Centurion University
Shaping Lives... Empowering Communities...

School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : SHA-256 in Action – Cryptographic Hashing

Objective/Aim:

To understand how SHA-256 cryptographic hashing works by generating hash values for different inputs and analyzing its key properties.

Apparatus/Software Used:

- Laptop/PC
- Web browser or Command Prompt/Terminal
- Internet connection (for online tools)
- Python/Command line utilities or Online SHA-256 hash generators

Theory/Concept:

SHA-256 (Secure Hash Algorithm 256-bit) is a cryptographic hash function from the SHA-2 family. It converts any input message into a **fixed 256-bit (32-byte) hash value**.

Key properties:

- **Deterministic:** Same input always gives the same output.
- **Fixed-length:** Output is always 256 bits (64 hex characters).
- **One-way:** Cannot retrieve the input from the hash.
- **Avalanche effect:** A small change in input produces a drastically different output.
- **Collision-resistant:** Extremely unlikely for two inputs to produce the same hash.

Applications:

- Blockchain (e.g., Bitcoin uses SHA-256 for proof-of-work)
- Digital signatures and certificates
- Password storage and verification
- Data integrity verification

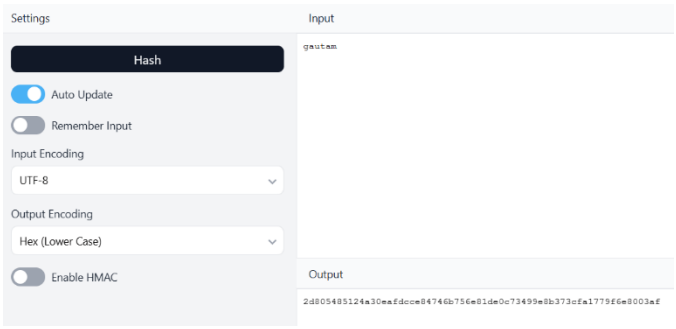
Procedure:

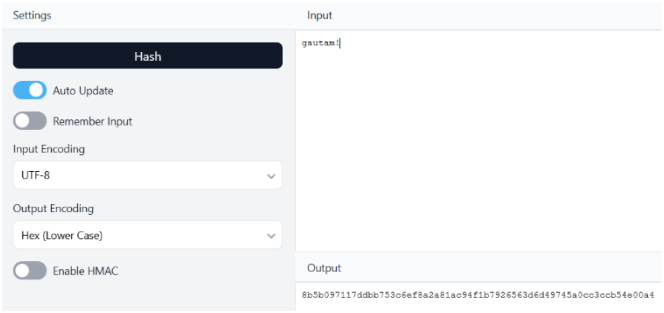
1. Open your **laptop/PC** and ensure Python or a terminal/command prompt is available.
2. Choose an input message or text string (e.g., **"Gautam"**) for hashing.
3. Use any one of the following methods to generate the SHA-256 hash:

- **Online tool:** Open a SHA-256 generator website and enter the text.
- **Command Prompt (Windows):**
    - Create a text file (e.g., message.txt) with your input.

a. Note down the **64-character (256-bit) hash output**.

4. Slightly change the input (e.g., **"Gautam! "**) and repeat the hashing process.
5. Compare the hash outputs of both inputs and observe how even a tiny change in input drastically changes the hash (avalanche effect).
6. Record the results and verify that the hash length remains constant (256 bits) for all inputs.





Observation

- Hash length is always **64 hexadecimal characters** (256 bits).
- A tiny change in input leads to a completely different hash (avalanche effect).

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:
Name :
Regn. No.

Signature of the Faculty: