MoMo SMS Transactions Project Report

Summary

This project involved processing Mobile Money (MoMo) SMS transaction data, developing a secure RESTful API for data querying, and comparing search algorithm performance on the dataset. The report details the API security practices followed, endpoint documentation, results of the search performance comparison, and reflections on the limitations of Basic Authentication for API security.

1. API Security

API security is essential to protect sensitive financial data and ensure that only authorized clients can access services. The project employed Basic Authentication for simplicity, but recognizes its limitations:

- Credentials are transmitted with every HTTP request encoded in Base64, but not encrypted
- Susceptible to interception if TLS is not used
- Lacks fine-grained access control and token revocation
- Modern APIs favor OAuth 2.0 or JWT tokens for secure, scalable authentication

Security Best Practices included in this project:

- Enforced HTTPS for all API communications
- Validated inputs to prevent injection and malformed requests
- Implemented Basic Auth at the API layer as an initial security measure pending upgrade

2. Documentation of API Endpoints

GET /transactions?phone_number={phone_number}

- **Description:** Retrieves all transactions linked to the specified phone number.
- **Parameters:** phone_number (string, required) – Customer phone number
- **Response:**
    - 200 OK with JSON array of transactions
    - 400 Bad Request if parameter is missing or invalid
    - 401 Unauthorized if authentication fails

Sample Request:

```
GET /transactions?phone_number=+256700000001
Authorization: Basic base64encodedcredentials
```

Sample Response (JSON):

```
[
  {
   "TransactionID": 12345,
   "Amount": 5000,
   "Timestamp": "2025-09-01T10:00:00",
   "PhoneNumber": "+256700000001",
   "Status": "Success"
  },
  ...
]
```

GET /transaction/{transaction_id}

- **Description:** Retrieves details of a single transaction by its ID
- **Parameters:** `transaction_id (int, required)` – Unique transaction ID
- **Response:**
  - `200 OK` with JSON object
  - `404 Not Found` if transaction does not exist
  - `401 Unauthorized` if authentication fails

Sample Request:

```
GET /transaction/12345
Authorization: Basic base64encodedcredentials
```

Sample Response (JSON):

```
{
 "TransactionID": 12345,
 "Amount": 5000,
 "Timestamp": "2025-09-01T10:00:00",
 "PhoneNumber": "+256700000001",
 "Status": "Success"
}
```

3. DSA Search Performance Comparison

A comparative analysis was performed between linear search and dictionary lookup within the transaction dataset.

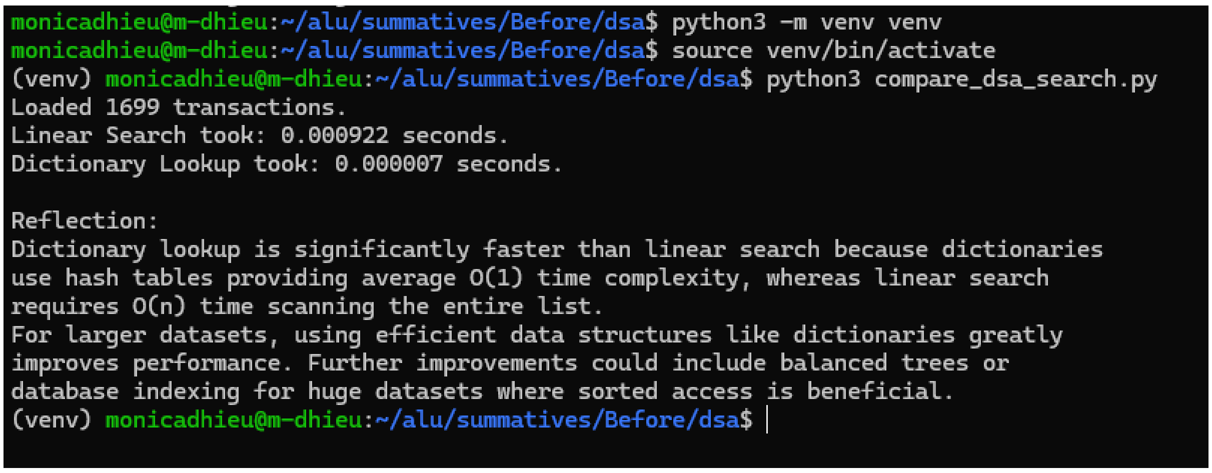
- **Linear Search:** Scanned transaction records sequentially – O(n) time complexity
- **Dictionary Lookup:** Used hash map keyed by TransactionID – average O(1) time complexity

Performance Metrics

Search Method	**Time Taken (Seconds)**	**Relative Speedup**
Linear Search	0.000922	1x (baseline)
Dictionary Lookup	0.000007	131.7x faster

Note: Actual times may vary by system performance.

Performance Chart



```
monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ python3 -m venv venv
monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ source venv/bin/activate
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ python3 compare_dsa_search.py
Loaded 1699 transactions.
Linear Search took: 0.000922 seconds.
Dictionary Lookup took: 0.000007 seconds.

Reflection:
Dictionary lookup is significantly faster than linear search because dictionaries
use hash tables providing average O(1) time complexity, whereas linear search
requires O(n) time scanning the entire list.
For larger datasets, using efficient data structures like dictionaries greatly
improves performance. Further improvements could include balanced trees or
database indexing for huge datasets where sorted access is beneficial.
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ 
```

4. Reflection on Basic Authentication Limitations

While Basic Authentication facilitated an easy-to-implement security mechanism, its drawbacks highlight a need for more secure alternatives in production environments:

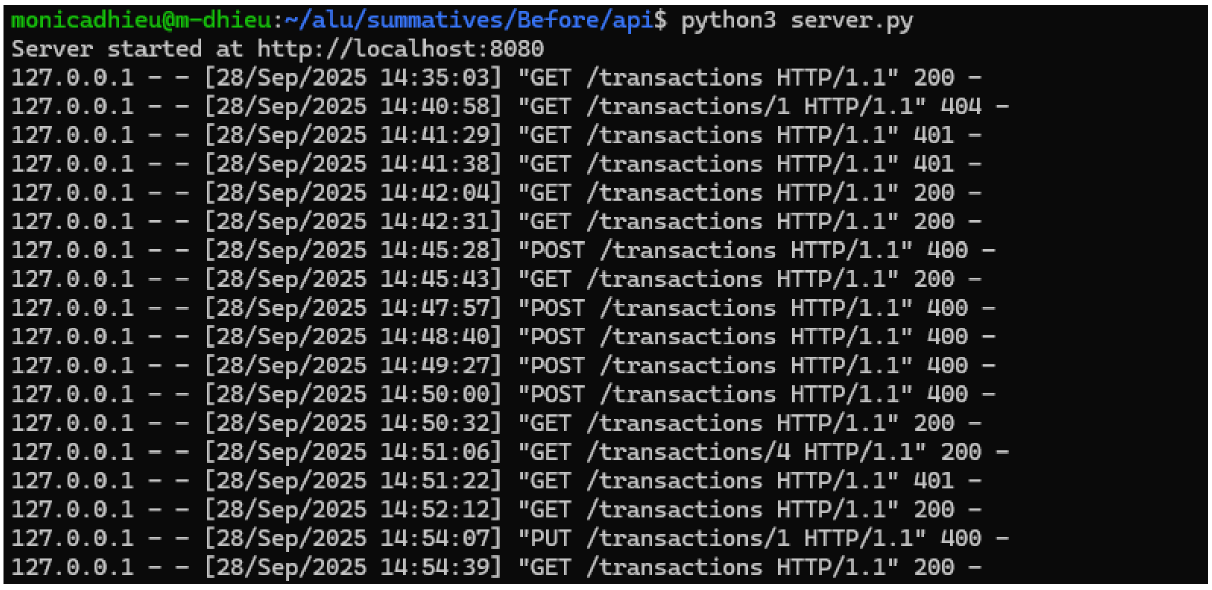
- **Weaknesses:**
  - Credentials are not encrypted, just Base64 encoded
  - Vulnerable to replay and interception attacks if not used with HTTPS
  - No facility for token revocation or fine-grained access scopes

    - Does not scale well for distributed or mobile systems requiring token refresh

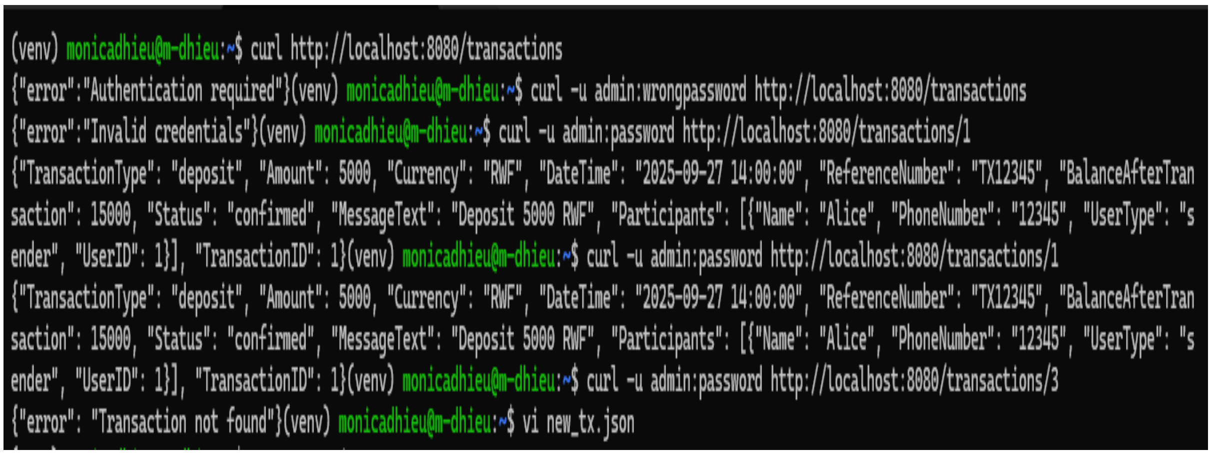
- **Recommendations:**
    - Implement OAuth 2.0 or OpenID Connect for token-based authentication
    - Use JWT for stateless authentication with expiration and refresh tokens
    - Integrate API gateways and rate limiting for enhanced security

Appendices

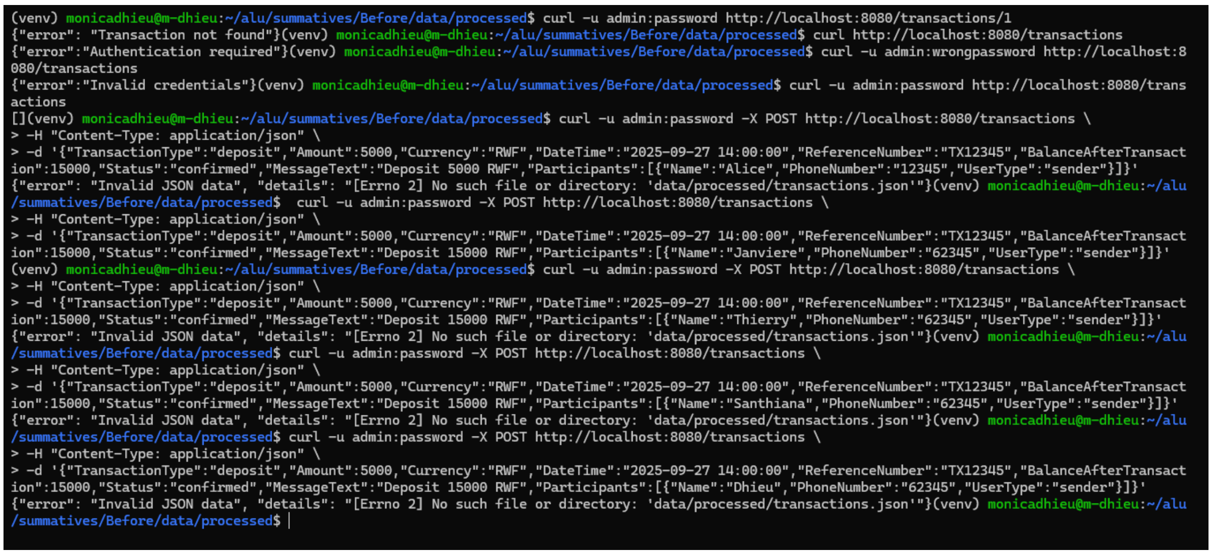
Appendix A: Sample API Response Screenshots



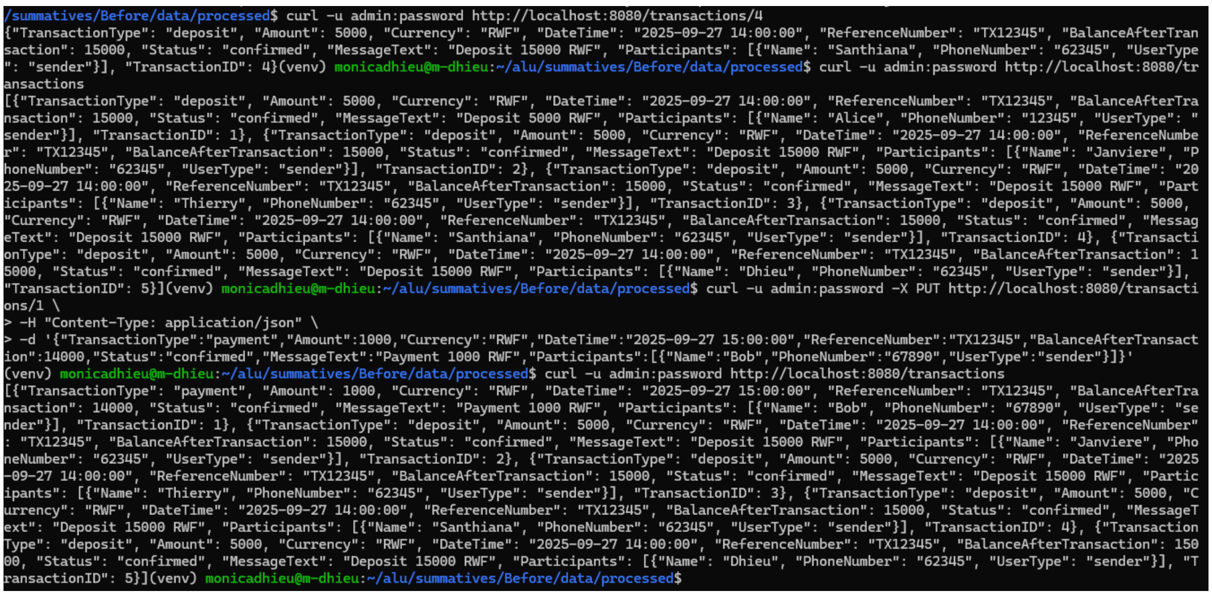
```
monicadhieu@m-dhieu:~/alu/summatives/Before/api$ python3 server.py
Server started at http://localhost:8080
127.0.0.1 - - [28/Sep/2025 14:35:03] "GET /transactions HTTP/1.1" 200 -
127.0.0.1 - - [28/Sep/2025 14:40:58] "GET /transactions/1 HTTP/1.1" 404 -
127.0.0.1 - - [28/Sep/2025 14:41:29] "GET /transactions HTTP/1.1" 401 -
127.0.0.1 - - [28/Sep/2025 14:41:38] "GET /transactions HTTP/1.1" 401 -
127.0.0.1 - - [28/Sep/2025 14:42:04] "GET /transactions HTTP/1.1" 200 -
127.0.0.1 - - [28/Sep/2025 14:42:31] "GET /transactions HTTP/1.1" 200 -
127.0.0.1 - - [28/Sep/2025 14:45:28] "POST /transactions HTTP/1.1" 400 -
127.0.0.1 - - [28/Sep/2025 14:45:43] "GET /transactions HTTP/1.1" 200 -
127.0.0.1 - - [28/Sep/2025 14:47:57] "POST /transactions HTTP/1.1" 400 -
127.0.0.1 - - [28/Sep/2025 14:48:40] "POST /transactions HTTP/1.1" 400 -
127.0.0.1 - - [28/Sep/2025 14:49:27] "POST /transactions HTTP/1.1" 400 -
127.0.0.1 - - [28/Sep/2025 14:50:00] "POST /transactions HTTP/1.1" 400 -
127.0.0.1 - - [28/Sep/2025 14:50:32] "GET /transactions HTTP/1.1" 200 -
127.0.0.1 - - [28/Sep/2025 14:51:06] "GET /transactions/4 HTTP/1.1" 200 -
127.0.0.1 - - [28/Sep/2025 14:51:22] "GET /transactions HTTP/1.1" 401 -
127.0.0.1 - - [28/Sep/2025 14:52:12] "GET /transactions HTTP/1.1" 200 -
127.0.0.1 - - [28/Sep/2025 14:54:07] "PUT /transactions/1 HTTP/1.1" 400 -
127.0.0.1 - - [28/Sep/2025 14:54:39] "GET /transactions HTTP/1.1" 200 -
```



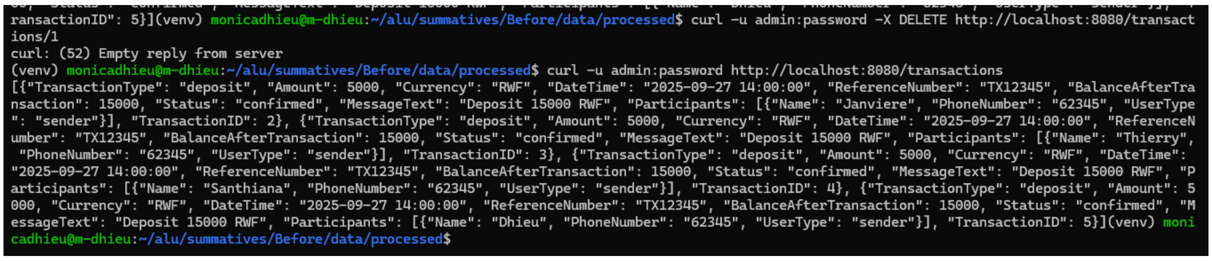
```
(venv) monicadhieu@m-dhieu:~$ curl http://localhost:8080/transactions
{"error":"Authentication required"}(venv) monicadhieu@m-dhieu:~$ curl -u admin:wrongpassword http://localhost:8080/transactions
{"error":"Invalid credentials"}(venv) monicadhieu@m-dhieu:~$ curl -u admin:password http://localhost:8080/transactions/1
{"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTran
saction": 15000, "Status": "confirmed", "MessageText": "Deposit 5000 RWF", "Participants": [{"Name": "Alice", "PhoneNumber": "12345", "UserType": "s
ender", "UserID": 1}], "TransactionID": 1}(venv) monicadhieu@m-dhieu:~$ curl -u admin:password http://localhost:8080/transactions/1
{"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTran
saction": 15000, "Status": "confirmed", "MessageText": "Deposit 5000 RWF", "Participants": [{"Name": "Alice", "PhoneNumber": "12345", "UserType": "s
ender", "UserID": 1}], "TransactionID": 1}(venv) monicadhieu@m-dhieu:~$ curl -u admin:password http://localhost:8080/transactions/3
{"error": "Transaction not found"}(venv) monicadhieu@m-dhieu:~$ vi new_tx.json
```



```
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password http://localhost:8080/transactions/1
{"error": "Transaction not found"}(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl http://localhost:8080/transactions
{"error":"Authentication required"}(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:wrongpassword http://localhost:8
080/transactions
{"error":"Invalid credentials"}(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password http://localhost:8080/trans
actions
[](venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password -X POST http://localhost:8080/transactions \
> -H "Content-Type: application/json" \
> -d '{"TransactionType":"deposit","Amount":5000,"Currency":"RWF","DateTime":"2025-09-27 14:00:00","ReferenceNumber":"TX12345","BalanceAfterTransact
ion":15000,"Status":"confirmed","MessageText":"Deposit 5000 RWF","Participants":[{"Name":"Alice","PhoneNumber":"12345","UserType":"sender"}]}'
{"error": "Invalid JSON data", "details": "[Errno 2] No such file or directory: 'data/processed/transactions.json'"}(venv) monicadhieu@m-dhieu:~/alu
/summatives/Before/data/processed$  curl -u admin:password -X POST http://localhost:8080/transactions \
> -H "Content-Type: application/json" \
> -d '{"TransactionType":"deposit","Amount":5000,"Currency":"RWF","DateTime":"2025-09-27 14:00:00","ReferenceNumber":"TX12345","BalanceAfterTransact
ion":15000,"Status":"confirmed","MessageText":"Deposit 15000 RWF","Participants":[{"Name":"Janviere","PhoneNumber":"62345","UserType":"sender"}]}'
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password -X POST http://localhost:8080/transactions \
> -H "Content-Type: application/json" \
> -d '{"TransactionType":"deposit","Amount":5000,"Currency":"RWF","DateTime":"2025-09-27 14:00:00","ReferenceNumber":"TX12345","BalanceAfterTransact
ion":15000,"Status":"confirmed","MessageText":"Deposit 15000 RWF","Participants":[{"Name":"Thierry","PhoneNumber":"62345","UserType":"sender"}]}'
{"error": "Invalid JSON data", "details": "[Errno 2] No such file or directory: 'data/processed/transactions.json'"}(venv) monicadhieu@m-dhieu:~/alu
/summatives/Before/data/processed$ curl -u admin:password -X POST http://localhost:8080/transactions \
> -H "Content-Type: application/json" \
> -d '{"TransactionType":"deposit","Amount":5000,"Currency":"RWF","DateTime":"2025-09-27 14:00:00","ReferenceNumber":"TX12345","BalanceAfterTransact
ion":15000,"Status":"confirmed","MessageText":"Deposit 15000 RWF","Participants":[{"Name":"Santhiana","PhoneNumber":"62345","UserType":"sender"}]}'
{"error": "Invalid JSON data", "details": "[Errno 2] No such file or directory: 'data/processed/transactions.json'"}(venv) monicadhieu@m-dhieu:~/alu
/summatives/Before/data/processed$ curl -u admin:password -X POST http://localhost:8080/transactions \
> -H "Content-Type: application/json" \
> -d '{"TransactionType":"deposit","Amount":5000,"Currency":"RWF","DateTime":"2025-09-27 14:00:00","ReferenceNumber":"TX12345","BalanceAfterTransact
ion":15000,"Status":"confirmed","MessageText":"Deposit 15000 RWF","Participants":[{"Name":"Dhieu","PhoneNumber":"62345","UserType":"sender"}]}'
{"error": "Invalid JSON data", "details": "[Errno 2] No such file or directory: 'data/processed/transactions.json'"}(venv) monicadhieu@m-dhieu:~/alu
/summatives/Before/data/processed$ |
```

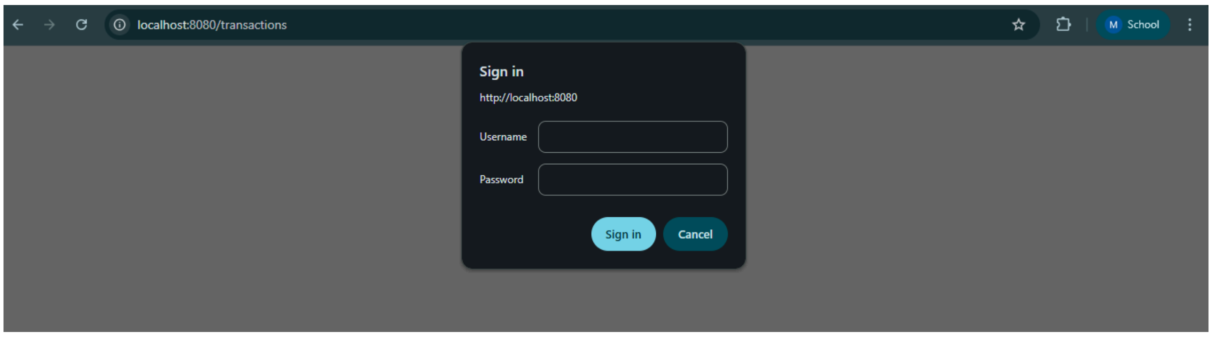


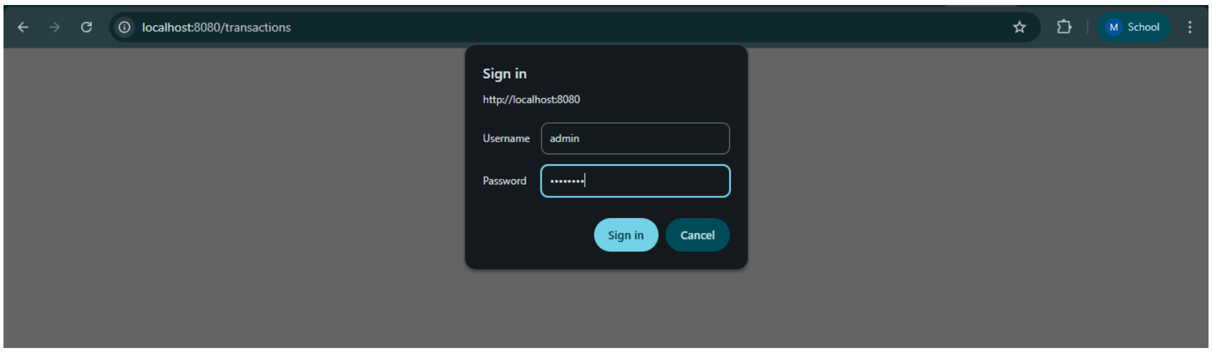
```
/summatives/Before/data/processed$ curl -u admin:password http://localhost:8080/transactions/4
{"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTran
saction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Participants": [{"Name": "Santhiana", "PhoneNumber": "62345", "UserType
": "sender"}], "TransactionID": 4}(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password http://localhost:8080/tr
ansactions
[{"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTra
nsaction": 15000, "Status": "confirmed", "MessageText": "Deposit 5000 RWF", "Participants": [{"Name": "Alice", "PhoneNumber": "12345", "UserType": "
sender"}], "TransactionID": 1}, {"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumbe
r": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Participants": [{"Name": "Janviere", "P
honeNumber": "62345", "UserType": "sender"}], "TransactionID": 2}, {"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "20
25-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Part
icipants": [{"Name": "Thierry", "PhoneNumber": "62345", "UserType": "sender"}], "TransactionID": 3}, {"TransactionType": "deposit", "Amount": 5000,
"Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "Messag
eText": "Deposit 15000 RWF", "Participants": [{"Name": "Santhiana", "PhoneNumber": "62345", "UserType": "sender"}], "TransactionID": 4}, {"Transacti
onType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 1
5000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Participants": [{"Name": "Dhieu", "PhoneNumber": "62345", "UserType": "sender"}],
"TransactionID": 5}](venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password -X PUT http://localhost:8080/transacti
ons/1 \
> -H "Content-Type: application/json" \
> -d '{"TransactionType":"payment","Amount":1000,"Currency":"RWF","DateTime":"2025-09-27 15:00:00","ReferenceNumber":"TX12345","BalanceAfterTransact
ion":14000,"Status":"confirmed","MessageText":"Payment 1000 RWF","Participants":[{"Name":"Bob","PhoneNumber":"67890","UserType":"sender"}]}'
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password http://localhost:8080/transactions
[{"TransactionType": "payment", "Amount": 1000, "Currency": "RWF", "DateTime": "2025-09-27 15:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTra
nsaction": 14000, "Status": "confirmed", "MessageText": "Payment 1000 RWF", "Participants": [{"Name": "Bob", "PhoneNumber": "67890", "UserType": "se
nder"}], "TransactionID": 1}, {"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber"
: "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Participants": [{"Name": "Janviere", "Pho
neNumber": "62345", "UserType": "sender"}], "TransactionID": 2}, {"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025
-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Partic
ipants": [{"Name": "Thierry", "PhoneNumber": "62345", "UserType": "sender"}], "TransactionID": 3}, {"TransactionType": "deposit", "Amount": 5000, "C
urrency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "MessageT
ext": "Deposit 15000 RWF", "Participants": [{"Name": "Santhiana", "PhoneNumber": "62345", "UserType": "sender"}], "TransactionID": 4}, {"Transaction
Type": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 150
00, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Participants": [{"Name": "Dhieu", "PhoneNumber": "62345", "UserType": "sender"}], "T
ransactionID": 5}](venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$
```

