Reg. No.: 23MCA1030

Name : Vinayak Singh

VIT®

Vellore Institute of Technology

Continuous Assessment Test (CAT – II) – October 2023

Programme	:	**MCA**	Semester	:	**Fall Semester 2023-24**
Course	:	**Full Stack Web Development**	Code	:	**PMCA601L**
			Class Nbr	:	**CH2023240101719**
Faculty	:	**Dr. Sandhya P**	Slot	:	**F1+TF1**
Time	:	**1 ½ hours**	Max. Marks	:	**50**

Answer all the Questions

1. As a web developer you are given the task of designing an online registration form for an event called "MasterChef Chennai" using HTML and JavaScript. The form must obtain the following details and perform validations as specified:

- Textbox – name of the contestant (cannot be empty)
- Number – age of the contestant (age in range of 20-60 inclusive)
- Email – email of the contestant (must be in correct format, if not display a custom error text "Correct mail-id please")
- Textbox – pin code (must be 6 digits, check using regular expression)
- Submit button

10

Note:

- The above validations must be performed using HTML attributes (3 marks)
- When there is invalid data, style respective controls with a box shadow of blue color and default black if valid. Perform this using appropriate CSS pseudo-selectors. (3 marks)
- E-mail validation must be performed using JavaScript validation DOM API to display the custom text. (4 marks)

2. Develop an online greeting website using jquery. Assume that the webpage has paragraph with attribute called "title", <h1> tag without text, <img> tag without "src" attribute and a <h5> tag with text "XXXX". The value of the "title" attribute is "Happy Birthday". The webpage must have a button called "Greet". On clicking the "Greet" button perform event handling as follows using jquery (2 marks):

- Assign the "title" attribute value to the text of <h1> tag. (3 marks)
- Change the styling of background color of the <h1> text to "pink". (2 marks)
- Set the "src" attribute of <img> tag to "C:/birthday.jpg". (2 marks)
- Replace the content of <h2> tag with text "XXXX" to "Kevin". (1 mark)

10

3. Develop a SPA (Single Page Application) for online verification of age eligibility to caste vote during the coming polls using Angular JS. The application consists of 3 components namely AppComponent (parent component) and two child components namely eligible and not-eligible. In the AppComponent obtain the name and age of the **15**

voter in textboxes and on clicking the "Verify" button display the voter's name in AppComponent in an ordered list. Simultaneously send the voter object (including name and age) as inputs to eligible and not-eligible components. If the age of the voter is greater than or equal to 18, then display the name of the voter in an ordered list in eligible component. If the age of the voter is lesser than 18, then display the name of the voter in an ordered list in not-eligible component. The two child components must be embedded in AppComponent such that AppComponent displays all the names, followed by eligible voter's names and finally not eligible voter's names. Perform the above using inter component communication.

Note:

- For all the three components write both the component class (typescript file) and template (html file).
- Write the appropriate interface to represent a voter.
- Write the steps for app creation and component creation in CLI.
- Assume <app-eligible> and <app-not-eligible> as selectors of eligible and not-eligible components respectively.

4. Eric's Pizzeria is a new Pizza shop in town. They have approached you to develop a SPA (Single Page Application) using Angular JS for online ordering. The web app consists of 3 components. The parent component (AppComponent) must display the cost of an order where it is initially set to zero. Create a child component to place order, namely "MenuComponent" that has buttons (+) to add one vegetarian pizza and (-) to remove one vegetarian pizza. Similarly, it has (+) and (-) buttons for adding and removing one non-vegetarian pizza. MenuComponent fetches initial cost as input from AppComponent. On clicking the (+) and (-) buttons compute the price of the order using event binding in MenuComponent. Assume the cost of one vegetarian pizza is Rs. 40 and non-vegetarian pizza is Rs. 80. Update the computed cost in AppComponent by sending cost as output from MenuComponent. Create another child component called "TotalComponent". "TotalComponent" must get cost as input from AppComponent. TotalComponent must have a button called "TotalCost" which on clicking will deduct Rs.10 if the cost is greater than or equal to Rs. 500 and retain the same amount if not. A GST of Rs. 20 must be added to cost in both cases. Update the final cost of the order in AppComponent by sending the final cost as output from "TotalComponent". Perform the above using inter component communication. **15**

Note:

- For all the three components write both the component class (typescript file) and template (html file).
- Write the steps for app creation and component creation in CLI.
- Assume <app-menu> and <app-total> as selectors of MenuComponent and TotalComponent class respectively.