MoMo SMS Transactions API Project Report

1. Introduction to API Security

Modern APIs must be protected against unauthorized access and data leaks. One basic method is Basic Authentication (Auth), where clients supply a username and password with each request. While easy to implement, Basic Auth has security limitations—credentials are only base64-encoded and can be intercepted if the connection is not encrypted with HTTPS. Protecting APIs properly is essential for any application handling sensitive data.

2. API Endpoint Documentation

All endpoints require Basic Authentication (admin/password123). Unauthorized requests receive 401 Unauthorized.

GET /transactions
List all SMS transactions
Example:

```
GET http://localhost:8000/transactions
Authorization: Basic
```

Response:


```
[
  {
    "id": "1715351458724",
  "service": "M-Money",
  "body": "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.",
```

```
  "transaction_type": "received",
  "amount": "2000",
  "sender": "Jane Smith",
  "receiver": "Your Account",
  "timestamp": "2024-05-10 16:30:51",
  "transaction_id": "76662021700"

  }
]
```


Error codes: 401 Unauthorized

GET /transactions/{id}
View single transaction
Error codes: 401 Unauthorized, 404 Not Found

POST /transactions
Add new transaction (JSON body required)
Error codes: 401, 404 Invalid endpoint

PUT /transactions/{id}
Update transaction by ID (JSON body)
Error codes: 401, 404 Not Found/Invalid endpoint

DELETE /transactions/{id}
Delete transaction
Error codes: 401, 404 Not Found/Invalid endpoint

3. Results of DSA (Data Structures & Algorithms) Comparison

For 20+ records, searching by transaction ID:

- Linear Search:
  Scans entire list for matching ID

- Dictionary Lookup:
  Uses dict[id] for instant access

Sample output:

```
Linear search time: 0.00001621s
Dictionary lookup time: 0.00000215s
```

Dictionary lookup is much faster than linear search (O(1) vs O(n)), especially as datasets grow.

4. Reflection on Basic Auth Limitations

Weaknesses of Basic Authentication:

- Credentials are sent (even over HTTPS, but unencrypted without HTTPS)
- Easy to intercept on insecure networks
- No account management, token expiration, or role-based access

Stronger alternatives:

- JWT (JSON Web Token):
  Secure, stateless tokens; supports claims, expiration, role checks
- OAuth2:
  Standard protocol for delegated authorization; widely adopted by major platforms
- API Key Authentication:
  Suitable for simple cases, better than Basic Auth but less flexible than JWT/OAuth2