**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **10 june 2020** | **Name:** | **Sanketh S Acharya** |
| **Course:** | **KiCad like a pro 2nd edition** | **USN:** | **4AL17EC084** |
| **Topic:** | **•Mounting holes.**  **• Create a library and put your own**  **• component in that library. Create PCB footprint component.**  **• Add Footprint search path**  **• Prepare production files.** | **Semester & Section:** | **6TH SEM & ‘B’ SEC** |
| **Github Repository:** |  |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **C:\Users\cw\Desktop\10 j1.png** |
| **Report –**  **PCB Design:**  **• Creating holes in the PCB for pins.**  **C:\Users\cw\Desktop\10  j2.png**  **• How to Create a library and put your own component in that library and created our ownlibrary for Resistor.**  **•Create PCB footprint component was shown.**  **• Parameters required to place an order for PCB printing.**  **C:\Users\cw\Desktop\10 j3.png** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:10 june 2020** |  | **Name:Sanketh S Acharya** |  | |
| **Course:MySql** |  | **USN:4AL17EC084** |  | |
| **Topic:** |  | **Semester & Section:6TH SEM & ‘B’ SEC** |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  **Hiding a sensitive data:**  When thinking about security within a MySQL installation, you should consider a wide range of possible procedures / best practices and how they affect the security of your MySQL server and related applications. MySQL provides many tools / features / plugins in order to protect your data including some advanced features like [Transparent Data Encryption aka TDE](http://dasini.net/blog/2018/04/10/mysql-security-mysql-enterprise-transparent-data-encryption/),  [Audit](http://dasini.net/blog/2018/04/04/mysql-security-mysql-enterprise-audit/), [Data Masking & De-Identification](http://dasini.net/blog/2019/03/19/mysql-security-mysql-enterprise-data-masking-and-de-identification/)  [Data Masking & De-Identification](http://dasini.net/blog/2019/03/19/mysql-security-mysql-enterprise-data-masking-and-de-identification/), [Firewall](http://dasini.net/blog/2018/04/16/mysql-security-mysql-enterprise-firewall/), [Password Management](http://dasini.net/blog/2018/03/07/mysql-security-password-management/), [Password Validation Plugin](http://dasini.net/blog/2018/03/01/mysql-security-password-validation-plugin/), etc…  In order to mitigate the effects of **data breaches**, and therefore the associated **risks for yourorganization**’s brand and reputation, popular regulations or standards including [**GDPR**](https://www.eugdpr.org/)**,**[**PCI DSS**](https://en.wikipedia.org/wiki/Payment_Card_Industry_Data_Security_Standard), [**HIPAA**](https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act),… recommand (among others things) **data masking** and **de-identification**. PHP double quotes vs single quotes Strings in PHP can be specified in four different ways: single quoted, double quoted, heredoc syntax and (since PHP 5.3.0) nowdoc syntax, the first two of them being by far the most frequently used.  It is important to know the difference between using single quotes and double quotes. In this post we will see the difference between them and which should be used when.  Single quoted strings are the easiest way to specify string. This method in used when we want to the string to be exactly as it is written. When string is specified in single quotes PHP will not evaluate it or interpret escape characters except single quote with backslash (‘) and backslash(\) which has to be escaped. PHP sprintf() FunctionDefinition and Usage The sprintf() function writes a formatted string to a variable.  The arg1, arg2, ++ parameters will be inserted at percent (%) signs in the main string. This function works "step-by-step". At the first % sign, arg1 is inserted, at the second % sign, arg2 is inserted, etc.  If there are more % signs than arguments, you must use placeholders. A placeholder is inserted after the % sign, and consists of the argument- number and "\$". See example two.   Related functions: [printf()](https://www.w3schools.com/php/func_string_printf.asp), [vprintf()](https://www.w3schools.com/php/func_string_vprintf.asp), [vsprintf()](https://www.w3schools.com/php/func_string_vsprintf.asp), [fprintf()](https://www.w3schools.com/php/func_string_fprintf.asp) and [vfprintf()](https://www.w3schools.com/php/func_string_vfprintf.asp)  **Syntax**  sprintf(format,arg1,arg2,arg++*)*  Replace the percent (%) sign by a variable passed as an argument: $number = 9; $str = "Beijing"; ...   1. Using the format value %f: $number = 123; $txt = sprintf("%f",$number); ... 2. Use of placeholders: $number = 123; ...A demonstration of string specifiers: $str1 = "Hello";  Introduction to User Defined Variables: MySQL supports user defined variables to have some data that can be used later part of your query. You can save a value to a variable using a SELECT statement and later you can access its value.  Unlike other RDBMSs, you do not need to declare the data type for a variable. The data type is automatically assumed when you assign a value. A value can be assigned to a variable using a SET command as shown below  SET @server\_type:='MySQL';  When you above command is executed, the value, MySQL is assigned to the variable called @server\_type. Now you can use this variable in the later part of the code. Suppose if you want to display the value, you can use SELECT statement | | | |