**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**TrTool**

Предмет: Въведение в скриптовите езици

Ученици : Научен ръководител:

*Борис Стоянов и Стоян Бабанов Даниел Димитров*

СОФИЯ

2 0 2 3

# Увод

Трейдването и криптовалутите са интригуваща сфера, която продължава да привлича още хора. С иновациите на блокчейн технологията и криптовалутите, се вижда растеж в областта на електронната търговията. Въпреки това, трейдването си остава изключително сложно и е нужно обучение и подготовка за потенциални печалби.

С разширяването на достъпа до интернет, уеб приложенията стават все по-голяма част от животите ни. Те ни дават възможността да се учим, работим, общуваме и забавляваме в дигиталния свят. Нашето приложение е в областта на трейдването и криптовалутите и има за цел да запълни липсата на ефективни, забавни и едновременно с това образователни инструменти за обучение.

Целта, която си поставяме, е да създадем уеб приложение, което да обучава потребителите на основно/базово ниво на трейдване както и на крипто света. Приложението ще бъде направено като платформа, включваща симулационна игра, в която потребителите могат да експериментират и да практикуват без реален риск. Освен това, ще съдържа курс за трейдинг, който ще представя теоретичните основи на търговията с криптовалути.

Задачите на проекта са следните:

Изграждане на уеб приложение с всички функции, които икаме да включим:

Разработване на симулационна игра за трейдване с криптовалути.

Създаване на обучителен курс за трейдинг.

Място за социализиране с други потребители.

Използване на известни стратегии, за запазването на потребителите.

#### II

Функционални изисквания:

Системата трябва да позволява на потребителите да създават акаунт / да влизат в него и да играят играта за търговия.

Системата трябва да симулира изменение на цената на акциите въз основа на различни сценарии и променливи, включително геополитически тегло, икономическо тегло, социално тегло, популярност и ограничения.

Приложението трябва да позволява на потребителите да играят един път на ден.

Системата трябва да показва таблица с резултатите от играта, като показва печалбата на всеки потребител.

Приложението трябва да има място/места за социализиране на потребителите с останалите

Сайтът трябва да съдържа стратегия за задържане на потребителите ( elo / ranked system)]

Развойна среда:

PHP - основният език за програмиране, използван за разработване на приложението.

Laravel - PHP framework, използван за по-лесно и ефективно създаване на уеб приложения.

Sqlite - система за управление на бази данни, използвана за съхранение на потребителска информация, данни за играта, баджовете, обществата.

Composer - инструмент за управление на зависимостите в PHP, използван за инсталиране на Laravel и други PHP пакети.

Git - система за контрол на версиите, използвана за следене на промените в кода по време на разработка.

Visual Studio Code - редактиране на кода.

Apache - уеб сървър за хостване на приложението.

В момента няма известни уеб приложения за симулация на пазар, а по-скоро се използва safe-trading на борсите, което представлява реално трейдване с фалшиви пари. Според нас, макар и това да го прави по-реалистично, хората имат по - малка вероятност да научат нещо, защото те не разбират реално защо цената се сменя и също не смятаме, че потребител би използвал този метод дълго, тъй като лесно ще се загуби мотивация и интерес. Ние се отличаваме точно с това, че показваме на потребителя фактори и събития, които играят роля на цената, както и създаваме среда на вид състезание/игра, което допълнително да задържи членовете.

Избрахме PHP/Laravel за back end, защото те предоставят една лесно разбираема среда за хора без опит в тази област - като нас. Добрата и обширна документация + не особено различен синтаксис от останалите езици и многото вградени функции ни вписаха в back end-а бързо и за кратко време успяхме да осъществим редица от целите ни. *Защо Laravel е по-добър от останалите избори - 1. MVC поддръжка и обектно-ориентиран подход*

*Първото и най-добро предимство на използването на рамката на Laravel е, че тя следва - базиран на модел, изглед и контролер архитектурен модел и има изразителен красив синтаксис, който го прави обектно-ориентиран.*

*2. Вградено удостоверяване и оторизация*

*Laravel предоставя конфигурация извън кутията за системата за удостоверяване и оторизация. Това означава, че само с няколко занаятчийски команди вашето приложение ще бъде оборудвано със сигурно удостоверяване и оторизация.*

*3. Система за опаковане*

*Системата за опаковане се занимава с множеството поддържащи софтуери или библиотеки, които помагат на уеб приложението да автоматизира процеса. Laravel използва композитор като мениджър на зависимости, който управлява цялата информация, необходима за управление на пакети. Пакетите са чудесен начин за ускоряване на разработката е да предоставим функционалността, от която се нуждаем, веднага. Изображение, Laravel Debug bar и Laravel IDE помощник са едни от най-добрите пакети на Laravel.*

*4.Множествена файлова система*

*Laravel също има вградена поддръжка за облачна система за съхранение като Amazon S3 и Rack space Cloud Storage и разбира се за локално съхранение. Удивително лесно е да превключвате между тези опции за съхранение, тъй като API остава един и същ за всяка система. Човек може да използва и трите системи в едно приложение, за да обслужва файлове от множество местоположения като в разпределена среда.*

*5. Конзола Artisan*

*Laravel има собствен интерфейс на командния ред, наречен Artisan. Обичайните употреби на Artisan включват публикуване на активи на пакети, управление на миграции на бази данни, зареждане и генериране на шаблонен код за нови контролери, модели и миграции. Тази функция освобождава разработчика от създаването на правилни кодови скелети. Човек може да разшири функционалността и възможностите на Artisan чрез внедряване на нови персонализирани команди.*

*6. Красноречив ORM*

*Eloquent ORM е вградената ORM реализация на Laravel. Laravel има най-добрия Object-relational Mapper в сравнение с другите рамки там. Това обектно-релационно съпоставяне ви позволява да взаимодействате с вашите обекти на базата данни и връзките на базата данни, като използвате изразителен синтаксис.*

*7. Двигател за шаблони*

*Laravel идва с вградената машина за шаблони, известна като Blade Template Engine. Механизмът за шаблони на Blade комбинира един или повече шаблони с модел на данни, за да създаде резултатни изгледи, като прави това чрез транспириране на шаблоните в кеширан PHP код за подобрена производителност. Blade също така предоставя набор от свои собствени контролни структури, като условни изрази и цикли, които са вътрешно картографирани към техните PHP аналогове.*

Избрахме sqlite, тъй като е най-лесен и не искахме да обръщаме толкова много внимание на този избор.

За tailwind css, тъй като дълго сме използвали pure css / html искахме да научим нещо ново, за да надградим и във front-end областта

*Въпреки че Tailwind CSS е доста нова рамка, тя става все по-популярна поради предимствата, които предлага. Тази популярна CSS рамка на първо място като помощна програма е много адаптивна.*

*Tailwind е известен с това, че ускорява процеса на разработка. За разлика от традиционния CSS, тази рамка не изисква да именувате класове и да превключвате между HTML и стилови файлове.*

Работа на проекта

За улеснение ще разграничим “еко системите” на уеб приложението

1.Трейдинг система

2.Експериментална система

3.Система - потребител

4.Баджове

5.Community система

6.Ranked

## I.трейдинг системата:

***index()***: Методът, който се изпълнява при отваряне на страницата на симулатора. Първо той проверява дали потребителят е логнат и дали е играл днес - ако е, потребителят се пренасочва към ранглистата със съобщение, че може да играе само веднъж на ден. След това инициализира сесията със стартови стойности, ако не е била инициализирана досега. Накрая връща изгледа на страницата на симулатора с актуалните данни.

***simulate()***: Методът, който се изпълнява при изпълнение на действие от потребителя, като закупуване или продажба на акции. Той пресмята новата цена на акцията на базата на различни теглови фактори и случайни елементи и обновява данните в сесията. Ако се стигне до 10-ия рунд, методът калкулира крайната печалба на потребителя, обновява данните на потребителя в базата данни и връща изгледа на крайната страница на симулатора.

