



Πανεπιστήμιο Πειραιώς  
Τμήμα Ψηφιακών Συστημάτων

Διαδικτυακά και Φορητά Πληροφοριακά Συστήματα  
Απαλλακτική Εργασία Σεπτεμβρίου 2021

## **ΤΕΧΝΙΚΗ ΠΕΡΙΓΡΑΦΗ**

Ονοματεπώνυμο  
ΣΕΜΠΑΣΤΙΑΝ ΑΝΤΡΙΣΑΝ

A.M.  
E18008

## Interface 1: Αρχική Σελίδα

Επεξήγηση interface1.html :

Το πρώτο interface , εκτός του navigation bar και του footer αποτελείται από ένα wrapper που περιέχει τα <div class="who-are-we"> , <div class="services"> , <div class="img-gallery-wrapper"> , <form class="register" action="./register.php" method="post"> .

Το who-are-we αποτελείται από έναν h1 τίτλο , ένα κείμενο που περιγράφει την εταιρία και ένα iframe σχετικό βίντεο από το youtube που ξεκινάει χωρίς ήχο όταν ανοίγει η σελίδα. Για καθένα από τα παραπάνω έχουν οριστεί τα αντίστοιχα class.....css

Το services αποτελείται από έναν h1 τίτλο και ακολουθούν 4 υπηρεσίες. Κάθε υπηρεσία έχει έναν h3 τίτλο και μια εικόνα με class="responsive" το οποίο εξηγείται παρακάτω . Μετά τις 4 υπηρεσίες υπάρχει ένα call-to-action button το οποίο στέλνει τον χρήστη στο δεύτερο interface δημιουργώντας νέο παράθυρο με την βοήθεια των href και target. Συγκεκριμένα :

```
<div class="button2">  
  <a href="./interface2.html" target="_blank">More About Us</a>  
</div>
```

Ακολουθεί νέο wrapper, το img-gallery-wrapper που περιέχει 12 φωτογραφίες που επιλέχθηκαν από το διαδίκτυο και καλούνται μέσω των ``. Το styling και responsiveness κάθε φωτογραφίας χρησιμοποιείται το `class="box"`.

Τέλος, υπάρχει η παρακάτω φόρμα εγγραφής στο Newsletter της εταιρίας :

```
<form class="register" action="./register.php" method="post">
  <h4 style="padding:10px;">Register to Gazprom's Newsletter</h4>
  <input style="margin:auto;" type="text" name="username" id="username" placeholder="Username">
  <input style="margin:auto;" type="text" name="name" id="name" placeholder="Full Name">
  <input style="margin:auto;" type="text" name="email" id="email" placeholder="Your email">
  <button style="height:50px; width:45%; margin:3px; margin:auto;" type="submit" name="submit">REGISTER</button>
</form>
```

Ο χρήστης, εφόσον επιθυμεί, μπορεί να συμπληρώσει τα πεδία Username, Full Name, Your email και να πατήσει το κουμπί REGISTER. Όταν πατιέται το REGISTER καλείται το register.php αρχείο με `method="post"`. Εντός της φόρμας χρησιμοποιείται εσωτερική css, μέσω του `style`, για την αλλαγή του μεγέθους του register button και `margin:auto` ώστε κάθε πεδίο να βρίσκεται στη μέση της φόρμας. Υπάρχει ακόμη αναφορά στην κλάση register στο εξωτερικό css όπου ορίζεται η φόρμα ως grid (θα εξηγηθεί παρακάτω).

Επεξήγηση register.php :

Δίνεται τιμή 0 στην μεταβλητή count.

Ελέγχεται για κάθε field αν δόθηκε τιμή από τον χρήστη. Εφόσον έχει δοθεί τιμή, εκχωρείται στην αντίστοιχη μεταβλητή ( \$username, \$name

ή \$email) και αυξάνεται η τιμή του count κατά 1.

```
if (empty($_POST["username"])) {  
    $usernameErr = "username is required";  
} else {  
    $username = $_POST['username'];  
    $count = $count +1;  
}
```

Στην συνέχεια ελέγχεται αν το count ισούται με 3, εάν δηλαδή έχουν συμπληρωθεί όλα τα input της φόρμας. Αν ναι, δημιουργείται mysqli σύνδεση με τον localhost για την βάση electricitydb :

```
$conn = new mysqli('localhost','root','','electricitydb');
```

Εφόσον γίνει επιτυχώς σύνδεση με την βάση δεδομένων εκτελείται ο παρακάτω κώδικας:

```
} else {  
    $stmt = $conn->prepare("insert into clients (name,username,email) values(?,?,?)");  
    $stmt->bind_param("sss",$name,$username,$email);  
    $stmt->execute();  
    $stmt->close();  
    $conn->close();  
}  
header("Location: ./interface1.html");
```

Εκτελείται το insert into της mySql στον πίνακα clients. Κάθε τιμή αντιστοιχεί σε ένα ? εντός της values και γίνεται εκχώρηση μέσω του bind\_param("sss", \$name,\$username,\$email) όπου το s σημαίνει πως η τιμή που θα περαστεί είναι τύπου string. Αφού εκτελεστεί το query, η μεταβλητή stmt δηλαδή, κλείνει η σύνδεση με τον localhost και η php μας ανακατευθύνει στην αρχική σελίδα interface1.html μέσω του header.

Σε περίπτωση που το count δεν είναι 3 εκτελείται ο παρακάτω κώδικας:

```

}else{
    ?>
    <script type="text/javascript">
        window.location = "./interface1.html";
        window.alert("EVERY FIELD MUST BE COMPLETE. PLEASE TRY AGAIN.");
    </script>
    <?php
}

```

Ο παραπάνω κώδικας κάνει έξοδο από την php και εκτελεί javascript script. Ανανεώνει την αρχική σελίδα interface1 και ενημερώνει τον χρήστη με alert-box μήνυμα πως πρέπει να συμπληρώσει όλα τα πεδία και να προσπαθήσει ξανά. Προτιμήθηκε η χρήση της javascript για να επιστρέφει το alert-box στον χρήστη.

Επεξήγηση interface1.css :

wrapper:

```

.wrapper{
    display: grid;
    grid-gap: 1em;
    grid-template-rows: auto auto auto auto;
    background-color: #ccf2ff;
    background-image: linear-gradient(rgba(4,9,30,0.1),rgba(4,9,30,0.1)),url(https://v
    background-position: center;
    background-size: cover;
    position: relative;
    width: 100%;
}

```

Ορίζεται grid wrapper που χωρίζει το περιεχόμενο σε 4 γραμμές . Το μέγεθος κάθε γραμμής εξαρτάται από το αντίστοιχο content που φιλοξενεί . Για φόντο προτιμήθηκε η χρήση εικόνας (από το διαδίκτυο) αντί απλού χρώματος.

Το who-are-we είναι grid 2 σειρών. Η πρώτη σειρά αποτελείται από

έναν μεγάλο τίτλο. Η δεύτερη σειρά φιλοξενεί στα αριστερά ένα κείμενο και στα δεξιά το video που είναι σταθερού μεγέθους.

Το services είναι grid 4 στηλών και 3 γραμμών. Στην πρώτη γραμμή υπάρχει μόνο ένα header. Η δεύτερη σειρά χωρίζεται σε 4 στήλες ίδιου μεγέθους, όπου η καθεμία έχει από μια υπηρεσία που προσφέρει η εταιρία (μια υπηρεσία αποτελείται από έναν τίτλο και μια αντίστοιχη εικόνα ). Στην 3<sup>η</sup> σειρά υπάρχει ένα κουμπί που βρίσκεται στο κέντρο της σελίδας.

Οι εικόνες των 4 παραπάνω υπηρεσιών ανήκουν στην κλάση responsive ώστε να αντιδρούν με τον ίδιο τρόπο όταν αλλάζει το μέγεθος της ιστοσελίδας.

responsive:

```
148 .responsive {  
149     width: 100%;  
150     height: auto;  
151     border-radius: 10px;  
152 }
```

Το κουμπί button2 που οδηγεί τον χρήστη στο 2<sup>ο</sup> Interface, εκτός του styling γραμματοσειράς, μεγέθους ,χρωμάτων και τοποθέτησης , έχει επίσης την ιδιότητα, όποτε γίνεται hover από πάνω του να ανάβει μπλε φως γύρω από τα γράμματα.

Το image-gallery είναι grid 4 στηλών ίδιου μεγέθους. Ο αριθμός των γραμμών εξαρτάται από τον αριθμό των εικόνων. Αν δηλαδή αντί για

12 είχαμε 13 εικόνες, θα φτιαχνόταν μια νέα σειρά ώστε να μπει η τελευταία εικόνα.

Στην τελευταία σειρά του γενικού wrapper έχουμε την φόρμα register. Χρησιμοποιήθηκαν διάφορες τιμές του margin ώστε να τοποθετηθεί στο κέντρο της σελίδας. Αποτελείται από τον τίτλο της φόρμας, τα input fields και το κουμπί που στέλνει την φόρμα. Το καθένα από τα παραπάνω καταλαμβάνουν ίσο χώρο στην φόρμα αφού ορίστηκε display: grid και μέγεθος κάθε σειράς 1fr.

Για να είναι responsive με mobile και tablet έγιναν αλλαγές στα column και rows των grid καθώς και αυξομειώσεις στα margin, padding, height και width των περισσότερων div.

## **Interface 2: Προφίλ Εταιρίας**

Επεξήγηση interface2.html :

Στο συγκεκριμένο interface έχουμε ένα wrapper που αποτελείται από το about-us , ένα image wrapper δύο εικόνων , το human-power (ανθρώπινο δυναμικό) και ένα call-to-action button.

Το about-us αποτελείται από τον τίτλο <<About Us>> και ακολουθούν 3 div ( philosophy, vision, mission ). Κάθε div αποτελείται από έναν τίτλο και κείμενο σχετικά με την θεματολογία.

Πριν το ανθρώπινο δυναμικό προστέθηκε ένα grid 2 εικόνων για τον εμπλουτισμό του interface.

Το human-power ακολουθεί την ίδια λογική με το about-us.

Το button3 είναι το ίδιο με αυτό του πρώτου interface.

### Interface 3: Υπολογισμός Κόστους Παροχής Ηλεκτρικού Ρεύματος

Κώδικας html του calculator:

```
<div class="container">
  <h1 style="padding-top:23px;">Electricity Cost Calculator</h1>
  <p style="font-size:14px;">cheapest electricity guaranteed!</p>
  <p>Consumption (kWh) :</p>
  <input id="consumption" type="number" min="0" value="1" onchange = "calculateCost()">
  <p>Total Days :</p>
  <input id="days" type="number" min="0" value="1" onchange = "calculateCost()">
  <p>Size of space &#13216 :</p>
  <input style="margin-bottom:34px;"id="size" min="0" value="1" type="number" onchange = "calculateCost()">
  <h1 id="total_cost" style="color: #ffff00; padding-bottom:23px;"></h1>
</div>
```

Εντός ενός container έχουμε αρχικά έναν τίτλο ακολουθούμενη από μια πρόταση. Στην συνέχεια ακολουθούν τα input πεδία. Κάθε πεδίο δέχεται αριθμούς ( type="number" ) , δέχεται ελάχιστη τιμή 0 και έχει



default 1. Γίνεται χρήση των id για την επικοινωνία με το script.js στο οποίο υπάρχει η μέθοδος υπολογισμού του κόστους. Το κόστος υπολογίζεται κάθε φορά που γίνεται αλλαγή σε κάποιο πεδίο ( `onchange= "calculateCost()"` ). Κάθε φορά που υπολογίζεται νέο κόστος εμφανίζεται στο τέλος του container το συνολικό κόστος, h1 με `id="total_cost"` .

Για την υλοποίηση του cost calculator χρησιμοποιήθηκε κώδικας javascript που συνδέεται με την html μέσω

```
<script src="./script.js" defer></script>
```

 και της χρήσης id στην html.

Η μέθοδος `calculateCost()` δέχεται τις τιμές που δίνει ο χρήστης την ίδια στιγμή που τις ορίζει, δηλαδή δεν μεσολαβεί κάποιο submit. Γίνεται μέσω χρήσης της DOM μεθόδου `getElementById`.

```
function calculateCost(){  
    var consumption = document.getElementById('consumption').value;  
    var days = document.getElementById('days').value;  
    var size = document.getElementById('size').value;
```

Στην συνέχεια με την βοήθεια if ορίζονται τιμές για μεταβλητές σχετικά με χρέωση ενέργειας, λοιπές χρεώσεις , ΥΚΩ και χρεώσεις δήμου με βάση τις τιμές που έδωσε ο χρήστης.

Τέλος, υπολογίζονται τα κόστη που αναφέρονται στην εκφώνηση και το `total_cost` που επιστρέφεται στον χρήστη.

```

    }
    else{
        var xr_energ = 0.00822;
        var loip_xr = 0.00011;
        var ykw = 0.0135;
    }

    var cost_dim = (size*xr_dim*days)/365;
    var total_ykw = (consumption*ykw*days)/365;
    var total_xr = consumption*(xr_energ+loip_xr);
    var total_cost = cost_dim+total_ykw+total_xr;
    total_cost = total_cost.toFixed(2);

    document.getElementById('total_cost').innerHTML = "Total Cost : "+ total_cost + "€";
}

```

## Interface 4: Επικοινωνία

Κώδικας html της φόρμας επικοινωνίας :

```

<form class="contact-form" method="post" action="./contact2.php">
  <h2 style="padding:10px;">CONTACT FORM</h2>
  <input type="text" name="name" id="name" placeholder="Full name">
  <input type="number" name="phone_number" id="phone" placeholder="Phone number (10-digit)">
  <input type="text" name="email" id="email" placeholder="Your email">
  <input type="text" name="subject" id="subject" placeholder="Subject">
  <textarea style="margin:auto;" name="message" rows="10" cols="80" placeholder="Message"></textarea>
  <input style="font-size:25px; border-radius:200px; height:60px; width:30%; margin-bottom:15px; background-color:#f0f0f0;" type="submit" name="submit" value="SUBMIT"></input>
</form>

```

Η φόρμα εκτελεί το contact2.php αρχείο κάθε φορά που ο χρήστης πατάει το submit.

Για κάθε input έχουν οριστεί placeholder, name, id και type.

Επεξήγηση contact2.php :

Αρχικοποίηση μεταβλητής \$count = 0 και \$ph\_ok = false . Η μεταβλητή \$count χρησιμοποιείται για να ελέγχει αν όλα τα πεδία της φόρμας είναι συμπληρωμένα.

Κάθε φορά που ο χρήστης πατάει το submit κουμπί , και εφόσον έχουμε ορίσει method= "POST" ο php κώδικας περνάει εντός της παρακάτω if :

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

 Εντός αυτής της if

ελέγχεται, για κάθε πεδίο της φόρμας , αν έχει οριστεί κάποια τιμή.

Εφόσον έχει συμπληρωθεί από τον χρήστη ένα πεδίο, καταχωρείται η τιμή σε αντίστοιχη php μεταβλητή. Ακόμη, αυξάνεται το count κατά 1.

