

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ

HTML5 - Bootstrap - jQuery

Σπυρίδων Γκέοργκ Κούστας
06.02.2018



Περιεχόμενα

1. Εισαγωγή

- 1.1 Επικοινωνία πελάτη - διακομιστή (Client - Server Communication).....
- 1.2 HTML5 και PHP.....
- 1.3 Bootstrap.....
- 1.4 jQuery.....

2. Προετοιμασία

- 2.1 XAMPP και Sublime.....
- 2.2 Προσθήκη Βιβλιοθηκών.....

3. Ανάπτυξη

- 3.1 Ανάπτυξη της σελίδας (HTML & CSS).....
- 3.2 Προσθήκη Λειτουργικότητας.....
- 3.3 Επικοινωνία με το διακομιστή και τη βάση δεδομένων.....
- 3.4 Αποθήκευση στο local & session storage.....

4. Συμπεράσματα και πληροφορίες

- 4.1 Συμπεράσματα.....
- 4.2 Χρήσιμες Πηγές.....

Εισαγωγή

Το συγκεκριμένο εγχειρίδιο είναι μέρος εργασίας εξαμήνου του μαθήματος “Διαχείριση Έργων Πληροφορικής” του Ζ εξαμήνου, του τμήματος πληροφορικής του Ιονίου Πανεπιστημίου.

Παρακάτω γίνεται ανάλυση και περιγραφή πώς υλοποιείται μια ιστοσελίδα στο κλασικό μοντέλο πελάτη - εξυπηρετητή (client - server), κάνοντας χρήση του αποθηκευτικού χώρου που προσφέρει η HTML5 (local και session storage).

Επίσης, θα γίνει χρήση της βιβλιοθήκης jQuery, της javascript.

Η jQuery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού.

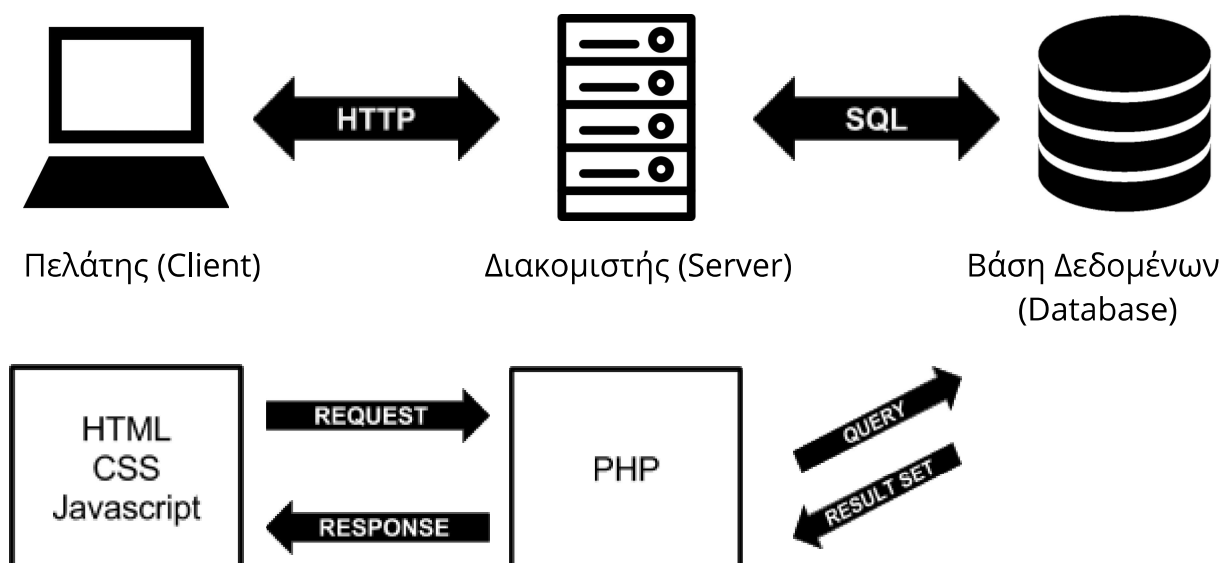
Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig).

Για ευκολία στη σχεδίαση της ιστοσελίδας και απλοποίηση του styling, θα χρησιμοποιηθεί η βιβλιοθήκη bootstrap.