Симулацията използва следните параметри за пресмятане на цената на акцията:

***"stabilityIndex"***: индекс на стабилността на цената на акцията.

***"crashLikelihood"***: вероятността за значително намаляване на цената на акцията.

***"geopoliticalWeight"***: влиянието на геополитически събития върху цената на акцията.

***"economicWeight"***: влиянието на икономически събития върху цената на акцията.

***"socialWeight"***: влиянието на социални събития върху цената на акцията.

***"popularityWeight"***: влиянието на популярността върху цената на акцията.

***"restrictionWeight"***: влиянието на регулаторни ограничения върху цената на акцията.

Тези параметри от своя страна дават шанс (%) на различни събития да се случат - Война (малка или голяма) Рестрикции (нови или премахване) Технологически иновации (малки или големи).

Събитията дават по-драстична промяна в цената, като може да има повече от едно събитие в рунд.

Индексът за стабилност пък определя “таван”, до който може да падне цената, тоест буквално определя стабилността и, а краш индекса дава шанс на събитието “crash”, което представлява голямо падане на цената.

Тези параметри/събития се използват и за пресмятане на новата цена на акцията въз основа на случайни фактори и техните теглови стойности. Крайната печалба на потребителя се пресмята като разликата между крайния баланс и стартовия баланс, плюс стойността на притежаваните акции.

***leaderboard()***: Методът, който се изпълнява при отваряне на страницата на ранглистата. Той извлича всички потребители с положителна печалба от базата данни, сортира ги в низходящ ред по печалба и връща изгледа на страницата на ранглистата с данните за потребителите.

### 

### II.Експериментала система

***index()***: Този метод е отговорен за отварянето на началната страница на симулатора на експеримент. Той просто връща изгледа на experiment-simulator.

**simulate()**: Този метод изпълнява самата симулация. Той приема като входни данни стойности за редица фактори, които оказват влияние върху цената на акцията. Всеки рунд се генерира случаен фактор, който влияе на цената на акцията, като отрицателните стойности представляват негативно влияние, а положителните - положително. За всеки фактор, ако генерираното случайно число е по-малко от тегловата стойност на фактора, се променя цената на акцията. В зависимост от размера на въздействието (малко или голямо) и направлението на промяната (увеличение или намаление), се увеличава или намалява цената на акцията. Стабилността на цената се регулира също, като при по-висок индекс на стабилност влиянието на случайния фактор се намалява наполовина. Накрая, ако цената на акцията превишава определен праг, тя се ограничава до този праг. Резултатите от всеки рунд се съхраняват в масив, който се изпраща към изгледа на experiment-simulator за визуализация.

#### III. Потребител

За потребителската система сме използвали включената на laravel (breeze) с малко промени, самата система работи така :

Модел User: В Laravel, моделите представляват базата данни и взаимодействието с нея. User моделът, който се намира в App\Models\User, представя таблицата users в базата данни. Той съдържа методи и свойства, които улесняват взаимодействието с данните на потребителите.

User Migration: Laravel използва миграции за изграждане на структурата на базата данни. users миграцията, която се намира в database/migrations, определя структурата на users таблицата, включително колоните и техните типове.

User Factory и Seeder: Laravel предоставя фабрики и сийдери за употреба при тестване или пълнене на базата данни с фиктивни данни. UserFactory създава фиктивни потребители, докато UserSeeder вкарва тези потребители в базата данни.

Controllers: Laravel използва MVC архитектура, където контролерите обработват логиката, свързана с всяка операция. LoginController, RegisterController, ForgotPasswordController, и ResetPasswordController обработват входа, регистрацията, забравената парола и ресетването на паролата съответно.

Middleware: Laravel използва междинен софтуер (middleware) за филтриране на заявките към приложението. Authenticate междинният софтуер проверява дали потребителят е влязъл в системата, преди да му позволи да достъпи определени ресурси.

Него използваме почти на всякъде, защото искаме потребителят да се е логнал преди да използва, която и да е от нашите системи

Authentication Views: Laravel предоставя готови изгледи (views) за формите за вход и регистрация.

За самия потребител ние пазим няколко неща в датабазата -

daily profit, печалбата за днес ( рестартира се всеки ден в 00:00)

highest profit - най-голяма печалба, която се изобразява на профила

elo - елото на играча, което определя и ранка му

Име/ Имейл/ Парола (хеширана чрез bycrypt)

Баджове, за които ще говорим отделно.

##### **IV.Баджове**

Баджовете са важен елемент от потребителския опит в нашата платформа. Те представляват визуални символи, отдаващи почит към достиженията на потребителите. Нашата система за баджове е базирана на три основни атрибута, които се съхраняват в нашата база данни - името на баджа, описанието му и уникален уникод, използван за представяне на баджа като емоджи.

Баджовете са дефинирани в нашата моделна структура като "много към много" отношение с потребителите, което означава, че един потребител може да има множество баджове, а един бадж може да бъде притежаван от множество потребители. Това отношение се поддържа чрез специална средна таблица в базата данни, която служи за свързване на потребителите с техните баджове.

Нашите баджове са разделени на два основни типа - автоматични и ръчни. Автоматичните баджове се присъждат автоматично на потребителите при постигане на определени постижения или събития в системата, като например заемане на първо място или постигане на значителна печалба. Ръчните баджове от друга страна се добавят ръчно от администраторите на системата.

Администраторите имат пълни права по отношение на управлението на баджовете - те могат да създават нови баджове, да дават баджове на потребители, да редактират съществуващи баджове и да изтриват баджове.

Баджът се визуализира като емоджи, използвайки уникалния си уникод. Уникодът е записан като част от баджа без 'U+', така че само числовата част се съхранява (например, за 'U+1621', ще съхраняваме само '1621'). След това използваме функция за декодиране, която използва '&#X' и ';' за форматиране на уникода в HTML елемент. Накрая, използваме функцията html\_entity\_decode, за да превърнем HTML съдържанието в съответния емоджи в UTF-8 формат.

Тази функционалност на баджовете добавя интерактивност на потребителския опит, предоставяйки възможности за придобиване на почетни символи и споделяне на своите постижения с общността.

###### V. Общности (communities)

С увеличаването на важността на социалните взаимодействия в онлайн пространството, ние в нашата платформа също приемаме подход, който цели да стимулира общността и да създава среда за обмен на мнения и дискусии. Този подход е вдъхновен от механиките, използвани от популярни социални мрежи като Reddit и се концентрира върху създаването на общности, които събират хора с общи интереси или цели.

Потребителите на нашата платформа имат възможността да създават свои собствени общности. Тези общности са публични и достъпни за всеки да ги прегледа. Съществуват две основни аспекти на всяка общност - името и описанието. Името представлява кратка идея за темата или целта на общността, докато описанието предлага по-подробен контекст и информация. Всяка общност също така има уникален идентификатор, който е автоматично генериран при създаването й.

Администраторът на общността има контрол върху съдържанието и може да изтрие цялата общност, ако това е необходимо. Това е функционалност, предназначена да помогне за поддържане на целостта и качеството на дискусиите в рамките на общността.

В рамките на тези общности, потребителите могат да публикуват съобщения или "постове". Всеки пост съдържа заглавие и тяло със съобщение, което обсъжда определена тема или въпрос. Отново, създателят на поста или администраторът на общността има право да изтрие поста, ако е необходимо.

Също така е въведена възможността за коментари към всяка публикация. Това допълнително обогатява диалога и предлага платформа за подробни дискусии и обмен на мнения. Коментарите могат да обхващат широк спектър от теми - от предложения за подобрение и посока на дискусията, до конкретни съвети и препоръки.

С тази нова функционалност се стремим да стимулираме активното участие на потребителите и да създадем виртуално пространство за обмен на идеи и информация, което в крайна сметка да подпомага растежа и успеха на нашата потребителска база.

