VIT®

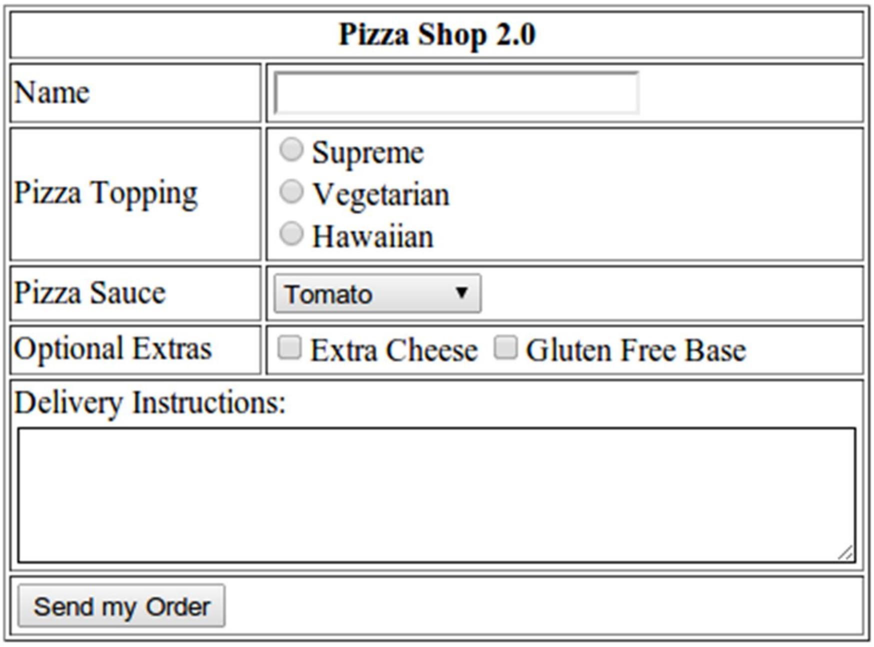
Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING
B.Sc (Computer Science) - Fall 2021-22
Lab Digital Assessment 4

Course: Web Development Lab (CSC4002) **Faculty Incharge: Dr.Mareeswari V**

1. Design the HTML form. Using JavaScript, validate all the form elements have value. Display all the values in table format after clicking Send my Order.



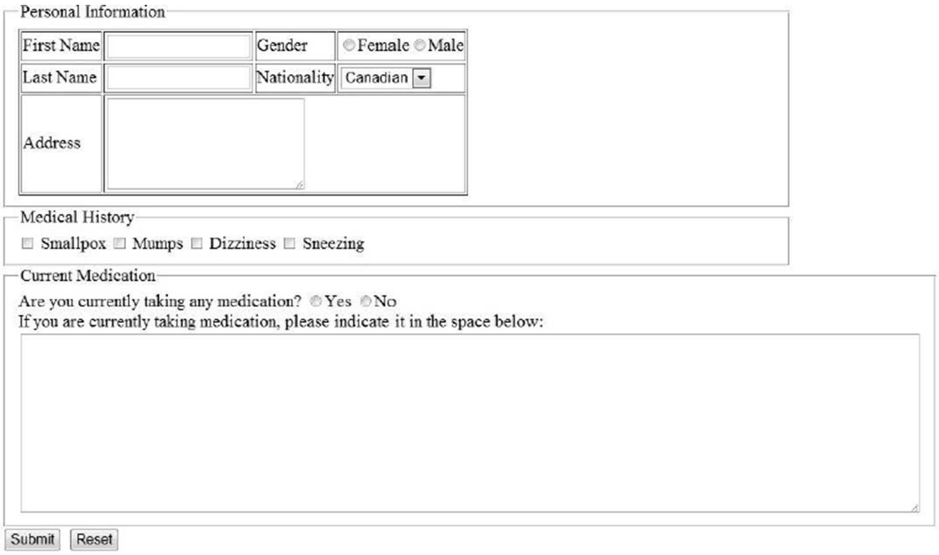
Pizza Shop 2.0	
Name	
Pizza Topping	○ Supreme ○ Vegetarian ○ Hawaiian
Pizza Sauce	Tomato ▼
Optional Extras	□ Extra Cheese □ Gluten Free Base
Delivery Instructions:	
Send my Order	

2. Develop an application using **HTML and JavaScript** code to design the form and validate it and display the respective validation error messages in **red** colour.

Input Name	**Input Type**	**Input Validations**
User Name	text	1. not empty 2. at least 5 characters long 3. can't be more than 25 characters long 4. must contain only numbers and letters 5. unique user name hence check the unavailability (array holds set of available user names)
Mail ID	text	1. not empty 2. valid format: vmareeswari@vit.ac.in 3. Ends with vit.ac.in

Zip Code	Text	(2 alphanumeric characters for country code), 3 digits for city code - 4 digits for area code Example: (A1)123-1234
Phone Number	Text	Number should start with +91followed by10 digits or +65 followed by 8 digits Example:+91 1234567890 , +65 12345678

3. Design and validate the given form elements.



Personal Information

First Name		Gender	○ Female ○ Male
Last Name		Nationality	Canadian ▾
Address			

Medical History

□ Smallpox □ Mumps □ Dizziness □ Sneezing

Current Medication

Are you currently taking any medication? ○ Yes ○ No

If you are currently taking medication, please indicate it in the space below:

The validation should satisfy the following rules:

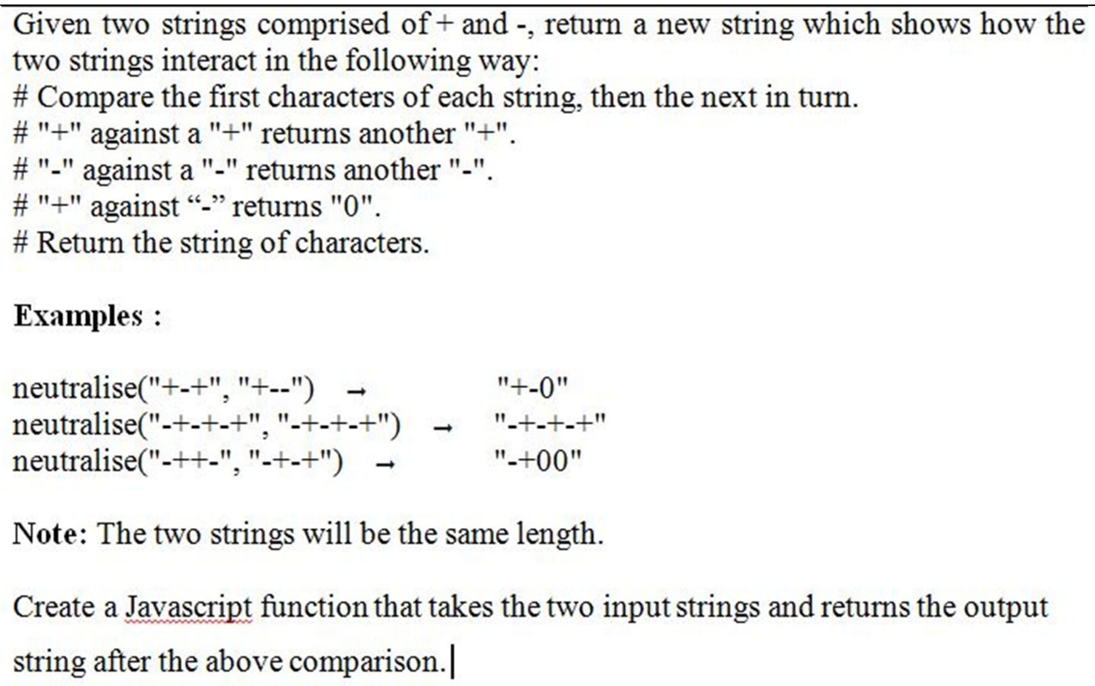
1. No empty input. All form elements should have an input.
2. Exception for rule one: If you choose "No" for "Current Medication", the textarea for "Current Medication" *must* be empty. Otherwise, there should be an input to indicate the details, i.e., if you choose "Yes", the textarea should not be empty.
3. Maximun input characters of "First name" or "Last name" are 50 characters.
4. Maximun input characters of "Address" are 300 characters.
5. For correct user inputs, show the user inputs in an alert window. Othersiwse, warn the user in an alert window.

