Q1:

We store the hash of the password so that even we do not know the password. We can use the hash to verify that the user knows the password by simply comparing the hashes.

We can store the hash of the user’s username and password to make each and every login unique instead of having possible repeated hashes (hash functions are one to one obviously) from repeated passwords.

Q2:

Mysqli\_connect() = opens a connection to the database

Mysqli\_select\_db = selects the default database for queries

Mysqli\_error() = returns a string description of the last error

Q3:

It is a good idea to use the stored procedure here so that we don’t have to deal with how the data from creating a post is stored. It is necessary because it allows us to treat creating a post as an atomic operation. Also allows for separation between the levels of schema. It additionally, in the case of database failure, is an atomic operation so we don’t have to worry about cleaning up if one insert was successful and the other was not. It would be a bad if for it was a procedure on a structure that was being constantly modified which means that the stored procedure itself must be update again and again.

Q4:

It allows the user to implement anything as a valid sql query on the successive lines via commenting out the rest of the procedure. It allows the user to perform queries that they should not be able to run. What happened was basic sql injection where they input their user name, salted a hash of ‘,’ and then commented out the rest of the line that the input was inside of (I.e. the rest of the register user query).

Q5:

It redirected to the imdb webpage given in the script. This is more dangerous because it could redirect the user to any website malicious or not without their consent causing any untold amount of harm to their browsing history.