###### VI. Ranked

Системата на ранкиране в нашата платформа е елегантна в своята простота и едновременно позволява точно измерване на уменията и постиженията на играчите. Идеята е базирана на концепцията за Ело рейтинг, широко използвана в шахa, но тук сме я адаптирали за нашите специфични нужди и условия.

Всеки нов играч започва с начално Ело от 600 точки. Това е вашата стартова линия - мястото, от което започва вашето пътуване. В зависимост от вашите постижения и загуби в нашата дневна игра, Ело рейтингът ви може да се покачи или спадне. Системата е справедлива и интуитивна: при печалба от 1$ вашето Ело се увеличава с 1 точка. Всеки долар печалба води до съответен ръст на вашия рейтинг. Въпреки това, за да запазим баланса и да предотвратим излишните колебания, в случай на печалба по-голяма от 25$, елото се увеличава с фиксирани 25 точки.

От друга страна, загубите в игрите водят до намаляване на Ело рейтинга ви. За всяка загуба от 1$, вашето Ело спада с 1 точка. Ако загубите повече от 30$, то вашият рейтинг ще спадне с максимално 30 точки, независимо от размера на загубата. Това ограничение помага да се предотврати драстичното спадане на рейтинга и дава възможност на играчите да се възстановят от неблагоприятни серии.

Вашето ело определя вашия ранк в нашата система. Има шест възможни ранга, които са разделени на две главни категории - ниско и високо ело.

Copper Cadet: < 610 ело

Silver Strategist: > 610, но < 670 ело

Gold Guardian: > 670, но < 720 ело

Ако вашият рейтинг достигне до 720 ело, влизате в елитния клуб на играчите с високо ело.

Platinum Prodigy: > 720, но < 800 ело

Diamond Dealer: > 800, но < 900 ело

Obsidian Overlord: > 900 ело

###### VII.Course

Вграденият курс, който предлагаме в нашата платформа, има за цел да помогне на потребителите да разберат и научат за следните аспекти, свързани с трейдването и криптовалутите:

Криптовалути: Курсът представя основни понятия и особености на криптовалутите като цифрови активи. Обяснява какво са криптовалутите, как работят транзакциите с тях и основните криптовалути, които могат да бъдат търгувани.

Блокчейн: Уроците се фокусират върху блокчейн технологията, която е основа на криптовалутите. Обяснява как функционира блокчейн, как се записват транзакции и как се осигурява сигурността и непроменимостта на данните в блокчейн мрежата.

Кога да продадем: Курсът обхваща стратегии и подходи за определяне на подходящия момент за продажба на криптовалути. Обяснява фактори, които могат да се вземат предвид при вземането на решение, като ценови анализ, технически индикатори и новини от крипто валутния пазар.

Нашите фактори: В курса се разглеждат конкретните фактори, които влияят на трейдването и криптовалутите в нашата платформа. Това може да включва различни параметри и показатели, които определят въз основа на нашите стратегии и модели.

Курсът има за цел да предостави на потребителите основни знания и умения, които са необходими за успешно търговия с криптовалути и разбиране на пазарните движения и стратегии.

**\*Admin**

Админ ще е отделно, защото не всеки потребител е админ.

Управление на потребители: Администраторът има възможността да преглежда и управлява потребителите в системата. Това включва преглед на списък с регистрираните потребители, промяна на ролите им. Така администраторът може да осигури контрол върху потребителите и да гарантира правилното функциониране на системата.

Присвояване на баджове: Администраторът има възможността да присвоява баджове на потребителите в системата. Баджовете представляват знакове на постижения и признание за потребителите и могат да бъдат предоставени за различни постижения или специални умения. Това дава възможност на администратора да мотивира и награди потребителите, както и да създаде по-интерактивно и ангажиращо потребителско изживяване.

###### Ръководство на потребителя

**Необходимо :**

Процесор (CPU): Двуядрен процесор с тактова честота от поне 2.0 GHz или по-висока.

Памет (RAM): Минимум 2 GB RAM.

Хард диск: Свободно дисково пространство от поне 300MB за инсталацията и работата на приложението.

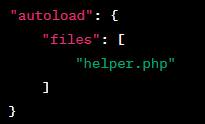
Графична карта: Вградена графична карта или дискретна графична карта с поддръжка на OpenGL 3.3 или по-нова версия.

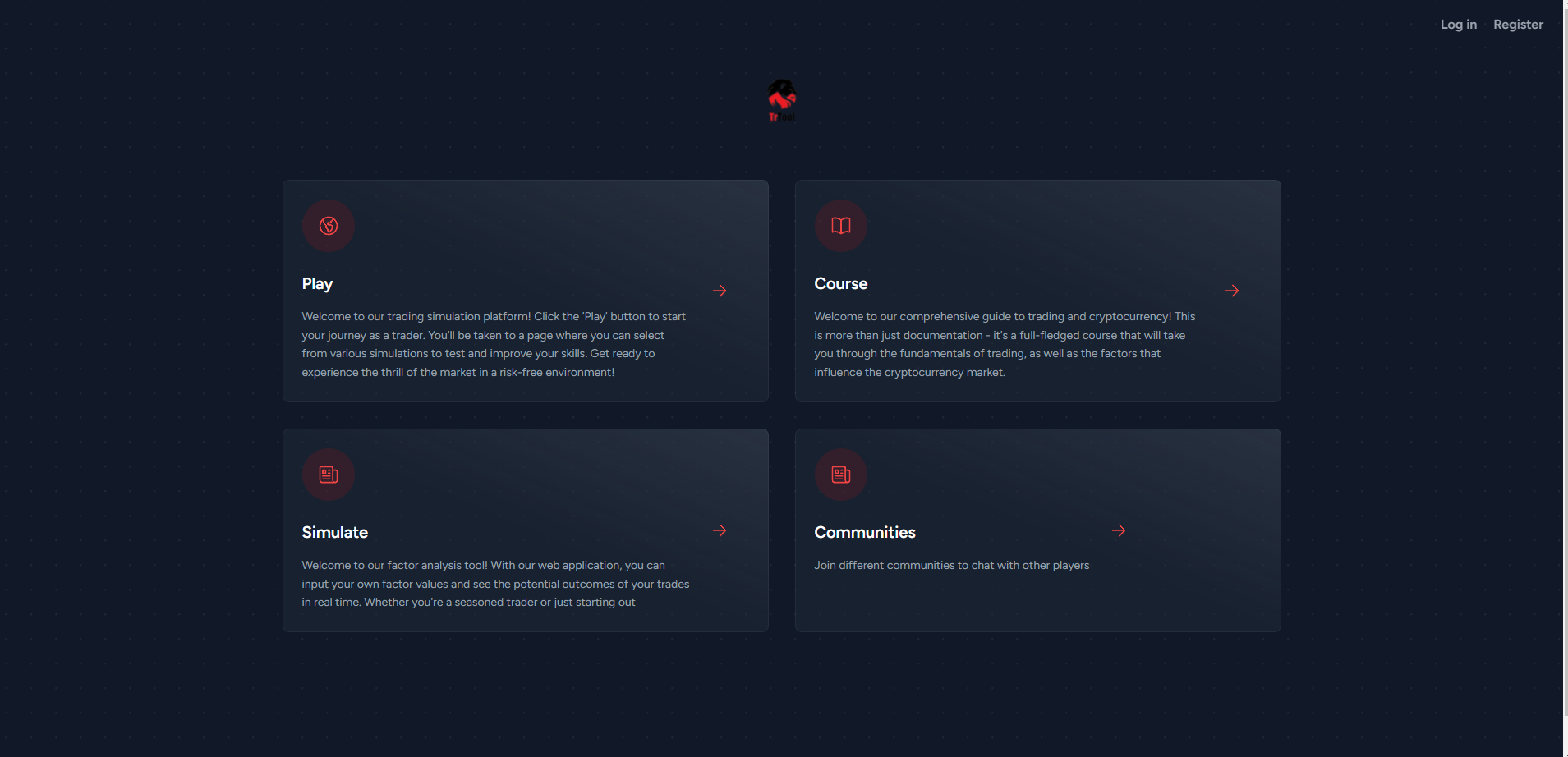
Операционна система: Поддържаните операционни системи включват Windows 7 или по-нова версия, macOS 10.12 Sierra или по-нова версия и Linux (различни дистрибуции).

Интернет връзка: Широколентова интернет връзка за сваляне на необходимите зависимости и актуализации на приложението.

Свалени PHP и Node.js

Стъпки за сваляне:

1. git clone https://github.com/BorisStoyanv/TrTool.git
2. cd TrTool
3. composer install
4. touch database/database.sqlite
5. конфигурирайте .env-example файл, като го прекръстите - .env
6. composer init
7. добавяте това в composer.json -
8. composer dump-autoload
9. php artisan key:generate
10. npm install
11. npm run dev
12. php artisan serve
13. <http://localhost:8000> :)

Ако всички стъпки минат по план ще сте на главната страница на TrTool

От тук избирате коя функция да използвате.

###### Заключение

В рамките на 4-месечен период, нашият екип успешно придоби познания и умения в областта на PHP и Laravel, които ни позволиха да реализираме значителна част от нашите идеи. Създадохме реалистична симулация с различни фактори и събития, предоставяйки на потребителите възможността да изпитат и тестват своите умения. Проектирахме и разработихме експериментална "зала", където потребителите могат да се насладят на играта и да получат опит в сферата на трейдването и криптовалутите.

Разработката включва и добавяне на реални стратегически елементи за трейдването и криптовалутите. Въведохме система за рангове, която предоставя потребителите с постижения и изгражда конкурентна общност. Създадохме и социално пространство, където потребителите могат да споделят знания, да обсъждат и да се взаимодействат.

Въпреки постигнатото, разработката ни оставя много възможности за подобрение, които сме решени да използваме в бъдеще. Някои от нашите планове включват:

Добавяне на още фактори и събития, за да подобрим реалистичността на симулацията. Желаем да предоставим на потребителите по-широк спектър от възможности и предизвикателства.

Въвеждане на възможност за игра срещу други потребители или срещу ботове/изкуствен интелект. Целта ни е да предоставим интерактивен и конкурентен опит на потребителите.

Предоставяне на възможност за получаване на баджове чрез специални кодове или линкове. Това ще позволи на потребителите да получават баджове в резултат на постижения или специални събития.

Разширяване на опциите на администраторите чрез създаване на административен панел - гъвкавост и удобство при управлението на системата, без да е необходимо да се докосваме до кода.

###### Използвана Литература

[**https://www.clariontech.com/blog/10-reasons-why-laravel-is-the-best-php-framework-for-2019**](https://www.clariontech.com/blog/10-reasons-why-laravel-is-the-best-php-framework-for-2019)

**на:**

[**-10 причини защо ларавел е най-добрият фреймуърк**](#jibthm2shsoa)

[**https://encircletechnologies.com/blog/advantages-of-tailwind-css/#Advantages\_of\_Using\_Tailwind\_CSS**](https://encircletechnologies.com/blog/advantages-of-tailwind-css/#Advantages_of_Using_Tailwind_CSS)

**на :**

* [**6 причини защо Taiwind css си струва**](#ygwwo4eoyqfa)

**Съдържание:**

1. [*Увод*](#_6q17x8x83k21)
2. [*Втора глава*](#_9bf2pq1ozuc5)
3. [*Трейдинг система*](#_y8o34ho3d45i)
4. [*Експериментална система*](#_p1bmrg3ko3xs)
5. [*Потребителска система*](#_5vristjsp8yz)
6. [*Баджове*](#_5kbej2qcd3sq)
7. [*Общности*](#_5aio6bayz5nn)
8. [*Ranked*](#_g5f864xayt4s)
9. [*Курс*](#_p3vuqjzo5jg)
10. [*Ръководство на потребител* / трета глава](#_uqpcjyb3ohr5)
11. [*Заключение*](#_666r3wxevx4e)
12. [*Използвана литература*](#_frx94euhiz7o)