

EDUCATION



# ΘΗΨ2

## HTML Forms & PHP Scripts

## Forms

- Γενικά οι φόρμες είναι στοιχεία που χρησιμοποιούμε για να επικοινωνεί ο επισκέπτης της ιστοσελίδας με τον εξυπηρετητή (Server).

**Είσοδος Μέλους**

Your Email:

Your Password :

Your Name :

- Συνήθως σε αυτήν περιλαμβάνονται διαφόρων τύπων πεδία εισαγωγής στοιχείων , βοηθητικό κείμενο και πλήκτρα ελέγχου

# EDUCATION

## Forms

- Γενική μορφή ετικέτας form στην Html5:

```
<form>  
  First name:<br>  
  <input type="text" name="firstname"><br>  
  Last name:<br>  
  <input type="text" name="lastname"><br>  
  <input type="radio" name="gender" value="male"  
    checked> Male <br>  
  <input type="radio" name="gender" value="female">  
  Female<br>  
</form>
```

First name:  
  
Last name:  
  
 Male  
 Female

# Διάφοροι τύποι πεδίων στοιχείου input

input type="text" - "Email" - "password" - "submit" - "reset" "number"

First name (text):

Last name (text):

E-mail:

Quantity (between 1 and 5):

User password:

Υποβολή  
καθάρισμα

```
65 <form id="formal">
66   First name (text):<br>
67   <input type="text" name="firstname" size="50"><br>
68   Last name (text):<br>
69   <input type="text" name="lastname"> <br>
70   E-mail:<br>
71   <input type="email" name="email"><br>
72   Quantity (between 1 and 5):<br>
73   <input type="number" name="quantity" min="1" max="5"><br>
74   User password:<br>
75   <input type="password" name="psw"><br>
76   <input type="submit" value="Υποβολή"><br>
77   <input type="reset" value="καθάρισμα"><br>
78 </form>
```

# Διάφοροι τύποι πεδίων στοιχείου input

input type="radio" - "checkbox" - "button"

☑ Male  
☐ Female  
☐ Other

Τα εξεταζόμενα μαθήματα είναι:

☐ Μαθηματικά  
☐ Ιστοσελίδα  
☐ Προγραμματισμός  
☐ Ελληνικά  
☐ Φυσική  
☐ Ιστορία

Button Click Me!



```
82 <form id="forma2">
83   <input type="radio" name="gender" value="male" checked> Male<br>
84   <input type="radio" name="gender" value="female"> Female<br>
85   <input type="radio" name="gender" value="other"> Other<br>
86   <br>
87   <h3>Τα εξεταζόμενα μαθήματα είναι:</h3>
88   <input type="checkbox" name="lesson1" value="maths">Μαθηματικά<br>
89   <input type="checkbox" name="lesson2" value="web">Ιστοσελίδα<br>
90   <input type="checkbox" name="lesson3" value="prog">Προγραμματισμός<br>
91   <input type="checkbox" name="lesson4" value="greek">Ελληνικά<br>
92   <input type="checkbox" name="lesson5" value="phisics">φυσική<br>
93   <input type="checkbox" name="lesson6" value="history">Ιστορία<br>
94   <br>
95   <input type="button" onclick="alert('Επιλέξτε τέσσερα εξεταζόμενα μαθήματα!')" value="Button Click Me!">
96 </form>
```

# Διάφοροι τύποι πεδίων στοιχείου input

input type="color" - "date" - "month" - "number" - "range"

Select your favorite color:

Birthday:

Enter a date before 1980-01-01:

Enter a date after 2000-01-01:

Birthday (month and year):

Quantity (between 1 and 5):

Βαθμολογία απο 1 εως 10:

```
<h2>input type="color" - "date" - "month" - "number" - "range"</h2>
```

```
<form id="forma3">
```

```
Select your favorite color:
```

```
<input type="color" name="favcolor"> <br>
```

```
Birthday:
```

```
<input type="date" name="bday"> <br>
```

```
<br>
```

```
Enter a date before 1980-01-01:
```

```
<input type="date" name="bday" max="1979-12-31"><br>
```

```
Enter a date after 2000-01-01:
```

```
<input type="date" name="bday" min="2000-01-02"><br>
```

```
Birthday (month and year):
```

```
<input type="month" name="bdaymonth"><br>
```

```
<br>
```

```
Quantity (between 1 and 5):
```

```
<input type="number" name="quantity" min="1" max="5"><br>
```

```
<br>
```

```
Βαθμολογία απο 1 εως 10:
```

```
<input type="range" name="points" min="0" max="10"><br>
```

```
</form>
```



# Στοιχείο Select



```
<h2>Select</h2>
<form id="forma4">
<select name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab" selected>Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
</form>
```

```
<select name="cars" size="4" multiple>
```

## Στοιχείο List



```
<h2>List - Datalist</h2>
<form id="Forma5" action="/action_page.php">
  <input list="browsers">
    <datalist id="browsers">
      <option value="Internet Explorer">
      <option value="Firefox">
      <option value="Chrome">
      <option value="Opera">
      <option value="Safari">
    </datalist>
  </form>
```

## Στοιχείο Button

### Button

Click Me!



```
<h2>Button</h2>
```

```
<button type="button" onclick="alert('Καλά Ευπνητούρια!')">Click Me!</button>
```

This page says

Καλά Ευπνητούρια!

OK



## Άσκηση 1

- Να δημιουργήσετε ένα αρχείο html το οποίο να εμφανίζει την φόρμα εισαγωγής στοιχείων του διπλανού σχήματος.

QUESTIONS    RESPONSES

### Contact information

Form description

Name \*

Short answer text

Email \*

Short answer text

Address \*

Long answer text

Phone number

Short answer text

Comments

Long answer text



## Άσκηση 2

- Να δημιουργήσετε ένα αρχείο html το οποίο να εμφανίζει την φόρμα εισαγωγής στοιχείων του διπλανού σχήματος.

### Customer Feedback

We would love to hear your thoughts or feedback on how we can improve your experience!

#### Feedback Type

- Comments
- Questions
- Bug Reports
- Feature Request

#### Feedback \*

Long answer text

#### Suggestions for improvement

Long answer text

#### Name

Short answer text

#### Email

Short answer text

# Forms

- Παράμετροι Action και Target :

```
<form action="/action_page.php"  
      target="_blank">
```

**action:** Ορίζει την ενέργεια που πρέπει να εκτελεστεί όταν υποβληθεί το έντυπο. Στο παράδειγμα, τα δεδομένα φόρμας αποστέλλονται σε μια σελίδα του διακομιστή που ονομάζεται "/action\_page.php". Αυτή η σελίδα περιέχει ένα σενάριο πλευράς διακομιστή που χειρίζεται τα δεδομένα φόρμας:

**Target:** Καθορίζει εάν το αποτέλεσμα που θα υποβληθεί θα ανοίξει σε μια νέα καρτέλα προγράμματος περιήγησης, ένα πλαίσιο ή στο τρέχον παράθυρο. Η προεπιλεγμένη τιμή είναι "\_self". Άλλες επιλογές : "\_blank", "\_parent", "\_top"

# Forms

Παράμετρος Method:

```
<form action="/action_page.php"  
      method="get">
```

```
<form action="/action_page.php"  
      method="post">
```

**Method** : Καθορίζει τη μέθοδο HTTP (GET ή POST) που θα χρησιμοποιηθεί κατά την υποβολή των δεδομένων φόρμας. Η προεπιλεγμένη μέθοδος κατά την υποβολή δεδομένων φόρμας είναι GET.

Ωστόσο, όταν χρησιμοποιείται το GET, τα δεδομένα της φόρμας που θα υποβληθούν θα είναι **ορατά** στο πεδίο διεύθυνσης σελίδας

# Forms

## Σημειώσεις για τη GET:

- Προσθέτει δεδομένα φόρμας στη διεύθυνση URL σε ζεύγη ονόματος / τιμής.  
Το μήκος μιας διεύθυνσης URL είναι περιορισμένο (περίπου 3000 χαρακτήρες)
- Ποτέ μην χρησιμοποιείτε το GET για την αποστολή ευαίσθητων δεδομένων! (θα είναι ορατό στη διεύθυνση URL)
- Χρήσιμο για υποβολές φόρμας όπου ένας χρήστης θέλει να κάνει σελιδοδείκτη του αποτελέσματος
- Η GET είναι καλύτερο για μη ασφαλή δεδομένα, όπως οι συμβολοσειρές ερωτήσεων στο Google

# Forms

## Σημειώσεις για τη POST:

- Χρησιμοποιείτε την POST, εάν τα δεδομένα της φόρμας περιέχουν ευαίσθητες ή προσωπικές πληροφορίες. Η μέθοδος POST δεν εμφανίζει τα δεδομένα φόρμας που υποβλήθηκαν στο πεδίο διεύθυνσης σελίδας.
- Η POST δεν έχει περιορισμούς μεγέθους και μπορεί να χρησιμοποιηθεί για την αποστολή μεγάλων ποσοτήτων δεδομένων.
- Οι υποβολές φορμών με POST δεν μπορούν να επισημανθούν με σελιδοδείκτη

# Forms

Παράμετρος name:

```
<form action="/action_page.php"
      method="post">
  <input type="text" value="Aristos"><br>
  <input type="text" name="lastname"
        value="Pashias"><br>
```

**name:** Κάθε πεδίο εισαγωγής πρέπει να έχει ένα χαρακτηριστικό όνομα που πρέπει να υποβληθεί. Εάν το χαρακτηριστικό όνομα παραλείπεται, τα δεδομένα αυτού του πεδίου εισαγωγής δεν θα αποσταλούν καθόλου. Αυτό το παράδειγμα θα υποβάλει μόνο το πεδίο εισαγωγής "lastname":

# Client site html form

## Index.html

```
<html>
```

```
<body>
```

```
<form action="welcome.php" method="post">
```

```
Name: <input type="text" name="name"><br>
```

```
E-mail: <input type="text" name="email"><br>
```

```
<input type="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

Name:

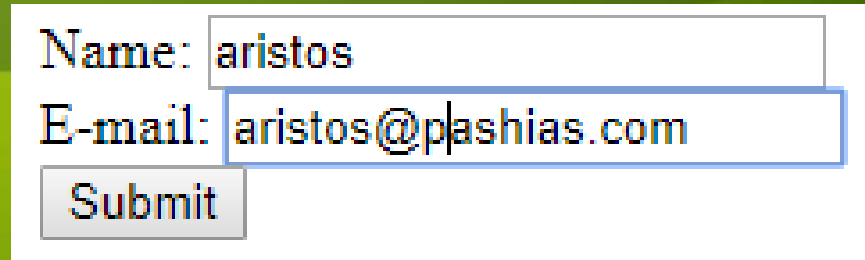
E-mail:

Submit

# Server site PHP Scripting code.

welcome.php (Post)

```
<html>  
<body>
```



Name:   
E-mail:

Welcome

```
<?php echo $_POST["name"]; ?><br>
```

Your email address is:

```
<?php echo $_POST["email"]; ?>
```

```
</body>  
</html>
```

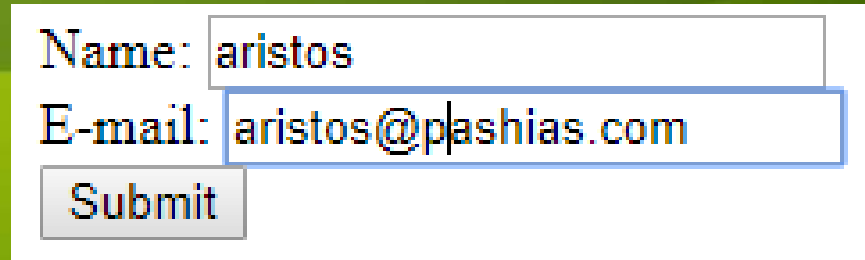
Welcome aristos

Your email address is: aristos@pashias.com

# Server site PHP Scripting code.

welcome.php (Get)

```
<html>  
<body>
```



A screenshot of a web form. It has two input fields: 'Name:' with the value 'aristos' and 'E-mail:' with the value 'aristos@pashias.com'. Below the fields is a 'Submit' button.

Welcome

```
<?php echo $_GET["name"]; ?><br>
```

Your email address is:

```
<?php echo $_GET["email"]; ?>
```

```
</body>  
</html>
```

Welcome aristos

Your email address is: aristos@pashias.com

# Παράδειγμα με GET.