4. Create a Javascript function to validate the given string as valid IP Address using regular expression. It takes a string (IPv4 address in standard dot-decimal format). Required pop up boxes should be used. If the IP address is valid then send the valid IPAddress to this mail ID: vmareeswari@vit.ac.in. If it is invalid, request the client to reenter the input.

Examples

isValidIP("1.2.3.4") → true

	isValidIP("1.2.3") ➞ false isValidIP("1.2.3.4.5") ➞ false isValidIP("123.45.67.89") ➞ true isValidIP("123.456.78.90") ➞ false isValidIP("123.045.067.089") ➞ false **Notes** IPv6 addresses are not valid. Leading zeros are not valid ("123.045.067.089" should return false). You can expect a single string for every test case. Numbers may only be between 1 and 255. The last digit may not be zero, but any other might.
5.	Create a Javascript function to validate the given a string as valid Postal Address. If the address is valid then send the valid Address to this mail ID: vmareeswari@vit.ac.in. If it is invalid, request the client to re-enter the input. Note: Required pop up boxes should be used. 1. **door number** is a number from1 to 100. 2. **street** could contain characters and numbers ended with street or road. 3. **city** could contain strings a to z (all cases accepted). 4. **pincode** is formatted value starts with 632 followed by 3 digits number. **Example:** 43, Pillayarkovil street, Katpadi,632014 **Example:** 67, VIT road, Tiruvalam,632123
6.	Create a Javascript function to validate the given a string as valid HTML GET request. If it is valid then redirect to the profile.html. Otherwise, display the error message "404 Request cannot be processed". Note: Required pop up boxes should be used. **Input String Format:** name1=val1&name2=val2&name3=val3&name4=val4 **Example #1** "val=Mareeswari&id=78&dept=SITE&time=23:59" **Example #2** "val=MAREES2021&id=100&dept=SSE&time=23:59" Each GET request has a specific value:

	1. **val** could contain string a to z and number 0 to 9 (any case accepted). 2. **id** is a number from 0 to 100. 3. **dept** could contain strings a to z. 4. **time** is formatted value (00:00 to 23:59).
7.	 Given two strings comprised of + and -, return a new string which shows how the two strings interact in the following way: # Compare the first characters of each string, then the next in turn. # "+" against a "+" returns another "+". # "-" against a "-" returns another "-". # "+" against "-" returns "0". # Return the string of characters. **Examples :** neutralise("+-+", "+--") → "+-0" neutralise("-+-+-+", "-+-+-+") → "-+-+-+" neutralise("-++-", "-+-+") → "-+00" **Note:** The two strings will be the same length. Create a Javascript function that takes the two input strings and returns the output string after the above comparison.
8.	Create a Javascript function to find the given strings rhyme sentences. It returns true if two lines **rhyme** and false otherwise. For the purposes of this exercise, two lines rhyme if the **last word** from each sentence contains the **same vowels**. Required pop up boxes should be used. **Examples** doesRhyme("Sam I am!", "Green eggs and ham.") → true doesRhyme("Sam I am!", "Green eggs and HAM.") → true // Capitalization and punctuation should not matter. doesRhyme("You are off to the races", "a splendid day.") → false doesRhyme("and frequently do?", "you gotta move.") → false **Notes:** Case insensitive.
9.	Create a Javascript function that takes a string of words and returns the highest scoring word. Each letter of a word scores points according to it's position in the alphabet: a = 1, b = 2, c = 3, etc. **Examples** word_rank("The quick brown fox.") → "brown"

	word_rank("Nancy is very pretty.") ➞ "pretty" word_rank("Check back tomorrow, man!") ➞ "tomorrow" word_rank("Wednesday is hump day.") ➞ "Wednesday" **Notes** • If two words score the same, return the word that appears first in the original string. • The returned string should only contain alphabetic characters (a-z). • Preserve case in the returned string (see 4th example above).
10.	Details of Product (Model Name, Price and Brand) is stored and maintained by company. Validate the entries of Product details. If it is, display the shipping charge based on price. Rs.100 for shipping charge is collected when the price is less than Rs.1000. Example: Product5 is lowest price than Product3. Develop the web application for the same using HTML and JavaScript.