# Лабораторна робота №3. Передача та обробка параметрів форми.

**Мета:** навчитися створювати HTML-форми, передаватися параметри методами GET та POST, обробляти отримані параметри на стороні сервера.

**Теоретичні відомості**

В класичному програмуванні, зазвичай, програма отримує від користувача набор вхідних даних або параметрів в залежності від яких алгоритм програми формує певний результат. Робота сценаріїв на стороні веб-сервера теж доволі часто залежить від параметрів, що передає користувач. Правила передачі вхідних параметрів та отримання результату к даному випадку регламентовані протоколом HTTP.

Обмін повідомленнями між клієнтом та сервером відбувається за схемою

«запит-відповідь» при цьому передається певна кількість службових даних (наприклад [кодування](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) веб-сторінки й/або запитати версію сторінки в потрібних мові/кодуванні і таке інше.) та, при необхідності, дані користувача.

Відправка параметрів відбувається двома різними методами.

**Метод GET** – передача параметрів виконуються безпосередньо у рядку URI (наприклад: [http://www.example.net/resource?param1=value1&param2=](http://www.example.net/resource?param1=value1&param2) value2). Згідно зі стандартом HTTP, запити типу GET вважаються *ідемпотентними* — багатократне повторення одного і того ж запиту GET повинне приводити до однакових результатів (за умови, що сам ресурс не змінився за час між запитами). Це дозволяє кешувати відповіді на запити GET.

**Метод POST** – передає призначені для користувача дані включеними в тіло запиту. На відміну від методу GET, метод POST не відображається у рядку запиту браузера та не вважається ідемпотентним, тобто багатократне повторення одних і тих же запитів POST може повертати різні результати.

**Веб-форма (форма)** — елемент [веб-сторінки](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0), який дає користувачам можливість вводити інформацію і відправляти її на [сервер](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80) для подальшої обробки.

Форма задається за допомогою тегу form, усередині якого розташовуються елементи керування. Крім загальних для HTML атрибутів , в form можуть бути присутніми наступні:

* + *action* – дія, обов'язковий атрибут , що містить URI обробника форми;
  + *method* – метод відправки форми, атрибут, який приймає значення GET (за замовчуванням) або POST;
  + *enctype* – тип кодування для вмісту;
  + *accept* – список MIME - типів для завантаження файлів;
  + *name* – ім'я форми;
  + *onsubmit* – обробник події «форма відправлена»;
  + *onreset* – обробник події «форма очищена»;
  + *accept - charset* список підтримуваних наборів символів.

У HTML визначені наступні елементи керування:

* + *input* наступних типів:
    - *submit*
    - *image*
    - *reset,*
    - *button*
    - *checkbox*
    - *radio*
    - *password*
    - *hidden*
    - *file*
  + *button;*
  + *textarea;*
  + *select* з елементами *optgroup* і *option* всередині;
  + *label.*

У HTML5 визначено деякі додаткові елементи управляння але не всі вони є кросбраузерними.

У мові програмування PHP не треба піклуватися про отримання, розкодування і будь-яку іншу обробку даних прийнятих від користувача. PHP автоматично заповнює ряд вбудованих масивів, які ще називають суперглобальними. Їх імена та призначення наступні:

*$\_SERVER* – серверні змінні;

$\_*ENV* – змінні середовища, в якій працює PHP;

$\_*COOKIE* – змінні, що передаються за допомогою cookies;

$\_*GET* – параметри, що передані за допомогою методу GET;

$\_*POST* – параметри, що переданих методом POST;

$\_*FILES* – дані про файли, що передані за допомогою методу POST;

$\_*REQUEST* – містить усередині себе масиви $\_*GET*, $\_*POST* та

$\_*COOKIE*;

$\_*SESSION* – зберігає параметрів сесії.

Таким чином для обробки даних переданих користувачем зазвичай використовують масиви $\_*GET*, $\_*POST* та $\_*REQUEST*.

