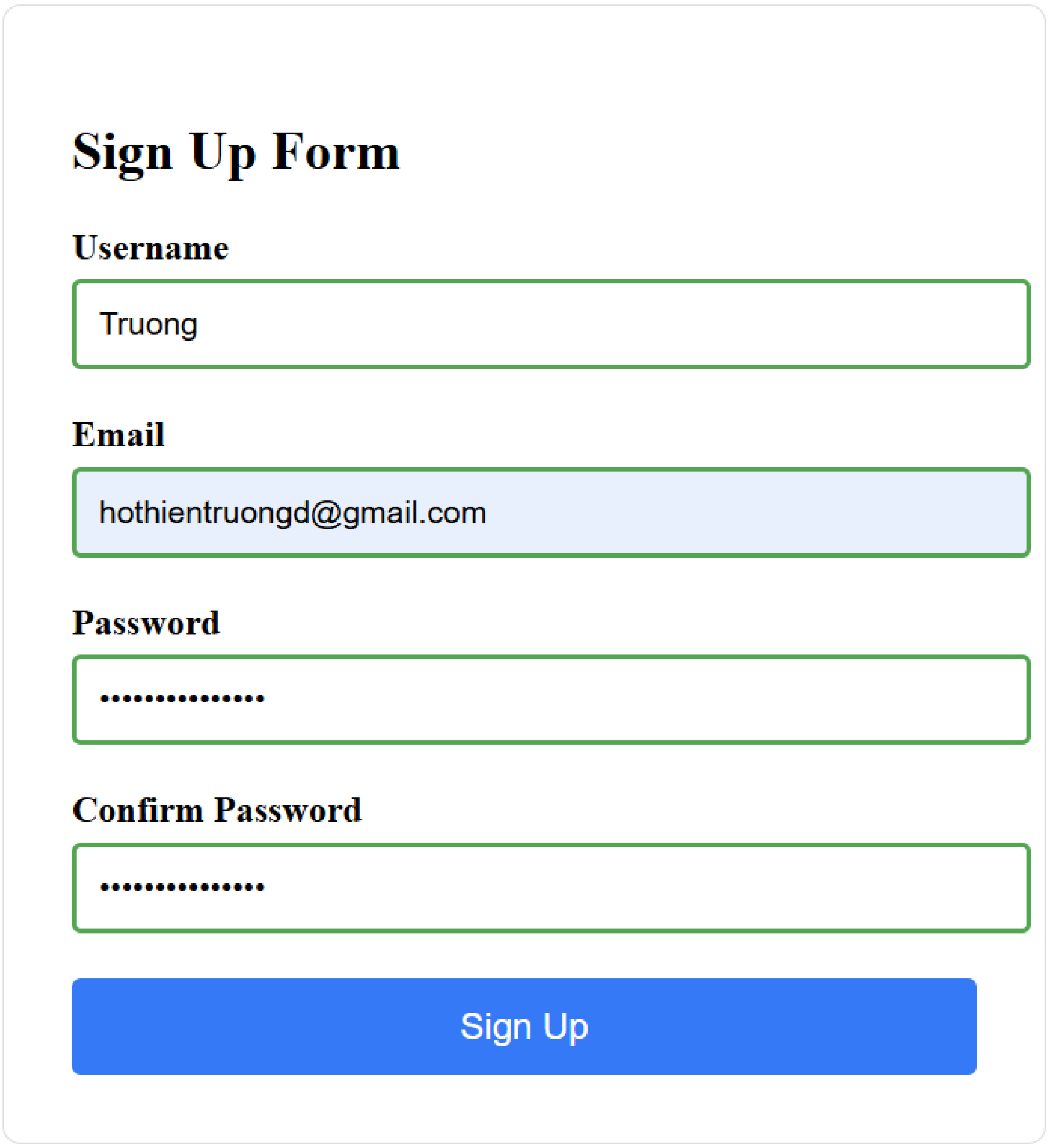
Student name: Hồ Thiên Trường – ITCSIU22306

Task 1.1

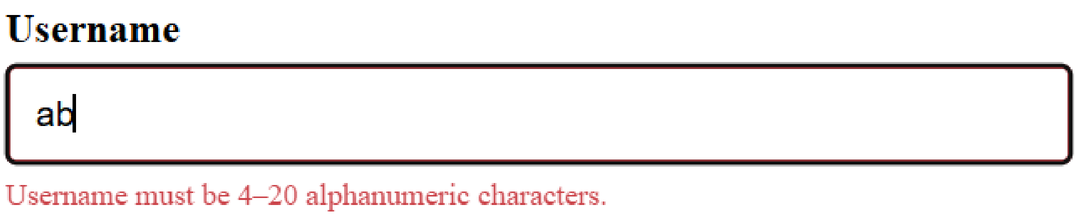
- Case1: all 4 form are approve



Sign Up Form

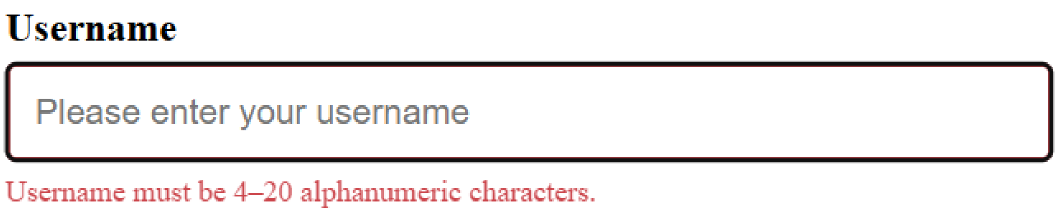
Username	Truong
Email	hothientruongd@gmail.com
Password	••••••••••••••
Confirm Password	••••••••••••••

