Projekt zaliczeniowy JS

Projekt na zaliczenie części **JS** z laboratorium **Technologie Internetowe**.

Uniwersytet Rzeszowski, Informatyka I st. II rok sem. zimowy

Wykonał: **Karol Bobowski**

Link do działającej wersji strony: https://prettyniceguy14.github.io/projekt_js/

Opis projektu

Projekt przedstawia przykładową implementacje struktury opartej o blockchain. Aplikacja pozwala na tworzenie i "wydobywanie" bloków zawierających dane. Pozwala również na zmiany w "wydobytych" wcześniej blokach i wizualizację konsekwencji, jakie niosą ze sobą owe zmiany tj. zmiana hash'u aktualnego i każdego następnego bloku, "zepsucie" samej struktury - (uznanie blockchain'u jako invalid) i konieczność ponownego "wydobycia" wszystkich invalid bloków.

Projekt opiera się o model **proof of work** gdzie blok uznawany jest jako **valid** w przypadku gdy jego hash rozpoczyna się od danej ilości zer równej trudności sieci w czasie wydobycia samego bloku. W tym przypadku aplikacja korzysta z algorytmu **SHA-1** do wyliczenia hash'u danego bloku w którego skład wchodzą:

```
- ID bloku
- Poprzedni hash bloku
- Data utworzenia bloku
- Dane bloku
- Trudność wydobycia
- NONCE
```

Wydobycie bloku polega na zwiększaniu wartości **NONCE**, kalkulacji hash'u i każdorazowym sprawdzeniu, czy owy zaczyna się od danej ilości zer ustalonej na podstawie trudności sieci w danej chwili. Należy pamiętać, że im większa trudność sieci, tym dłużej zajmie "wydobycie" danego bloku tj. znalezienie hash'u spełniającego warunek. Na czas "wydobycia" danego bloku ma również wpływ specyfikacja sprzętu, który dokonuje obliczeń.

Aplikacja pozwala na ręczną inkrementacje i dekrementacje trudności sieci która defaultowo ustawiona jest na **2**. (To znaczy: hash każdego **valid** bloku musi zaczynać się od dwóch zer).

Znalezienie odpowiedniego **nonce** bloku (tj. jego wydobycie) przy trudności sieci równej **2** jest stosunkowo szybkie (mniej niż sekunda) na PC jak i na telefonie, zwiększenie trudności do **5** drastycznie zwiększa czas wydobycia do kilkudziesięciu sekund, natomiast przy wartości **6** wydobycie trwa nawet kilka minut (testowane na i9-9900k).

`Każda kolejna inkrementacja trudności sieci ma gargantuiczny wpływ na czas wydobycia bloku. W przypadku modyfikacji któregoś z już wydobytych bloków, każdy następny blok musi zostać ponownie 'wydobyty' co niesie za sobą ogromny koszt czasowy.`

Struktura projektu

Strona składa się z czystego HTML, CSS i JS bez użycia jakichkolwiek bibliotek czy też gotowych komponentów.

Zawartość plików i katalogów:

> `index.html` - aplikacja.
>
> `blockchain.js` - plik zawierający klasę bloku, klasę blockchain'a oraz asynchroniczną funkcję liczącą SHA-1 i funkcję do generowania aktualnej daty w przystępnym formacie.
>
> `script.js` - plik zawierający instancję klasy blockchain (`boboChain`), wszystkie eventListenery w aplikacji oraz funkcję która renderuje bloki na stronie (`function render(...){...}`).
>
> `style.css` - plik zawierający wszystkie css'y na stronie.
>
> `../js/` - katalog zawierający skrypty.
>
> `../img/` - katalog zawierający wszystkie grafiki umieszczone na stronie.

Kawałki kodu

Funkcja licząca SHA-1 (`crypto.*` jest wbudowane w przeglądarke):

```
async function sha1(message) {
    const msgBuffer = new TextEncoder().encode(message);
    const hashBuffer = await crypto.subtle.digest('SHA-1', msgBuffer);
    const hashArray = Array.from(new Uint8Array(hashBuffer));
    const hashHex = hashArray.map(b => b.toString(16).padStart(2, '0')).join('');
    return hashHex;
}
```

Klasa bloku oraz metody do kalkulacji aktualnego hash'u i wydobycia bloku:

```
class Block{
    constructor(index, time, data, diff, previousHash = ''){
        this.index = index;
        this.time = time;
        this.data = data;
        this.block_difficulty = diff;
        this.previousHash = previousHash;
        this.hash = "";
        this.nonce = 0;
    }

    async calculateHash(){
        var result =  await sha1(this.index + this.previousHash + this.time + this.data + this.diff + this.nonce);
        return result;
    }

    async mineBlock(diff){
        this.nonce = 0;
        this.hash = await this.calculateHash();
        while(this.hash.substring(0, diff) !== Array(diff + 1).join("0")){
            this.nonce++;
            this.hash = await this.calculateHash();
        }
    }
}
```

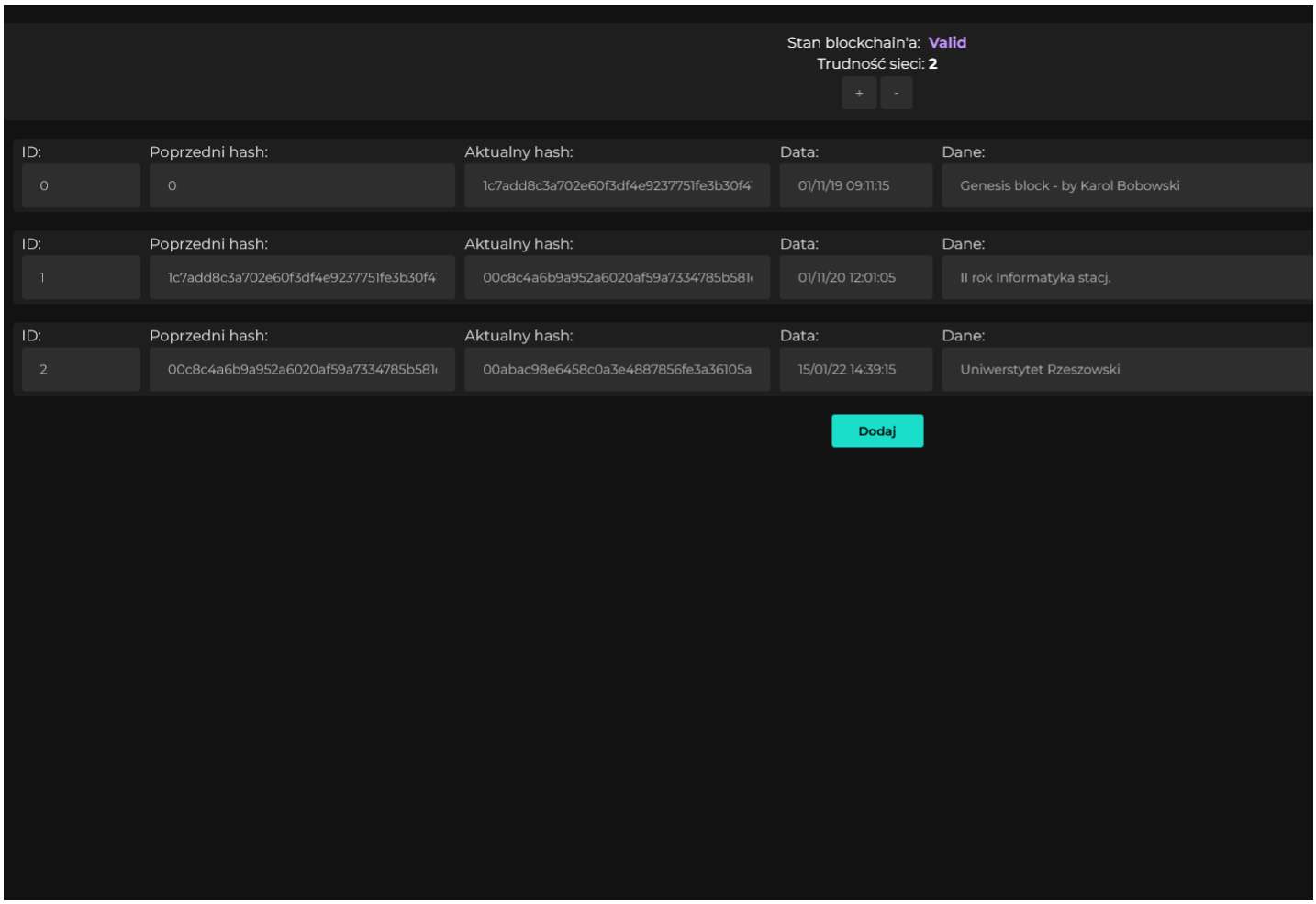
Event listener zapięty na pole zawierające dane bloku:

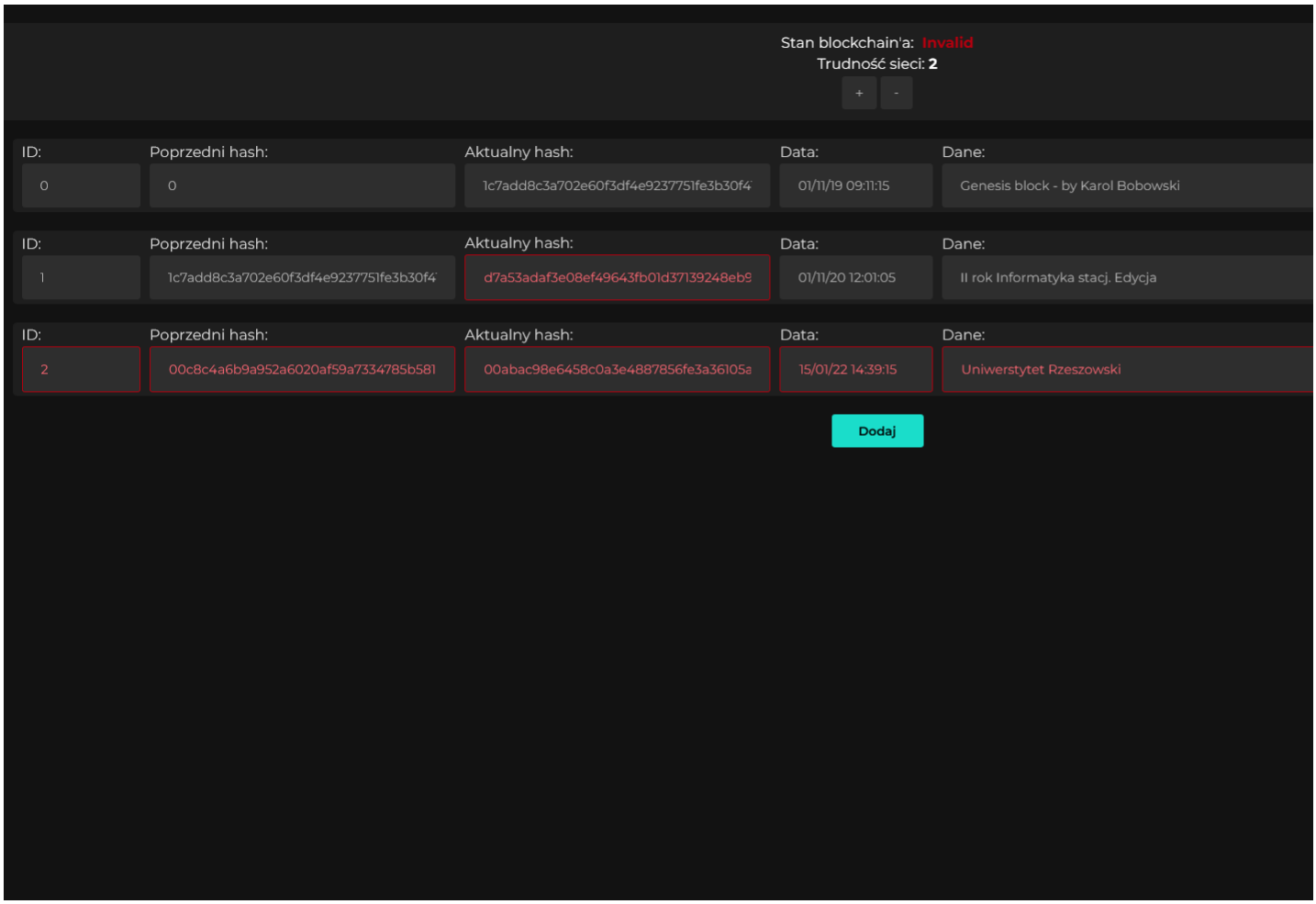
```
newDiv.children[4].children[1].addEventListener("blur", function(r) {
            const newValue = r.target.value;
            const blockID = r.target.parentElement.parentElement.children[0].children[1].value;
            chain.chain[blockID].data = newValue;
            chain.chain[blockID].calculateHash().then(newHash => {
                chain.chain[blockID].hash = newHash;
                render(chain);
            });
        });
```

Strona jest w pełni zgodna ze standardami W3C:

> https://validator.w3.org/nu/?doc=https%3A%2F%2Fprettyniceguy14.github.io%2Fprojekt_js%2Findex.html

UI / Wygląd Desktop





Stan blockchain'a: **Invalid**

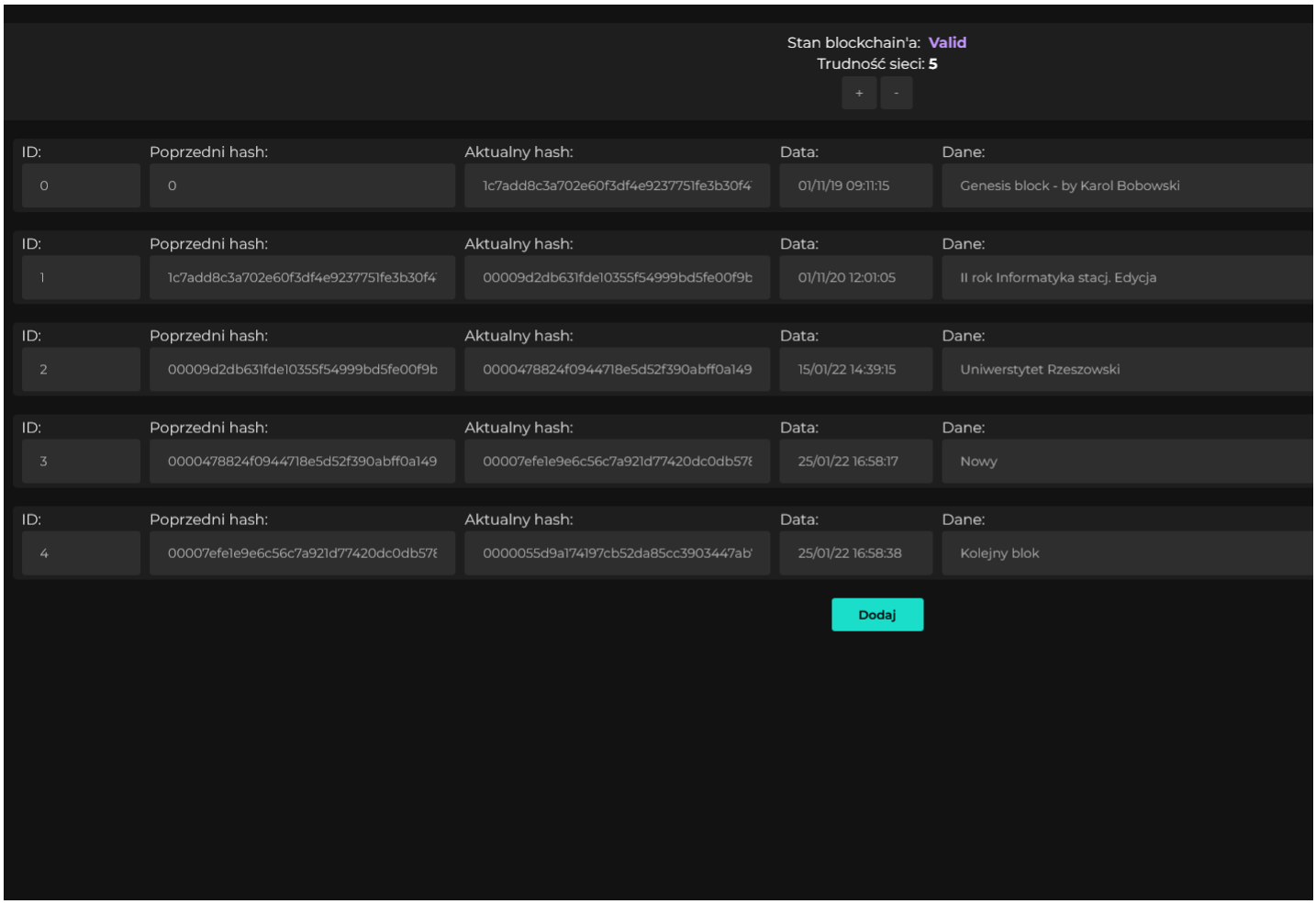
Trudność sieci: **4**

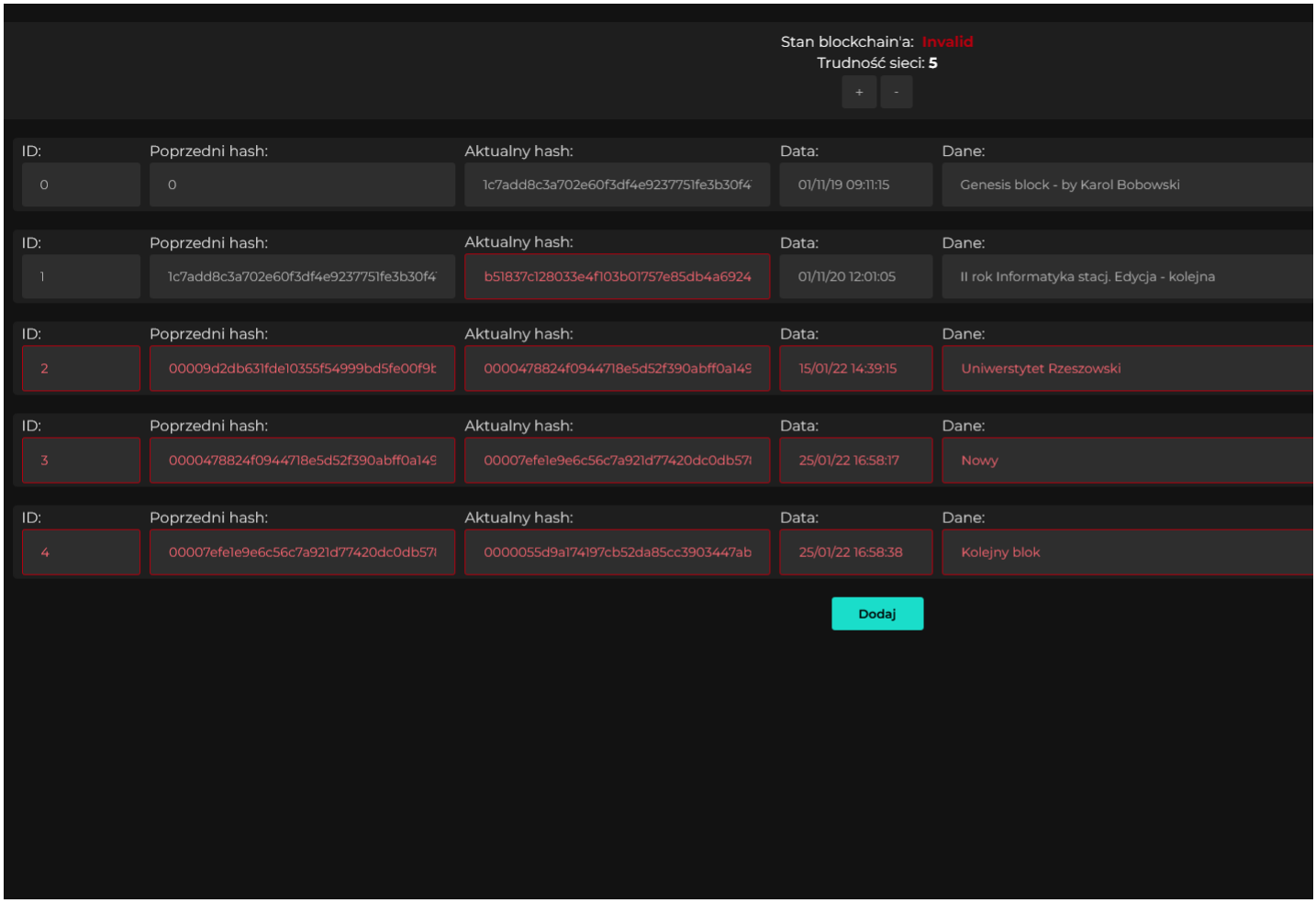
\+ -

