جامعة دمشـــق

كلية الهندسة المعلوماتية

قسم هندسة البرمجيات

تطبيق ويب لمحاكاة عملBlockchain

إعداد الطالبة:بشرى عمر المحمد

البيئة المستخدمة:

- ***php FrameworkLaravel***
- ***DB MySql***
- ***FrontWeb Html,Css***

:BlockChain

تكنولوجيا الـ **Blockchain** هي تقنية تسجيل مشترك (Distributed Ledger Technology) تستخدم لتسجيل وتخزين المعلومات بشكل آمن وموثوق به.

يعتبر الـ **Blockchain** نوعًا من قواعد البيانات الموزعة حيث يتم توزيع البيانات عبر عدة أجهزة متصلة في شبكة بدلاً من تخزينها في مكان واحد مركزي.

يتكون الـ **Blockchain** من سلسلة متتالية من الكتل (Blocks) التي تحتوي على المعلومات المختلفة.

ترتبط الكتل معًا بواسطة وظيفة التجزئة الهاش(Hash Function) ، حيث يتم حساب هاش الكتلة الحالية بناءً على محتواها وهاش الكتلة السابقة، وبالتالي يتم إنشاء رابط دائم بين الكتل.

كيف تم تحقيق BlockChain ضمن المشروع ؟

تم تحقيق تقنية **BlockChain** في المشروع السابق باستخدام لغة برمجة Laravel وقاعدة بيانات MySQL.

ماهي خوارزمية التشفير المستخدمة ضمن تقنية **BlockChain** ,ومادورها ؟

تم استخدام خوارزمية التجزئة SHA-256 :عملية حساب هاش الكتلة والتحقق من صحة البيانات..

وتعتبر **SHA-256 (Secure Hash Algorithm 256-bit)** واحدة من أكثر خوارزميات التجزئة أمانًا وشيوعًا.

تستخدم لتوليد قيمة هاش فريدة للبيانات المدخلة.

دور خوارزميةSHA-256 هو تحويل بيانات الإدخال إلى سلسلة ثابتة وفريدة من الأحرف والأرقام بطول 256 بت (32 بايت).

يتم استخدام هذه السلسلة المولدة (هاش) كتوقيع رقمي للبيانات المدخلة.

وتتميز SHA-256 بالخصائص التالية:

- (Continuity): تغيير بيانات الإدخال حتى بتغيير بسيط يؤدي إلى تغيير كبير في الهاش الناتج.

- (Non-reversible): من الصعب جدًا استنتاج البيانات الأصلية من الهاش المولد، مما يعني أنه لا يمكن استعادة البيانات الأصلية من الهاش.

- (Collision resistance): من الصعب جدًا أن تكون هناك بيانتان مختلفتان تولدان نفس الهاش، مما يجعلها مناسبة لاستخدامها في ضمان سلامة البيانات.

مااهو مفهوم الـ Proof-of-Work (PoW) :

هو آلية مستخدمة في تقنية البلوكشين لتأكيد وتأمين العمليات وإضافة الكتل الجديدة إلى سلسلة الكتل.

يستفيد من **PoW** فيما يلي:

1. ضمان أمان الشبكة: يستخدم **PoW** لحماية الشبكة من هجمات الاحتيال والتلاعب بالبيانات.

يتطلب حل الألغاز المعقدة في **PoW** الكثير من الوقت والموارد الحسابية، مما يجعل من الصعب على المهاجمين السيطرة على الشبكة وتعديل الكتل بشكل غير مشروع.

2. توزيع العمل: يعمل **PoW** على توزيع العمل بين المشاركين في الشبكة.

يجب على المشاركين حل الألغاز المعقدة لإنشاء الكتل الجديدة، وهذا يتطلب موارد حسابية.

يتم منح المشارك الذي يحل اللغز بشكل صحيح حق إضافة الكتلة إلى السلسلة وجمع المكافأة المقدمة.

3. إنشاء الكتل الجديدة: يستخدم **PoW** لإنشاء الكتل الجديدة في سلسلة الكتل.

تم التحقيق من خلال مجموعة من التوابع:

بعد تهيئة البيئة وانشاء ال table ,Model ,Controller اللازمة لانشاء الكتل وتخزينها لدينا مجموعة من التوابع للقيام بذلك:

1-تابع index من أجل عرض جميع الكتل المدخلة:

```php
public function index()
{
    $blocks = Block::all();

    return view('blocks', compact('blocks'));
}
```

2-تابع getLastBlockHash: من أجل استرجاع قيمة هاش الكتلة السابقة لاستخدامها في عملية حساب هاش الكتلة الجديدة.

```php
private function getLastBlockHash()
{
    $lastBlock = Block::latest()->first();

    if ($lastBlock)
    {

        return $lastBlock->hash;
    }
    return ' No exist  just for the first block .';
}
```

```php
private function calcBlockHash($block)
{
    $data = $block->data . $block->previous_hash . $block->timestamp . $block->nonce;

    return hash('sha256', $data);
}
```

3-تابع calcBlockHash : تقوم بحساب هاش (hash) للبلوكة بناءً على مجموعة من البيانات المحددة ، وهي:

1. previousBlockHash: هو هاش البلوكة السابقة، ويستخدم لربط البلوكات في سلسلة الكتل. .

