

ЛАБОРАТОРНА РОБОТА №9

Тема: Робота з формами.

Мета: навчитися обробляти форми, валідувати їх та відправляти без перезавантаження сторінки за допомогою JavaScript.

Вимоги та самостійна робота:

- [Властивості та методи форми](#)
- [FormData](#)
- [Validating forms using JavaScript](#)
- [Sending forms through JavaScript](#)
- [JavaScript Validation](#)
- [Регулярні вирази](#)
- [Regular expressions](#)
- [Regular expression tester](#)

Практична частина

Форми — основа вебу. Грубо кажучи, типовий застосунок здебільшого складається із форм та списків.

Форми повинні бути:

- зручними: користувач чітко розуміє, яку дію має зробити, чи що ввести. Якщо ввід користувача неправильний, це повинно бути повідомлено в зрозумілій формі. Поля вводу і форма повинні реагувати на дії користувача надаючи чітке розуміння, що відбувається, і що потрібно зробити (чи поле активне, сфокусоване, чи очікується введення даних тощо).
- доступними: користувач часто заповнює форму не на комп'ютері, а на мобільному пристрої, не за допомогою мишки, а за допомогою клавіатури замість неї, іноді з допомогою зчитувача екрану. Сучасні веб-застосунки повинні бути зручними в усіх перерахованих випадках.
- гарними і функціональними: стандартний вигляд полів вводу може сильно відрізнятися від дизайну застосунку. Їх стилізація деяких полів може бути доволі складна, оскільки потрібно повністю зберігати функціональність.

Необхідно реалізувати форму для авторизації користувачів та їх реєстрації, яка буде відображатися у вигляді табів на вебсторінці.

Форма реєстрації має наступні поля:

- First Name (обов'язкове, повинно бути менше 3-х та не більше 15-ти символів).
- Last Name (обов'язкове, повинно бути менше 3-х та не більше 15-ти символів).
- Email (обов'язкове, повинно бути валідно відповідно до шаблону мейлу @._ (реалізувати через regular expression).
- Password (обов'язкове, повинно бути не менше 6-ти символів).
- Confirm password (обов'язкове, повинно збігатися з полем Password). Поле Password та Confirm password містить іконку у вигляді ока, клік по якій дозволяє користувачеві показати/приховати пароль який він вводить).
- Phone (обов'язкове поле, повинно бути валідно відповідно до шаблону мобільного оператора України +380....).
- Date Birth (обов'язкове поле, якщо користувачу менше 12 років, вивести відповідне повідомлення, що він не зможе зареєструватися, дата народження не може бути у майбутньому).
- Sex (male, female) (обов'язкове поле).
- Country (обов'язкове поле, випадаючий список країн).
- City (обов'язкове поле, випадаючий список міст відповідної країни. Поле не активне, поки поле Country не обране).

Форма авторизації має наступні поля:

- Username (обов'язкове поле).
- Password (обов'язкове поле, повинно бути не менше 6-ти символів). Поле Password містить іконку у вигляді ока, клік по якій дозволяє користувачеві показати/приховати пароль який він вводить).
- Remember Me (true, false) (не обов'язкове поле, використовується для запам'ятовування сесії користувача).

Технічні вимоги:

- HTML, CSS, JavaScript.
- Адаптивність.
- Розробити привабливий та інтуїтивно зрозумілий дизайн форми.
- Використовувати яскраві та привабливі елементи дизайну для полегшення сприйняття користувачем.
- Обов'язкова валідація усіх полів форм з використанням JavaScript.
- Якщо поле валідне - додати йому зелений border, якщо не валідне - червоний. Показувати текстову помилку користувачеві. (див. рисунок для прикладу нижче).
- Обидві форми повинні валідуватися по сабміту форми.

- Сторінка під час відправлення даних не має перезавантажуватися;
- У разі успішного відправлення - показати користувачеві, що його успішно зареєстровано та очистити форму.

Приблизний вигляд форми й помилок

The image displays two user interface mockups for a registration and login system, followed by a detailed view of form validation errors.

Mockup 1: Create an account

- Buttons: Signup, Login
- Title: Create an account
- Fields: Username (Enter username), Email (Enter email), Password (Enter password with eye icon), Confirm Password (Enter password with eye icon)
- Button: Signup

Mockup 2: Login here

- Buttons: Signup, Login
- Title: Login here
- Fields: Username (Enter username), Password (Enter password)
- Button: Login

Form Validation Examples

Example 1: Successful Input

- First name: Mark (Looks good! ✓)
- Last name: Otto (Looks good! ✓)
- Username: @ (Please choose a username. ❗)
- City: (Please provide a valid city. ❗)
- State: Choose... (Please select a valid state. ❗)
- Zip: (Please provide a valid zip. ❗)
- Agree to terms and conditions: ☐ (You must agree before submitting. ❗)
- Button: Submit form

Example 2: Various Errors

- First name: Mark (Looks good! ✓)
- Last name: Otto (Looks good! ✓)
- Username: @ (Please choose a username. ❗)
- City: (Please provide a valid city. ❗)
- State: Choose... (Please select a valid state. ❗)
- Zip: (Please provide a valid zip. ❗)
- Agree to terms and conditions: ☐ (You must agree before submitting. ❗)
- Button: Submit form

Контрольні запитання:

1. Що таке властивість "action" форми? Як вона використовується?
2. Яка роль властивості "method" в формі?
3. Що таке метод "submit()" для форми і як його викликати?
4. Як використовувати метод "reset()" для очищення значень в формі?
5. Яка різниця між атрибутами й властивостями форми в JavaScript?

6. Як можна перевірити введені дані в форму перед її відправленням?
7. Як можна створити форму динамічно за допомогою JavaScript?
8. Що таке об'єкт FormData в JavaScript?
9. Які переваги має використання об'єкта FormData порівняно з традиційним методом отримання даних форми?
10. Як додати дані до об'єкта FormData за допомогою JavaScript?
11. Які методи доступні для роботи з об'єктом FormData?
12. Як відправити дані з об'єктом FormData на сервер за допомогою AJAX-запиту?
13. Як перебрати всі пари ключ-значення в об'єкті FormData?