

# Configuration Manual

MSc Research Project  
Practicum

Vishnuprasanth Sivakumar  
Student ID: X23293161

School of Computing  
National College of Ireland

Supervisor: Michael Prior

**National College of Ireland**  
**MSc Project Submission Sheet**



**School of Computing**

**Student Name:** Vishnuprasanth Sivakumar  
**Student ID:** X23293161  
**Programme:** Cybersecurity **Year:** 2024-2025  
**Module:** Practicum  
**Supervisor:** Michael Prior  
**Submission Due Date:** 11/08/2025  
**Project Title:** The Role of blockchain in enhancing data security and privacy

**Word Count:** 650 **Page Count:** 12

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Vishnuprasanth Sivakumar

**Date:** 11/08/2025

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Vishnuprasanth Sivakumar  
X23293161

## 1 Business Problem

Organizations face rising threats from data breaches, identity theft, and unauthorized access. Single sources of failure are a common problem with traditional centralized systems.

This project explores how a Python-based blockchain system can offer enhanced data security and privacy using distributed ledgers, cryptographic techniques, and transparent recordkeeping mechanisms.

## 2 System Requirements

Operating System:

- Windows 10 or Windows 11 (64-bit)

Python Version:

- Python 3.10 or higher



Development Environment:

- Visual Studio Code (VS Code)



Hardware:

- RAM: Minimum 8000 MB (Recommended: 16000 MB)
- Disk Space: Minimum 5000 MB free

Libraries/Dependencies:

- hashlib, datetime, json, pandas, numpy, matplotlib, seaborn, scikit-learn, jupyterlab, dotenv

### **3 Installation Instructions**

Step 1- Upgrade pip:

```
python -m pip install --upgrade pip
```

```

PS C:\Users\Vishnuprasanth S\Desktop\Blockchain code> python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\program files\windowsapps\pythonsoftwarefoundation.python.3.11_3.11.2544.0_x64__qbz5n2kfra8p0\lib\site-packages (24.0)
Collecting pip
  Downloading pip-25.2-py3-none-any.whl.metadata (4.7 kB)
  Downloading pip-25.2-py3-none-any.whl (1.8 MB)
    1.8/1.8 MB 7.0 MB/s eta 0:00:00
Installing collected packages: pip
  WARNING: The scripts pip.exe, pip3.11.exe and pip3.exe are installed in 'C:\Users\Vishnuprasanth S\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-25.2

```

Step 2- Create and activate the virtual environment:

```

python -m venv blockchain_env
blockchain_env\Scripts\activate

```

```

PS C:\Users\Vishnuprasanth S\Desktop\Blockchain code> python -m venv blockchain_env
PS C:\Users\Vishnuprasanth S\Desktop\Blockchain code> .\blockchain_env\Scripts\Activate
(blockchain_env) PS C:\Users\Vishnuprasanth S\Desktop\Blockchain code>

```

Step 3 - Install Required Libraries:

```

pip install pandas numpy matplotlib cryptography

```

```

(blockchain_env) PS C:\Users\Vishnuprasanth S\Desktop\Blockchain code> pip install pandas matplotlib cryptography
Collecting pandas
  Using cached pandas-2.3.1-cp311-cp311-win_amd64.whl.metadata (19 kB)
Collecting matplotlib
  Downloading matplotlib-3.10.5-cp311-cp311-win_amd64.whl.metadata (11 kB)
Collecting cryptography
  Downloading cryptography-45.0.6-cp311-abi3-win_amd64.whl.metadata (5.7 kB)
Collecting numpy>=1.23.2 (from pandas)

```

```

Installing collected packages: pytz, tzdata, six, pyparsing, pycparser, pillow, packaging, numpy, kiwisolver, fonttools, cyler, python-dateutil, contourpy, cffi, pandas, matplotlib, cryptography
Successfully installed cffi-1.17.1 contourpy-1.3.3 cryptography-45.0.6 cyler-0.12.1 fonttools-4.59.0 kiwisolver-1.4.8 matplotlib-3.10.5 numpy-2.3.2 packaging-25.0 pandas-2.3.1 pillow-11.3.0 pycparser-2.22 pyparsing-3.2.3 python-dateutil-2.9.0.post0 pytz-2025.2 six-1.17.0 tzdata-2025.2
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip

```

## 4 Dataset & Blockchain Configuration

The system uses simulated blockchain logic built in Python to mimic block creation, hashing, validation, and anomaly detection. Sample datasets are tailored to key domains:

### Healthcare Sample Data:

Patient_ID	Name	Age	Blood	Med Condition	Doctor
P1001	Bobby Jackson	30	B-	Cancer	Matthew Smith
P1002	Leslie Terry	62	A+	Obesity	Samatha Davies

```

data_processor.py | healthcare_dataset.csv X
healthcare_dataset.csv > data
1 patient_id,Name,Age,Gender,Blood Type,Medical Condition,Date of Admission,Doctor,Hospital,
2 1,Bobby Jackson,30,Male,B-,Cancer,1/31/2024,Matthew Smith,Sons and Miller,Blue Cross,18856
3 2,Leslie Terry,62,Male,A+,Obesity,8/20/2019,Samantha Davies,Kim Inc,Medicare,33643.32729,20
4 3,Danny Smith,76,Female,A-,Obesity,9/22/2022,Tiffany Mitchell,Cook PLC,Aetna,27955.09608,20
5 4,Andrew Watts,28,Female,O+,Diabetes,11/18/2020,Kevin Wells,"Hernandez Rogers and Vang",M
6 5,Adrienne Bell,43,Female,AB+,Cancer,9/19/2022,Kathleen Hanna,White-White,Aetna,14238.3178
7 6,Emily Johnson,36,Male,A+,Asthma,12/20/2023,Taylor Newton,Nunez-Humphrey,UnitedHealthcare
8 7,Edward Edwards,21,Female,AB-,Diabetes,11/3/2020,Kelly Olson,Group Middleton,Medicare,195

```

**Finance Sample Data:**

Symbol	Name	Amount
BRK-B	Berkshire Hathway	321.26
V	Visa Inc	230.76

```

yahoo Finance_industry - Stock Market.csv > data
1 Symbol,Name,Price,Change,market cap,PE_ratio
2 BRK-B,Berkshire Hathway Inc.,321.26,-1.38,709.361B,N/A
3 V,Visa Inc.,230.76,-0.25,483.304B,30.81
4 JPM,JPMorgan Chase & Co.,133.89,-2.16,392.325B,9.90
5 MA,Mastercard Incorporated,380.21,-3.18,359.338B,37.85
6 JPM-PD,JPMorgan Chase & Co.,24.57,-0.08,326.427B,1.82
7 JPM-PC,JPMorgan Chase & Co.,25.02,-0.09,328.207B,1.86
8 BAC-PK,Bank of America Corporation,24.38,-0.04,244.596B,6.94
9 BAC-PL,Bank of America Corporation,"1,144.76",+0.79,0,325.77
10 BML-PG,Bank of America Corporation,20.02,-0.03,270.35B,5.70
11 BML-PH,Bank of America Corporation,19.90,+0.05,269.812B,5.66
12 BAC,Bank of America Corporation,27.06,-0.33,215.526B,8.10
13 BML-PL,Bank of America Corporation,20.20,+0.05,248.946B,5.75
14 BAC-PE,Bank of America Corporation,20.15,+0.12,239.34B,5.73
15 BML-PJ,Bank of America Corporation,20.76,-0.01,240.083B,5.91
16 BAC-PB,Bank of America Corporation,24.53,-0.04,245.183B,6.98
17 WFC,Bank of America Corporation,25.01,+1.45,201.020B,5.18

```

**Supply Chain Sample Data:**

Month	Sales	Supplier Name
1	3675.25	Walmart
2	4588.39	Walmart

```
simple_vegetable_retail-chains-sales_data.csv > data
1  Month,Sales(kg),Supplier_Name
2  1,3675.25,Walmart
3  2,4588.39,Walmart
4  3,5079.92,Walmart
5  4,5541.03,Walmart
6  5,6134.35,Walmart
7  6,6564.08,Walmart
8  7,7156.3,Walmart
9  8,7073.38,Walmart
10 9,7322.49,Walmart
```

## 5 Evaluation Settings

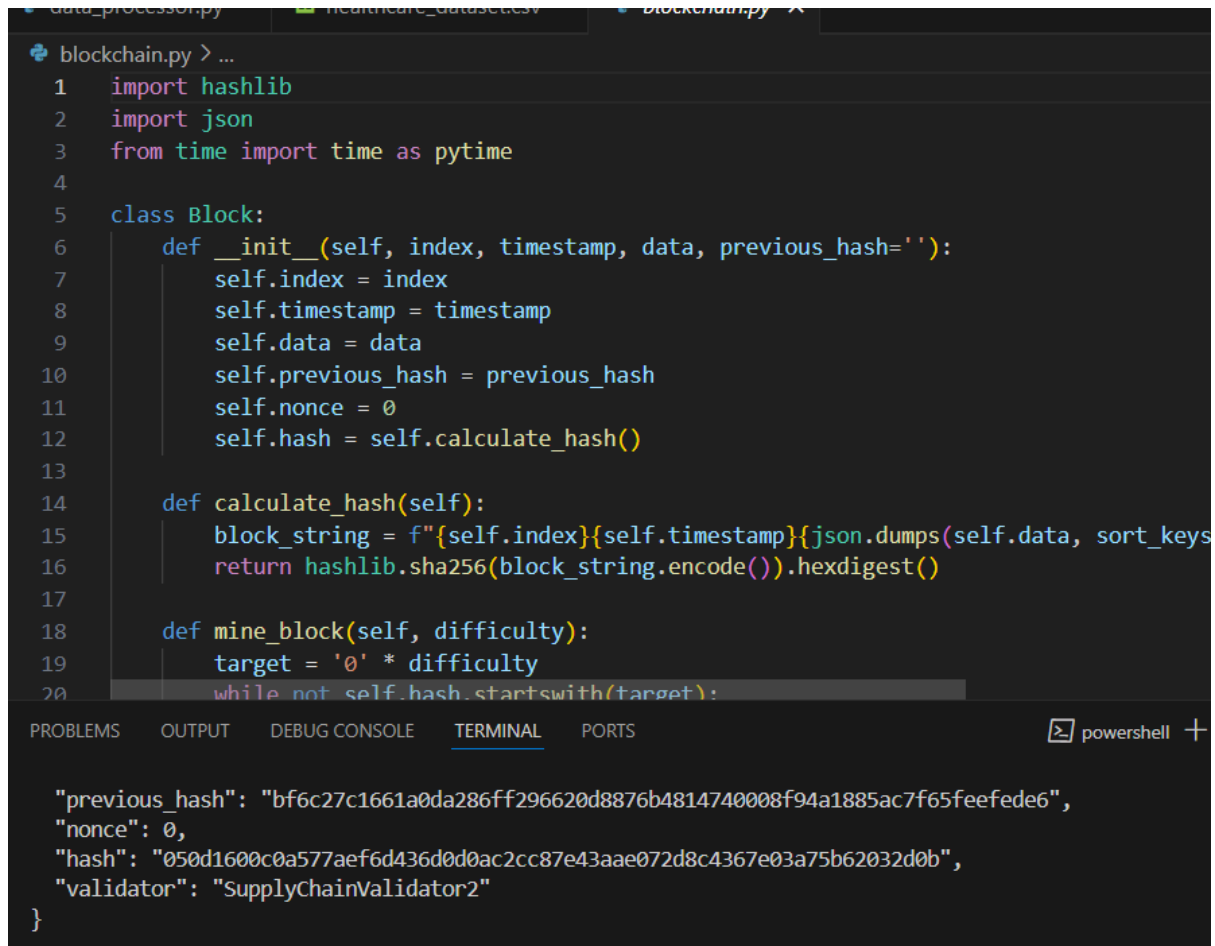
Blockchain Performance Evaluation:

- TPS (Transactions per Second)
- Anomaly Detection
- Block Validation Time

```
=====
✅ Evaluation Complete
This implementation demonstrates:
1. Hybrid blockchain architecture (permissioned chains)
2. Sector-specific compliance and validation
3. Anomaly detection and security measures
4. Performance metrics and evaluation
```

## 6 Sample Python Logic

Block Structure:



```
blockchain.py > ...
1  import hashlib
2  import json
3  from time import time as pytime
4
5  class Block:
6      def __init__(self, index, timestamp, data, previous_hash=''):
7          self.index = index
8          self.timestamp = timestamp
9          self.data = data
10         self.previous_hash = previous_hash
11         self.nonce = 0
12         self.hash = self.calculate_hash()
13
14         def calculate_hash(self):
15             block_string = f"{self.index}{self.timestamp}{json.dumps(self.data, sort_keys=True)}"
16             return hashlib.sha256(block_string.encode()).hexdigest()
17
18         def mine_block(self, difficulty):
19             target = '0' * difficulty
20             while not self.hash.startswith(target):
                self.nonce += 1
                self.hash = self.calculate_hash()
                # ... (rest of the loop)
        }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell +

{"previous_hash": "bf6c27c1661a0da286ff296620d8876b4814740008f94a1885ac7f65feefede6",
 "nonce": 0,
 "hash": "050d1600c0a577aef6d436d0d0ac2cc87e43aae072d8c4367e03a75b62032d0b",
 "validator": "SupplyChainValidator2"
}
```

Blockchain Initialization:

class Blockchain:

class Hybrid Blockchain:

def \_\_init\_\_(self, difficulty=3):

self.permissioned\_chain = [self.create\_genesis\_block\_permissioned()]

self.permissionless\_chain = [self.create\_genesis\_block\_permissionless()]

self.difficulty = difficulty

self.authorized\_validators = ["AuthorityNode1", "AuthorityNode2"]

```
def create_genesis_block_permissioned(self):
    return PermissionedBlock(0, pytime(), "Permissioned Genesis", "0",
validator="AuthorityNode1")
```

```
def create_genesis_block_permissionless(self):
    return (0, pytime(), "Permissionless Genesis", "0")
```

```
def get_latest_permissioned(self):
    return self.permissioned_chain[-1]
```

```
def get_latest_permissionless(self):
    return self.permissionless_chain[-1]
```

```
class HybridBlockchain:
    def __init__(self, difficulty=3):
        self.permissioned_chain = [self.create_genesis_block_permissioned()]
        self.permissionless_chain = [self.create_genesis_block_permissionless()]
        self.difficulty = difficulty
        self.authorized_validators = ["AuthorityNode1", "AuthorityNode2"]

    def create_genesis_block_permissioned(self):
        return PermissionedBlock(0, pytime(), "Permissioned Genesis", "0", validator="Author

    def create_genesis_block_permissionless(self):
        return Block(0, pytime(), "Permissionless Genesis", "0")

    def get_latest_permissioned(self):
        return self.permissioned_chain[-1]

    def get_latest_permissionless(self):
        return self.permissionless_chain[-1]

    def add_permissioned_block(self, block):
        block.previous_hash = self.get_latest_permissioned().hash
        if block.validate_authority(self.authorized_validators):
            block.hash = block.calculate_hash()
            self.permissioned_chain.append(block)
            print(f"Permissioned block {block.index} added by {block.validator}")
```

TPS Measurement:

```
82     start_time = time.time()
83     processed_count = 0
84     anomaly_count = 0
85     compliance_violations = 0
86
87     # Process each record
88     for i, (_, row) in enumerate(df.head(10).iterrows()): # Process first 10 records for te
89         try:
90             # Process record
91             processed_record = data_processor.process_record(row.to_dict(), sector)
92             if not processed_record:
93                 print(f"Error processing record {i}")
94                 continue
95
96             # Check for anomalies
97             if anomaly_detection(processed_record):
98                 anomaly_count += 1
99                 print(f"Anomaly detected in record {i}")
100
101             # Add to blockchain with appropriate validator
102             if not blockchain.add_industry_block(processed_record):
103                 print(f"Compliance violation in record {i}")
104                 compliance_violations += 1
105                 continue
106
107             processed_count += 1
108
109         except Exception as e:
110             print(f"Error processing record {i}: {str(e)}")
111             compliance_violations += 1
112
113     # Calculate processing speed
114     end_time = time.time()
115     processing_time = end_time - start_time
```

## 7 Troubleshooting

Problem: Slow TPS

Resolution: Optimize data structure or reduce logging overhead

```

print(f"Loaded {sector} dataset with {len(df)} records")

# Initialize metrics
start_time = time.time()
processed_count = 0
anomaly_count = 0
compliance_violations = 0

# Process each record
for i, (_, row) in enumerate(df.head(10).iterrows()): # Process first 10 records for t
    try:

```

Problem: Block not validating

Resolution: Check hash function and timestamp consistency

```

65
66     def is_chain_valid(self, chain):
67         for i in range(1, len(chain)):
68             current = chain[i]
69             previous = chain[i - 1]
70             if current.hash != current.calculate_hash():
71                 print(f"Block {i} hash invalid!")
72                 return False
73             if current.previous_hash != previous.hash:
74                 print(f"Block {i} previous hash invalid!")
75                 return False
76         return True

```

Problem: Duplicate hashes

Resolution: Ensure unique timestamp or include nonce/random seed

```

self.hash = self.calculate_hash()

def calculate_hash(self):
    block_string = f"{self.index}{self.timestamp}{json.dumps(self.data, sort_keys=True)}"
    return hashlib.sha256(block_string.encode()).hexdigest()

```

Problem: Outliers in anomaly scores

Resolution: Recalibrate thresholds or retrain detection model

```
'healthcare': {
  'suspicious_keywords': ['hack', 'unauthor
  'thresholds': {
    'billing_amount': 10000,      # Thresh
    'age': 120,                  # Maximu
    'admission_frequency': 0.3,  # Thres
    'test_results': ['Critical', 'Emergen
  }
},
'finance': {
  'suspicious_keywords': ['fraud', 'transfe
  'thresholds': {
    'price_change': 0.1,        # Thresh
    'pe_ratio': 50,            # Thresh
    'price': 1000,             # Maximu
    'market_cap': ['B', 'M']   # Expect
```