ID:	Poprzedni hash:	Aktualny hash:	Data:	Dane:
0	0	1c7add8c3a702e60f3df4e9237751fe3b30f4	01/11/19 09:11:15	Genesis block - by Karol Bobowski
1	1c7add8c3a702e60f3df4e9237751fe3b30f4	00009d2db631fde10355f54999bd5fe00f9b	01/11/20 12:01:05	II rok Informatyka stacj. Edycja
2	00009d2db631fde10355f54999bd5fe00f9b	0000478824f0944718e5d52f390abff0a149	15/01/22 14:39:15	Uniwerstytet Rzeszowski
3	00abac98e6458c0a3e4887856fe3a36105a	be315f705976103d59c9a60dea4db416370(	25/01/22 16:58:17	Nowy
4	be315f705976103d59c9a60dea4db416370(	64b3c9a0994817c65c7670a0f37988516e2(	25/01/22 16:58:38	Kolejny blok

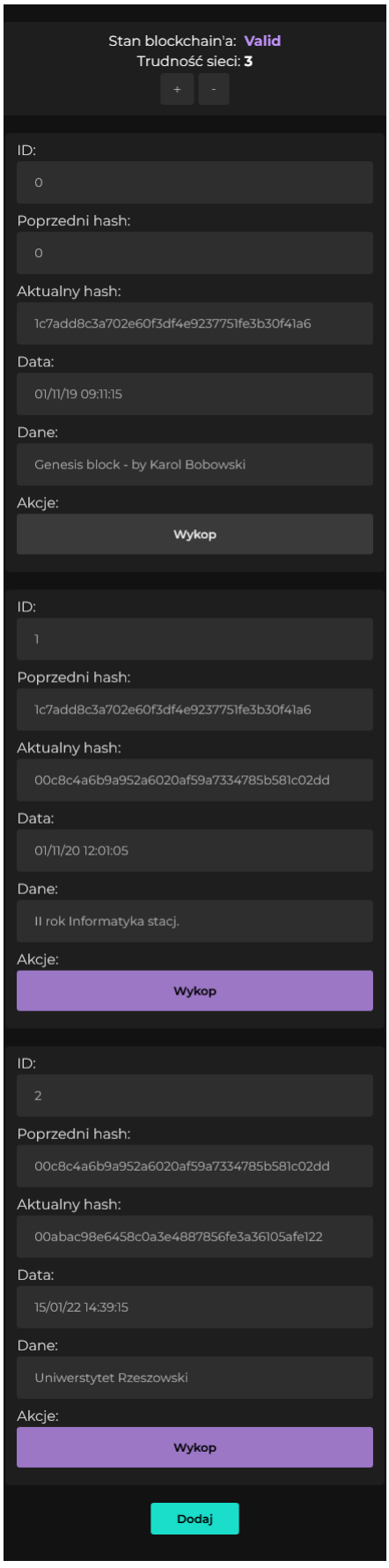
Dodaj

Wydobywanie...