2. data: هو البيانات التي يتم تخزينها في البلوكة، مثل المعاملات أو المعلومات الأخرى المرتبطة بالتطبيق.

3. nonce: هو القيمة المستخدمة في عملية التعدين (Proof-of-Work)، والتي يتم تجريب قيم مختلفة لها للعثور على القيمة المناسبة التي تنتج هاش البلوكة المطلوب وتستوفي شرط الصعوبة المحددة.

```php
private function mineBlockNonce($block)
{
    $zero_number = 5;

    while (true)

    {
        $block->nonce++;

        $hash = $this->calcBlockHash($block);

        if (substr($hash, 0, $zero_number) === str_repeat('0', $zero_number))
        {
            return $block->nonce;
        }
    }
}
```

4-تابع mineBlockNonce: هو المسؤول عن حل الألغاز المعقدة (Proof-of-Work) للحصول على قيمة nonce المناسبة للكتلة" تقوم بعملية التعدين الفعلية لإيجاد القيمة المناسبة لـ nonce التي تلبي شرط صعوبة العملية، مما يساهم في حماية البلوكشين وضمان أمانه وتوثيق العمليات المتماثلة. "

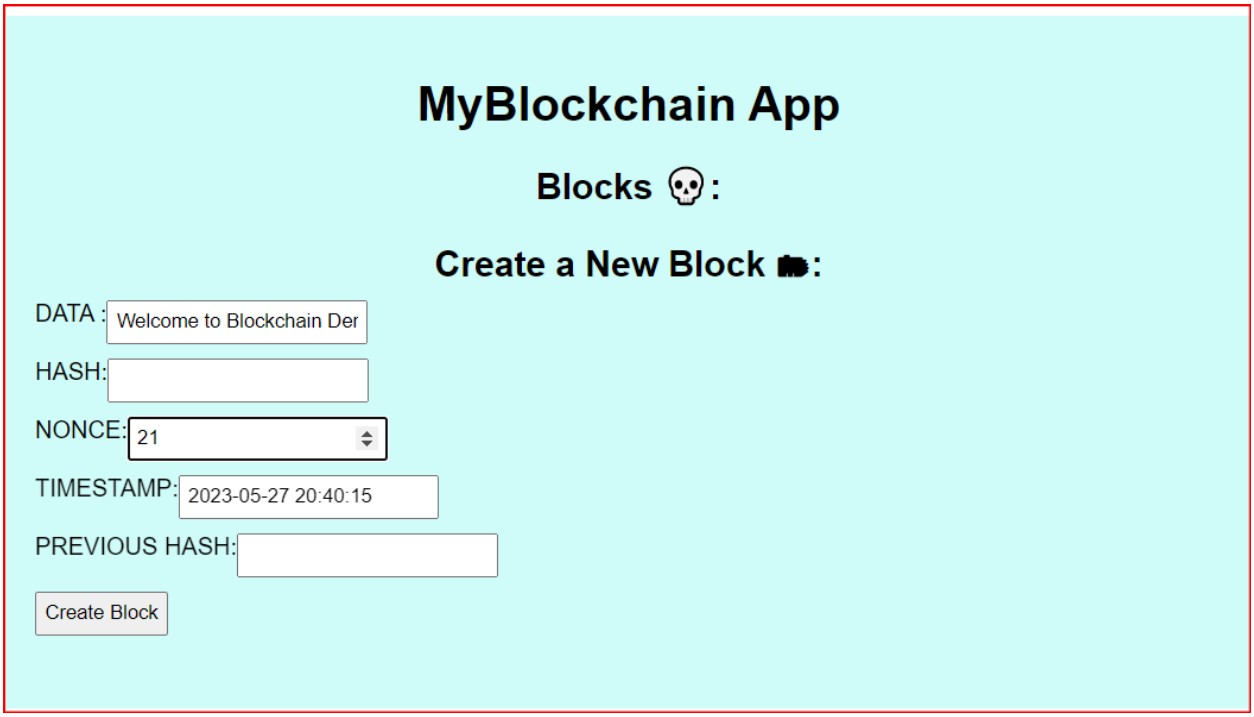
ويقوم بتجريب قيم مختلفة لـ nonce وحساب هاش الكتلة المتعلقة بهذه القيمة، ثم التحقق مما إذا كان هاش الكتلة يفي بشرط الصعوبة المحددة.

بحيث يتم تجريب قيم الـ nonce بشكل تتابعي حتى يتم العثور على القيمة التي تولد هاش الكتلة المطلوب والذي يبدأ بعدد معين من الصفرات ويستوفي شرط الصعوبة المحددة.

```php
public function storeBlock(Request $request)
{
    $data = $request->input('data');

    $block = new Block();

    $block->data = $data;

    $block->previous_hash = $this->getLastBlockHash();

    $block->nonce = $this->mineBlockNonce($block);

    $block->hash = $this->calcBlockHash($block);

    $block->timestamp = now();

    $block->save();

    return response()->json([

        'message' => 'Block created successfully for block ' ,

        'block' => $block,
    ]);
}
```

5- تابع storeBlock : هي تخزين البيانات الخاصة بالبلوكة، مثل الهاش (hash) للبلوكة والهاش السابق (previous hash) والبيانات الأخرى المهمة.

التنفيذ:



- قام بإعادة النتيجة ك JSON:

{"message":"Block created successfully for block ","block":{"data":"Welcome to Blockchain Demo 2.0!","previous_hash":" No exist just for the first block
.","nonce":689666,"hash":"00000f300c1dcdd65f62d2c2c5b83f4193f42eadb330a6673688183ea5c2d7b0","timestamp":"2023-05-27T20:40:55.463724Z","updated_at":"2023-05-27T20:40:55.000000Z","created_at":"2023-05-27T20:40:55.000000Z","id":1}}

- قام بانشاء 2blocks اعتمادا على قيمة الهاش السابقة :

MyBlockchain App

Blocks 💀:

Hash 🤛: 00000f300c1dcdd65f62d2c2c5b83f4193f42eadb330a6673688183ea5c2d7b0
Previous Hash 🔒: No exist just for the first block .
Timestamp 🕐: 2023-05-27 20:40:55

Hash 🤛: 00000e4614fb7cdbad0ffcc2f4c50aad30e7a88538e17fe18568713b995efe5c
Previous Hash 🔒: 00000f300c1dcdd65f62d2c2c5b83f4193f42eadb330a6673688183ea5c2d7b0
Timestamp 🕐: 2023-05-27 20:42:26

Create a New Block [redacted]:

DATA : name is boushra almouham
HASH:
NONCE: 21
TIMESTAMP: 2023-05-27 20:42:28
PREVIOUS HASH:

Create Block

MyBlockchain App

Blocks 💀:

Hash 🔐: 00000f300c1dcdd65f62d2c2c5b83f4193f42eadb330a6673688183ea5c2d7b0
Previous Hash 🔒: No exist just for the first block .
Timestamp 🕐: 2023-05-27 20:40:55

Hash 🔐: 00000e4614fb7cdbad0ffcc2f4c50aad30e7a88538e17fe18568713b995efe5c
Previous Hash 🔒: 00000f300c1dcdd65f62d2c2c5b83f4193f42eadb330a6673688183ea5c2d7b0
Timestamp 🕐: 2023-05-27 20:42:26

Hash 🔐: 00000407da8391d7cf32712408a43ab54e621d08dcb108e0c960c279921822cd
Previous Hash 🔒: 00000e4614fb7cdbad0ffcc2f4c50aad30e7a88538e17fe18568713b995efe5c
Timestamp 🕐: 2023-05-27 20:43:55

Create a New Block 👉:

DATA : I study ITE

HASH:

NONCE: 21

TIMESTAMP: 2023-05-27 20:43:57

PREVIOUS HASH:

Create Block

- البيانات مخزنة ضمن الداتا بيز:

	id	hash	data	previous_hash	timestamp	nonce	create
Edit Copy Delete	1	00000f300c1dcdd65f62d2c2c5b83f4193f42eadb330a66736...	Welcome to Blockchain Demo 2.0!	No exist just for the first block .	2023-05-27 20:40:55	689666	2023-0 20:40:
Edit Copy Delete	2	00000e4614fb7cdbad0ffcc2f4c50aad30e7a88538e17fe185...	name is boushra almouhammad	00000f300c1dcdd65f62d2c2c5b83f4193f42eadb330a66736...	2023-05-27 20:42:26	848716	2023-0 20:42:
Edit Copy Delete	3	00000407da8391d7cf32712408a43ab54e621d08dcb108e0c9...	I study ITE	00000e4614fb7cdbad0ffcc2f4c50aad30e7a88538e17fe185...	2023-05-27 20:43:55	618197	2023-0 20:43: