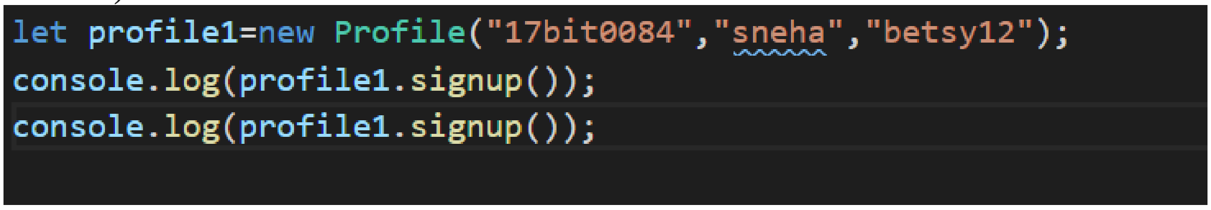
OUTPUT SNIPPETS WITH EXPLANATION

FEATURE 1: REMOVAL OF FAKE ACCOUNTS FROM SOCIAL MEDIA.

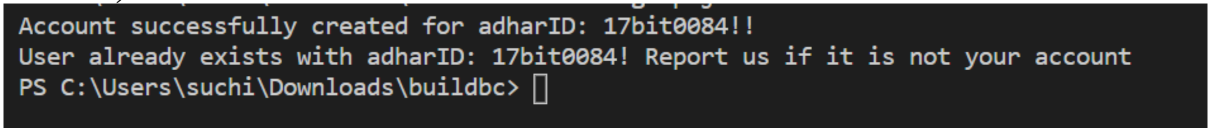
To achieve this, every user will be asked to enter their Adhar ID details along with their username and password during the signup process. Once a user tries to create an account with us, model checks if the unique Aadhaar ID has already been used. If so, the server notifies them that an account already exists and asks them to report if that wasn't their account. This technique is something that is blooming right now. Digital Identity verification has become something very important with security and fake account issues growing dramatically.

1. a) SIGN UP CODE :



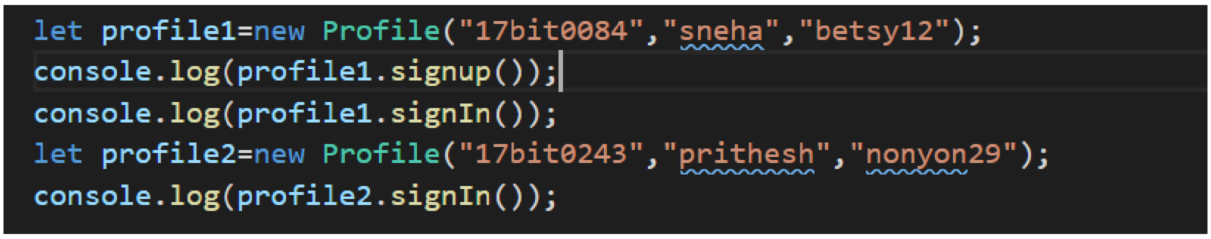
```
let profile1=new Profile("17bit0084","sneha","betsy12");
console.log(profile1.signup());
console.log(profile1.signup());
```

b) SAMPLE OUTPUT:



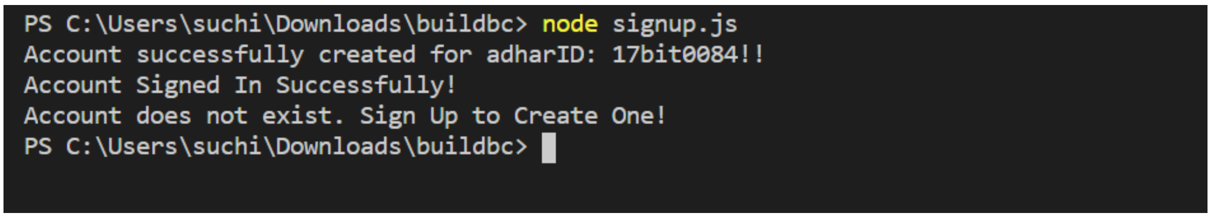
```
Account successfully created for adharID: 17bit0084!!
User already exists with adharID: 17bit0084! Report us if it is not your account
PS C:\Users\suchi\Downloads\buildbc>
```

2. a) Sign in:



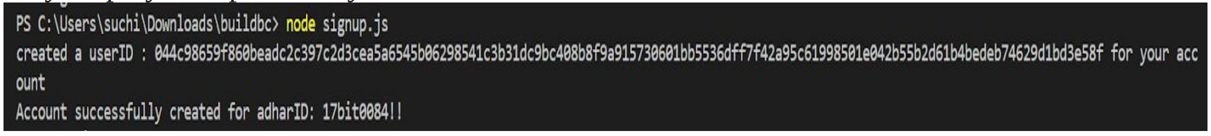
```
let profile1=new Profile("17bit0084","sneha","betsy12");
console.log(profile1.signup());
console.log(profile1.signIn());
let profile2=new Profile("17bit0243","prithesh","nonyon29");
console.log(profile2.signIn());
```

b) SAMPLE OUTPUT:



```
PS C:\Users\suchi\Downloads\buildbc> node signup.js
Account successfully created for adharID: 17bit0084!!
Account Signed In Successfully!
Account does not exist. Sign Up to Create One!
PS C:\Users\suchi\Downloads\buildbc>
```

Once a user creates an account on this website, a User-ID will be generated using an elliptic curve algorithm. This userID will later be used to sign posts that are created by the user which ensures non repudiation. The system makes sure that every user's userID is unique and only displays the public key to the users.



```
PS C:\Users\suchi\Downloads\buildbc> node signup.js
created a userID : 044c98659f860beadc2c397c2d3cea5a6545b06298541c3b31dc9bc408b8f9a915730601bb5536dff7f42a95c61998501e042b55b2d61b4bedeb74629d1bd3e58f for your acc
ount
Account successfully created for adharID: 17bit0084!!
```

When the user tries to share a post, all the new posts will be added to a pending post ledger and later added to our blockchain if they are valid.

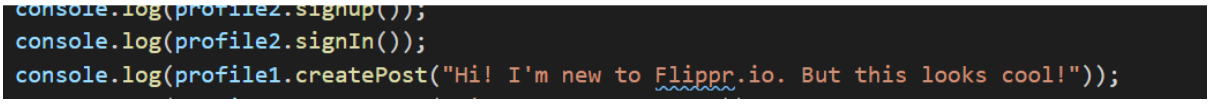
<u>FEATURE 2: VERIFICATION OF AUTHENTICITY OF THE POST.</u>

The authenticity of a post is verified using the following conditions.

- If the userID is null, then the post is invalid since it is assumed that a user is trying to share a fake post on the website.
- If userID is not null, a hash key for this post is generated using the user's ID, system time, his private key and the content he wants to post. Later, the signature for the post will be generated using this hash key. Once a signature is created for the post, it can never be modified. This lets us verify posts when a user mines a block to understand if the post's data was edited by some third party entity in the meanwhile.
- Later, when these posts are mined, model checks if the signature of the post and the generated signature matches and that signature is not null.

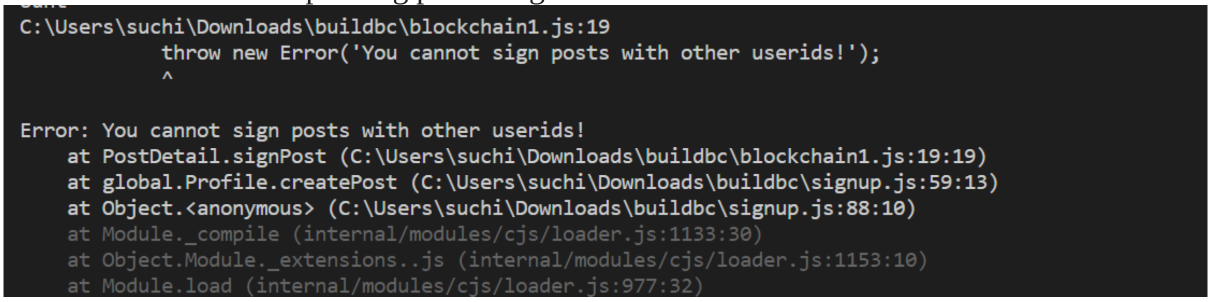
If all of these conditions satisfy, one can understand that the post is valid and add it to our pending posts ledger.

TEST CASE 1: When a user tried changing his signing user ID and tried to create a new post



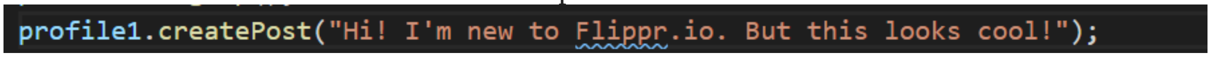
```
console.log(profile2.signup());
console.log(profile2.signIn());
console.log(profile1.createPost("Hi! I'm new to Flippr.io. But this looks cool!"));
```

OUTPUT: Since it's invalid the system throws an error and the post has been invalidated and was never added to the pending posts ledger.



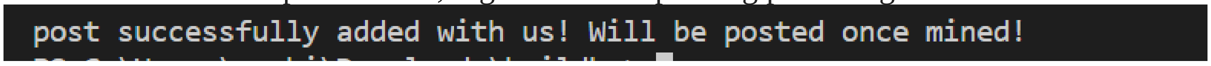
```
C:\Users\suchi\Downloads\buildbc\blockchain1.js:19
            throw new Error('You cannot sign posts with other userids!');
            ^

Error: You cannot sign posts with other userids!
    at PostDetail.signPost (C:\Users\suchi\Downloads\buildbc\blockchain1.js:19:19)
    at global.Profile.createPost (C:\Users\suchi\Downloads\buildbc\signup.js:59:13)
    at Object.<anonymous> (C:\Users\suchi\Downloads\buildbc\signup.js:88:10)
    at Module._compile (internal/modules/cjs/loader.js:1133:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1153:10)
    at Module.load (internal/modules/cjs/loader.js:977:32)
```

TEST CASE 2: When user creates a valid post



```
profile1.createPost("Hi! I'm new to Flippr.io. But this looks cool!");
```

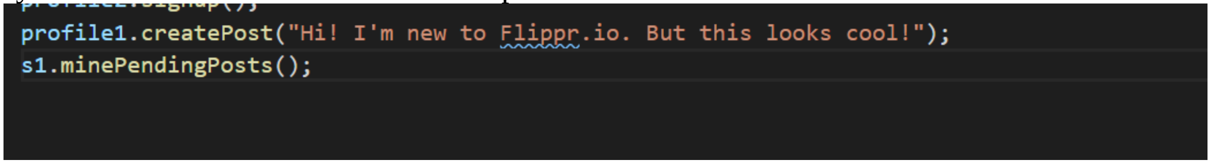
OUTPUT: Since the post is valid, it gets added to pending posts ledger.



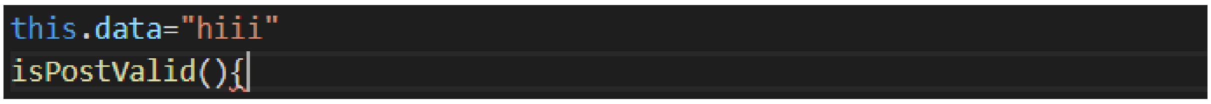
```
post successfully added with us! Will be posted once mined!
```

When a miner tries to mine the pending posts, he needs to submit his proof of work and then the pending posts block gets added to the blockchain as a new linked block and the miner gets rewarded. When the pending posts ledger needs to be added to the blockchain, model again checks before adding if all signatures have not been changed. Change in a signature means that the post has been corrupted and will be removed from the ledger and the user will get notified.

TEST CASE 3: When a user has created a post and in the meanwhile someone has hacked the system and edited his data he wants to post.

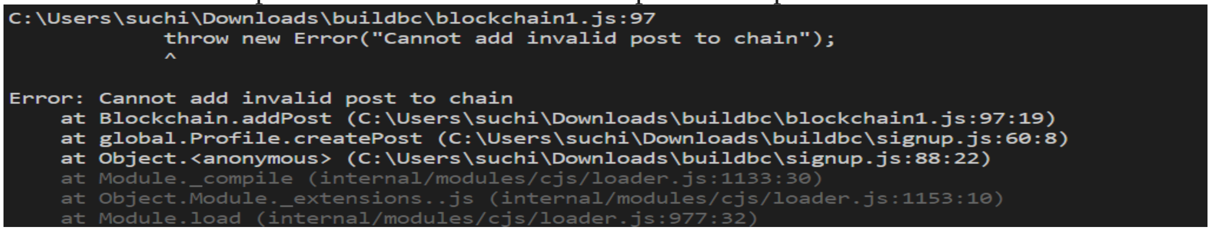


```
profile1.createPost("Hi! I'm new to Flippr.io. But this looks cool!");
s1.minePendingPosts();
```



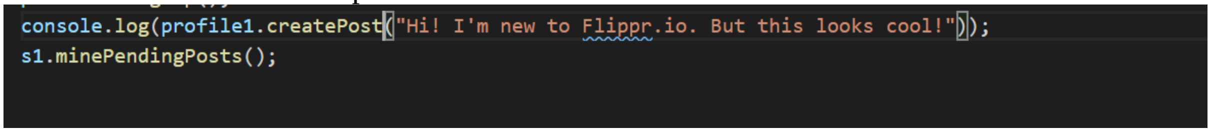
```
this.data="hiii"
isPostValid(){
```

OUTPUT: Invalid post error is thrown since the post is tampered.



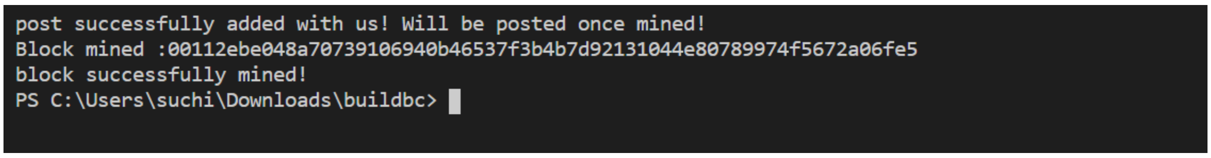
```
C:\Users\suchi\Downloads\buildbc\blockchain1.js:97
            throw new Error("Cannot add invalid post to chain");
            ^

Error: Cannot add invalid post to chain
    at Blockchain.addPost (C:\Users\suchi\Downloads\buildbc\blockchain1.js:97:19)
    at global.Profile.createPost (C:\Users\suchi\Downloads\buildbc\signup.js:60:8)
    at Object.<anonymous> (C:\Users\suchi\Downloads\buildbc\signup.js:88:22)
    at Module._compile (internal/modules/cjs/loader.js:1133:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1153:10)
    at Module.load (internal/modules/cjs/loader.js:977:32)
```

TEST CASE 4: When the post is valid.



```
console.log(profile1.createPost("Hi! I'm new to Flippr.io. But this looks cool!"));
s1.minePendingPosts();
```

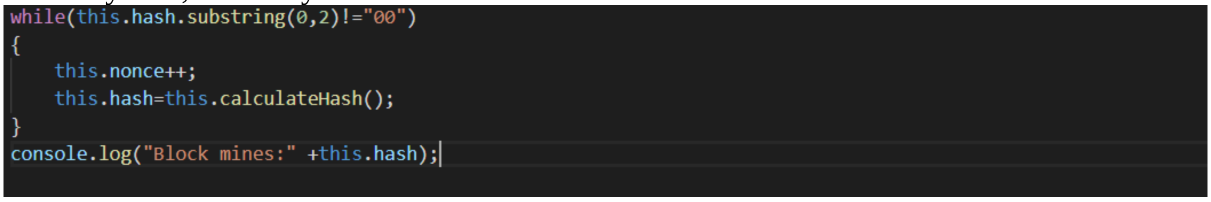
OUTPUT: The block is mined and its added to the chain.



```
post successfully added with us! Will be posted once mined!
Block mined :00112ebe048a70739106940b46537f3b4b7d92131044e80789974f5672a06fe5
block successfully mined!
PS C:\Users\suchi\Downloads\buildbc>
```

