

QUESTION-1 : ANDERS-BLOCKCHAIN

1. Checking the Hash value for some random data

The screenshot shows a web browser window with the URL `anderbrownworth.com/blockchain/hash`. The page has a dark navigation bar with the text "Blockchain Demo" and several menu items: "Hash", "Block", "Blockchain", "Distributed", "Tokens", and "Coinbase". The "Hash" menu item is highlighted. The main content area is titled "SHA256 Hash". It features a "Data:" label next to a large text input field containing the text "This is a test data to check for the Hash value as part of Assignment-3 for BlockChain". Below the input field, a "Hash:" label is followed by a text box displaying the resulting hash: `6ad005e522cbb58ce04a4e8b0ff79c38836054e8f8cfd124b79da73d1555f348`.

2. Checking for how to mine if we are changing the data

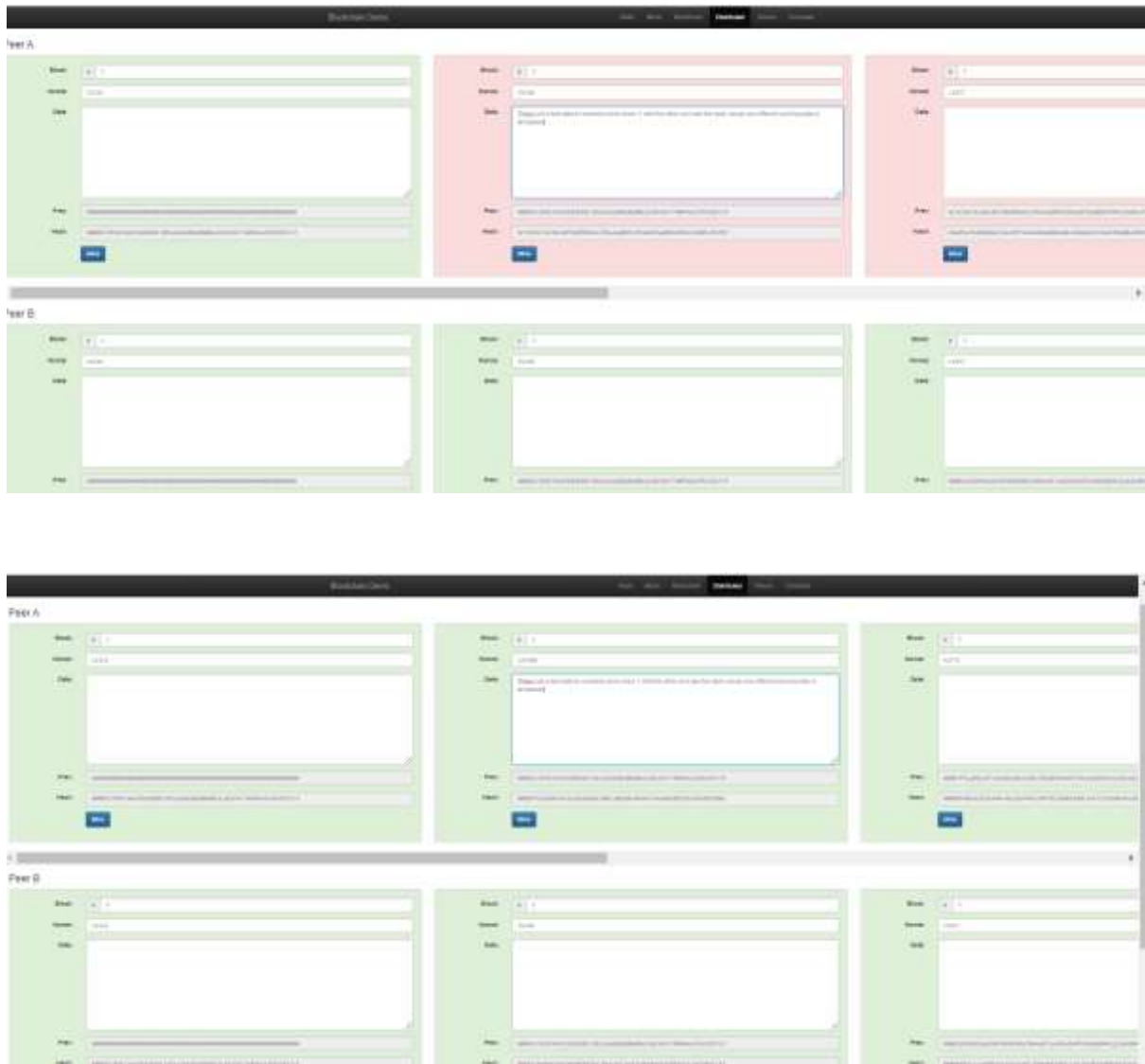
The screenshot shows the "Block" page of the Blockchain Demo website. The navigation bar is the same as in the previous screenshot, but the "Block" menu item is highlighted. The main content area is titled "Block" and has a light green background. It contains a "Block:" label with a dropdown menu showing "# 1". Below this is a "Nonce:" label with a text input field containing "72688". The "Data:" label is next to a large text input field. Below the input field, a "Hash:" label is followed by a text box displaying the hash: `800bf727854b50b05c254b09c3fe5c92e5b0cfa4bcb5dc270f55aa95a165e5a`. At the bottom of the form is a blue button labeled "Mine".

This screenshot shows the "Block" page after the "Mine" button was clicked. The "Block:" dropdown still shows "# 1". The "Nonce:" text input field now contains "23348". The "Data:" text input field now contains the text "This is a test data and now I have to mine it to get the required hash value by clicking mine button". The "Hash:" text box now displays a new hash: `800bf5142960da4800149cfff8cc92b0d0f4d98bd4bf317a92ae2fa5a92b5c90`. The blue "Mine" button remains at the bottom.

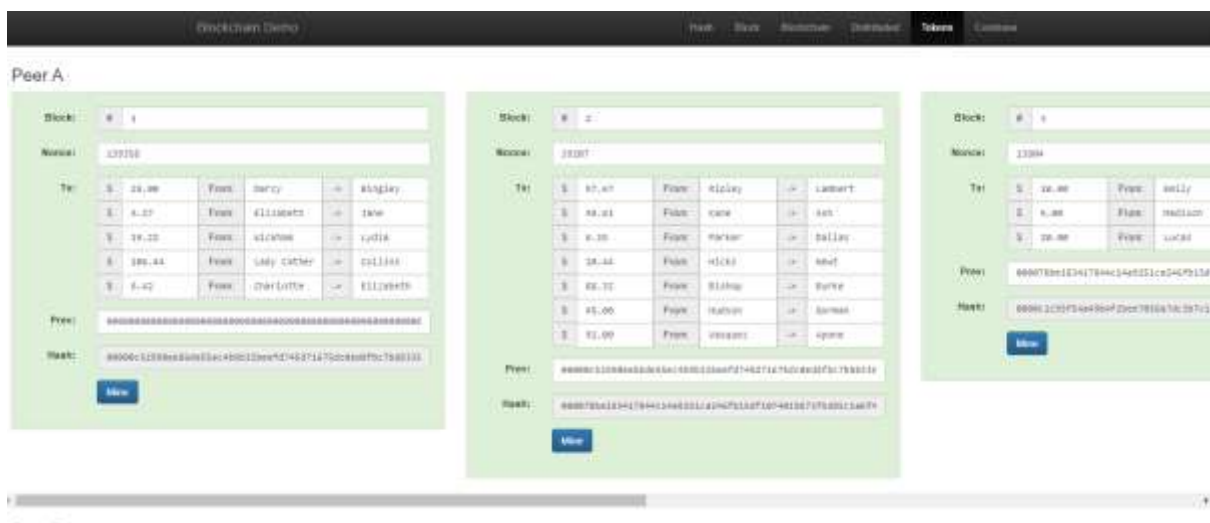
- Seeing various blocks and how their values change, when there is change in any one block data or value

The image shows two screenshots of a 'Blockchain Demo' application. The top screenshot shows three blocks in a chain. Block 1 (green) has a nonce of 11136 and a hash of 00001578b754259d382017d01a36d2060860e2cbb. Block 2 (pink) has a nonce of 35236 and a hash of 100009610ea2edcfd1a7452f740c6d1f7935d10ea5. Block 3 (pink) has a nonce of 12937 and a hash of 0d94f9b4006deF83867a79cc10a3bf6c3054. The bottom screenshot shows the same three blocks, but Block 2's data has been updated to 'Block-2 has now been update and now we have to mine each block to correct the data'. This change has propagated to Block 3, which now has a nonce of 142782 and a hash of 000010ac9a3772b94aeF2b0453a3bfbd7d122a099. The application interface includes a navigation bar with 'Hash', 'Block', 'Blockchain', 'Distributed', 'Tickets', and 'Database' tabs, and a 'Mine' button for each block.

- Checking the distributed network to show how we can compare one blockchain with other chains over the network to identify the data has been tampered and needs correction



5. Checking for the tokens in order to get an idea for how the data transactions are handled in real scenarios



6. This shows how Coin base Transactions are taking place

The screenshot displays the Blackboard Desktop application interface, which is divided into three main sections: a top navigation bar, a central workspace, and a bottom status bar.

Top Navigation Bar: Contains tabs for 'Home', 'Stack', 'Bookmarks', 'Downloads', 'Files', and 'Crawler'.

Central Workspace: Displays three transaction windows, each with a green header and a white body.

- Transaction Window 1 (Left):**
 - Amount: 100.00
 - Counterparty: 100.00 (with a dropdown arrow)
 - To: [Empty field]
 - From: [Empty field]
 - Hash: [Empty field]
 - Max button
- Transaction Window 2 (Middle):**
 - Amount: 100.00
 - Counterparty: 100.00 (with a dropdown arrow)
 - To: [Empty field]
 - From: [Empty field]
 - Hash: [Empty field]
 - Max button
- Transaction Window 3 (Right):**
 - Amount: 100.00
 - Counterparty: 100.00 (with a dropdown arrow)
 - To: [Empty field]
 - From: [Empty field]
 - Hash: [Empty field]
 - Max button

Bottom Status Bar: Displays 'Peer B' and 'Peer C' with their respective transaction details.

- Peer B:**
 - Stack: 4
 - Bookmarks: 100.00
 - Downloads: 100.00 (with a dropdown arrow)
- Peer C:**
 - Stack: 4
 - Bookmarks: 100.00
 - Downloads: 100.00 (with a dropdown arrow)

QUESTION-2: Make a Sample Contract and Publish it

DEPLOY & RUN TRANSACTIONS

Transactions recorded: 1

All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment, e.g. Transactions created in Javascript VM can be replayed in the Injected Web3.

Deployed Contracts

LAND AT OX41_EPCAD_BLOCKCHAIN

Land1 "Denny"

setDetails

setNewOwner: Denny

Transact

getDetails

to: string Denny

Low level interactions

CALLDATA

Transact

```

1  pragma solidity >=0.4.17 <=0.7.0;
2
3  contract land{
4      string ownerOfTheLand;
5
6      function land1 (string memory newOwner) public{
7          ownerOfTheLand = newOwner;
8      }
9
10     function setDetails(string memory setNewOwner) public{
11         ownerOfTheLand = setNewOwner;
12     }
13
14     function getDetails () public view returns(string memory){
15         return ownerOfTheLand;
16     }
17 }

```

listen on network

Search with transaction hash or address

[block: 830295 txindex: 17] from: 0x087...658d1 to: land.Land1(string) msg32...epcad value: 0 wei data: 0x25b...0000

Code is as Below

```
pragma solidity >=0.4.17 <0.7.0;
```

```
contract Land{
    string ownerOfTheLand;
```

```
function Land1 (string memory newOwner) public{
    ownerOfTheLand = newOwner;
}

function setDetails(string memory setNewOwner) public{
    ownerOfTheLand = setNewOwner;
}

function getDetails () public view returns(string memory){
    return ownerOfTheLand;
}
}
```