

✓ Experiment 3: JavaScript Client-Side Validation

Objective

Add client-side form validation using JavaScript for login and registration forms.

Technologies Used

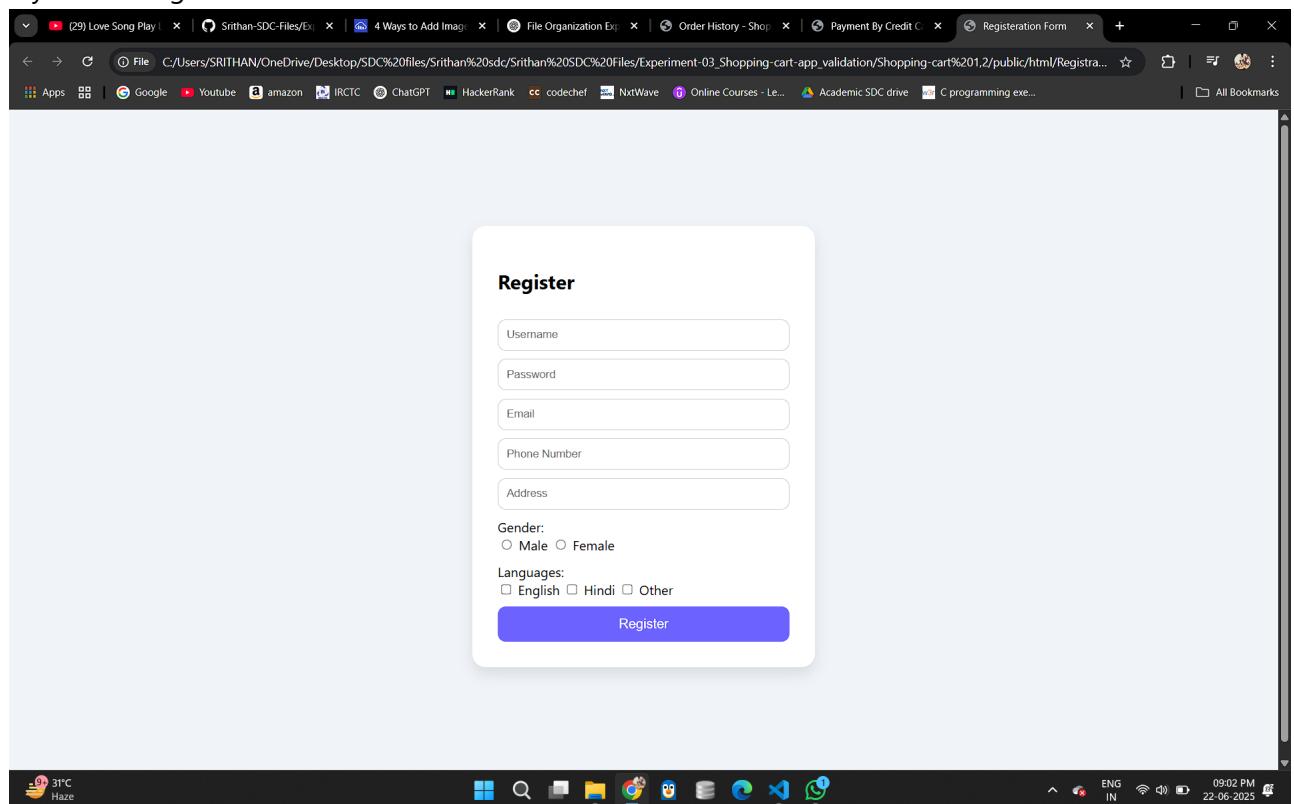
- HTML
- JavaScript (Form validation)

Features

- Email format check
- Password rules
- Real-time feedback

Steps to Execute

1. Open `index.html` in a browser.
2. Try submitting invalid forms and observe alerts.



```

1 # Experiment 3: JavaScript Client-Side Validation
11 ## Features
12 - Real-time feedback
13
14 ## Steps to Execute
15 1. Open `index.html` in a browser.
16 2. Try submitting invalid forms and observe alerts.
17 ! [alt text](./images/main_page.png)
18
19 ## Folder Contents
20 - `index.html`: Login/Register forms
21 - `validate.js`: Validation logic
22
23
24
25 Experiment 3: Shopping Cart Application (Client-Side Validation with JavaScript)
26 Folder Name (from image): Experiment-03_Shopping-cart-app_valid... (assuming it's
Experiment-03_Shopping-cart-app_validation)
27
28 Description (from document): "Use JavaScript for doing client - side validation of the pages implemented in
experiment 1 and experiment 2."
29
30 README.md for Experiment 3:
31
32 Markdown

```

git push origin main

Folder Contents

- **index.html**: Login/Register forms
- **validate.js**: Validation logic

Experiment 3: Shopping Cart Application (Client-Side Validation with JavaScript) Folder Name (from image):
Experiment-03_Shopping-cart-app_valid... (assuming it's Experiment-03_Shopping-cart-app_validation)

Description (from document): "Use JavaScript for doing client - side validation of the pages implemented in experiment 1 and experiment 2."

README.md for Experiment 3:

Markdown

Experiment 3: Shopping Cart Application (Client-Side Validation with JavaScript)

This project enhances the shopping cart application from Experiments 1 and 2 by implementing robust **client-side form validation using JavaScript**. The objective is to prevent invalid data submissions to the server, provide immediate feedback to users, and improve the overall user experience on pages such as Registration and Login.

Features

- **Client-Side Validation**: All form inputs (e.g., username, password, email, quantities) on the Registration and Login pages are validated directly in the browser using JavaScript before submission.

- **Immediate Feedback:** Users receive instant visual cues (e.g., error messages, changing border colors) if their input does not meet the required criteria.
- **Common Validation Rules:**
 - **Required Fields:** Ensures essential fields are not left empty.
 - **Minimum/Maximum Length:** Checks string lengths for fields like username and password.
 - **Email Format:** Validates email addresses using regular expressions.
 - **Password Confirmation:** Ensures password and confirm password fields match.
 - **Numeric Input:** Validates age, quantity, etc., are valid numbers.
 - **Custom Rules:** (Optional) More specific rules as needed (e.g., username cannot contain special characters).
- **Improved User Experience:** Reduces server load and provides a smoother interaction by catching errors early.
- **Integration with Existing UI:** Validation seamlessly integrates with the CSS3/Flexbox/Grid (from Exp 1) or Bootstrap (from Exp 2) styled forms.

Technologies Used

- HTML5
- CSS3 (from Exp 1 or Exp 2)
- JavaScript (ES5/ES6 for validation logic)
- (Optional: Bootstrap, if integrating with Exp 2's UI)

Setup and Running

1. Clone the Repository (or download the project files):

```
git clone [https://github.com/your-username/Experiment-03_Shopping-cart-app_validation.git](https://github.com/your-username/Experiment-03_Shopping-cart-app_validation.git)
cd Experiment-03_Shopping-cart-app_validation
```

2. Review Dependencies (if any):

- Ensure the HTML files correctly link to the JavaScript validation file.
- If building on Experiment 2, ensure Bootstrap CSS/JS are still linked.

3. Open in Browser:

- Navigate to the project directory.
- Open `registration.html` or `login.html` (the pages containing forms for validation) in your preferred web browser.
- **No server is required** as this is a purely frontend application.

Project Structure

```
. └── css/ | └── style.css # CSS from Exp 1/2, possibly with validation styles └── js/ | └── validation.js # Core JavaScript file for client-side validation logic └── images/ # Optional: directory for product images, logos etc. └── registration.html # Registration page with validation └── login.html # Login page with validation └── catalog.html # (May or may not have validation, depending on forms) └── cart.html # (May or may not have validation, depending on forms) └── README.md
```