FEATURE 3: REMOVAL OF MIDDLE MAN ATTACKS INTO OUR SYSTEM

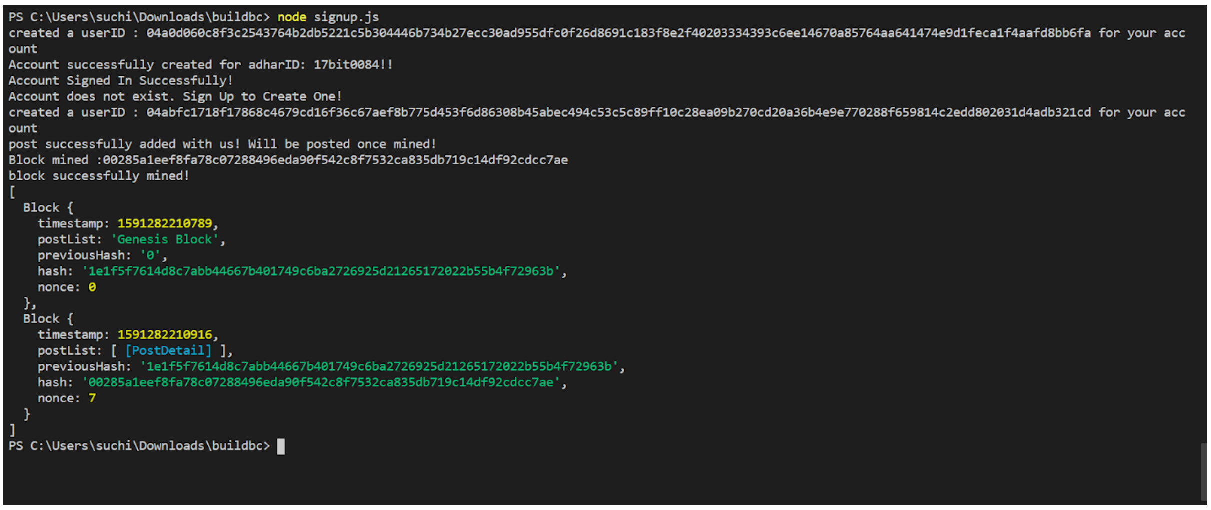
Miner has to mine a block and add pending blocks to the blockchain to show his proof of work. In order to maintain higher security as to let no one alter our chain, I have maintained previous hash and current hash in every block to verify that the chain hasn't been tampered.
Also, when a user tries to mine, model can customise and set difficulty level to ensure the blockchain is secure. This difficulty level will slow the process a tad more than normal processes but offers higher security. Someone who achieves this and mines successfully, gets rewarded to his account.
In this system, difficulty level has been set 2.



```
while(this.hash.substring(0,2)!="00")
{
    this.nonce++;
    this.hash=this.calculateHash();
}
console.log("Block mines:" +this.hash);
```

This "00" indicates that the hash key must always start with two zeroes.

FINAL BLOCKCHAIN STRUCTURE:

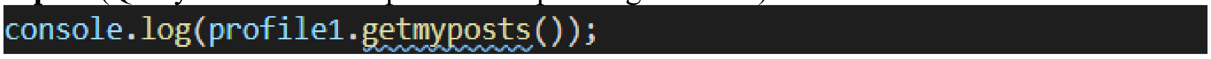


```
PS C:\Users\suchi\Downloads\buildbc> node signup.js
created a userID : 04a0d060c8f3c2543764b2db5221c5b304446b734b27ecc30ad955dfc0f26d8691c183f8e2f40203334393c6ee14670a85764aa641474e9d1feca1f4aafd8bb6fa for your acc
ount
Account successfully created for adharID: 17bit0084!!
Account Signed In Successfully!
Account does not exist. Sign Up to Create One!
created a userID : 04abfc1718f17868c4679cd16f36c67aef8b775d453f6d86308b45abec494c53c5c89ff10c28ea09b270cd20a36b4e9e770288f659814c2edd802031d4adb321cd for your acc
ount
post successfully added with us! Will be posted once mined!
Block mined :00285a1eef8fa78c07288496eda90f542c8f7532ca835db719c14df92cdcc7ae
block successfully mined!
[
  Block {
    timestamp: 1591282210789,
    postList: 'Genesis Block',
    previousHash: '0',
    hash: '1e1f5f7614d8c7abb44667b401749c6ba2726925d21265172022b55b4f72963b',
    nonce: 0
  },
  Block {
    timestamp: 1591282210916,
    postList: [ [PostDetail] ],
    previousHash: '1e1f5f7614d8c7abb44667b401749c6ba2726925d21265172022b55b4f72963b',
    hash: '00285a1eef8fa78c07288496eda90f542c8f7532ca835db719c14df92cdcc7ae',
    nonce: 7
  }
]
PS C:\Users\suchi\Downloads\buildbc>
```

ADDITIONAL FEATURES:

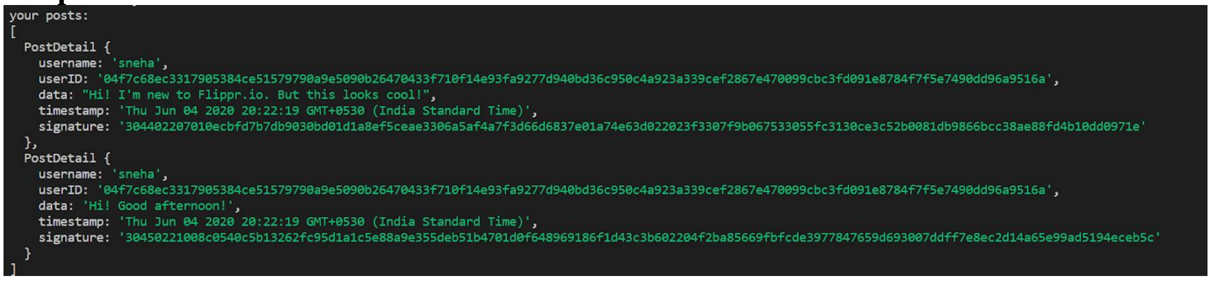
To view user's posts:

Input: (Query to view user's posts corresponding to a user)



```
console.log(profile1.getmyposts());
```

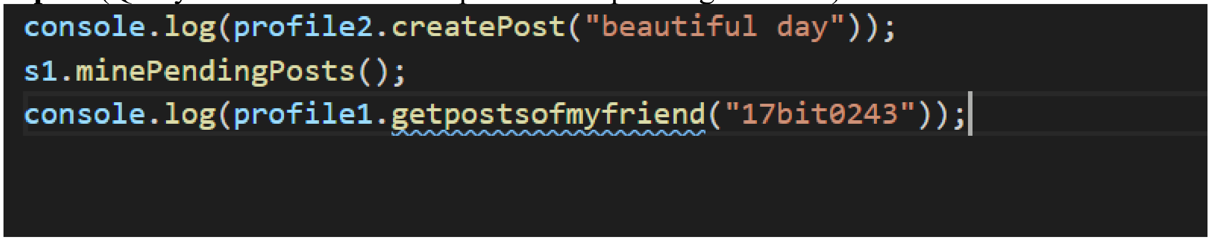
Output:



```
your posts:
[
  PostDetail {
    username: 'sneha',
    userID: '04f7c68ec3317905384ce51579790a9e5090b26470433f710f14e93fa9277d940bd36c950c4a923a339cef2867e470099cbc3fd091e8784f7f5e7490dd96a9516a',
    data: "Hi! I'm new to Flippr.io. But this looks cool!",
    timestamp: 'Thu Jun 04 2020 20:22:19 GMT+0530 (India Standard Time)',
    signature: '30440220701 0ecbfd7b7db9030bd01d1a8ef5ceae3306a5af4a7f3d66d6837e01a74e63d022023f3307f9b067533055fc3130ce3c52b0081db9866bcc38ae88fd4b10dd0971e'
  },
  PostDetail {
    username: 'sneha',
    userID: '04f7c68ec3317905384ce51579790a9e5090b26470433f710f14e93fa9277d940bd36c950c4a923a339cef2867e470099cbc3fd091e8784f7f5e7490dd96a9516a',
    data: 'Hi! Good afternoon!',
    timestamp: 'Thu Jun 04 2020 20:22:19 GMT+0530 (India Standard Time)',
    signature: '30450221008c0540c5b13262fc95d1a1c5e88a9e355deb51b4701d0f648969186f1d43c3b602204f2ba85669fbfcde3977847659d693007ddff7e8ec2d14a65e99ad5194eceb5c'
  }
]
```

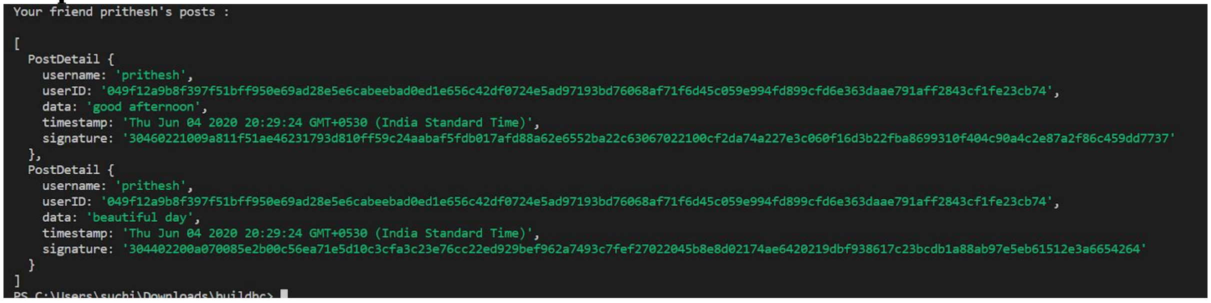
To View follower's posts:

Input: (Query to view other users' posts corresponding to a user)



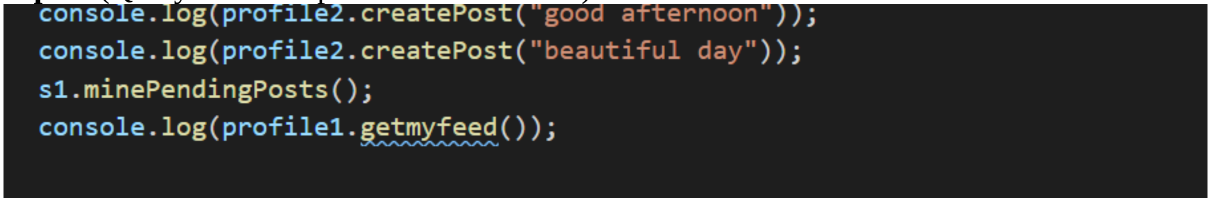
```
console.log(profile2.createPost("beautiful day"));
s1.minePendingPosts();
console.log(profile1.getpostsofmyfriend("17bit0243"));
```

Output:



```
Your friend prithesh's posts :

[
  PostDetail {
    username: 'prithesh',
    userID: '049f12a9b8f397f51bff950e69ad28e5e6cabeebad0ed1e656c42df0724e5ad97193bd76068af71f6d45c059e994fd899cfd6e363daae791aff2843cf1fe23cb74',
    data: 'good afternoon',
    timestamp: 'Thu Jun 04 2020 20:29:24 GMT+0530 (India Standard Time)',
    signature: '30460221009a811f51ae46231793d810ff59c24aabaf5fdb017afd88a62e6552ba22c63067022100cf2da74a227e3c060f16d3b22fba8699310f404c90a4c2e87a2f86c459dd7737'
  },
  PostDetail {
    username: 'prithesh',
    userID: '049f12a9b8f397f51bff950e69ad28e5e6cabeebad0ed1e656c42df0724e5ad97193bd76068af71f6d45c059e994fd899cfd6e363daae791aff2843cf1fe23cb74',
    data: 'beautiful day',
    timestamp: 'Thu Jun 04 2020 20:29:24 GMT+0530 (India Standard Time)',
    signature: '304402200a070085e2b00c56ea71e5d10c3cfa3c23e76cc22ed929bef962a7493c7fef27022045b8e8d02174ae6420219dbf938617c23bcdb1a88ab97e5eb61512e3a6654264'
  }
]
PS C:\Users\suchi\Downloads\buildbc>
```

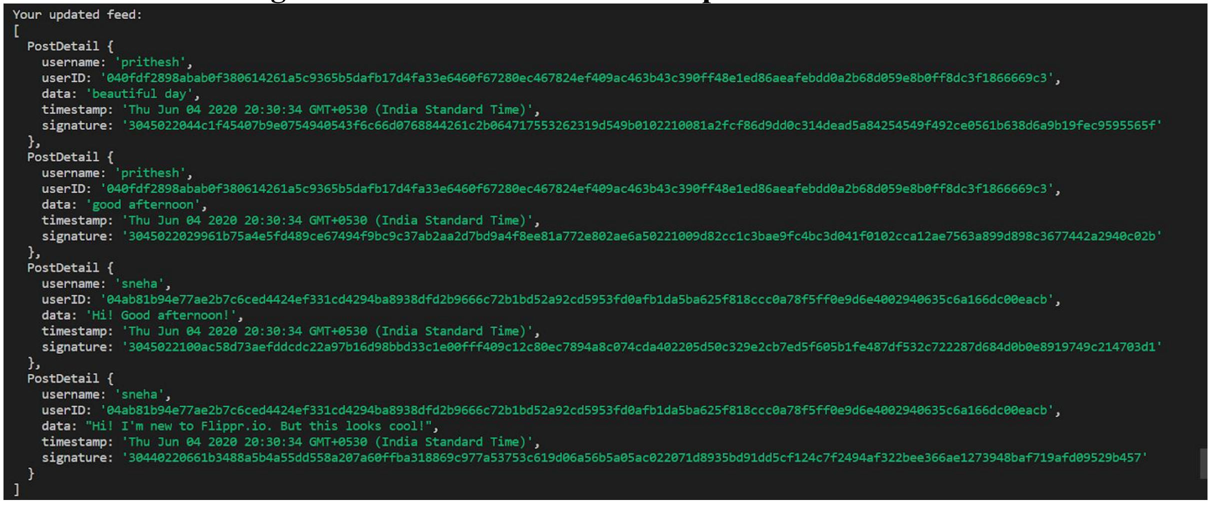
Input: (Query to view updated feed to a user)



```
console.log(profile2.createPost("good afternoon"));
console.log(profile2.createPost("beautiful day"));
s1.minePendingPosts();
console.log(profile1.getmyfeed());
```

OUTPUT:

A user's latest feed gets loaded based on the recent posts added to blockchain:



```
Your updated feed:
[
  PostDetail {
    username: 'prithesh',
    userID: '040fdf2898abab0f380614261a5c9365b5dafb17d4fa33e6460f67280ec467824ef409ac463b43c390ff48e1ed86aeafebdd0a2b68d059e8b0ff8dc3f1866669c3',
    data: 'beautiful day',
    timestamp: 'Thu Jun 04 2020 20:30:34 GMT+0530 (India Standard Time)',
    signature: '3045022044c1f45407b9e0754940543f6c66d0768844261c2b064717553262319d549b0102210081a2fcf86d9dd0c314dead5a84254549f492ce0561b638d6a9b19fec9595565f'
  },
  PostDetail {
    username: 'prithesh',
    userID: '040fdf2898abab0f380614261a5c9365b5dafb17d4fa33e6460f67280ec467824ef409ac463b43c390ff48e1ed86aeafebdd0a2b68d059e8b0ff8dc3f1866669c3',
    data: 'good afternoon',
    timestamp: 'Thu Jun 04 2020 20:30:34 GMT+0530 (India Standard Time)',
    signature: '3045022029961b75a4e5fd489ce67494f9bc9c37ab2aa2d7bd9a4f8ee81a772e802ae6a50221009d82cc1c3bae9fc4bc3d041f0102cca12ae7563a899d898c3677442a2940c02b'
  },
  PostDetail {
    username: 'sneha',
    userID: '04ab81b94e77ae2b7c6ced4424ef331cd4294ba8938dfd2b9666c72b1bd52a92cd5953fd0afb1da5ba625f818ccc0a78f5ff0e9d6e4002940635c6a166dc00eacb',
    data: 'Hi! Good afternoon!',
    timestamp: 'Thu Jun 04 2020 20:30:34 GMT+0530 (India Standard Time)',
    signature: '3045022100ac58d73aefddcdc22a97b16d98bbd33c1e00fff409c12c80ec7894a8c074cda402205d50c329e2cb7ed5f605b1fe487df532c722287d684d0b0e8919749c214703d1'
  },
  PostDetail {
    username: 'sneha',
    userID: '04ab81b94e77ae2b7c6ced4424ef331cd4294ba8938dfd2b9666c72b1bd52a92cd5953fd0afb1da5ba625f818ccc0a78f5ff0e9d6e4002940635c6a166dc00eacb',
    data: "Hi! I'm new to Flippr.io. But this looks cool!",
    timestamp: 'Thu Jun 04 2020 20:30:34 GMT+0530 (India Standard Time)',
    signature: '30440220661b3488a5b4a55dd558a207a60ffba318869c977a53753c619d06a56b5a05ac022071d8935bd91dd5cf124c7f2494af322bee366ae1273948baf719afd09529b457'
  }
]
```

KEY FEATURES

1. DATA INTEGRITY : The structure of the blocks in the blockchain ensures integrity of posts shared along with its metadata (eg: TIME,USER ID etc)
2. NON REPUDIATION : Every valid post is signed with the user's ID which is then stored in the block chain permanently.
3. REWARD SYSTEM: Users get rewarded which maximises participation.
4. HIGH DATA SECURITY: The blocks in a blockchain containing the post details can't be tampered with.

5. ANTI SPAM PREVENTION: By setting the level of difficulty the rate of mining of new blocks can be controlled.
6. DATA RECOVERY: Since the data is decentralised it’s nearly impossible to delete the contents unless one has access to 50% of the network which is not practically possible.
7. TRANSPARENCY: Since the regulations of the application and the reward system is open to all and the chain is not dependent on a single entity, which keeps the system more reliable and accountable.
8. ENCRYPTION: This ensures the confidentiality of the data when a unauthorised entity tries to gain access and read the contents of the data stored;
9. AUTHENTICATION: The combination of USER ID that is generated with a unique Aadhar ID along with the password guarantees authenticity.