Class Activity: Database Hashing and Encryption

Introduction

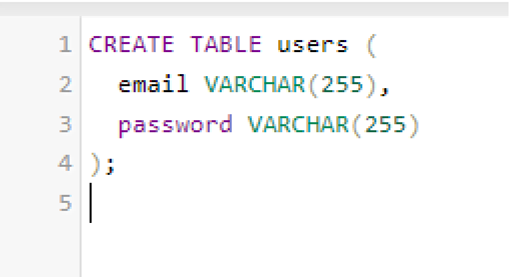
This report documents the steps and results of a class activity focused on database hashing and encryption. The objective was to create a secure database to store user credentials and card details, hash the passwords, and encrypt the card details. Additionally, we attempted to crack the hashed passwords using online tools.

Tools and Software

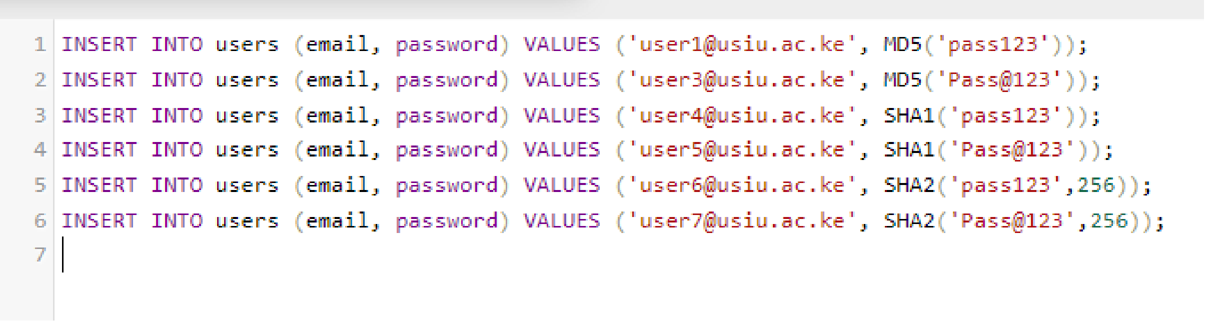
- **XAMPP**: A local server environment used for testing and development.
- **phpMyAdmin**: A web-based database management tool for MySQL.
- **CrackStation**: An online hash cracking tool.

We created 2 tables and inserted values into the tables

1. **User**

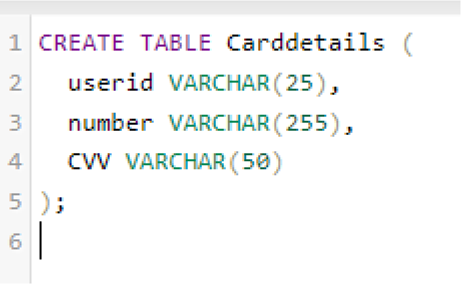


```
CREATE TABLE users (
  email VARCHAR(255),
  password VARCHAR(255)
);
```

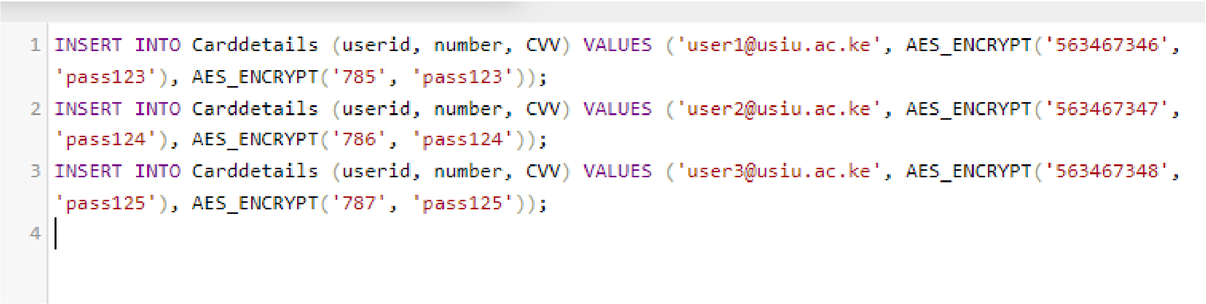


```
INSERT INTO users (email, password) VALUES ('user1@usiu.ac.ke', MD5('pass123'));
INSERT INTO users (email, password) VALUES ('user3@usiu.ac.ke', MD5('Pass@123'));
INSERT INTO users (email, password) VALUES ('user4@usiu.ac.ke', SHA1('pass123'));
INSERT INTO users (email, password) VALUES ('user5@usiu.ac.ke', SHA1('Pass@123'));
INSERT INTO users (email, password) VALUES ('user6@usiu.ac.ke', SHA2('pass123',256));
INSERT INTO users (email, password) VALUES ('user7@usiu.ac.ke', SHA2('Pass@123',256));
```

