**VETRI TECHNOLOGY SOLUTIONS**  
  
**PRACTICAL EVALUATION -2**

**MODULE NAME:HTML (EACH 2 MARKS) MARKS:10**

**1.Job Application Form (HTML Forms Validation)**

**Requirements:**

* Use HTML5 form elements
* Apply validation without JavaScript
* Required fields for Name, Email, Phone, Resume Upload
* Email validation, Pattern validation for phone number

## 2.Personal Portfolio Website

**Requirements:**

* Include an **About Me** section
* Add **Projects**, **Skills**, and **Contact Form**
* Use **semantic HTML** for better SEO
* Make it **mobile-friendly** with proper structure

**3.Online Quiz Application**

**Requirements:**

* Display multiple-choice questions
* Allow users to select an answer
* Show results after submission
* Use **radio buttons** for answer selection

**4.Student Registration Form**

**Requirements:**

* **Full Name, Email, Phone, Course Selection**
* **Dropdown** for selecting a course
* **Validation** for required fields

5.**Online Resume**

**Requirements:**

* Use **semantic HTML** tags
* Include **sections** like Summary, Education, Work Experience, and Skills
* Add **contact details**

**VETRI TECHNOLOGY SOLUTIONS**  
  
**PRACTICAL EVALUATION -2**

**MODULE NAME:CSS (EACH 3 MARKS) MARKS:30**

**1.Create a responsive, multi-column web layout using CSS Grid.**

**2.Build a complex layout using Flexbox with multiple rows and columns.**

**3.Create a smooth text fade-in animation when the page loads.**

**4.Style the scrollbar for a webpage.**

**5.Implement a parallax scrolling effect for the background of a section.**

**6.Design a dropdown menu that appears with an animation.**

**7. Build a multi-column layout that adjusts to the screen size using Flexbox.**

**8.Create a tooltip that appears when hovering over an element.**

**9.Create a 3D effect on hover for an element such as a card or image.**

**10.Create a responsive layout for a blog using CSS Grid.**

**VETRI TECHNOLOGY SOLUTIONS**  
  
**PRACTICAL EVALUATION -2**

**MODULE NAME: JAVASCRIPT (EACH 5 MARKS) MARKS:60**

### **1.SHOPPING CART FUNCTIONALITY**

Create a shopping cart that allows users to add products, view the cart, and update quantities.

**Requirements:**

Each product should have an "Add to Cart" button.

Display the cart items with product names, quantities, and prices.

Allow users to update the quantity of each item in the cart.

Recalculate the total price as items are added or quantities updated.

### **2.REMOVE ITEMS FROM CART**

Implement the functionality to remove items from the shopping cart.

**Requirements:**

Add a "Remove" button next to each item in the cart.

When clicked, remove the item from the cart and update the total price.

### **3. CART TOTAL CALCULATION**

Create a function that calculates the total price of all items in the shopping cart, taking into account their quantities.

**Requirements:**

Display the total price of the items in the cart.

Recalculate the total price when items are added, removed, or updated.

### **4. PRODUCT FILTER BY PRICE RANGE**

Implement a price range filter for the products. Users should be able to select a minimum and maximum price, and the products should be filtered accordingly.

**Requirements:**

Create two input fields for the minimum and maximum price.

Filter the displayed products based on the selected price range.

Only show products that fall within the specified range.

### **5. CHECKOUT FORM VALIDATION**

Create a checkout form with fields like name, address, and payment information. Use JavaScript to validate the form before submission.

**Requirements:**

Validate that all fields are filled out properly.

Ensure the email format is correct.

Ensure the payment details (credit card, for example) are valid.

### **6. APPLY DISCOUNT CODE**

Implement a discount code feature that allows users to enter a discount code during checkout. If the code is valid, apply a discount to the total price.

**Requirements:**

Create an input field for the discount code.

When a valid code is entered, apply a discount (e.g., 10% off) to the total price.

Show a message indicating whether the code is valid or invalid.

### **7. PRODUCT RATING SYSTEM**

Implement a star rating system for each product, allowing users to rate products from 1 to 5 stars.

**Requirements:**

Display a rating system with 5 stars.

Users can click to select a rating for each product.

Display the average rating for the product below the product name.

### **8. WISHLIST FUNCTIONALITY**

Implement a wishlist feature where users can add products they are interested in without adding them to the cart.

**Requirements:**

Add a "Add to Wishlist" button next to each product.

Display the products that have been added to the wishlist on a separate "Wishlist" page.

Allow users to remove items from the wishlist.

### **9. MULTI-STEP CHECKOUT PROCESS**

Implement a multi-step checkout process where users go through several steps (e.g., Address, Payment, Review) before completing their purchase.

**Requirements:**

Split the checkout process into multiple steps.

Users should navigate between steps using "Next" and "Previous" buttons.

At the final step, show a summary of the order with all selected items and total price.

### **10. ORDER SUMMARY PAGE**

After the user completes the checkout process, display an order summary page that includes order details, shipping information, and payment summary.

**Requirements:**

Show the list of products purchased with their quantity and total price.

Display shipping address and payment method.

Display the final total amount for the order.

### **11.LANGUAGE TOGGLE FOR INTERNATIONAL USERS**

* **Task:** Implement a language toggle to switch between multiple languages on the website.
* **Requirements:**
  + Provide buttons to toggle between languages (e.g., English, Spanish, French).
  + Dynamically change all text content on the page based on the selected language.

### **12. INTERACTIVE CALENDAR FOR CLASS SCHEDULES**

* **Task:** Implement an interactive calendar to display class schedules and upcoming events.
* **Requirements:**
  + Use JavaScript to display the calendar with class timings.
  + Allow users to click on dates to see details about the class or event.
  + Implement color-coding for different types of events (e.g., exams, lectures, office hours).

**\*\*\*\*\*ALL THE BEST\*\*\*\*\***