REST API Report

1. Introduction to API Security

APIs (Application Programming Interfaces) allow systems to communicate and exchange data. Securing APIs is critical because they often expose sensitive resources such as personal data and financial transactions. Authentication ensures that only authorized users can access the API. In this project, we used Basic Authentication (username and password). However, Basic Auth only encodes credentials in Base64 and does not encrypt them, making it insecure without HTTPS. For production, stronger methods like OAuth2, JWT, or API keys are recommended.

2. Documentation of Endpoints

- **GET /transactions**: Returns all transactions.
- **GET /transactions/<id>**: Returns a specific transaction by ID.
- **POST /transactions**: Creates a new transaction (requires JSON body).
- **PUT /transactions/<id>**: Updates an existing transaction by ID.
- **DELETE /transactions/<id>**: Deletes a transaction by ID.

3. DSA Comparison: Linear Search vs Dictionary Lookup

We implemented two methods to search for transactions: - Linear Search: Iterates through all records until a match is found. Time complexity is O(n). - Dictionary Lookup: Uses transaction IDs as dictionary keys for instant retrieval. Time complexity is O(1). Results showed that dictionary lookups are significantly faster, especially when working with large datasets.

4. Reflection on Basic Auth Limitations

While Basic Authentication is simple to implement, it has several limitations: - Credentials are only Base64 encoded, not encrypted. - Without HTTPS, they can be intercepted and reused (replay attacks). - Does not support token expiration or session management. - Reuse of static credentials increases risk. For secure applications, stronger authentication mechanisms should be used.