2. **Carddetails**

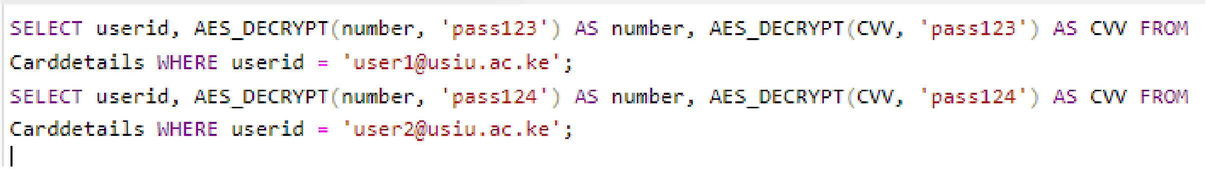


```
CREATE TABLE Carddetails (
  userid VARCHAR(25),
  number VARCHAR(255),
  CVV VARCHAR(50)
);
```



```sql
INSERT INTO Carddetails (userid, number, CVV) VALUES ('user1@usiu.ac.ke', AES_ENCRYPT('563467346',
'pass123'), AES_ENCRYPT('785', 'pass123'));
INSERT INTO Carddetails (userid, number, CVV) VALUES ('user2@usiu.ac.ke', AES_ENCRYPT('563467347',
'pass124'), AES_ENCRYPT('786', 'pass124'));
INSERT INTO Carddetails (userid, number, CVV) VALUES ('user3@usiu.ac.ke', AES_ENCRYPT('563467348',
'pass125'), AES_ENCRYPT('787', 'pass125'));
```

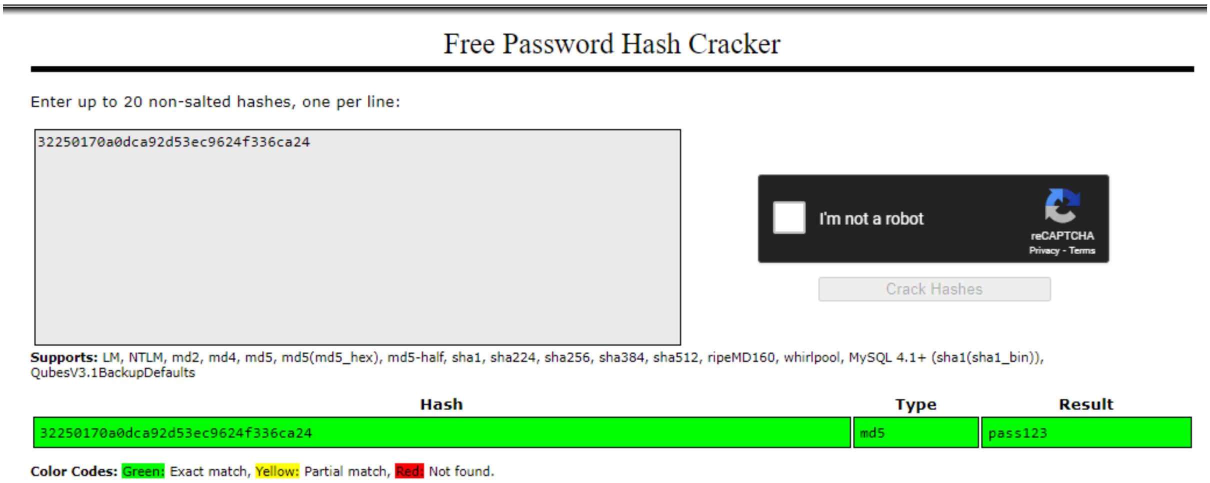
Retrieve and decrypt values from the Carddetails table:



```sql
SELECT userid, AES_DECRYPT(number, 'pass123') AS number, AES_DECRYPT(CVV, 'pass123') AS CVV FROM
Carddetails WHERE userid = 'user1@usiu.ac.ke';
SELECT userid, AES_DECRYPT(number, 'pass124') AS number, AES_DECRYPT(CVV, 'pass124') AS CVV FROM
Carddetails WHERE userid = 'user2@usiu.ac.ke';
```

Crack hash values: Using CrackStation, the following passwords were successfully cracked as shown in the screenshots below

- 32250170a0dca92d53ec9624f336ca24 (MD5) -> pass123
- f91e15dbec69fc40f81f0876e7009648 (MD5) -> Pass@123
- aafdc23870ecbcd3d557b6423a8982134e17927e (SHA1) -> pass123
- f63036841208c85f367cbb2680dea8125d001372 (SHA1) -> Pass@123
- 9b8769a4a742959a2d0298c36fb70623f2dfacda8436237df08d8dfd5b37374c (SHA256) -> pass123
- b6bc7b58510319a151d168ba3d5aecb3ac0a9708d06dd930f37fbc89b6cdc697 (SHA256) -> Pass@123



Free Password Hash Cracker

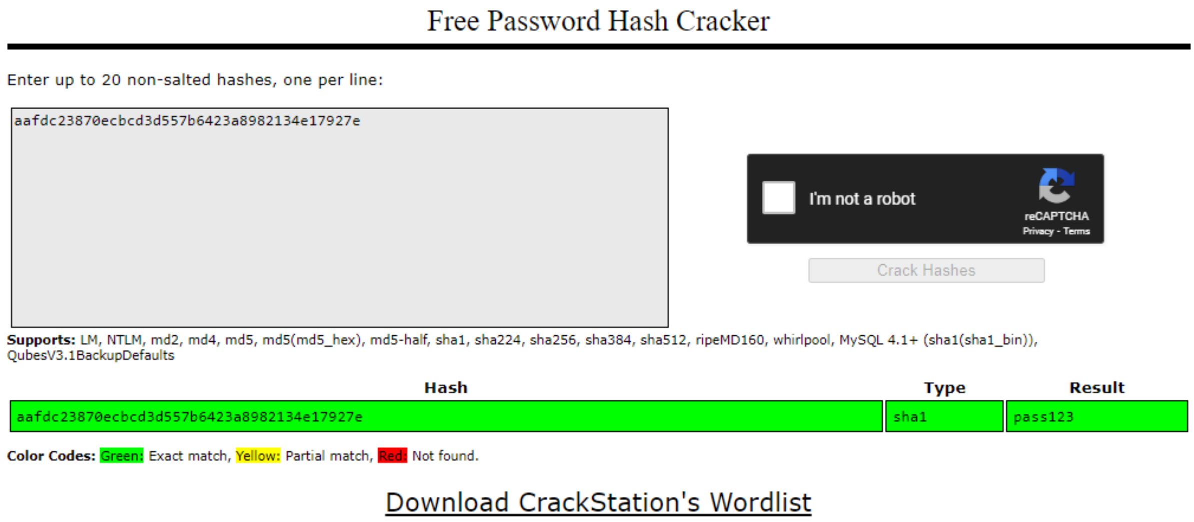
Enter up to 20 non-salted hashes, one per line:

```
32250170a0dca92d53ec9624f336ca24
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
32250170a0dca92d53ec9624f336ca24	md5	pass123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

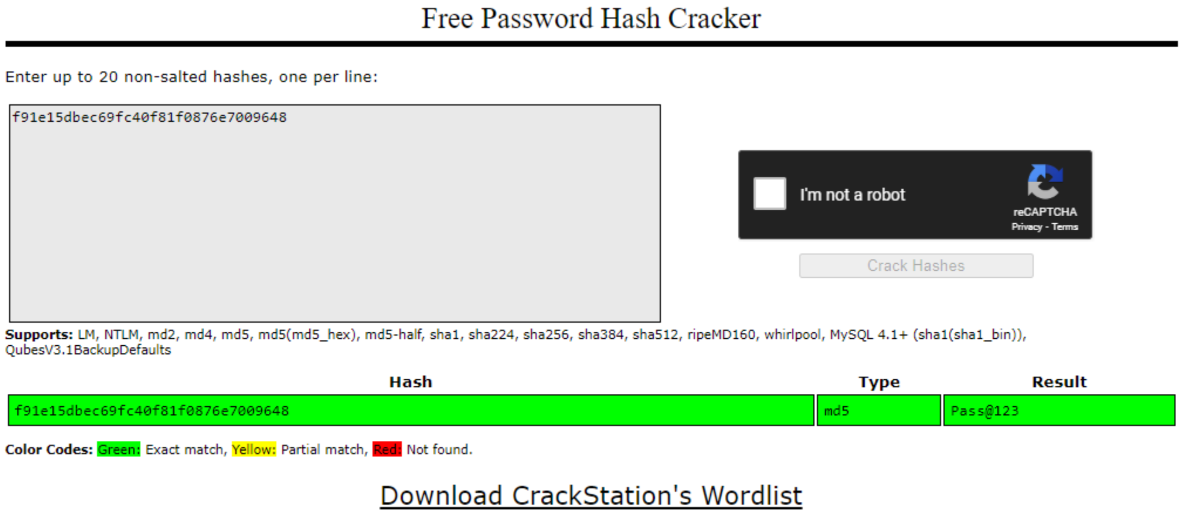
```
aafdc23870ecbcd3d557b6423a8982134e17927e
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
aafdc23870ecbcd3d557b6423a8982134e17927e	sha1	pass123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

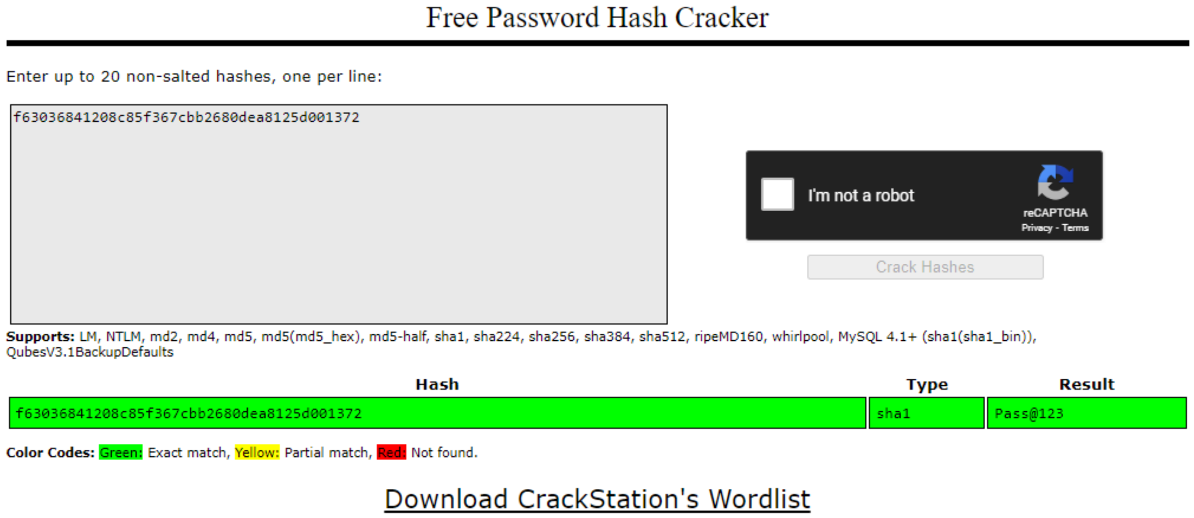
```
f91e15dbec69fc40f81f0876e7009648
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
f91e15dbec69fc40f81f0876e7009648	md5	Pass@123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

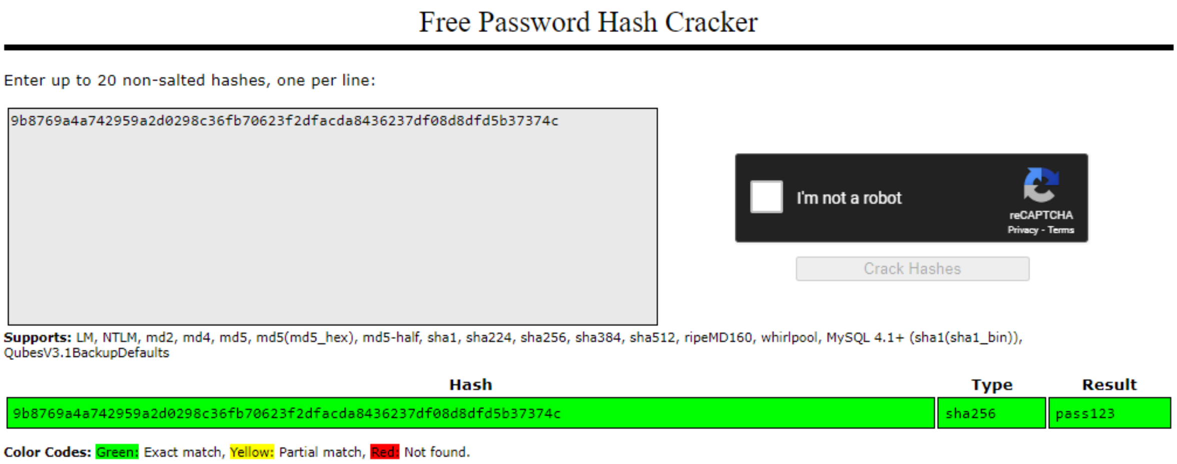
```
f63036841208c85f367cbb2680dea8125d001372
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
f63036841208c85f367cbb2680dea8125d001372	sha1	Pass@123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist



Free Password Hash Cracker

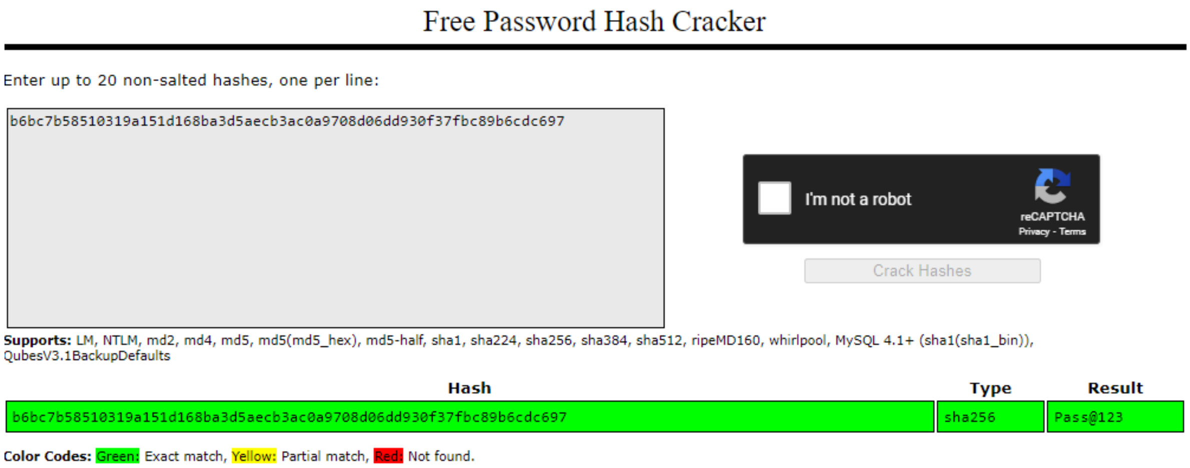
Enter up to 20 non-salted hashes, one per line:

```
9b8769a4a742959a2d0298c36fb70623f2dfacda8436237df08d8dfd5b37374c
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
9b8769a4a742959a2d0298c36fb70623f2dfacda8436237df08d8dfd5b37374c	sha256	pass123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
b6bc7b58510319a151d168ba3d5aecb3ac0a9708d06dd930f37fbc89b6cdc697
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
b6bc7b58510319a151d168ba3d5aecb3ac0a9708d06dd930f37fbc89b6cdc697	sha256	Pass@123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Observations and Conclusion

- **Security Insight**: Hashing algorithms like MD5 and SHA1 are relatively easy to crack with modern tools. Stronger algorithms like SHA256 offer better security but are still vulnerable to dictionary attacks if weak passwords are used.
- **Encryption**: AES encryption for card details ensures data confidentiality. Proper management of encryption keys is crucial for maintaining security.
- **Password Security**: Users should employ strong, unique passwords to enhance security. Systems should implement measures like salting and iterative hashing to protect passwords further.