Для виконання перегляду значень змінних будь яких типів зручно використовувати допоміжні функції PHP *print\_r()* та *var\_dump().*

Наприклад, якщо маємо запит «<http://name.labs/test.php?a=10&b=hello>», то для перегляду прийнятих параметрів у скрипті test.php необхідно написати:

<?php

echo “<pre>”;

print\_r($\_GET); echo “</pre>”;

?>

**Завдання**

1. Розробити скрипт *form.php,* що відображає HTML-форму профілю користувача, що задовольняє наступним вимогам:
   * форма повинна містити всі основні елементи керування;
   * обов'язково повинне буті питання, що дає можливість вибрати один варіант відповіді з декількох запропонованих;
   * дані на сервер повинні буті передані одним з методів – GET або POST.
2. Оформити макет форми (рис. 3.1) використовуючи CSS framework – Bootstrap.

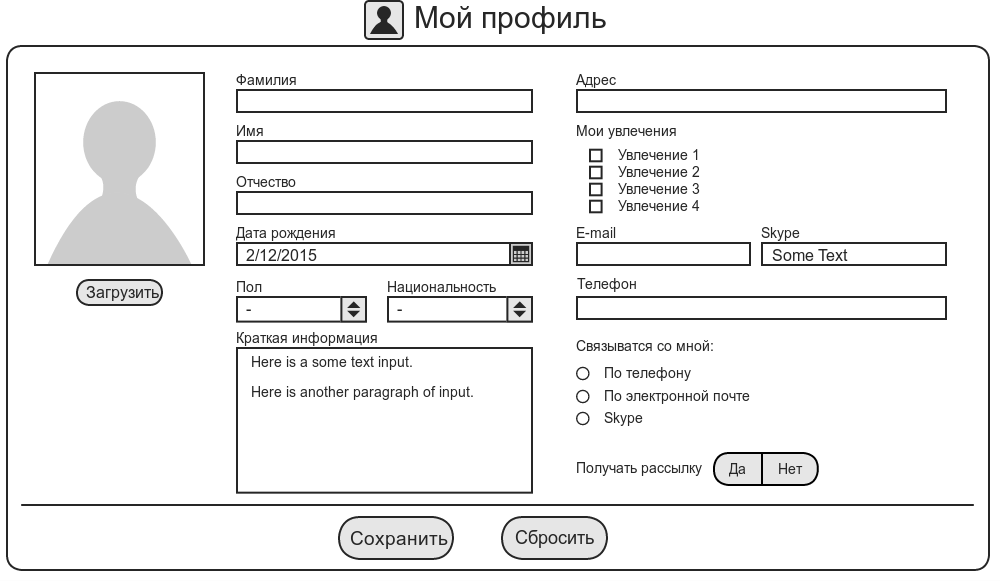


Рисунок 3.1 – Макет форми.

1. Передбачити можливість відображення форми на пристроях з різною роздільною здатністю.
2. При відображенні на пристроях шириною менших 720px поля типу textarea суховати.
3. Всі поля форми повинні мати перевірку на не пусте значення та відповідність формату вводу даних.
4. Поля другої колонки перевірити на відповідність формату використовуючи засоби HTML5 та JavaScript.
5. Поля третьої колонки перевірити на відповідність формату використовуючи PHP та у разі помилки відобразити відповідне повідомлення (використати відповідний компонент Bootstrap) не втративши даних, що вже заповнив користувач.
6. При завантаженні фото використати JavaScript об’єкт FileReader та показати фото профілю до відправки на сервер.
7. Відправити форму скрипту form.php та зберегти у файлі (самостійно обрати формат).
8. Передбачити неможливість повторного відправлення форми при оновленні сторінки (виконати перенаправлення сторінки).
9. При відкритті форми відобразити збережені у файлі дані.
10. Отриману веб-сторінки та пов'язані з нею дані (папка з картинками, файл стилів) скопіювати в папку «lab3» та розмістити її в кореневій директорій веб-сервера.
11. Протестувати роботу сторінок на віддаленому сервері.
12. Оформити звіт про виконання лабораторної роботи.

Додатково: Виконати обрізку та масштабування завантаженого фото перед збереженням.

**Контрольні питання**

1. Який html тег відповідає формі?
2. Які функції php використовуються при роботі з файлами?
3. Які типи передачі параметрів серверу існують?
4. Яким чином php – скрипт отримує доступ до переданих серверу параметрів?