CMPE 273 Lab 1

Name: Vishwanath Patil

Student ID: 012526410

Questions:

1. Explain the encryption algorithm used in your application. Mention different encryption

algorithms available and the reason for your selection of the algorithm used.

* The encryption algorithm used in this application the Bcrypt algorithm. This algorithm provides with building a password security platform that can evolve around hardware technology to guard against the threats. One of the main reasons for using Bcrypt is that it requires salt by default, it uses a 128-bit salt and encrypts a 192-bit value. Also, bcrypt is pretty easy to use since it comes with the npm package. Bcrypt works in two phases as mentioned below-

Phase-1:

“A function called EksBlowfishSetup is setup using the desired cost, the salt, and the password to initialize the state of eksblowfish. Then, bcrypt spends a lot of time running an expensive key schedule which consists of performing a key derivation where we derive a set of subkeys from a primary key. Here, the password is used as the primary key. In case that the user selected a bad or short password, we stretch that password/key into a longer password/key. The aforementioned practice is also known as key stretching.

Phase-2:

“The magic value is the 192-bit value OrpheanBeholderScryDoubt. This value is encrypted 64 times using eksblowfish in [ECB mode](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Electronic_Codebook_(ECB)) with the state from the previous phase. The output of this phase is the cost and the 128-bit salt value concatenated with the result of the encryption loop.”

Other encryption algorithms available are-

* Crypto
* AES
* Md5
* Pbkdf2
* Argon2

1. Compare the results of graphs with and without in-built MySQL connection pooling of database. Explain the result in detail and describe the connection pooling algorithm if you need to implement connection pooling on your own.

* MySQL has a built in connection pooling strategy by using this we can reduce the latency in the response times of the servers and results can be fetched much faster. In nodejs server there are predefined pooling strategies so that connections can be served much faster. Other connection pooling algorithms can be designed and implemented using JDBC connection pooling algorithms in order to reduce the maximum latency time.

One such connection pooling strategy is as shown below.



1. What is SQL caching? What all types of SQL caching are available, and which suits your code the most. You don’t need to implement the caching, write pseudo code or explain in detail.

* SQL Caching is the storing of result sets generated as a result of a query against a database and using the copy of the result for later results. Then after every time a request is made to a server if the result is already there in the cache it is retrieved from there itself. That, being said, there is a lot more to this issue than just “caching”. The real issue isn’t caching, it is the invalidation that is the hard part. Storing information isn’t hard, knowing when it is invalid and when it is valid is required. Sometimes the data stored may end up being useless. Thus, we need to store the data in such a way that it is useful most of the time.
* Types of caching available are-
* Result Caching
* Query Caching
* Auto Parameterized Caching

#) The efficient type of caching available for canvas application can be Result caching. Since every time user makes request to the DB. Instead of fetching the results from the Database they can be easily retrieved from the cache memory.

* “Is your session strategy horizontally scalable? If YES, explain your session handling strategy. If NO, then explain how you can achieve it.
* The session handling in this Canvas application has been horizontally scaled. The cookie here is stored in the client side and is encrypted. When a request is made the cookie is sent by the client to the server. Both server and client have two keys which are used to decrypt the cookie. If the decrypting results are same of the cookie, the server validates the cookies and processes the request.