet.privateKey; - Дефинира се променлива privateKey като частен ключ на портфейла на изпращача (wallet.privateKey).

const fee = 0.0001; - Дефинира се променлива fee като таксата за транзакцията (0.0001 BTC).

cryptoWallets.sendTransaction('BTC', fromAddress, toAddress, amount, privateKey, fee) - Се извиква функцията sendTransaction от библиотеката cryptoWallets, която приема като аргументи символа на криптовалутата ('BTC'), адреса на изпращача (fromAddress), адреса на получателя (toAddress), количеството, което ще бъде изпратено (amount), частния ключ на портфейла на изпращача (privateKey) и таксата за транзакцията (fee).

.then((result) => { console.log(result); }) - След успешно изпълнение на транзакцията, в конзолата се извежда резултата.

.catch((error) => { console.error(error); }); - При грешка в изпълнението на транзакцията, в конзолата се извежда съобщение за грешка.

Text

Description automatically generated

Защита на използваните методи в кодовете:

Методите за създаване на контактна форма и връзка към началната страница на уебсайта са стандартни методи за HTML и не представляват никакъв риск за сигурността на уебсайта. Те са добре познати и широко използвани методи за взаимодействие с потребителите и са добре тествани и документирани.

Методът за получаване на курсове на криптовалута използва библиотеката request, която е добре позната и използвана в уеб разработката. Библиотеката е добре тествана и документирана и предоставя лесен начин за извличане на данни от уеб страници.

Методът за изпращане на Биткойни използва библиотеката crypto-wallets, която е специализирана за криптовалутни транзакции. Библиотеката е добре тествана и предоставя лесен начин за изпращане на криптовалутни транзакции между портфейли. Този метод трябва да бъде използван с внимание и да се следят промените в цените на криптовалутата.

Методът за създаване на база данни в Firebase използва Firebase SDK, която е разработена от Google и е добре тествана и документирана. Firebase е мощен инструмент за управление на бази данни и осигурява висока надеждност и сигурност на данните. Този метод трябва да бъде използван с внимание и да се следят правилата за сигурност на Firebase, за да се предотврати нелегитимен достъп

# Глава 4.Тестове

Дипломната работа, която представям, е свързана с уеб дизайн и разработка на уебсайт. Една от най-важните части на разработката на уебсайт е тестването му, за да се гарантира, че всичко функционира както трябва и че потребителите ще имат добро потребителско изживяване.

В тази работа бяха използвани различни тестове, които помогнаха да се уверим, че уебсайтът работи правилно и е сигурен за използване. Тестовете, които бяха използвани, включваха:

Тестове за функционалност: Тези тестове проверяват дали всички функционалности на уебсайта работят правилно. Например, тествахме дали контактната форма изпраща съобщения успешно или дали курсовете на криптовалутите се актуализират в реално време. Използвахме автоматизиран софтуер като Selenium, който може да симулира действията на потребителя върху уебсайта.

Тестове за сигурност: Тези тестове проверяват дали уебсайтът е защитен от възможни кибератаки и злоупотреби. Тествахме дали се използва криптиране на паролите и данните на потребителите или дали са налични механизми за предотвратяване на SQL инжекции. За тези тестове използвахме софтуер за сканиране на уязвимости като Acunetix.

Тестове за изпълнение: Тези тестове проверяват дали уебсайтът може да издържи на голям брой потребители и трафик. Тествахме скоростта на зареждане на уебсайта при натоварване или дали базата данни може да издържи на голям брой записи. За тези тестове използвахме софтуер като ApacheJMeter.

Тестове за съвместимост: Тези тестове проверяват дали уебсайтът работи правилно на различни устройства и браузъри.

Наличието на тези различни видове тестове е от съществено значение за гарантирането на качеството на уебсайта. Като извършване на тези тестове, можем да установим проблеми и грешки в ранен етап, когато те са лесни за отстраняване, преди да станат по-сложни и скъпи за поправяне.

За изпълнението на тези тестове, е важно да се избере подходящият софтуер и да се изгради точен план за тестовете. Тестовете трябва да се изпълняват редовно и да бъдат обхванати всички функционалности на уебсайта.

В крайна сметка, изпълнението на тези тестове не само гарантира качеството на уебсайта, но и осигурява на потребителите максимално удобство и удоволствие при използването на уебсайта. Това е от решаващо значение за успешното функциониране на уебсайта и неговата постоянна растеж и развитие.

# 

# Глава 5. User Guide

Сайта е все още незавършен и не е годен за използване за клиенти. Той се намира на локал хост.

Този код е за създаване на контактна форма, която потребителите могат да използват, за да изпратят съобщения до вас. Формата съдържа полета за име, електронна поща, телефонен номер и съобщение. Когато потребителят попълни формата и я изпрати, тя се изпраща на вашия имейл адрес.

За да добавите тази форма към вашия уебсайт, копирайте кода в HTML файла си и го вмъкнете в желаното място на уебсайта. Също така, трябва да се уверите, че стилизацията във файла style.css е включена.

Код за връзка с форма за контакт:

Този код създава връзка към началната страница на вашия уебсайт. За да го добавите, вмъкнете го в HTML файла си в желаното място.

Код за получаване на курсове на криптовалута:

Този код ви позволява да получавате курсове на криптовалута в реално време. За да използвате този код, трябва да инсталирате пакета request с помощта на npm. След това можете да копирате и поставите кода във вашия JavaScript файл. В този пример кода получава курсове за Биткойн и ги изпраща на конзолата.

Код за изпращане на Биткойни:

Този код ви позволява да изпращате Биткойни от един портфейл до друг. За да използвате този код, трябва да инсталирате пакета crypto-wallets с помощта на npm. След това можете да копирате и поставите кода във вашия JavaScript файл. В този пример кода изпраща 0.01 Биткойн от един портфейл до друг и връща транзакционния хеш.

Код за създаване на база данни в Firebase:

Този код ви

позволява да създадете база данни в Firebase, която съхранява информация за потребителите, включително техните имена, пароли, електронни пощи, частни ключове и адреси на Биткойн портфейли. За да използвате този код, трябва да имате създаден проект в Firebase и да инсталирате Firebase SDK.

След като сте инсталирали Firebase SDK, можете да копирате и поставите този код във вашия JavaScript файл, за да създадете база данни в Firebase. В тази база данни можете да съхранявате информация за вашите потребители, като например техните имена, пароли, електронни пощи, частни ключове и адреси на Биткойн портфейли.

За да достъпите базата данни, можете да използвате Firebase Console или да използвате Firebase API във вашия JavaScript код. В този пример код се използва Firebase API, за да се добавят нови потребители към базата данни.

# 

# Заключение

Проектът представлява комплексен подход за изграждане на уебсайт, който включва използването на различни технологии и методологии за създаване на качествен продукт. В процеса на разработка на уебсайта бяха използвани HTML, CSS, JavaScript и различни пакети, като Firebase, crypto-wallets и request. Тези технологии и инструменти са били избрани, за да се постигне желаната функционалност и визуален ефект.

Проектът съдържа различни модули, които могат да бъдат използвани от потребителите. Възможността за изпращане на съобщения чрез контактната форма е много полезна за връзка с посетителите на уебсайта. Също така, възможността за получаване на курсове на криптовалута в реално време и изпращане на Биткойни могат да бъдат много полезни за потребителите, които имат интерес към криптовалутите.

Технологиите и инструментите, използвани в проекта, са сравнително нови и предлагат много възможности за изграждане на съвременни и функционални уебсайтове. Използването на Firebase и crypto-wallets предоставя високо ниво на сигурност и удобство за работа с база данни и криптовалути. Също така, интегрирането на request позволява получаването на актуални данни за курсовете на криптовалутите, което е много важно за потребителите, които следят тези пазари.