Κώδικας:

```
if (empty($_POST["email"])) {  
    $email = "Email is required";  
} else {  
    $email = test_input($_POST["email"]);  
    $count = $count +1;  
}
```

Ο παραπάνω κώδικας είναι ίδιος για όλα τα πεδία της φόρμας εκτός του phone number όπου ελέγχεται επίσης αν η τιμή που δόθηκε είναι αριθμός 10-ψηφίων. Αν συμφωνεί με τις παραπάνω παραμέτρους αλλάζουμε την τιμή του \$ph\_ok σε true.

Κώδικας:

```
if (empty($_POST["phone_number"])) {  
    $phone_number = "Phone number is required";  
} else {  
    $phone_number = test_input($_POST["phone_number"]);  
    if(preg_match("/^[0-9]{10}$/", $phone_number)){  
        $ph_ok = true;  
    }  
    $count = $count +1;  
}
```

Στην συνέχεια, ελέγχεται αν το count είναι 5 ( δόθηκε τιμή σε κάθε μεταβλητή) και αν ph\_ok είναι true.

Αν περάσουν τον παραπάνω έλεγχο γίνεται mysqli connection με τον localhost στην βάση electricitydb.

Εφόσον γίνει επιτυχώς η σύνδεση, ορίζεται μεταβλητή που περιέχει το sql query , γίνονται bind οι παράμετροι μέσω του bind\_param όπου s είναι string και i είναι integer και εκτελείται. Διαφορετικά εμφανίζεται μήνυμα πως απέτυχε η mysqli σύνδεση.

Κώδικας:

```
if($count == 5){  
    if($ph_ok){  
        $conn = new mysqli('localhost','root','','electricitydb');  
        $sql = "INSERT INTO requests (name ,phone_number ,email,subject,message) VALUES (?,?,,?,?)";  
  
        if($conn->connect_error){  
            echo "$conn->connect_error";  
            die("Connection Failed : ". $conn->connect_error);  
        } else {  
            $stmt = $conn->prepare($sql);  
            $stmt->bind_param("sisss",$name,$phone_number,$email,$subject,$message);  
            $execval = $stmt->execute();  
            $stmt->close();  
            $conn->close();  
        }  
    }  
}
```

Παρακάτω γίνεται έξοδος από την php και χρησιμοποιείται javascript.

```

    ?>
    <script type="text/javascript">
    window.location = "./interface4.html";
    window.alert("FORM SENT!");
    </script>
    <?php
}else{
    ?>
    <script type="text/javascript">
    window.location = "./interface4.html";
    window.alert("PHONE NUMBER MUST BE 10-DIGIT. PLEASE TRY AGAIN.");
    </script>
    <?php
}

}else{
    ?>
    <script type="text/javascript">
    window.location = "./interface4.html";
    window.alert("EVERY FIELD MUST BE COMPLETE. PLEASE TRY AGAIN.");
    </script>
    <?php
}

```

Στην περίπτωση που όλα τα πεδία έχουν συμπληρωθεί και το αριθμός κινητού είναι 10-ψηφία εμφανίζεται FORM SENT! και η σελίδα κάνει refresh.

Με τον ίδιο τρόπο δουλεύουν τα υπόλοιπα 2 script εμφανίζοντας διαφορετικά alert-box μηνύματα ανάλογα την περίπτωση.

Επιτυχής Καταχώρηση:

1)

**CONTACT FORM**

Sebastian Andrisan

6987044121

sebyyyandr@gmail.com

Testing This Form

Lorem Epsum blah  
mkdsamkosadmkoasmkoddkasotsbnnsd  
fiopdfsnhi

SUBMIT

2)

Ο ιστότοπος localhost λέει  
FORM SENT!

OK

3)

name	phone_number	email	subject	message
Sebastian Andrisan	6987044121	sebyyyandr@gmail.com	Testing This Form	Lorem Epsum blah mkdsamkosadmkoasmkoddkasotsbnnsd fiopdfsnhi

## Interface 5: Σελίδα Διαχείρισης

Στο κάτω μέρος του footer υπάρχει ένα μπλε κουμπί που οδηγεί τον χρήστη σε νέο interface στο οποίο καλείται να συνδεθεί με τον admin123 κωδικό.

