VIT Vellore Institute of Technology

Reg. No. : 23MCA1030

**Final Assessment Test (FAT) - November/December 2023**

| | | | |
|---|---|---|---|
| Programme | **M.C.A.** | Semester | **FALL SEMESTER 2023 - 24** |
| Course Title | **FULL STACK WEB DEVELOPMENT** | Course Code | **PMCA601L** |
| Faculty Name | **Prof. Sandhya P** | Slot | **F1+TF1** |
| | | Class Nbr | **CH2023240101719** |
| Time | **3 Hours** | Max. Marks | **100** |

**PART-A (7 X 10 Marks)**

**Answer all questions**

01. Design a HTML webpage for an online supermarket with the following: **[10]**
    a. Display 4 images of fruits in a single row (use table) (1 Marks)
    b. Create a hyperlink to navigate internally to the bottom of the page (1 Marks)
    c. Create a form that displays 5 items with respective dropdown to choose the quantity of order. (quantity values will be 1 to 3) (1 Marks)
    d. Create a feedback form. (2 Marks)
    e. At the top of the page use CSS animation such that the logo of the supermarket scrolls from left to right. (5 Marks)

02. Create an application using DOM, JavaScript and HTML for displaying an online school report card. Design with the following features: Each (5 Marks) **[10]**
    a. Provide a text box to enter the name of the student. When "add" button is clicked the name of the student entered in the textbox must be dynamically added as text to a paragraph and it must be in bold. The paragraph and bold elements must be dynamically generated.
    b. Enter marks in 3 subjects in 3 respective textboxes. On clicking "mark" button calculate the total marks. If the marks is greater than or equal to 150 display in green color with in a <h2> tag. If the marks is less than 150 display in red with in a <h2> tag.
    Note: Strictly perform the above two using DOM

03. Design an application using PHP and AJAX for an online health monitoring system. The system must fetch sugar level of a patient through a HTML form. On submission the PHP script must evaluate if sugar is greater than or equal to 120. Using AJAX respond "Risk" if sugar level is higher than or equal to 120 and "Normal" otherwise, without reloading the entire page. **[10]**

04. Develop an online event driven system for an automated door in a mall using node.js. When a person reaches the door an "open" event must be emitted. When the event occurs two event handlers must respond. The first handler must display the count of the person entered. The second handler must display "Welcome".
    Note: Write appropriate class module and reuse it in the code. **[10]**

05. Develop a login application using node.js and express middleware for an online banking application. As a web developer create a webpage (login page) to enter username and login password. Create a session. If the username and login password is "Kevin" and "Eric" respectively, then display a welcome message. The webpage must also have a logout link which on clicking must redirect to the login page. If the credentials are invalid redirect to the login page. Set the session timeout period as one minute. **[10]**

06. Develop an online certificate generation application using React. Create a stateful class component that displays "Certificate given to name from University". The name and University must be rendered in a paragraph using props. In the AppComponent generate certificate for two students considering name and University as attributes. Below the certificate of each student add the signature of the coordinator as an image. **[10]**

07. Create a collection using MongoDB for a student management system and perform the following: Each (2 marks) **[10]**

a. Add the register number and name of a student.

b. Delete a student with register number "101".

c. Search for a student with name starting with 'S'.

d. Update name of student with register number '102' to 'Tim'.

e. Display the names of all students.

**PART-B (2 X 15 Marks)**

**Answer all questions**

08. Design an online application for Engineering University admission using Angular. In the initial screening process enter the roll number and 12th standard marks of the student in two textboxes respectively. The textboxes and the submit button must be a part of the AppComponent. If the marks is greater than or equal to 50 then list the name of the candidate in an ordered list in a child component called "eligible". If the marks is lesser than 50 then display name in an ordered list in another child component called "not-eligible". Display the parent, eligible and not-eligible components in a sequence. **[15]**

09. Create an online application using Angular service that provides an array of item objects in a shop. An item object has itemid, name and quantity available. Create two child components for online shopping and stock management respectively. In the shopping component display name and quantity by fetching the details from the service. In the stock management component display the itemid and quantity of items with quantity less than 5. Integrate the components to the AppComponent. **[15]**

⇔⇔⇔