```
<form action="examp2.php" method="GET">
  <div><label for="firstname">First name:
    <input type="text" name="firstname" id="firstname"/>
  </label>
</div>
  <div><input type="submit" value="Submit"/></div>
</form>
```

```
<html>
<body>
<?php
    $theName = $_GET['firstname'];
    echo "Καλησπέρα, ";
    echo $theName;
?>
</body>
</html>
```

# Post Vs Get

Τόσο η GET όσο και η POST δημιουργούν έναν πίνακα ο οποίος περιέχει ζεύγη κλειδιών / τιμών (key/value), όπου keys είναι τα ονόματα των στοιχείων ελέγχου και value είναι τα δεδομένα εισόδου από το χρήστη.

(Array (key0 => value0, key1 => value1, ... key10 => value10) ).

Ο Τόσο η GET όσο και η POST αντιμετωπίζονται ως superglobals πίνακες μεταβλητών με ονόματα \$\_GET και \$\_POST, που σημαίνει ότι είναι πάντα προσβάσιμα από οποιαδήποτε λειτουργία, τάξη ή αρχείο χωρίς να χρειάζεται να κάνετε κάτι ιδιαίτερο.

Το \$\_GET είναι ένας πίνακας μεταβλητών που μεταφέρονται στο τρέχον σενάριο μέσω των παραμέτρων διεύθυνσης URL.

Το \$\_POST είναι ένας πίνακας μεταβλητών που μεταφέρονται στο τρέχον σενάριο μέσω της μεθόδου POST HTTP.

# Post Vs Get

## GET

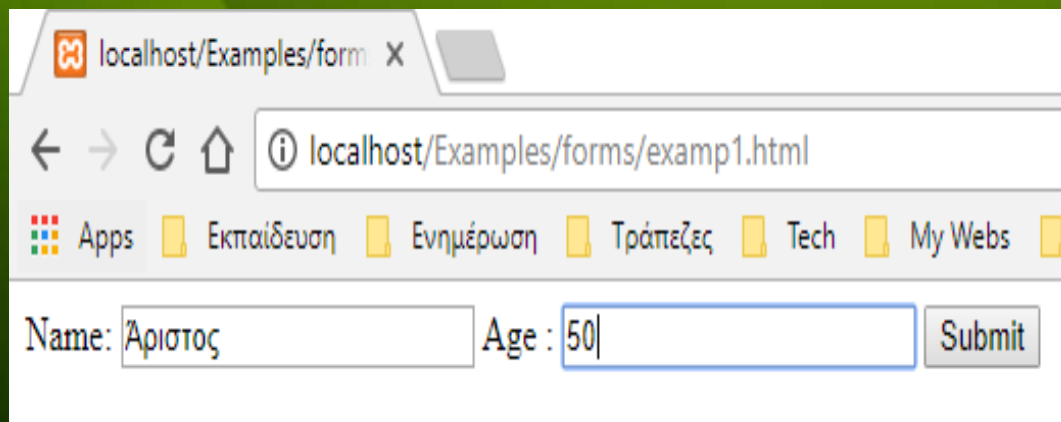
- μπορούν να αποθηκευτούν προσωρινά
- παραμένουν στο ιστορικό του προγράμματος περιήγησης
- μπορούν να επισημανθούν με σελιδοδείκτη
- δεν θα πρέπει ποτέ να χρησιμοποιούνται για μεταφορά ευαίσθητων δεδομένων
- έχουν περιορισμούς μήκους

## POST

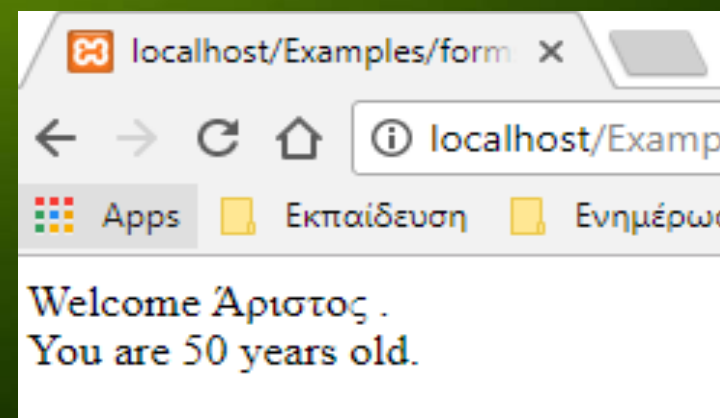
- Δεν αποθηκεύονται προσωρινά
- Δεν παραμένουν στο ιστορικό του προγράμματος περιήγησης
- Δεν μπορούν να επισημανθούν με σελιδοδείκτη
- Μπορούν να χρησιμοποιούνται για μεταφορά ευαίσθητων δεδομένων
- Δεν έχουν περιορισμούς μήκους

## Άσκηση 3

Να γράψετε ένα αρχείο html με το οποίο να εμφανίζετε την φόρμα της εικόνας 1 και να το συνδέσετε με ένα δεύτερο αρχείο PHP, το οποίο να διαβάζει τα πεδία που πληκτρολογήθηκαν στη φόρμα και να εμφανίζει το ανάλογο μήνυμα όπως φαίνεται στην εικόνα 2. Να χρησιμοποιήσετε την μέθοδο POST



Εικόνα 1



Εικόνα 2

## Άσκηση 3 Λύση

```
1 <html>
2 <body>
3   <form action="example.php" method="POST">
4     Name: <input type="text" name="name"/>
5     Age : <input type="text" name="age"/>
6     <input type="submit"/>
7   </form>
8 </body>
9 </html>
```

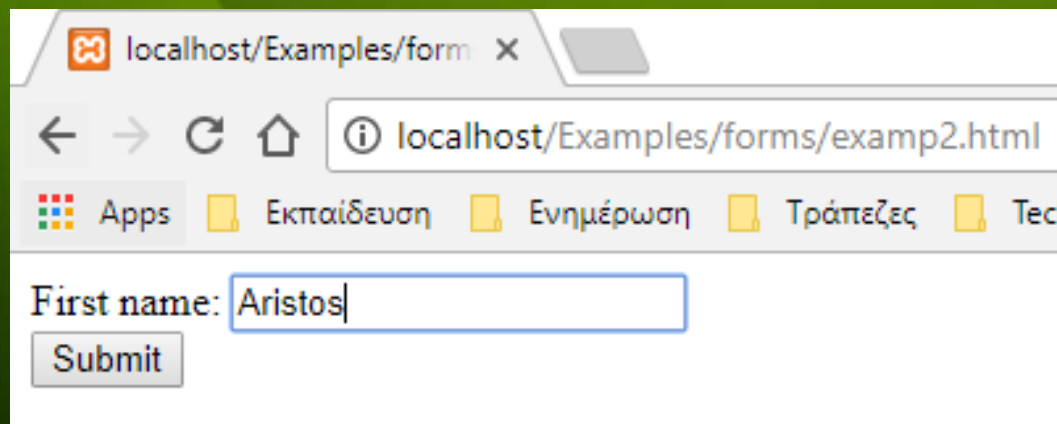
Αρχείο Html

