

Лабораторна робота №6

Тема: Асинхронні запити на сервер та їх обробка

Мета роботи: ознайомитися з асинхронними запитами на сервер та їх обробкою

Завдання на лабораторну роботу

Завдання 0: Ретельно вивчити теорію за посиланням:

[PHP and JSON](#)

 [Лекція. Асинхронні запити на сервер та їх обробка](#)

Завдання 1: Система керування користувачами

Опис завдання: Створіть веб-додаток для системи керування користувачами, який використовує асинхронні запити для взаємодії між клієнтом та сервером. У цьому завданні передбачено дві основні функціональності: реєстрація нового користувача та перегляд списку користувачів.

1. Реєстрація користувача (фронтенд):

- Створіть HTML-форму для реєстрації нового користувача з полями для ім'я, електронної адреси та пароля.
- Використовуйте JavaScript для збору даних з форми та відправки їх на сервер за допомогою fetch API.
- Перевірте, чи всі поля заповнені перед відправкою на сервер.

2. Реєстрація користувача (бекенд):

- Створіть PHP-скрипт для обробки запиту на реєстрацію нового користувача.
- Перевірте правильність введених даних, такі як унікальність електронної адреси та мінімальна довжина пароля.
- Збережіть дані нового користувача в базі даних MySQL

3. Отримання списку користувачів (фронтенд):

- Створіть кнопку або посилання для відображення списку користувачів.
- Використовуйте JavaScript для відправки запиту на сервер для отримання списку користувачів.
- Виведіть отриманий список користувачів на веб-сторінці.

4. Отримання списку користувачів (бекенд):

- Створіть PHP-скрипт, який вибирає всіх користувачів з бази даних.
- Поверніть список користувачів у форматі JSON, щоб його можна було легко обробити на клієнті.
- Додайте можливість входу користувача за допомогою електронної адреси та пароля.
- Реалізуйте можливість видалення користувача.
- Додайте можливість редагування даних користувача.

Завдання 2: Реалізація заміток (Notes App) за допомогою FETCH та PHP

Опис завдання: Створіть веб-додаток, який дозволяє користувачам додавати, переглядати, оновлювати та видаляти замітки. Користувачі можуть створювати замітки, вказуючи заголовок та текст. Всі замітки повинні зберігатися на сервері. Взаємодія між клієнтом та сервером повинна здійснюватися за допомогою AJAX-запитів та PHP на серверній стороні.

Ключові вимоги:

1. **Додавання заміток:** Користувач повинен мати можливість додавати нові замітки, вказуючи заголовок та текст, та надсилати їх на сервер за допомогою асинхронних запитів.
2. **Перегляд заміток:** Усі замітки, збережені на сервері, повинні відображатися на головній сторінці. Замітки повинні бути відсортовані за датою створення (новіші зверху).
3. **Оновлення заміток:** Користувач повинен мати можливість оновлювати замітки, змінюючи їх заголовок або текст. Зміни повинні відображатися на сервері після натискання відповідної кнопки або події.
4. **Видалення заміток:** Користувач повинен мати можливість видаляти замітки за допомогою відповідної опції або кнопки. Видалення заміток повинно бути асинхронним, без перезавантаження сторінки.
5. **Валідація даних на клієнті та сервері:** Перед надсиланням даних на сервер, вони повинні бути перевірені на клієнті (заголовок та текст мають бути заповнені) та сервері (захист від SQL-ін'єкцій та інших атак).
6. **Відправка та отримання даних за допомогою fetch API:** Взаємодія з сервером повинна здійснюватися за допомогою fetch API на клієнтській стороні для відправки та отримання даних.
7. **Зберігання даних на сервері з використанням PHP:** Замітки повинні зберігатися на сервері за допомогою PHP. Використовуйте MySQL або іншу базу даних для зберігання інформації про замітки.
8. **Асинхронність та безперервність додатку:** Усі операції (додавання, оновлення, видалення заміток) повинні відбуватися без перезавантаження сторінки, з використанням асинхронних запитів.

Надайте доступ до репозиторію Вашим викладачам.