**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **11/6/2020** | **Name:** | **M V Ramya** |
| **Course:** | |  | | --- | | **PCB design using Ki cad** | | **USN:** | **4AL17EC045** |
| **Topic:** | **Up and Running**   * **Silk-screen and copper pour** * **Mounting holes** * **Create a library and put your own** * **component in that library** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **MV-Ramya-045** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of the session** |
| **REPORT-**  **Silk Screen**   * The silkscreen is printed to the external surface of a PCB to aid in component identification and orientation. Typically this layer contains the component RefDes to locate components on the board after assembly. * KiCad refers to the silkscreen layers as:   F.SilkS - Front silkscreen layer.  B.SilkS - Back silkscreen layer.  **Copper Pour**   * A copper pour or fill refers to an area on a printed circuit board where the original copper is not etched away, and remains in place, usually electrically connected to the Ground signal, producing a “Ground Plane”. * This has a number of advantages, including decreasing the amount of etching fluid required during manufacturing, as well as reducing the amount of electrical noise and signal crosstalk experienced by the circuit elements.   **Virtual Components**   * Virtual components are those which have a footprint on the PCB (and may additionally have a schematic symbol) but do not have an associated physical component which needs to be loaded onto the board during assembly. * Examples of virtual components include:   Mounting holes, Solder bridges ,Net ties, Fiducial markings, Test points  **Custom libraries** can be created in KiCad using the existing libraries which can be edited howerer the user wants it to be and make use of the custom library in the projects.       |  | | --- | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **11/06/2020** | **Name:** | **M V Ramya** | |
| **Course:** | **MySQL** | **USN:** | **4AL17EC045** | |
| **Topic:** | **Insert data into Mysql, Mysql joins, inner join ,right join ,left join** | **Semester & Section:** | **6th A** | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **REPORT-**  **Insert Data Into MySQL :**  After a database and a table have been created, we can start adding data in them.  Here are some syntax rules to follow:   * The SQL query must be quoted in PHP * String values inside the SQL query must be quoted * Numeric values must not be quoted * The word NULL must not be quoted   The INSERT INTO statement is used to add new records to a MySQL table:  INSERT INTO table\_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)  In this we created an empty table named "MyGuests" with five columns: "id", "firstname", "lastname", "email" and "reg\_date". Now, let us fill the table with data.  The following examples add a new record to the "MyGuests" table  <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  $sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('John', 'Doe', 'john@example.com')";  if ($conn->query($sql) === TRUE) {   echo "New record created successfully"; } else {   echo "Error: " . $sql . "<br>" . $conn->error; }  $conn->close(); ?>  **MySQL  Joins:**  This MySQL tutorial explains how to use MySQL JOINS (inner and outer) with syntax, visual illustrations.  **Description:**  MySQL JOINSare used to retrieve data from multiple tables. A MySQL JOIN is performed whenever two or more tables are joined in a SQL statement.  There are different types of MySQL joins:   * MySQL INNER JOIN (or sometimes called simple join) * MySQL LEFT OUTER JOIN (or sometimes called LEFT JOIN) * MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)   **INNER JOIN (simple join)**  Chances are, you've already written a statement that uses a MySQL INNER JOIN. It is the most common type of join. MySQL INNER JOINS return all rows from multiple tables where the join condition is met.  This MySQL INNER JOIN example would return all rows from the suppliers and orders tables where there is a matching supplier\_id value in both the suppliers and orders tables.  Let's look at some data to explain how the INNER JOINS work:  We have a table called suppliers with two fields (supplier\_id and supplier\_name).  **LEFT JOIN :**  The LEFT JOIN keyword returns all records from the left table and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.  LEFT JOIN Syntax  SELECT column\_name(s) FROM table1 LEFT JOIN table2 ON table1.column\_name *=* table2.column\_name*;*  **RIGHT JOIN:**  The RIGHT JOIN keyword returns all records from the right table, and the matched records from the left table .The result is NULL from the left side, when there is no match.  RIGHT JOIN Syntax  SELECT column\_name(s) FROM table1 RIGHT JOIN table2 ON table1.column\_name *=* table2.column\_name*;* | | | |