```
1 <html>
2 <body>
3   Welcome
4   <?php
5     $n=$_POST["name"];
6     echo $n;
7     echo $_POST["name"];
8   ?> . <br/>
9   You are
10  <?php
11    echo $_POST["age"];
12  ?>
13  years old.
14 </body>
15 </html>
```

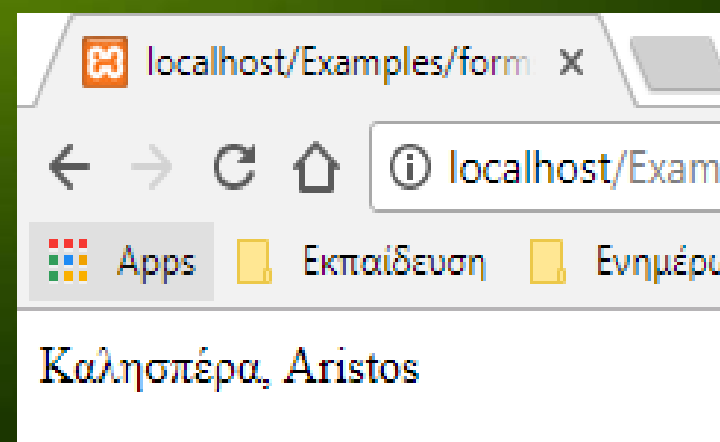
Αρχείο PHP

## Άσκηση 4

Να γράψετε ένα αρχείο html με το οποίο να εμφανίζετε την φόρμα της εικόνας 1 και να το συνδέσετε με ένα δεύτερο αρχείο PHP, το οποίο να διαβάζει τα πεδία που πληκτρολογήθηκαν στη φόρμα και να εμφανίζει το ανάλογο μήνυμα όπως φαίνεται στην εικόνα 2. Να χρησιμοποιήσετε την μέθοδο GET



Εικόνα 1



Εικόνα 2

## Άσκηση 4 Λύση

```
1 <form action="examp2.php" method="GET">
2   <div><label for="firstname">First name:
3     <input type="text" name="firstname" id="firstname"/></label>
4   </div>
5   <div><input type="submit" value="Submit"/></div>
6 </form>
```

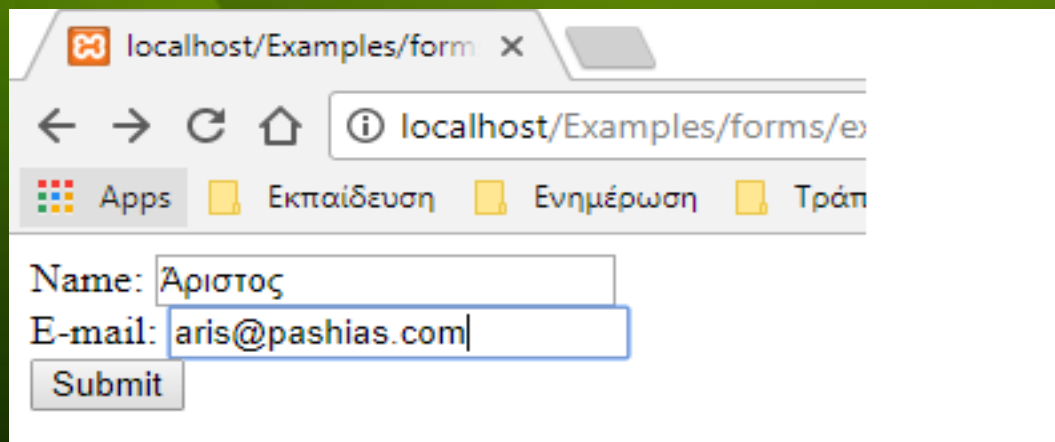
Αρχείο Html

```
1 <html>
2 <body>
3 <?php
4   $theName = $_GET['firstname'];
5   echo "Καλησπέρα, ";
6   echo $theName;
7 >?>
8 </body>
9 </html>
```

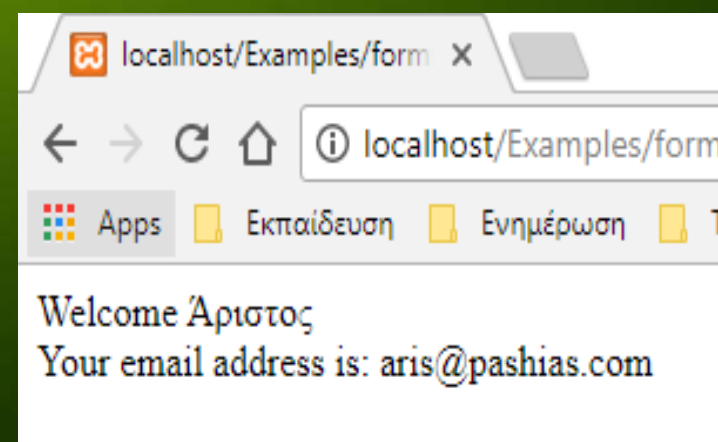
Αρχείο PHP

## Άσκηση 5

Να γράψετε ένα αρχείο html με το οποίο να εμφανίζετε την φόρμα της εικόνας 1 και να το συνδέσετε με ένα δεύτερο αρχείο PHP, το οποίο να διαβάζει τα πεδία που πληκτρολογήθηκαν στη φόρμα και να εμφανίζει το ανάλογο μήνυμα όπως φαίνεται στην εικόνα 2. Να χρησιμοποιήσετε την μέθοδο POST



Εικόνα 1



Εικόνα 2

## Άσκηση 5 Λύση

```
1 <html>
2 <body>
3 <form action="examp3.php" method="post">
4     Name :
5     <input type="text" name="name"><br>
6     E-mail:
7     <input type="text" name="email"><br>
8     <input type="submit">
9 </form>
10 </body>
11 </html>
```

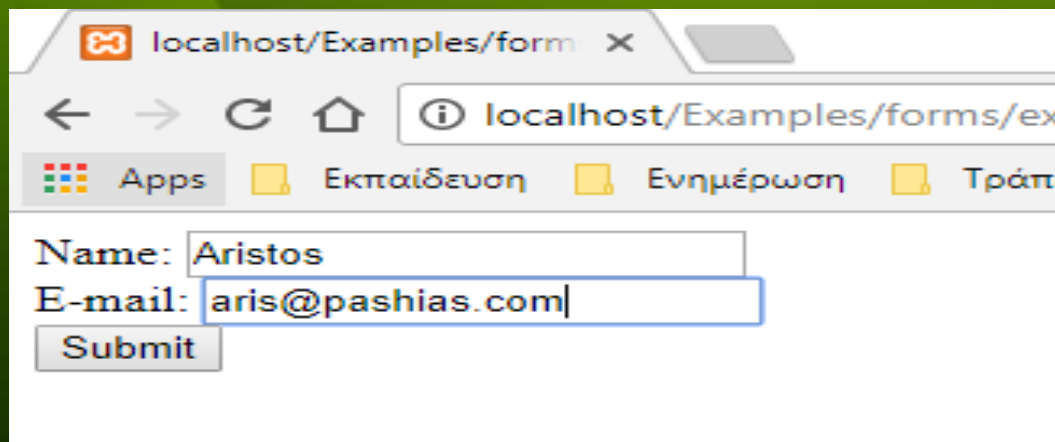
Αρχείο Html

```
1 <html>
2 <body>
3     Welcome
4     <?php
5     echo $_POST["name"]; <br>
6     echo 'Your email address is:'
7     echo $_POST["email"];
8     ?>
9 </body>
10 </html>
```

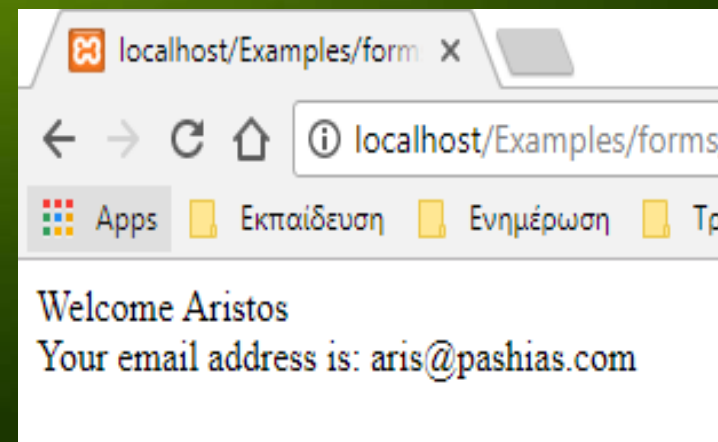
Αρχείο PHP

## Άσκηση 6

Να γράψετε ένα αρχείο html με το οποίο να εμφανίζετε την φόρμα της εικόνας 1 και να το συνδέσετε με ένα δεύτερο αρχείο PHP, το οποίο να διαβάζει τα πεδία που πληκτρολογήθηκαν στη φόρμα και να εμφανίζει το ανάλογο μήνυμα όπως φαίνεται στην εικόνα 2. Να χρησιμοποιήσετε την μέθοδο GET



Εικόνα 1



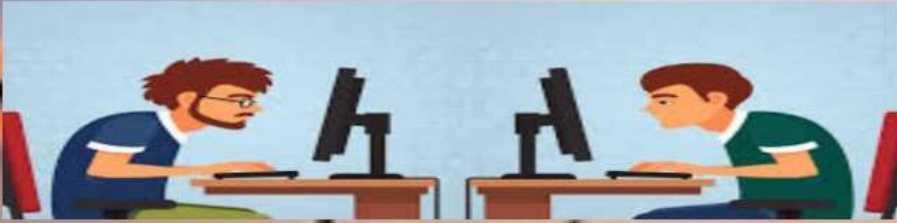
Εικόνα 2



## Άσκηση 7

Να δημιουργήσετε τα απαραίτητα αρχεία για την επικοινωνία χρήστη – εξυπηρετητή σύμφωνα με τα στοιχεία της φόρμας στα δεξιά. (Άσκηση PHP-FORMS ιστοσελίδας)

Λέσχη "Ερασιτέχνες Προγραμματιστές"



Εγγραφή μέλους

First Name (required):

Last name (required):

Birth day:

Gender:

Male  
 Female  
 Other

E-mail:

Address:

Town:

Programming in:

C++  
 PHP  
 Python  
 Javascript  
 Perl  
 None of the above

User Alias:

User password:

Επιστροφή φόρμας   
Αποδοχή και υποβολή

# Άσκηση 7 PHP file

```
1 <!DOCTYPE HTML>
2 <html>
3 <head></head>
4 <body>
5 <?php
6 // define variables and set to empty values
7 $fname = $lname = $email = $gender = $address
8 = $town = $bday = $alias = $psw= "";
9
10     $fname = $_POST["firstname"];
11
12     $lname = $_POST["lastname"];
13
14     $gender = $_POST["gender"];
15
16     $email = $_POST["email"];
17
18     $bday = $_POST["bday"];
19
20     $address = $_POST["address"];
21
22     $town = $_POST["town"];
23
24     $alias = $_POST["alias"];
25
26     $psw = $_POST["psw"];
27 <?>
```