Κώδικας html:

```
<div class="admin_page">
  <h1>ADMIN PASSWORD VERIFICATION</h1>
  <input id="password" type="password" name="password" value="" placeholder="admin-password">
  <input class="submit_input" type="submit" name="button" value="Connect" id="btn"></input>
</div>
```

Η ορθότητα του κωδικού ελέγχεται με την βοήθεια js script εντός της html.

Κώδικας ελέγχου κωδικού :

```
<script type="text/javascript">
  btn.onclick = function() {
    const x = document.getElementById("password").value;
    if(x == "admin123"){
      window.location.replace("./admin_page.php");
    }
    else
    {
      alert('Sorry, you are not the admin of this application.');
```

admin\_page.php αρχείο. Διαφορετικά εμφανίζεται σχετικό alert μήνυμα.

Επεξήγηση admin\_page.php :

Αρχικά γίνεται mysqli σύνδεση με την electricitydb του localhost.

Εφόσον γίνει επιτυχώς η σύνδεση, κάνουμε select \* from clients και μέσω του mysqli\_fetch\_array εμφανίζουμε μία-μία κάθε σειρά του \$result. Το \$result περιέχει όλα τα δεδομένα που επέστρεψε το παραπάνω sql query.

```
while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['username'] . "</td>";
    echo "<td>" . $row['name'] . "</td>";
    echo "<td>" . $row['email'] . "</td>";
    echo "</tr>";
}
echo "</table>";
```

Τέλος, εξωτερικά της php χρησιμοποιείται html με εσωτερική css για τον σχεδιασμό του background, των margin και ενός Return button που επιστρέφει τον admin στο interface 5.

Πίνακας των clients :



The screenshot shows a web application interface with a table containing client data. Below the table is a yellow button labeled 'Return'.

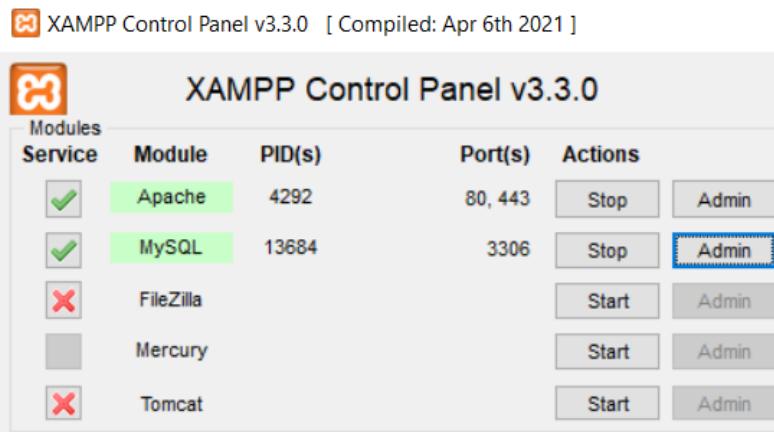
username	name	email
marios123	Marios Andrew	maranw@test.com
new_user180	New One	test@newtest.gr
seb123	Sebastian Andrisan	seb@seb.ro

[Return](#)



## Απαραίτητες ενέργειες για την σωστή λειτουργία της εφαρμογής :

- Ανοίγω το xampp ως Administrator και ενεργοποιώ τα Apache και MySQL services όπως φαίνεται παρακάτω :



- Ο κώδικας της εφαρμογής βρίσκεται στον path του xampp, εντός του φακέλου htdocs. Εκτελείται στον browser γράφοντας το path όπως παρακάτω :



*Προσοχή: Αν χρησιμοποιηθεί local file access για την εκτέλεση της εφαρμογής, τα ρηρ δεν λειτουργούν σωστά.*