- Case2: Invalid name


Username

ab

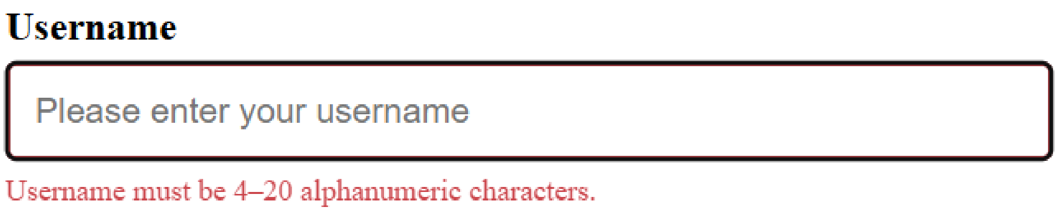
Username must be 4–20 alphanumeric characters.


Username

Please enter your username

Username must be 4–20 alphanumeric characters.

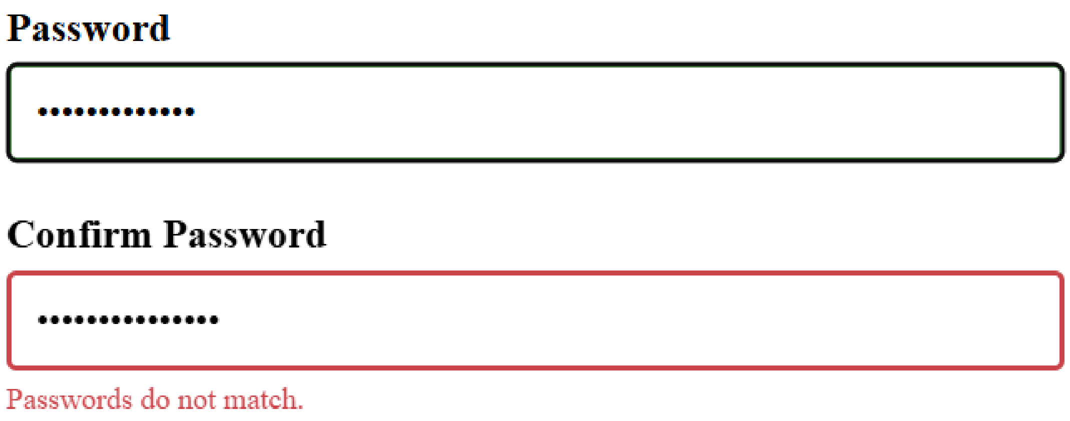
- Case3: Invalid Email:


Username

Please enter your username

Username must be 4–20 alphanumeric characters.

- Case4 : PassW and Confirm Password do not match


Password

•••••••••••••

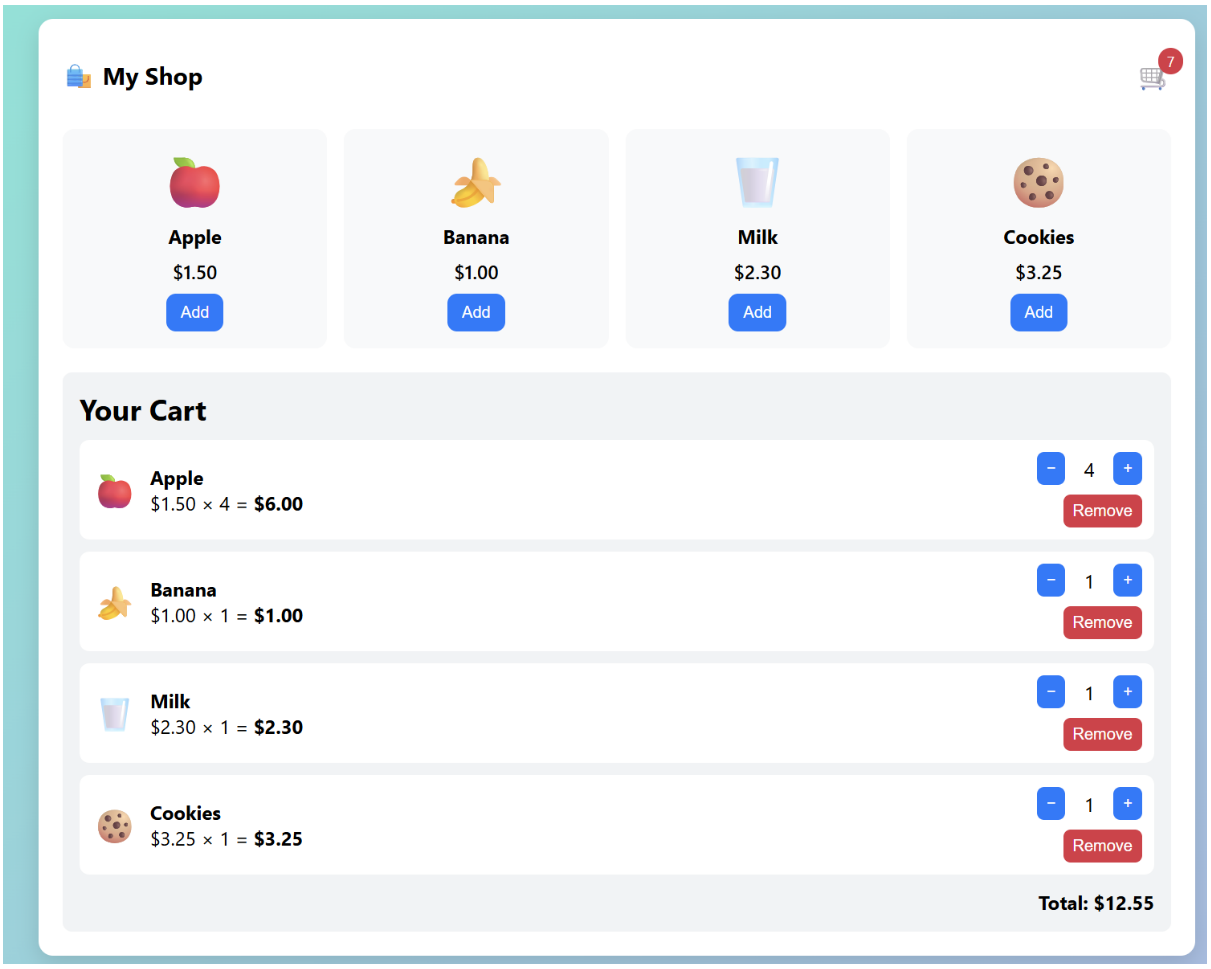
Confirm Password

•••••••••••••••

Passwords do not match.

- This is a form-validation web application in real-time, written in HTML,CSS and JavaScript. It has a user registration form containing four columns namely: username, email, password, and confirm password. The fields are verified during typing. The JavaScript enforces rules by the use of patterns: usernames have to be 4-20 letters or numbers, emails have to resemble actual emails, passwords have to contain at least one capital letter and a number and need to be at least eight characters in length. The passwords are configured to match in the so-called confirm- password field. When an entry is incorrect, a red border and an error message would be displayed below the field but when correct the border would be green and the error message would just vanish. An object maintains the record of which the fields are correct and therefore the Sign Up button only becomes effective, when all the fields are valid. The addEventListener listener input event to validateForm provides instant feedback on the form with each key press and the submit event prevents the form sending it in case any field is invalid. To put it in another way, this code is a convenient check that can be run in the browser and enhances the accuracy of the data without the use of external libraries.

Task 1.2:
- My Output


My Shop

7

Apple
$1.50
Add

Banana
$1.00
Add

Milk
$2.30
Add

Cookies
$3.25
Add

Your Cart

Apple
$1.50 × 4 = **$6.00**
− 4 +
Remove

Banana
$1.00 × 1 = **$1.00**
− 1 +
Remove

Milk
$2.30 × 1 = **$2.30**
− 1 +
Remove

Cookies
$3.25 × 1 = **$3.25**
− 1 +
Remove

Total: $12.55

- It is a web application that allows you to do online shopping within your browser. It makes use of pure HTML, CSS, and JavaScript. Four products with the emoji, name and price are displayed in a grid that is responsive to any screen size. The Add button can be found in every product updating the cart immediately. The cart displays the items you are choosing, number of each item, the price of each item and the accumulated amount. You are able to add or remove the amount or cancel things and all changes will be reflected immediately.
-

- The JavaScript script relies on individual small functions, addToCart, removeFromCart, changeQuantity, calculateTotal, renderProducts, and renderCart. Instead of having clicks all over, the code considers them in a single location, thus making the page faster and the code less messy. Both functions modify the display of the page as well as the array that maintains the current cart. The cart is stored in localStorage in the app meaning that the items remain in the local storage even when the page is revisited. It also includes ARIA roles and labels to assist the screen readers.
-
- The style uses modern CSS. The backdrop is the light blue-to-purple gradient. Boxes will be rounded and the layout will adapt to smaller screens. Provisions A media query to modify padding and column spacing on phones. It is mobile-friendly as it utilizes flexboxes and CSS grid designs.
-
- Ultimately, this code presents a bare and unadulterated front-end shopping cart, which is functional, can be accessed, and requires no external libraries.