**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 28th May 2020 | **Name:** | Rashmi K B |
| **Course:** | Logic design | **USN:** | 4AL16EC056 |
| **Topic:** | **Boolean equations for digital circuits. Combinational circuits: Conversion of MUX and Decoders to logic gates.**  **Design of 7 segment decoder with common anode display** | **Semester & Section:** | 8th B |
| **Github Repository:** | rashmikb |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**    **l2.PNG** |
| **Report:**  Boolean algebra is based upon binary scheme.  The two values may be expressed in many ways, such as true or false, 1 or 0, and "on" or "off".  **Boolean laws:**  **Commutative Law:**  (a) A + B = B + A (b) A B = B A    **Associate Law:**  (a) (A + B) + C = A + (B + C) (b) (A B) C = A (B C)    **Distributive Law:**  (a) A (B + C) = A B + A C (b) A + (B C) = (A + B) (A + C)    **Identity Laws:**  (a) A + A = A (b) A A = A  (a) AB +ANOT B = A (b) (A+B)(A+NOT B) = A    **Redundancy Laws:**  (a) A + A B = A (b) A (A + B) = A  (a) 0 + A = A (b) 0 A = 0  (a) 1 + A = 1 (b) 1 A = A  (a) A+NOT A = 1 (b) A NOT A = 0  (a) A+NOT AB = A+B (b) A(NOT A+B) = AB  **Involution Law:**  (a) NOT A = A    **De Morgan's Theorem:**  (a) NOT A or B = NOT A NOT B    (b) NOT A = NOT A + NOT B  Note: NOT A NOT B is different from NOT A Logic Gates Logic gates may have two or more inputs and, except in some special cases, they have a single output. The status of the input and output terminals can only be in one of the two binary conditions, either low (0) or high (1), represented by two different voltage levels, typically 0 volts for logic 0, and around 3 to 5 volts for logic 1.  OR Gates **The Multiplexer:**  The multiplexer is a combinational logic circuit designed to switch one of several input lines to a single common output line    **Design of 7 segment decoder with common anode display:**  A **Digital Decoder** IC, is a device which converts one digital format into another and one of the most commonly used devices for doing this is called the Binary Coded Decimal (BCD) to 7-Segment Display Decoder.  7-segment LED (Light Emitting Diode) or LCD (Liquid Crystal Display) type displays, provide a very convenient way of displaying information or digital data in the form of numbers, letters or even alpha-numerical characters.  Typically 7 segment consist of seven individual coloured LED’s (called the segments), within one single display package. In order to produce the required numbers or HEX characters from 0 to 9 and A to F respectively, on the display the correct combination of LED segments need to be illuminated and **BCD to 7-segment Display Decoders** such as the 74LS47 do just that.  bcd display decoder |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | 28th May 2020 | **Name:** | Rashmi K B | |
| **Course:** | Udemy | **USN:** | 4AL16EC056 | |
| **Topic:** | Python | **Semester & Section:** | 8th B | |
| **AFTERNOON SESSION DETAILS** | | | |
| **p3.PNG**  **Report:**  **Front end:**  Front-end Web Development refers to building web interfaces, specifically the parts of the website that the user will interact with. When you’re browsing the web, everything you see, from images and headings to sliders and buttons is made using HTML, CSS and JavaScript, the main components to any website.  **Front End Frameworks and Libraries:**   * **Bootstrap**: Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. * **AngularJS:** AngularJs is a JavaScript open source front-end framework that is mainly used to develop single page web applications (SPAs). It is an open source project which can be freely used and changed by anyone. It extends HTML attributes with Directives, and data is bound with HTML. * **React.js:** It is maintained by Facebook. React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. * **jQuery:** jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.   **Back-end Development** :  The back end handles application logic, algorithms, database interaction and the processing of user requests.  **Back End Frameworks:**   * The list of back end frameworks are: Express, Django, Rails, Laravel, Spring, etc. * The other back end program/scripting languages are: C#, Ruby, REST, GO etc.   **Back end Languages:** The back end portion is built by using some languages which are discussed below:   * **PHP:** PHP is a server-side scripting language designed specifically for web development. Since PHP code executed on the server side so it is called server-side scripting language. * **C++:** It is a general purpose programming language and widely used now a days for competitive programming. It is also used as backend language. * **Java:** Java is one of the most popular and widely used programming language and platform. It is highly scalable. Java components are easily available. * **Python:**Python is a programming language that lets you work quickly and integrate systems more efficiently. * **JavaScript:**Javascript can be used as both (front end and back end) programming languages. * **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser.   **Connecting the Frontend to theBackend**  # Head of Tkinter application  master =Tk()  master.title("Network Automation")  # configuration for the labels and entry  Label(master, text="Device : ").grid(row=0)  Label(master, text="User ID : ").grid(row=1)  Label(master, text="Password : ").grid(row=2)  e1 =Entry(master)  e2 =Entry(master)  e3 =Entry(master, show='\*')  e1.grid(row=0, column=1)  e2.grid(row=1, column=1)  e3.grid(row=2, column=1,)  # configuration for the button  Button(master, text='Quit', command=master.destroy).grid(row=4, column=0, sticky=W,pady=4)  Button(master, text='Harden', command=show\_entry\_fields).grid(row=4, column=1, sticky=W,pady=4) | | | |