

FSD Assignment - 2

Aim: Develop a web application using javascript to implement sessions, cookies, DOM. Perform validation such as checking for emptiness, only numbers for phone number, special character requirement for password, regular expressions for certain format of the field etc. Use the my SQL database.

Objective:

- 1) To understand what form validation is
- 2) To learn basic functioning of DOM objects.
- 3) To learn how to apply various techniques to implement it
- 4) To learn how to apply various techniques to implement it.

Theory :

1) Explain the role of regular expressions. Why are they a suitable tool for validating data formats like a phone number or checking for the presence of specific characters in a password?

Regular expressions (regex) are sequences of characters that define a search pattern. They are a powerful tool for validating data from formats because they provide a concise and flexible way to match and manipulate strings based on specific rules.

2) Suitability for validation:

Phone number validation: A regex can ensure that a phone number contains only numeric value, has a specific length and optionally follows a certain format (eg: (xxx) xxx-xxx-xx)

of the IS manager

Password validation: Regex can enforce rules like minimum length, presence of uppercase letters, lowercase letters, digits and special characters and exclusion of certain patterns.

Pattern Matching: They allow for complex pattern matching, making it easy to define rules for various data types.

Q) Explain fundamental difference b/w session and a cookie in the context of web application development. How do they work together to maintain a user's logged-in state?

Fundamental difference:

Session

A session is a server-side mechanism to store user-specific data for a limited time, typically tied to a unique ID; it's more secure, as data is not directly exposed on the client side.

→ Working together to maintain logged-in state:
When a user logs in, the server creates a session for that user and generates a unique session ID. This session ID is then sent to the client, typically stored in a cookie. In subsequent requests, the browser sends this cookie back to the server.

Cookie

A cookie is a small piece of data stored on the client-side (user's browser) by the website. It's primarily used to remember stateful information or user preferences across multiple requests.

3) What is the purpose of performing both client-side and server-side validation? Describe a scenario where relying solely on client-side validation could lead to a security vulnerability:

→ Purpose:

Client-side validation:

Provides immediate feedback to the user, improving user experience and reducing server load by catching simple errors before submission.

Server-side validation:

Essential for security and data integrity, as it cannot be bypassed by malicious users and ensures that data conforms to business rules before being stored in the database.

→ Scenario:

If a web application relies solely on client-side validation for a form field like price input, a malicious user could bypass this validation and submit a negative or extremely high price value directly to the server, potentially leading to incorrect calculations, financial discrepancies or even system crashes.

4) Provide a simple example of how a JavaServer can interact with the DOM to dynamically change the content of a web page after a user action such as a form submission.

```
<!DOCTYPE html>
<html>
<head>
<title> DOM Example </title>
</head>
<body>
<p id="greeting">Hello, User!</p>
<button onclick="changeText()">Change Greeting</button>
<script>
function changeText() {
    greetingElement.innerHTML = "Welcome back, Java";
}
</script>
</body>
</html>
```

- 5) Give 8 steps for connectivity from Frontend using HTML CSS JS to MySQL.
- 1) HTML/CSS/JS (Frontend): The frontend is responsible for user interface and interaction. JavaScript handles user input, client-side validation, and making requests to the backend.
 - 2) AJAX/Fetch API: JavaScript uses technologies like AJAX or the Fetch API to send HTTP requests to a backend server.
 - 3) Backend Server (e.g. Node.js, Python, PHP): The backend server receives the HTTP request from frontend. It processes the request, performs server-side validation and interacts with the database.
 - 4) Database Driver/ORM: The backend uses a database driver or an Object Relational Mapper (ORM) library (e.g.: mysql package in Node.js, SQLAlchemy in Python) to establish a connection with MySQL database.
 - 5) SQL Queries: The backend constructs and executes SQL queries (e.g: INSERT, SELECT, UPDATE, DELETE) to interact with MySQL database.
 - 6) Data Retrieval/Storage: Data is retrieved from or stored in MySQL database based on MySQL query.

State is managed within component itself

7) Response to Frontend: The backend sends a response back to the frontend which then updates the UI accordingly.

#FAQ's:

Q1) Write 3 reasons why form validation are important.

1) Data Integrity:

Ensures that the data submitted by user is in the correct format and meets specified criteria, preventing invalid or incomplete data from being stored.

2) User Experience:

Provides immediate feedback to user about incorrect input, guiding them to correct errors and improving overall usability of the form.

3) Security:

Helps prevent malicious input by sanitizing and validating data before it reaches the server and database.

Q2) Give an example of how to modify an attribute value using DOM?

HTML DOM element `setAttribute()`

1) Example: Add a `class` attribute to an element's element.

```
setAttribute("class", "democlass");
```

2) Change an input field to an input button: `myInput`.

```
setAttribute("type", "button");
```

3) Add a href attribute to an `a` element: `myAnchor`.

```
setAttribute("href", "https://www.w3schools.com");
```

What are different features of Javascript?

Javascripts offer a wide array of features that contributes to its versatility and widespread used in web development and beyond.

Key features include:

→ Client-side Scripting:

Javascript primarily runs in the user's browser enabling interactive and dynamic web page without constant server communication. This lead to faster response times and a more fluid user experience

→ Dynamic Typing:

variables in Javascript are dynamically typed meaning their data type is determined at runtime rather than being declared explicitly. This offers flexibility in coding.

→ Object-Oriented Capabilities:

Javascript is a prototype-based object oriented language. It supports object creation, inheritance through prototypes and the use of objects as first-class citizens

(M)
signature