

1. ЛАБОРАТОРНІ РОБОТИ

1.1 Лабораторна робота 1. Робота з формами в PHP.

Завдання: Створити форму реєстрації на сервері з перевіркою введених даних на коректність.

Приклад виконання аналогічного завдання [8].

Початкова форма показана на Рис.58.

PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male *

Your Input:

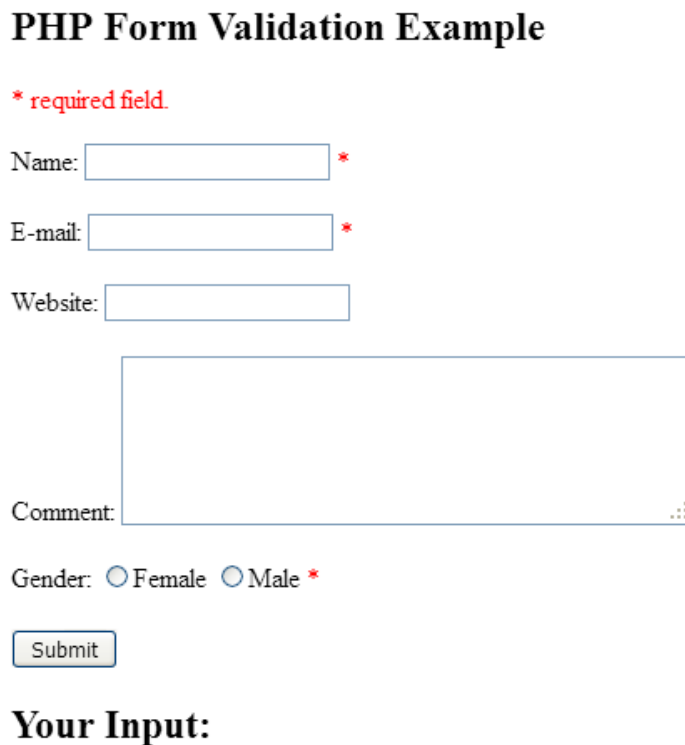


Рис.58. Початкова форма

Обробку форми виконує наступний скрипт:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
```

```

$name = test_input($_POST["name"]);
// check if name only contains letters and whitespace
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
    $nameErr = "Only letters and white space allowed";
}
}
if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = test_input($_POST["email"]);
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $emailErr = "Invalid email format";
    }
}
if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid
    if (!preg_match("/\b(?:https?|ftp):\/\/[www\.]?[-a-z0-9+&@#\/%?~_!:\.~]*[-a-z0-9+&@#\/%?~_]/i",$website)) {
        $websiteErr = "Invalid URL";
    }
}
if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $comment = test_input($_POST["comment"]);
}
if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = test_input($_POST["gender"]);
}
}

```

```

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

```

<h2>PHP Form Validation Example</h2>

```

<p><span class="error">* required field.</span></p>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    Name: <input type="text" name="name">
    <span class="error">* <?php echo $nameErr;?></span>
    <br><br>
    E-mail: <input type="text" name="email">

```

```

<span class="error">* <?php echo $emailErr;?></span>
<br><br>
Website: <input type="text" name="website">
<span class="error"><?php echo $websiteErr;?></span>
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<span class="error">* <?php echo $genderErr;?></span>
<br><br>
<input type="submit" name="submit" value="Submit">
</form>

<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>

</body>
</html>

```

1.2.Лабораторна робота 2. Використання MySQL за допомогою PHP.

Завдання: Модифікувати лабораторну роботу 1, додавши збереження даних реєстрації (профіля користувача) в базі даних MySQL. Необхідно забезпечити можливість перегляду й редагування профілів.

Методичні вказівки.

Треба запустити phpMyadmin, створити базу даних users з полями, що відповідають полям реєстраційної форми. Далі при написанні скрипта для обробки форми вибрати спосіб спілкування з MySQL (бібліотеки mysqli або PDO) і використати матеріали лекції 5.

1.3.Лабораторна робота 3. Простий сайт з авторизацією.

Завдання: Використовуючи результати лабораторних робіт 1 і 2, створити сайт, що складається з п'яти сторінок:

- 1) Головна сторінка, що містить форму входу в систему.
- 2) Сторінка перегляду або зміни персональних даних.
- 3) Сторінка реєстрації.

- 4) Гостьова книга.
- 5) Сторінка "Про сайт".

У профілі користувача повинне бути присутнім поле "Фотографія", через яке користувач завантажує на сервер свою аватарку, яка відображається біля повідомлень даного користувача в гостьовій книзі.

Ідентифікаційну інформацію про користувача потрібно зберігати в cookie.

Пароль у базі даних зберігається в хешованому вигляді.

Гостьова книга повинна бути доступна тільки аутентифікованим користувачам.

Сайт повинен бути організований із простим поділом дизайну й змісту (верхній блок, основний зміст, нижній блок).

Всі часто використовувані функції повинні бути винесені в окремий файл із розширенням inc, який сторінки будуть підключати за допомогою команди require_once.

1.4 Лабораторна робота 4. Поліпшений сайт.

Завдання: До сайту, розробленому в лабораторній роботі 3 необхідно додати: 1) шаблон дизайну сайту; 2) роль адміністратора в профілі користувача; 3) модерування гостьової книги адміністратором сайту (видалення окремих повідомлень, видалення й блокування нових повідомлень порушника, фільтрація по IP адресі).

1.5 Лабораторна робота 5. Розробка CMS для адміністрування сайту.

Завдання: Додати до сайту систему керування вмістом, доступну тільки користувачам, що мають роль адміністратора.

CMS повинна забезпечити:

1. Можливість вибору шаблону дизайну сайту.
2. Редагування динамічної зміни тексту сторінок (в TextArea або вбудова-ному HTML редакторі, наприклад, CKEditor), які зберігаються в базі даних.
3. Систему статей (або новин) з їхнім редагуванням в адмінській частині та зберіганням у базі даних.
4. Адміністрування користувачів (видалення, зміна пароля).
5. Перегляд і оновлення статистики відвідувань сайту.