

Mine It – Basic Proof-of-Work Simulation



**Objective/Aim:**

To simulate the Proof-of-Work (PoW) mining process by finding a nonce such that the SHA-256 hash of a block (header + data + nonce) begins with a chosen number of leading zeros (difficulty).

**Apparatus/Software Used:**

* Laptop / PC
* PoW Simulator (offline demo tool or simple Python script)
* Word/Docs for documentation

**Theory/Concept:**

* What is Proof-of-Work?  
   PoW is a consensus mechanism that requires miners to solve a computationally expensive puzzle before adding a block to the blockchain.
* Nonce (Number used once):  
   A variable number that miners adjust to find a valid hash matching the difficulty criteria (e.g., hash starting with N zeros).
* Properties of PoW:
* Determinism: Same input always gives the same hash.
* Avalanche Effect: Even a minor input change produces a completely different hash.
* Security: Mining requires effort, making attacks costly.
* Block Components in Simulation:
* Block Header (metadata)
* Data (transaction or payload)
* Previous Block Hash
* Nonce
* Timestamp



**Procedure:**

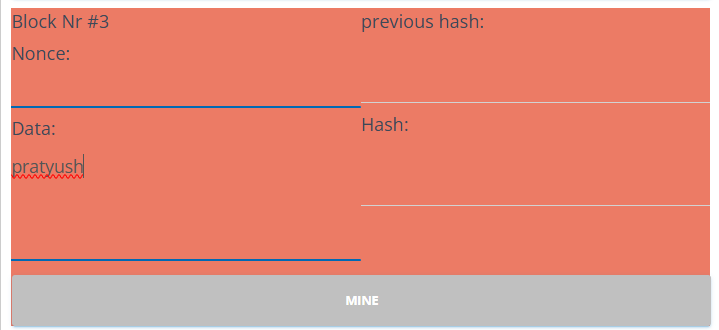
**Step 1: -** Open the browser

**Step 2: -** There is a proof of work simulator where in realtime you can enter the data and mine a block

**at: https://blockchain-academy.hs-mittweida.de/2021/05/proof-of-work-simulator/**

**Step 4:** There are blocks where you can give the input the data and mine it

**Step 5:** One by one give data and mine all the block.



**Observation:**

* For the same input it will generate the same hash but if single alphabet or

Number or space changes then it changes the hash even if the change.

* The SHA-256 algorithm provides a one-way hash—it is not possible to retrieve the original input from the hash, ensuring data confidentiality.



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |