**Case**

1. Database Servers in Docker (30%)
   * Docker consists of: a database
   * *Database* which will be created, will have another username because it is requested by the Marketing division with the username: Marketing\_001, the password is free, but marketing can only be used from LOCALHOST.

Please attach:

1. *Step by step screenshot* work and explanation
2. The final result of the command:

select user, host from mysql.user

1. PHP: Webserver in Docker (40%)
   * Docker image created as 'trucorp-web'
   * The Docker container is running on port 80 which is forwarded to port 8000
   * Applications can be accessed from http: // <server-ip>: 8000
   * The Docker Image must use the dockerfile
   * *Image* using PHP 7.4.x + Web Server
   * Make sure the access control list for the ownership of 'u' and 'g' is set to www-data, and remove the 'w' and 'x' permissions for others.
   * Content from the web must be copied using commands from within the dockerfile
   * *Command* **docker exec** should not be executed at all
   * *Command* **-v** from docker run should not be used, everything can be run via dockerfile

Please Attach:

1. *Step by step screenshot* work and explanation.
2. *Screenshot* from the contents of the dockerfile and explain line by line of the syntax used.
3. Network in Docker (30%)
   * The PHP files for connection are as follows:

<? php

echo "hello world";

$ link = mysqli\_connect ("[DatabaseIPNetwork]", "[databaseUsername]", "[databasePassword]");

if (! $ link) {

echo "Cannot connect to MySQL". PHP\_EOL; exit;

}

echo "Success: Connection established!" . PHP\_EOL;

echo "Host Information:". mysqli\_get\_host\_info ($ link). PHP\_EOL; mysqli\_close ($ link);

?>

* + *Container* from PHP: Webserver and Database must be united into one docker

*network* with a free name

* + PHP will be considered connected if the web display displays a successful message
  + *Config* **not** must be in the dockerfile.

Please Attach:

* 1. *Step by step screenshot* work and explanation.
  2. *Screenshot* the result of the final website appearance.
  3. *Screenshot* results of docker network inspect [networkName]