UI / Wygląd Mobile


Stan blockchain'a: Valid
Trudność sieci: 3
+
-
ID:
0
Poprzedni hash:
0
Aktualny hash:
1c7add8c3a702e60f3df4e9237751fe3b30f41a6
Data:
01/11/19 09:11:15
Dane:
Genesis block - by Karol Bobowski
Akcje:
Wykop
ID:
1
Poprzedni hash:
1c7add8c3a702e60f3df4e9237751fe3b30f41a6
Aktualny hash:
00c8c4a6b9a952a6020af59a7334785b581c02dd
Data:
01/11/20 12:01:05
Dane:
II rok Informatyka stacj.
Akcje:
Wykop
ID:
2
Poprzedni hash:
00c8c4a6b9a952a6020af59a7334785b581c02dd
Aktualny hash:
00abac98e6458c0a3e4887856fe3a36105afe122
Data:
15/01/22 14:39:15
Dane:
Uniwerstytet Rzeszowski
Akcje:
Wykop
Dodaj


Stan blockchain'a: Invalid
Trudność sieci: 4
+
-
ID:
0
Poprzedni hash:
0

Aktualny hash:

1c7add8c3a702e60f3df4e9237751fe3b30f41a6

Data:

01/11/19 09:11:15

Dane:

Genesis block - by Karol Bobowski

Akcje:

Wykop

ID:

1

Poprzedni hash:

1c7add8c3a702e60f3df4e9237751fe3b30f41a6

Aktualny hash:

c6a8c828c37490009f9c12b67a3d7c5e802aafab

Data:

01/11/20 12:01:05

Dane:

II rok Informatyka stacj. Zmiana!

Akcje:

Wykop

ID:

2

Poprzedni hash:

00c8c4a6b9a952a6020af59a7334785b581c02dd

Aktualny hash:

00abac98e6458c0a3e4887856fe3a36105afe122

Data:

15/01/22 14:39:15

Dane:

Uniwerstytet Rzeszowski

Akcje:

Wykop

ID:

3

Poprzedni hash:

00abac98e6458c0a3e4887856fe3a36105afe122

Aktualny hash:

4c40772178d1dc864b40ea3f4662512c857662a6

Data:

25/01/22 17:05:19

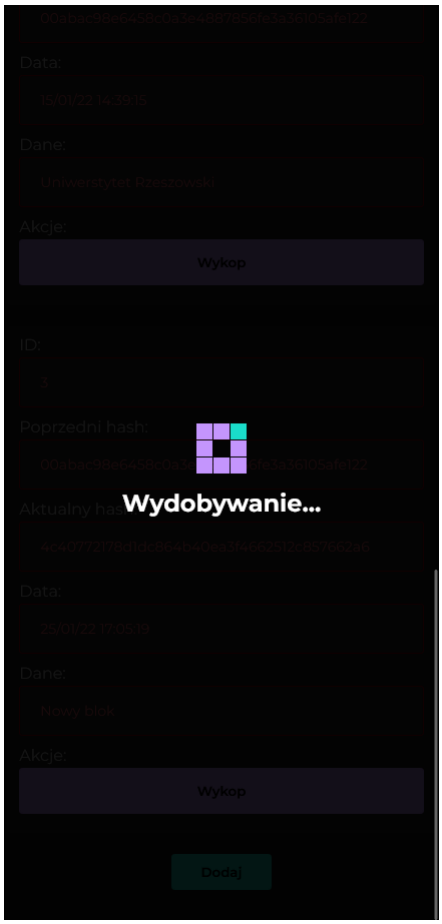
Dane:

Nowy blok

Akcje:

Wykop

Dodaj


00aba
Data:
15/01/22 14:39:15
Dane:
Uniwersytet Rzeszowski
Akcje:
Wykop
ID:
3
Poprzedni hash:
00aba
Wydobywanie...
Aktualny ha
4c4077
Data:
25/01/22 17:05:19
Dane:
Nowy blok
Akcje:
Wykop
Dodaj