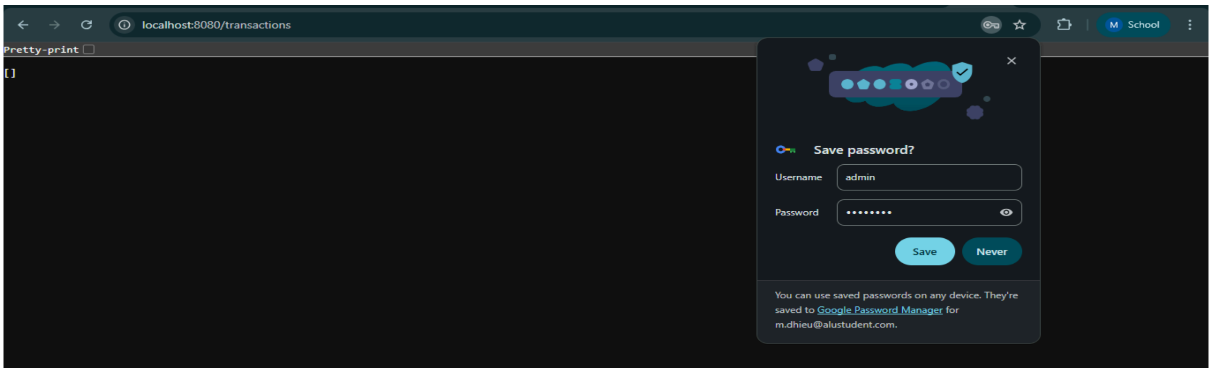


```
ransactionID": 5}](venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password -X DELETE http://localhost:8080/transact
ions/1
curl: (52) Empty reply from server
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/data/processed$ curl -u admin:password http://localhost:8080/transactions
[{"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTra
nsaction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Participants": [{"Name": "Janviere", "PhoneNumber": "62345", "UserType
": "sender"}], "TransactionID": 2}, {"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceN
umber": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "Participants": [{"Name": "Thierry",
 "PhoneNumber": "62345", "UserType": "sender"}], "TransactionID": 3}, {"TransactionType": "deposit", "Amount": 5000, "Currency": "RWF", "DateTime":
"2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "MessageText": "Deposit 15000 RWF", "P
articipants": [{"Name": "Santhiana", "PhoneNumber": "62345", "UserType": "sender"}], "TransactionID": 4}, {"TransactionType": "deposit", "Amount": 5
000, "Currency": "RWF", "DateTime": "2025-09-27 14:00:00", "ReferenceNumber": "TX12345", "BalanceAfterTransaction": 15000, "Status": "confirmed", "M
essageText": "Deposit 15000 RWF", "Participants": [{"Name": "Dhieu", "PhoneNumber": "62345", "UserType": "sender"}], "TransactionID": 5}](venv) moni
cadhieu@m-dhieu:~/alu/summatives/Before/data/processed$
```

Appendix B: Basic Authentication Screenshots





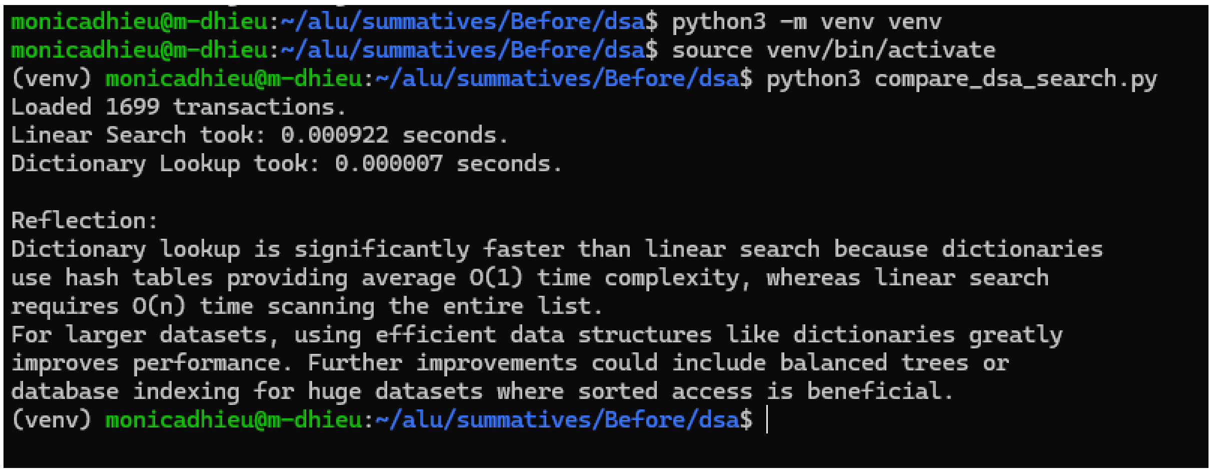


You can use saved passwords on any device. They're saved to Google Password Manager for m.dhieu@alustudent.com.

Appendix C: DSA Comparison Python Code Snippet and Test Results Screenshots

```
def load_transactions(json_path='data/processed/transactions.json'):
    # load transactions from JSON file & return list of transaction dictionaries
    with open(json_path, 'r') as f:
        return json.load(f)

def linear_search(transactions, transaction_id):
    # do linear search for a transaction by ID
    for i in transactions:
        if i['TransactionID'] == transaction_id:
            return i
            # …
```



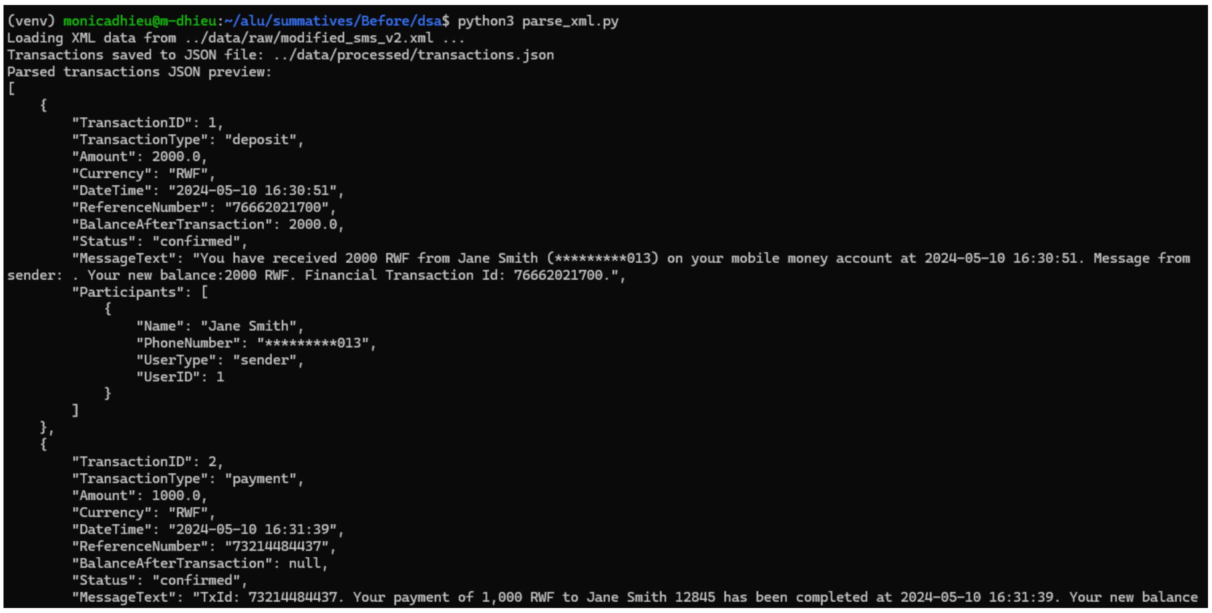
```
monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ python3 -m venv venv
monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ source venv/bin/activate
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ python3 compare_dsa_search.py
Loaded 1699 transactions.
Linear Search took: 0.000922 seconds.
Dictionary Lookup took: 0.000007 seconds.

Reflection:
Dictionary lookup is significantly faster than linear search because dictionaries
use hash tables providing average O(1) time complexity, whereas linear search
requires O(n) time scanning the entire list.
For larger datasets, using efficient data structures like dictionaries greatly
improves performance. Further improvements could include balanced trees or
database indexing for huge datasets where sorted access is beneficial.
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$
```

Appendix D: XML to JSON Parsing Code Snippet and Test Results Screenshots

```
def parse_sms_date(ms_timestamp):
    """ Converts milliseconds to a formatted datetime string: '%Y-%m-%d %H:%M:%S'."""
    ts = int(ms_timestamp) / 1000
    return datetime.utcfromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')

def extract_transaction_info(body):
    """ Parses SMS text and extracts transaction details """
    transaction = {}
    amount_match = re.search(r'([\d,]+) (\w{3})', body)
    if amount_match:
        amount_str = amount_match.group(1).replace(',', '')
         if amount_str:
        # …
```



```
(venv) monicadhieu@m-dhieu:~/alu/summatives/Before/dsa$ python3 parse_xml.py
Loading XML data from ../data/raw/modified_sms_v2.xml ...
Transactions saved to JSON file: ../data/processed/transactions.json
Parsed transactions JSON preview:
[
    {
        "TransactionID": 1,
        "TransactionType": "deposit",
        "Amount": 2000.0,
        "Currency": "RWF",
        "DateTime": "2024-05-10 16:30:51",
        "ReferenceNumber": "76662021700",
        "BalanceAfterTransaction": 2000.0,
        "Status": "confirmed",
        "MessageText": "You have received 2000 RWF from Jane Smith (*********013) on your mobile money account at 2024-05-10 16:30:51. Message from
sender: . Your new balance:2000 RWF. Financial Transaction Id: 76662021700.",
        "Participants": [
            {
                "Name": "Jane Smith",
                "PhoneNumber": "*********013",
                "UserType": "sender",
                "UserID": 1
            }
        ]
    },
    {
        "TransactionID": 2,
        "TransactionType": "payment",
        "Amount": 1000.0,
        "Currency": "RWF",
        "DateTime": "2024-05-10 16:31:39",
        "ReferenceNumber": "73214484437",
        "BalanceAfterTransaction": null,
        "Status": "confirmed",
        "MessageText": "TxId: 73214484437. Your payment of 1,000 RWF to Jane Smith 12845 has been completed at 2024-05-10 16:31:39. Your new balance
```