```
27
28 <?php
29 echo "<h2>Your Input is:</h2>";
30 echo "Name : " . $fname . " " . $lname;
31 echo "<br>";
32 echo "gender : " . $gender;
33 echo "<br>";
34 echo "email : " . $email;
35 echo "<br>";
36 echo "Bertday : " . $bday;
37 echo "<br>";
38 echo "Address : " . $address;
39 echo "<br>";
40 echo "Town : " . $town;
41 echo "<br>";
42 echo "Alias : " . $alias;
43 echo "<br>";
44 echo "Password : " . $psw;
45
46 ?>
47
48 </body>
49 </html>
```

# Έλεγχος στοιχείων φόρμας

- ❖ Συνάρτηση **empty()**: Ελέγχει αν η μεταβλητή έχει ως τιμή ( empty string , 0, NULL ή False) . Επιστρέφει FALSE αν η μεταβλητή έχει non-empty ή non-zero τιμή.
- ❖ Συνάρτηση **isset()**: Ελέγχει αν η μεταβλητή έχει πάρει τιμή, η οποία μπορεί να είναι ακόμη και False , 0 ή Empty string, αλλά όχι NULL. Επιστρέφει TRUE αν η μεταβλητή υπάρχει και FALSE σε οποιαδήποτε άλλη περίπτωση.
- ❖ Συνάρτηση **unset()**: μπορείτε να ελευθερώσετε μια μεταβλητή \$\_POST από την τιμή που έχει πάρει.

# Έλεγχος στοιχείων φόρμας - Παράδειγμα

```
<?php
$var = 0; //$var = $_GET['myage']
if (empty($var)) {
    echo '$var is either 0, empty, or not set
    at all';
} //Επιστρέφει true αφού η $var είναι 0
if (isset($var)) {
    echo '$var is set even though it is
    empty';
} // Επιστρέφει true αφού η $var υπάρχει
unset($_POST['variable_name']); ?>
```

# Επικύρωση στοιχείων φόρμας

- ❖ Μεταβίβαση όλων των μεταβλητών μέσω της λειτουργίας `htmlspecialchars ()` της PHP ώστε αν ένας χρήστης προσπαθήσει να υποβάλει ένα script σε ένα πεδίο κειμένου πχ:

```
<script> location.href ('http://www.hacked.com') </ script>
```

να μην εκτελεστεί, επειδή θα αποθηκευτεί ως :

```
& lt; script & gt; location.href ('http://www.hacked.com') & lt; /  
script & gt;
```

- ❖ Διαγραφή των περιττών χαρακτήρων από τα δεδομένα εισόδου του χρήστη με τη συνάρτηση PHP `trim ()`
- ❖ Αφαίρεση των backslash (\) από τα δεδομένα εισόδου του χρήστη με τη λειτουργία PHP `stripslashes ()`

# Επικύρωση στοιχείων φόρμας - Παράδειγμα

```
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

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

# Παράδειγμα 1α

1. Να δημιουργήσετε ένα Html αρχείο και να συμπληρώσετε σε αυτό τον απαραίτητο κώδικα ώστε να εμφανίζει την φόρμα:

## PHP Form Validation Example

Name:

E-mail:

Website:

Comment:

Gender:  Female  Male  Other

## Λύση 1α

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 </head>
5 <body>
6 <h2>PHP Form Validation Example</h2>
7 <form method="post" action="Valit.php">
8   Name: <input type="text" name="name"><br><br>
9   E-mail: <input type="text" name="email"><br><br>
10  Website: <input type="text" name="website"><br><br>
11  Comment: <textarea name="comment" rows="5" cols="40"></textarea>
12    <br><br>
13    Gender:
14    <input type="radio" name="gender" value="female">Female
15    <input type="radio" name="gender" value="male">Male
16    <input type="radio" name="gender" value="other">Other
17    <br><br>
18    <input type="submit" name="submit" value="Submit">
19 </form>
20 </body>
21 <html>
22 </html>
```

# Παράδειγμα 1β

1. Να δημιουργήσετε το αρχείο `valid.php` και να εισάγετε σε αυτό τον απαραίτητο κώδικα `php` ώστε να διαβάζει και να εμφανίζει τα πεδία της φόρμας του παραδείγματος 1α.

```
1 <?php
2 // define variables and set to empty values
3 $name = $email = $gender = $comment = $website = "";
4
5 if ($_SERVER["REQUEST_METHOD"] == "POST") {
6     $name = $_POST["name"];
7     $email = $_POST["email"];
8     $website = $_POST["website"];
9     $comment = $_POST["comment"];
10    $gender = $_POST["gender"];
11 }
12
13 ?>
14 <?php
15 echo "<h2>Your Input:</h2>";
16 echo $name;
17 echo "<br>";
18 echo $email;
19 echo "<br>";
20 echo $website;
21 echo "<br>";
22 echo $comment;
23 echo "<br>";
24 echo $gender;
25 ?>
```

**PHP Form Validation Example**

Name:

E-mail:

Website:

Comment:

Gender:  Female  Male  Other

# Παράδειγμα 1γ

1. Να διορθώσετε το αρχείο valid.php και να εισάγετε σε αυτό τον απαραίτητο κώδικα php ώστε να μπορεί να επικυρώσει τα στοιχεία που διαβάζει με τη χρήση των εντολών trim(); stripslashes(); htmlspecialchars();

```
1  <?php
2  // define variables and set to empty values
3  $name = $email = $gender = $comment = $website = "";
4
5  if ($_SERVER["REQUEST_METHOD"] == "POST") {
6      $name = test_input($_POST["name"]);
7      $email = test_input($_POST["email"]);
8      $website = test_input($_POST["website"]);
9      $comment = test_input($_POST["comment"]);
10     $gender = test_input($_POST["gender"]);
11 }
12
13 function test_input($data) {
14     $data = trim($data);
15     $data = stripslashes($data);
16     $data = htmlspecialchars($data);
17     return $data;
18 }
19 ?>
```

## Παράδειγμα 1γ συνέχεια

Πάντα λοιπόν μεταβιβάζουμε όλες τις μεταβλητές μέσω της λειτουργίας `htmlspecialchars ()` της PHP. Έτσι εάν ένας χρήστης προσπαθήσει να υποβάλει τα παρακάτω σε ένα πεδίο κειμένου:

```
<script> location.href ('http://www.hacked.com') </ script>
```

αυτό δεν θα εκτελεστεί, επειδή θα αποθηκευτεί ως Κωδικός HTML, όπως παρακάτω:

```
& lt; script & gt; location.href ('http://www.hacked.com') & lt; / script & gt;
```

Ο κωδικός είναι πλέον ασφαλής να εμφανίζεται σε μια σελίδα ή μέσα σε ένα μήνυμα ηλεκτρονικού ταχυδρομείου. Επίσης, κάνουμε και άλλα δύο πράγματα όταν ο χρήστης υποβάλει τη φόρμα: Διαγραφή των περιττών χαρακτήρων (πρόσθετος χώρος, καρτέλα, νέα γραμμή) από τα δεδομένα εισόδου του χρήστη (με τη συνάρτηση PHP `trim ()`) Αφαιρούμε τα `backslash (\)` από τα δεδομένα εισόδου του χρήστη με τη λειτουργία PHP `stripslashes ()`

# Παράδειγμα 1γ συνέχεια

1. Να διορθώσετε το αρχείο valid.php και να εισάγετε σε αυτό τον απαραίτητο κώδικα php ώστε να μπορεί να επικυρώσει τα στοιχεία που διαβάζει με τη χρήση των εντολών trim(); stripslashes(); htmlspecialchars();

```
1  <?php
2  // define variables and set to empty values
3  $name = $email = $gender = $comment = $website = "";
4
5  if ($_SERVER["REQUEST_METHOD"] == "POST") {
6      $name = test_input($_POST["name"]);
7      $email = test_input($_POST["email"]);
8      $website = test_input($_POST["website"]);
9      $comment = test_input($_POST["comment"]);
10     $gender = test_input($_POST["gender"]);
11 }
12
13 function test_input($data) {
14     $data = trim($data);
15     $data = stripslashes($data);
16     $data = htmlspecialchars($data);
17     return $data;
18 }
19 ?>
```

# Παράδειγμα 1δ

1. Επιπλέον θέλουμε να γίνεται έλεγχος , και να ενημερώνεται ο χρήστης ότι τα στοιχεία που εισάγει είναι σωστά σύμφωνα με τον πιο κάτω πίνακα

## PHP Form Validation Example

\* required field

Name:  \*

E-mail:  \*

Website:

Comment:

Gender:  Female  Male  Other \*

**Your Input:**

Πεδίο	Κανόνες Επικύρωσης
Name	Απαραίτητο. + Περιλαμβάνει μόνο Γράμματα και κενό διάστημα
E-mail	Απαραίτητο. + Περιλαμβάνει έγκυρη διεύθυνση email ( @ .)
Website	Προαιρετικό. Αν υπάρχει να είναι έγκυρο URL
Comment	Προαιρετικό.
Gender	Απαραίτητο. Υποχρεωτική επιλογή ενός από τα τρία

## Λύση 1δ

Για να μπορεί ο χρήστης να παίρνει μηνύματα για τυχόν λανθασμένη καταχώρηση θα πρέπει ο κώδικας ελέγχου `php` να βρίσκεται στο ίδιο αρχείο με την φόρμα εισαγωγής. Γι' αυτό θα χρησιμοποιήσουμε την superglobal μεταβλητή `$_SERVER["PHP_SELF"]` η οποία επιστρέφει το όνομα του τρέχοντος αρχείου σεναρίου. Συνεπώς αποστέλλει τα δεδομένα της φόρμας που υποβάλλονται στην ίδια τη σελίδα, αντί να μεταπηδά σε διαφορετική σελίδα. Με αυτόν τον τρόπο, ο χρήστης θα λάβει μηνύματα σφάλματος στην ίδια σελίδα με τη φόρμα. Οπότε η ετικέτα `<form>` παίρνει τη μορφή:

```
<form method="post"
action="<?php echo ($_SERVER["PHP_SELF"]);?>">
```

Απομένει βέβαια να προσθέσουμε το σενάριο για έλεγχο των καταχωρήσεων

# Λύση 1δ συνέχεια...

```
<?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"]); }

    if (empty($_POST["email"])) { $emailErr = "Email is required"; }
    else { $email = test_input($_POST["email"]); }

    if (empty($_POST["website"])) { $website = ""; }
    else { $website = test_input($_POST["website"]); }

    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"]); }
}
?>
```

# Λύση 1ε htmlspecialchars()

Απομένει να προστατεύσουμε το σενάριο μας από επίδοξους Χακερ που ενδέχεται να αξιοποιήσουν την συνάρτηση `$_SERVER["PHP_SELF"]`. Επιστρατεύουμε τη συνάρτηση **htmlspecialchars ()** η οποία μετατρέπει τους ειδικούς χαρακτήρες σε οντότητες HTML. Αυτό σημαίνει ότι θα αντικαταστήσει χαρακτήρες HTML όπως `<` και `>` με `&lt;` και `&gt;`; Αυτό εμποδίζει τους εισβολείς να εκμεταλλευτούν τον κώδικα, εισάγοντας κώδικες HTML ή Javascript (επιθέσεις διαδικτυακών σεναρίων σε φόρμες).

Πχ αν ο χρήστης πληκτρολογήσε στη γραμμή διευθύνσεων:

```
http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert('UR hacked')%3C/script%3E
```

Θα έδινε το αποτέλεσμα:

```
<form method="post" action="test_form.php/"><script>alert('UR hacked')</script>
```

# Λύση 1ε htmlspecialchars()

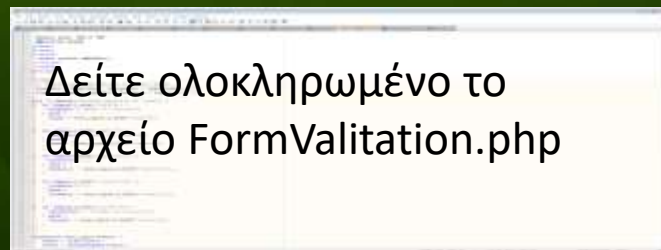
Αν όμως συντάξουμε την ετικέτα <form> όπως φαίνεται πιο κάτω

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

Θα έδινε ως αποτέλεσμα:

```
<form method="post" action="test_form.php/&quot;&gt;
&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

Που προφανώς δεν επιτρέπει στον χακερ να δημιουργήσει πρόβλημα





# Δείτε ολοκληρωμένο το αρχείο FormValitation.php

## PHP Form Validation Example

\* required field

Name:  \*

E-mail:  \*

Website:

Comment:

Gender:  Female  Male  Other \*

**Your Input:**

## PHP Form Validation Example

\* required field

Name:  \*

E-mail:  \*

Website:

Comment:

Gender:  Female  Male  Other \*

**Your Input:**

Aristos  
aristos@Pashias.com  
www.pashias.net  
Congrats  
male

# Δείτε ολοκληρωμένο το αρχείο FormValitation.php

```
2 Result Size: 945 x 788
3 <!DOCTYPE HTML>
4 <html>
5 <head>
6 <style>
7 .error {color: #FF0000;}
8 </style>
9 </head>
10 <body>
11
12 <?php
13 // define variables and set to empty values
14 $nameErr = $emailErr = $genderErr = $websiteErr = "";
15 $name = $email = $gender = $comment = $website = "";
16
17 if ($_SERVER["REQUEST_METHOD"] == "POST") {
18     if (empty($_POST["name"])) {
19         $nameErr = "Name is required";
20     } else {
21         $name = test_input($_POST["name"]);
22     }
23
24     if (empty($_POST["email"])) {
25         $emailErr = "Email is required";
26     } else {
27         $email = test_input($_POST["email"]);
28     }
29
30     if (empty($_POST["website"])) {
31         $website = "";
32     } else {
33         $website = test_input($_POST["website"]);
34     }
35
36     if (empty($_POST["comment"])) {
37         $comment = "";
38     } else {
39         $comment = test_input($_POST["comment"]);
40     }
41
42     if (empty($_POST["gender"])) {
43         $genderErr = "Gender is required";
44     } else {
45         $gender = test_input($_POST["gender"]);
```

```
46     }
47 }
48
49 function test_input($data) {
50     $data = trim($data);
51     $data = stripslashes($data);
52     $data = htmlspecialchars($data);
53     return $data;
54 }
55 <?>
56
57 <h2>PHP Form Validation Example</h2>
58 <p><span class="error">* required field</span></p>
59 <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
60     Name: <input type="text" name="name">
61     <span class="error">* <?php echo $nameErr;?></span>
62     <br><br>
63     E-mail: <input type="text" name="email">
64     <span class="error">* <?php echo $emailErr;?></span>
65     <br><br>
66     Website: <input type="text" name="website">
67     <span class="error"><?php echo $websiteErr;?></span>
68     <br><br>
69     Comment: <textarea name="comment" rows="5" cols="40"></textarea>
70     <br><br>
71     Gender:
72     <input type="radio" name="gender" value="female">Female
73     <input type="radio" name="gender" value="male">Male
74     <input type="radio" name="gender" value="other">Other
75     <span class="error">* <?php echo $genderErr;?></span>
76     <br><br>
77     <input type="submit" name="submit" value="Submit">
78 </form>
79
80 <?php
81 echo "<h2>Your Input:</h2>";
82 echo $name;
83 echo "<br>";
84 echo $email;
85 echo "<br>";
86 echo $website;
87 echo "<br>";
88 echo $comment;
89 echo "<br>";
90 echo $gender;
91 <?>
```

## Αυστηρότερος έλεγχος επικύρωσης - Όνομα

Η συνάρτηση `preg_match ()` αναζητά σε ένα αλφαριθμητικό μια σειρά χαρακτήρες, και επιστρέφει **True** εάν υπάρχει διαφορετικά επιστρέφει **False**.

```
1 <?php
2 $my_url = "www.pashias.net";
3 if (preg_match("/pashias/", $my_url))
4 {
5     echo "the url $my_url contains pashias";
6 }
7 else
8 {
9     echo "the url $my_url does not contain pashias";
10 }
11 ?>
```

Εκπαίδευση

Ενημέρωση

Τράπεζα

the url www.pashias.net contains pashias

```
$name = $_POST["name"];
if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
    $nameErr = "Only letters and white space allowed";
}
```

## Αυστηρότερος έλεγχος επικύρωσης - E-mail - URL

```
$email = $_POST["email"];  
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
    $emailErr = "Invalid email format";  
}
```

```
$website = $_POST["website"];  
if (!preg_match("/\b(?:(:https?|ftp):\\V\\V|www\\.)([-a-z0-9+&@#\\V/%?=#~_|!:,.;]*[-a-z0-9+&@#\\V/%=#~_|])/i",$website)) {  
    $websiteErr = "Invalid URL";  
}
```

# Keep The Values in The Form

```
Name: <input type="text" name="name" value="<?php echo $name;?>">
E-mail: <input type="text" name="email" value="<?php echo $email;?>">
Website: <input type="text" name="website" value="<?php echo $website;?>">
Comment: <textarea name="comment" rows="5" cols="40"><?php echo $comment;?></textarea>
Gender:
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="female") echo "checked";?>
value="female">Female
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="male") echo "checked";?>
value="male">Male
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="other") echo "checked";?>
value="other">Other
```

# Άσκηση

Να δημιουργήσετε την ιστοσελίδα ενός εστιατορίου της Λεμεσού η οποία θα επιτρέπει στους επισκέπτες της, αφού καταχωρήσουν το όνομα και το σταθερό τηλέφωνο τους να επιλέξουν και να παραγγείλουν φαγητό από τον τιμοκατάλογο ο οποίος περιλαμβάνει τέσσερις επιλογές (Σουβλάκι €8, Μπριζόλα €11, Σεφταλιά €9 και Γύρο €14).

Η εφαρμογή θα πρέπει να ελέγχει αν το τηλέφωνο που καταχωρήθηκε είναι σταθερό τηλέφωνο της Λεμεσού. Αν όχι θα εμφανίζει το ανάλογο μήνυμα και θα επιτρέπει στον επισκέπτη να δοκιμάσει ξανά. Αν ναι θα επιτρέπει στον επισκέπτη να καταχωρήσει την παραγγελία του. Με το πάτημα από τον επισκέπτη του κουμπιού «καταχώρηση», θα εμφανίζει το πιο κάτω μήνυμα με το όνομα και το τηλέφωνο του πελάτη επισκέπτη.

**«Πέτρο, Η παραγγελία σας καταχωρήθηκε. Σας ευχαριστούμε για την προτίμηση. Μόλις ετοιμαστεί θα ενημερωθείτε με μήνυμα στο τηλέφωνο 25444444. Θα πληρώσετε 8€»**

Σημείωση1: Μπορείτε να γράψετε τον κώδικα αλληλεπίδρασης χρησιμοποιώντας Javascript και PHP.

Σημείωση 2: Οι υπογραμμισμένες λέξεις στα μηνύματα να αντικατασταθούν από τις τιμές των δικών σας αντίστοιχων μεταβλητών

Σημείωση3: Όνομα αρχείου: efarmogi4.php