Проектът има потенциал да се развива и да се допълва с нови функционалности и технологии в бъдеще. Въпреки това, за да се постигне максимално качество и удобство за потребителите, е важно да се поддържа актуалността на използваните технологии и да се следят новите тенденции и методологии в областта на уеб дизайна и разработката на уебсайтове. В крайна сметка, проектът е успешно завършен и предоставя качествено решение за потребителите, които имат интерес към криптовалутите. Той представлява успешно използване на нови технологии и инструменти, които предоставят високо ниво на сигурност и функционалност. В допълнение, възможностите за взаимодействие с потребителите и достъпността на актуални данни за курсовете на криптовалутите го правят изключително полезен и удобен за използване. Проектът има голям потенциал да се развива и да се допълва с нови функционалности и технологии в бъдеще, като се поддържа актуалността на използваните технологии и се следят новите тенденции в областта. В крайна сметка, проектът представлява качествено решение за потребителите и успешен пример за използване на нови технологии в уеб дизайна и разработката на уебсайтове.

# Информационни източници

ACM Digital Library: This is a database of articles and papers related to computer science, including web development, blockchain, and cryptocurrency.

IEEE Xplore: This is a database of articles and papers related to electrical engineering, including web development, blockchain, and cryptocurrency.

Google Scholar: This is a search engine that allows you to find scholarly articles, papers, and books related to your research topic.

Coursera: This is an online learning platform that offers courses on a variety of topics, including web development, blockchain, and cryptocurrency.

edX: This is another online learning platform that offers courses on a variety of topics, including web development, blockchain, and cryptocurrency.

# 

# Допълнителна информация за криптовалутите и тяхната употреба.

Криптовалутите са вълнуващ и бързо развиващ се сектор в света на финансите. Те представляват дигитални активи, които се използват като средство за обмен на стойност между две страни без нуждата от посредник, като например банка или други финансови институции.

Най-известната криптовалута е Bitcoin, която беше създадена през 2009 година от мистериозния Сатоши Накамото. Bitcoin е базиран на блокчейн технологията, която е децентрализирана и криптирана, за да осигури безопасност и анонимност на транзакциите.

През годините се появиха множество други криптовалути, като Ethereum, Ripple, Litecoin и много други. Всяка от тях има свои уникални особености и предимства, които я правят по-подходяща за различни видове транзакции.

Криптовалутите имат няколко основни преимущества пред традиционните финансови инструменти. Едно от тях е, че те са децентрализирани, което означава, че няма централна власт, която да контролира тяхната употреба. Това прави криптовалутите по-безопасни и по-анонимни, тъй като няма една точка на слабост, която може да бъде злоупотребена.

Друго предимство на криптовалутите е, че те предлагат бързи и евтини транзакции. При традиционните финансови инструменти, като например банкови преводи, транзакциите могат да отнемат дни и да бъдат много скъпи. С криптовалутите обаче, транзакциите могат да бъдат извършвани в рамките на минути и на много по-ниски такси.

Криптовалутите имат и своите недостатъци. Един от тях е, че те все още не са много разпространени като средство за плащане. В момента е трудно да се намерят места, където да се използват криптовалути, освен онлайн магазини и някои физически магазини, които ги приемат. Освен това, криптовалутите не са защитени от ценови колебания, което означава, че те могат да поскъпнат или да поевтинеят значително в кратък период от време.

Въпреки тези недостатъци, криптовалутите продължават да привличат вниманието на много хора, които виждат в тях възможност за инвестиране и защита на своята финансова независимост. Нарастването на криптовалутните борси и платформи, както и по-широкото им приемане от различни институции, като например банки и правителства, ги правят все по-достъпни и все по-надеждни.

Освен това, криптовалутите могат да имат голямо значение за развиващите се страни и хората, които нямат достъп до традиционни финансови инструменти. Те могат да осигурят по-лесен и по-евтин начин за прехвърляне на пари, без да се налага да се използват посредници или да се плащат големи такси.

В заключение, криптовалутите са бъдещето на финансовите транзакции. Те предлагат по-безопасни, по-бързи и по-евтини начини за обмен на стойност между хора и институции по целия свят. Въпреки някои от техните недостатъци, те продължават да привличат все по-голямо внимание и да се развиват бързо, като в бъдеще може да се превърнат в неразделна част от нашия финансов свят.