1.1 Επικοινωνία πελάτη - διακομιστή

Στην επιστήμη των υπολογιστών το μοντέλο αρχιτεκτονικής λογισμικού πελάτη- διακομιστή αποτελεί μία συνήθη μέθοδο ανάπτυξης λογισμικού στην οποία ο πελάτης (ένα τμήμα λογισμικού) ζητά κάτι (π.χ. έναν πόρο, τα αποτελέσματα ενός υπολογισμού κ.ο.κ.) και ένα άλλο τμήμα λογισμικού, ο διακομιστής (ή εξυπηρετητής), του το επιστρέφει. Κάθε διακομιστής μπορεί να εξυπηρετεί πολλαπλούς πελάτες.

Στο σημείο αυτό είναι σημαντικό να ξεχωρίσουμε λίγο ποια αρχεία τρέχουν στον πελάτη και ποια στον διακομιστή και πως αυτά επικοινωνούν.



Όπως περιγράφει και το παραπάνω διάγραμμα, ο πελάτης μέσω του περιηγητή (web browser), όταν επιθυμεί να επισκεφθεί μία ιστοσελίδα (π.χ. di.ionio.gr) κάνει ένα HTTP αίτημα (HTTP Request) στο διακομιστή που φιλοξενεί την ιστοσελίδα. Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP) είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client).

Εάν ο πελάτης ζητήσει κάποια δεδομένα τα οποία είναι αποθηκευμένα σε μία βάση δεδομένων, τότε ο διακομιστής θα τα ζητήσει από την βάση, μέσω της SQL και θα τα εμφανίσει στον client.

1.2 HTML5 και PHP

Η HTML5 είναι μια υπό ανάπτυξη γλώσσα σήμανσης για τον παγκόσμιο Ιστό που όταν ετοιμαστεί θα είναι η επόμενη μεγάλη έκδοση της HTML (Γλώσσα Υπερκειμένου, HyperText Markup Language). Η HTML5 προορίζεται για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιόκτητα plug-in και πλούσιες διαδικτυακές εφαρμογές όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX.

Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστοτόπων. Είναι μια γλώσσα η οποία βρίσκεται σχεδόν σε κάθε διακομιστή. Διάσημες εφαρμογές που κάνουν εκτενή χρήση της PHP είναι το γνωστό Σύστημα Διαχείρισης Περιεχομένου (Content Management System, WordPress και το Drupal).

Όπως είδαμε και από το παραπάνω διάγραμμα στην ενότητα 1.1, τα αρχεία τύπου *.html, *.css, *.js είναι αρχεία τα οποία επιβαρύνουν τον εκάστοτε χρήστη. Αντιθέτως τα αρχεία τύπου *.php είναι αρχεία τα οποία επιβαρύνουν τον διακομιστή. Επομένως, στην ανάπτυξη των διαδικτυακών εφαρμογών, οφείλουμε να επιβαρύνουμε όσο το δυνατόν λιγότερο το διακομιστή και να εκτελεί ο χρήστης τις περισσότερες λειτουργίες. Η χρήση του server θα πρέπει να γίνεται για να επιβεβαιώνει τα αποτελέσματα.

1.3 Bootstrap

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα (Ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Είναι το πιο δημοφιλές πρόγραμμα στο GitHub και έχει χρησιμοποιηθεί από τη NASA και το MSNBC, μεταξύ άλλων.

Σύστημα πλέγματος (Grid System) και δυναμικός σχεδιασμός (responsive design).

Το Bootstrap έρχεται σταθερό με 940 pixel πλάτος. Εναλλακτικά, ο προγραμματιστής μπορεί να χρησιμοποιήσει μία μεταβλητού πλάτους διάταξη. Και για τις δύο περιπτώσεις, η εργαλειοθήκη έχει τέσσερις παραλλαγές για χρήση των διαφόρων ψηφισμάτων και τύπους συσκευών κινητά τηλέφωνα, πορτρέτα και το τοπία, tablets και υπολογιστές με χαμηλή και υψηλή ανάλυση. Κάθε παραλλαγή ρυθμίζει το πλάτος των στηλών.

Η κατανόηση του CSS στυλ.

Το Bootstrap παρέχει ένα σύνολο στυλ που παρέχουν βασικούς ορισμούς στυλ για όλα τα βασικά στοιχεία HTML. Αυτά παρέχουν ενιαία, σύγχρονη εμφάνιση για πίνακες, μορφοποίηση κειμένου, καθώς και στοιχεία μιας φόρμας.

Η Χρήση Bootstrap γίνεται ως εξής:

Μπορεί να συμπεριληφθεί σε μια ιστοσελίδα παρέχοντας το αρχείο CSS τοπικά ή έχοντας ένα σύνδεσμο σε ένα διακομιστή που το φιλοξενούν.

Αν ο προγραμματιστής θέλει να χρησιμοποιήσει τα στοιχεία JavaScript, θα πρέπει να αναφέρονται μαζί με τη βιβλιοθήκη jQuery στο HTML έγγραφο.

```
<!-- Latest compiled and minified CSS -->
<link
rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">

<!-- Latest compiled JavaScript -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootst
rap.min.js"></script>
```

1.4 jQuery

Η jQuery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα.

Τα χαρακτηριστικά της jQuery είναι τα εξής:

- DOM element επιλογές χρησιμοποιώντας την ανοιχτού κώδικα μηχανή επιλογής πολλαπλών φυλλομετρητών *Sizzle*.
- DOM διάσχιση και τροποποίηση (υποστηρίζοντας CSS 1-3)
- χειρισμός DOM βασισμένος σε CSS επιλογείς που χρησιμοποιεί τα id και class σαν κριτήρια για να κατασκευάσει επιλογείς.
- Events
- Εφέ και κινητά στοιχεία
- AJAX
- Επεκτασιμότητα μέσω plug-ins
- Μεθόδους συμβατότητας που είναι εγγενώς διαθέσιμα σε σύγχρονα προγράμματα περιήγησης.
- Υποστήριξη πολλαπλών φυλλομετρητών.

Η Χρήση της jQuery γίνεται ως εξής:

Η jQuery είναι ένα αρχείο JavaScript, που περιέχει όλες τις λειτουργίες. Μπορεί να συμπεριληφθεί σε μια ιστοσελίδα παρέχοντας το αρχείο τοπικά.

```
<script type="text/javascript" src="jquery.js"></script>
```

ή έχοντας ένα σύνδεσμο σε ένα από τους πολλούς διακομιστές που τη φιλοξενούν.

```
<script  
  src="http://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">  
</script>
```

Προετοιμασία

2.1 XAMPP και Sublime

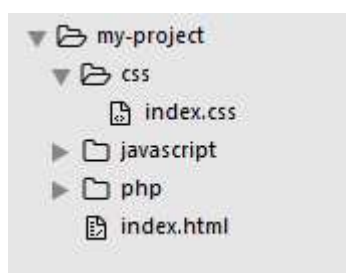
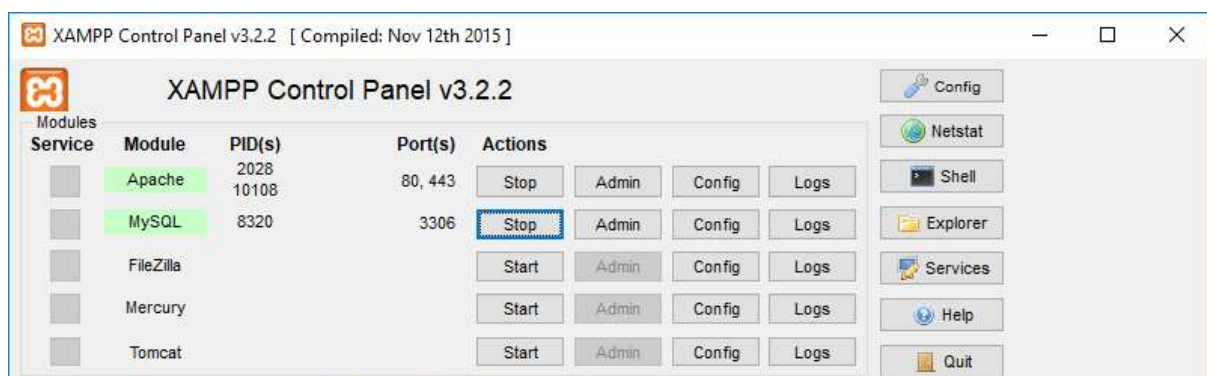
Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- **X** (αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- **A**pache HTTP εξυπηρετητής
- **M**ySQL
- **P**HP
- **P**erl

Το Sublime είναι ένας δωρεάν text editor που θα χρησιμοποιηθεί στα παρακάτω παραδείγματα.

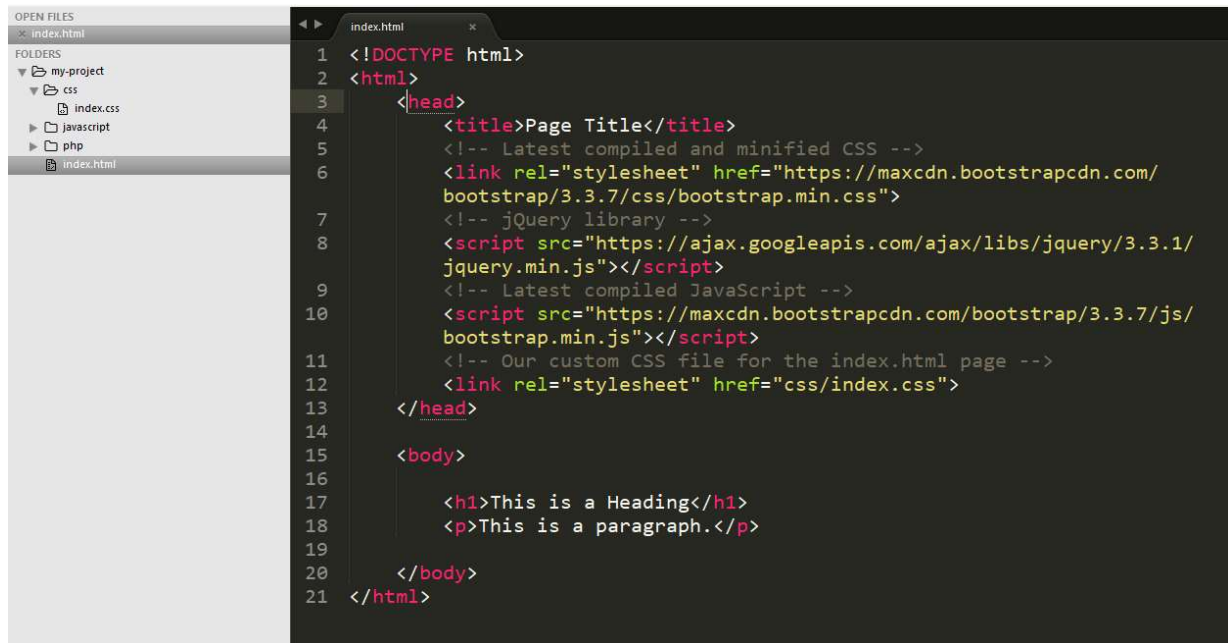
Αφού ανοίξουμε το XAMPP Control Panel και ανοίξουμε τον Apache και τη MySQL, πηγαίνουμε στο root folder πατώντας στο Explorer και επείτα στο φάκελο htdocs. Εκεί, δημιουργούμε ένα φάκελο με το όνομα του project μας.



Μέσα στο φάκελο του project μας (my-project), φτιάχνουμε μια δομή όπως στην διπλανή εικόνα. Με κάποιους κύριους φακέλους (css, javascript και php) και το αρχείο index.html και index.css. Το αρχείο index.html είναι και η αρχική μας σελίδα.

2.2 Προσθήκη Βιβλιοθηκών

Η προσθήκη των βιβλιοθηκών, γίνεται όπως είδαμε παραπάνω όταν εισάγαμε τις έννοιες jQuery και Bootstrap. Προσθέτουμε και κάποια βασικά tags της html στο αρχείο index.html και προκύπτει το εξής αρχείο.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Page Title</title>
5     <!-- Latest compiled and minified CSS -->
6     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
7       bootstrap/3.3.7/css/bootstrap.min.css">
8     <!-- jQuery library -->
9     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/
10       jquery.min.js"></script>
11     <!-- Latest compiled JavaScript -->
12     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/
13       bootstrap.min.js"></script>
14     <!-- Our custom CSS file for the index.html page -->
15     <link rel="stylesheet" href="css/index.css">
16   </head>
17   <body>
18     <h1>This is a Heading</h1>
19     <p>This is a paragraph.</p>
20   </body>
21 </html>
```

Στη γραμμή 12 του κώδικά μας, έχουμε προσθέσει το αρχείο index.css ως <link>. Σε αυτό το αρχείο θα κάνουμε τις αλλαγές που αφορούν το αρχείο index.html. Πλέον, έχουμε κάνει όλες τις απαραίτητες προσθήκες για να ξεκινήσουμε την ανάπτυξη της ιστοσελίδας μας.

Στις επόμενες ενότητες θα δούμε πως μπορούμε να εμφανίζουμε και να κρύβουμε δυναμικά, μέρη της ιστοσελίδας χωρίς να χρειάζεται να φορτώσουμε νέο αρχείο τύπου *.html, καθώς και πως γίνεται η επικοινωνία με το διακομιστή και τη βάση δεδομένων.

Ανάπτυξη

3.1 Ανάπτυξη της σελίδας (HTML & CSS)

Αυτή τη στιγμή, εάν επισκεφθούμε την ιστοσελίδα μας πληκτρολογώντας στον φυλλομετρητή `http://localhost/my-project/`, θα δούμε μία κενή html σελίδα με μία επικεφαλίδα και ένα κειμενάκι.

Έστω ότι θέλουμε να προσθέσουμε ένα κουμπί (button) στην ιστοσελίδα μας. Το bootstrap μας εφοδιάζει με αρκετές λύσεις, έτσι μπορούμε να διαλέξουμε από τα παρακάτω προκαθορισμένα κουμπιά:

Button Styles

Bootstrap provides different styles of buttons:



Εάν επιθυμούμε να αλλάξουμε κάποιο από αυτά, μπορούμε να το επεξεργαστούμε στο αρχείο `*.css` ή μέσω inline styling στο αρχείο `*.html`. Ο τρόπος με τον οποίο το bootstrap διαχωρίζει τα παραπάνω κουμπάκια, είναι κάνοντας χρήση διαφόρων κλάσεων. Έτσι τα παραπάνω κουμπιά αντιστοιχούν στις εξής κλάσεις:

```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
```

Έστω ότι θέλουμε να προσθέσουμε ένα κουμπί τύπου `btn-primary` στην σελίδα μας, τότε ο κώδικας θα τροποποιηθεί ως εξής:

```
17 <h1>This is a Heading</h1>
18 <p>This is a paragraph.</p>
19
20 <button type="button" class="btn btn-primary">Primary</button>
21
22 </body>
```

Αποθηκεύοντας τις αλλαγές μας, βλέπουμε ότι στην σελίδα μας έχει εμφανιστεί το κουμπί που επιθυμούμε.

This is a Heading

This is a paragraph.

Primary

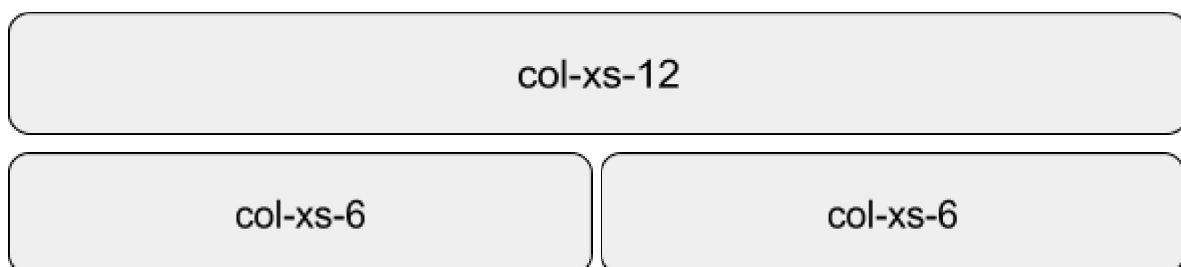
Για να αλλάξουμε τη στοίχιση των στοιχείων μέσα στη σελίδα μας, πρέπει να χωρίσουμε το περιεχόμενο κάνοντας χρήση του `<div>` tag και μετά μπορούμε να προσθέσουμε το χαρακτηριστικό (attribute) `align`.

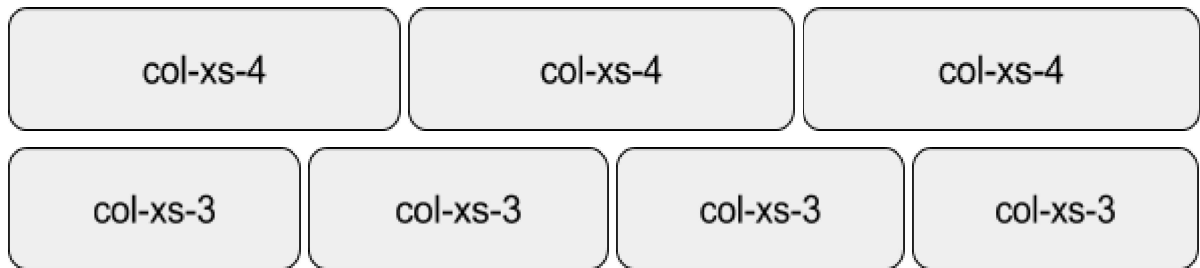
```
17 <div align="center" class="container">
18   <div class="col-xs-6">
19     <h1>This is a Heading</h1>
20     <p>This is a paragraph.</p>
21   </div>
22   <div class="col-xs-6">
23     <h1>This is a Bootstrap Button</h1>
24     <button type="button" class="btn btn-primary">Primary</button>
25   </div>
26 </div>
```

Στο παραπάνω κομμάτι κώδικα βλέπουμε να γίνεται χρήση της κλάσης `class="col-xs-6"` και `class="container"`.

Εάν παρατηρήσουμε και την στοίχιση του κώδικα βλέπουμε πως το `<div>` στη σειρά 17, αποτελεί ένα "δοχείο" (container), το οποίο περιέχει τα άλλα δύο `<div>`. Έτσι η κλάση `container`, προσθέτει μία μορφοποίηση με τις κατάλληλες αποστάσεις από τα υπόλοιπα δομικά στοιχεία που θα εμφωλευτούν σε αυτήν. Η κλάση `col-xs-6` καθορίζει τη δυναμικότητα (responsiveness) και το πλάτος που θα καταλαμβάνει το συγκεκριμένο `<div>`. Περισσότερες πληροφορίες σχετικά με το grid system του bootstrap θα βρείτε στην επίσημη ιστοσελίδα τους. Για το δικό μας παράδειγμα, αρκεί να γνωρίζουμε ότι το `col-xs-6` προορίζεται για δυναμικό design σε συσκευές έως και 320 pixel πλάτος (που είναι και η μικρότερη αναγνωρισμένη διάσταση πλάτους στην αγορά αυτή τη στιγμή) και επίσης καταλαμβάνει 6 στήλες από τις 12.

Μια ιστοσελίδα στο πλήρες πλάτος της αντιστοιχεί σε 12 στήλες, επομένως μπορούμε να έχουμε διάφορους συνδυασμούς όπως τους παρακάτω:





Οι παραπάνω συνδυασμοί μπορούν να φτάσουν έως και col-xs-1 και έτσι να έχουμε χωρίσει την οθόνη σε 12 ίσα μέρη.

Αν δούμε την ιστοσελίδα μας μετά τις αλλαγές θα παρατηρήσουμε το εξής:

This is a Heading

This is a paragraph.

This is a Bootstrap Button

Primary

Τα <div> έχουν στοιχηθεί στο κέντρο και η οθόνη μας έχει χωριστεί σε δύο ίσα μέρη.

Ας μιλήσουμε λίγο για Cascading Style Sheets (CSS). Στην css μπορούμε να προσπελάσουμε ένα tag είτε μέσω της κλάσης του, με αποτέλεσμα όλα τα στοιχεία με αυτή τη κλάση να επηρεαστούν ή μέσω ενός μοναδικού id. Κάποια δομικά στοιχεία όπως π.χ. το <body> μπορούν να επεξεργαστούν ως έχουν:

```
body {  
    background-color: lightblue;  
}
```

Σε γενικές γραμμές όμως είναι καλό να χρησιμοποιούμε κλάσεις και ids. Για παράδειγμα, εάν θέλουμε να αλλάξουμε το φόντο μόνο από το πρώτο μας <div> θα κάνουμε το εξής:

```
18 ▾ <div id="div1" class="col-xs-6">
```

Και τώρα μπορούμε να προσπελάσουμε στο αρχείο index.css που δημιουργήσαμε πιο πριν:

```
index.css  
1  #div1  
2  {  
3      background: green;  
4  }
```

Σημαντικό είναι να πούμε ότι θέλουμε να αναφερθούμε σε id στην css χρησιμοποιούμε: #{όνομα} ενώ όταν θέλουμε να αναφερθούμε σε κλάση: .{όνομα}.

Το αποτέλεσμα της παραπάνω αλλαγής είναι το εξής:



Έστω τώρα ότι θέλουμε όλες οι επικεφαλίδες τύπου `<h1>` να είναι με κόκκινη γραμματοσειρά. Τότε θα δημιουργήσουμε μία κλάση στο `index.html` και θα την επεξεργαστούμε στη συνέχεια στο αρχείο `index.css` ως εξής:

```
17 <div align="center" class="container">
18   <div id="div1" class="col-xs-6">
19     <h1 class="my-header">This is a Heading</h1>
20     <p>This is a paragraph.</p>
21   </div>
22   <div class="col-xs-6">
23     <h1 class="my-header">This is a Bootstrap Button</h1>
24     <button type="button" class="btn btn-primary">Primary</button>
25   </div>
26 </div>
```

και στο αντίστοιχο `*.css`:

```
6 .my-header
7 {
8   color: red;
9 }
```

Το αποτέλεσμα είναι το αναμενόμενο. Όλες οι επικεφαλίδες τύπου `<h1>` είναι πλέον με κόκκινη γραμματοσειρά, όπως φαίνεται και στην ιστοσελίδα μας.



3.2 Προσθήκη Λειτουργικότητας

Αφού είπαμε κάποια βασικά χαρακτηριστικά του bootstrap και της html/css πάμε να δούμε ένα πιο συγκεκριμένο παράδειγμα, προσθέτοντας και κάποια λειτουργικότητα μέσω της jQuery. Αρχικά ας αλλάξουμε τον κώδικά μας στο αρχείο index.html:

```
17 <div align="left" class="container">
18   <div class="col-xs-6">
19     <h1>This is a Bootstrap Form</h1>
20     <p>Here we will learn how to use jQuery.</p>
21   </div>
22   <div class="col-xs-6">
23     <h1>Press Button to login</h1>
24     <button type="button" class="btn btn-primary">LOGIN</button>
25   </div>
26 </div>
```

Το αποτέλεσμα είναι το εξής:

This is a Bootstrap Form

Here we will learn how to use jQuery.

Press Button to login

LOGIN

Πάμε να δημιουργήσουμε μια φόρμα login η οποία θα εμφανίζεται όταν ο χρήστης πατάει το κουμπί LOGIN.

Δημιουργούμε της φόρμα με τη βοήθεια του bootstrap και δηλώνουμε τη μέθοδο (GET) με την οποία θα αποστείλουμε τα δεδομένα μας το attribute action θα εκτελέσει την συνάρτηση myfunction() όταν πατηθεί το κουμπί Submit της φόρμας.

```
26 <div id="login-form" class="col-xs-12">
27   <form method="GET" action="javascript:myfunction()">
28     <div class="form-group">
29       <label for="email">Email address:</label>
30       <input type="email" class="form-control" id="email">
31     </div>
32     <div class="form-group">
33       <label for="pwd">Password:</label>
34       <input type="password" class="form-control" id="pwd">
35     </div>
36     <button type="submit" class="btn btn-default">Submit</button>
37   </form>
38 </div>
```


Το αποτέλεσμα είναι το εξής:

This is a Bootstrap Form

Here we will learn how to use jQuery.

Email address:

Password:

Submit

Press Button to login

LOGIN

Πάμε τώρα με την βοήθεια της jQuery να κρύψουμε την φόρμα και να την εμφανίζουμε μόνο με το πάτημα του κουμπιού LOGIN. Για να το καταφέρουμε αυτό, κάθε φορά που θα φορτώνει η σελίδα μας (attribute: onload) θα τρέχει μια συνάρτηση που θα την ονομάσουμε initialization().

Προσθέτουμε στη γραμμή 15 το εξής:

```
15 <body onload="initialization()">
```

και στη γραμμή 24:

```
24 <button type="button" class="btn btn-primary" onclick="show()">LOGIN</button>
```

και στο τέλος του κώδικά μας τις δύο javascript συναρτήσεις που θέλουμε:

```
42 <script type="text/javascript">
43   function initialization(){
44     $("#login-form").hide();
45   }
46
47   function show(){
48     $("#login-form").show();
49   }
50 </script>
```

Στις γραμμές 44 και 48 βλέπουμε για πρώτη φορά τη χρήση της jQuery η οποία κάνοντας χρήση του # αναφέρεται σε id, ενώ με τη χρήση της τελείας (.) αναφέρεται σε κλάσεις, μας επιτρέπει μια μία εύκολη εντολή να κρύβουμε και να

εμφανίζουμε το κομμάτι κώδικα που μας ενδιαφέρει, στη συγκεκριμένη περίπτωση την φόρμα login. Ο,τι γίνεται με την βοήθεια της jQuery θα μπορούσε να γίνει και με απλή javascript, απλά η jQuery μας προσφέρει πιο εύκολες και σύντομες εντολές.

Το αποτέλεσμα του κώδικά μας είναι ότι όταν φορτώνει η σελίδα μας η φόρμα δεν φαίνεται και όταν πατήσουμε το κουμπί LOGIN εμφανίζεται:

Όταν φορτώσει η σελίδα:

This is a Bootstrap Form

Here we will learn how to use jQuery.

Press Button to login

LOGIN

Αφού πατήσουμε το κουμπί LOGIN:

This is a Bootstrap Form

Here we will learn how to use jQuery.

Press Button to login

LOGIN

Email address:

Password:

Submit

Μπορεί το συγκεκριμένο παράδειγμα να μοιάζει απλό, αλλά σημασία έχει η λειτουργικότητα που παρέχει, αφού κάνοντας χρήση του `show()` και `hide()` μπορούμε να έχουμε γρήγορη διεπαφή με τον χρήστη και δεν χρειάζεται να φορτώνουμε νέα αρχεία *.html τα οποία καταναλώνουν πόρους και χρόνο. Έτσι, θα μπορούσαμε να έχουμε στήσει ένα μεγάλο μέρος της ιστοσελίδας μας με `<div>` να οποία είναι κρυμμένα και να τα εμφανίζουμε μετά από συγκεκριμένες ενέργειες του χρήστη.

3.3 Επικοινωνία με το διακομιστή και τη βάση δεδομένων

Έστω ότι θέλουμε να επικοινωνήσουμε με το διακομιστή και η φόρμα login να επιστρέφει δεδομένα όταν τα στοιχεία που έβαλε ο χρήστης είναι αληθή. Γράφουμε την εξής συνάρτηση myfunction():

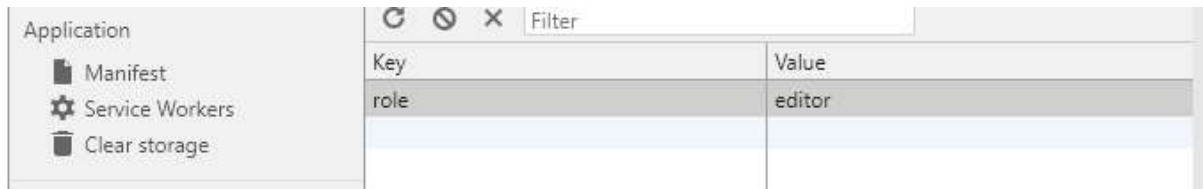
```
51 function myfunction(){
52     var email = document.getElementById("email").value;
53     var password = document.getElementById("pwd").value;
54
55     var obj, dbParam;
56     xmlhttp = new XMLHttpRequest();
57     obj = {"email":"'"+email+"'", "password":"'"+password+"'"};
58     dbParam = JSON.stringify(obj);
59     xmlhttp.onreadystatechange = function() {
60         if (this.readyState == 4 && this.status == 200) {
61             var myObj = JSON.parse(this.responseText);
62             // console.log(myObj);
63             if (myObj == "WRONG USER CREDENTIALS!") {
64                 alert("Wrong Credentials!");
65             } else {
66                 sessionStorage.role = myObj[0].role;
67                 window.location = "new.html";
68             }
69         }
70     };
71     xmlhttp.open("GET", "php/validation.php?x=" + dbParam, true);
72     xmlhttp.send(dbParam);
73 }
```

Στην παραπάνω συνάρτηση ορίζονται δύο μεταβλητές email και password, οι οποίες παίρνουν τις τιμές τους από τα πεδία της φόρμας μας.

Στη γραμμή 56 αρχικοποιούμε ένα νέο XMLHttpRequest και μετά στη γραμμή 57 και 58 δημιουργούμε ένα JSON object το οποίο περιέχει τις τιμές των μεταβλητών. Στόχος είναι να στείλουμε το JSON αντικείμενο στο διακομιστή, αυτός να ελέγξει εάν υπάρχει ζεύγος email και password στη βάση δεδομένων που επαληθεύουν τα στοιχεία του χρήστη, και να μας επιστρέψει το ρόλο του χρήστη (π.χ. user, editor, administrator).

Στις γραμμές 59-71 γίνεται η διαδικασία αποστολής του JSON object. Στη γραμμή 70 δείχνουμε σε ποιο αρχείο *.php θα το στείλουμε (θα το δημιουργήσουμε αργότερα) και με ποιο τρόπο (POST). Στις γραμμές 60-69 ελέγχουμε την απάντηση από το διακομιστή, για ενημερώνουμε το χρήστη. Στη γραμμή 66 βλέπουμε για πρώτη φορά το sessionStorage το οποίο αποθηκεύει την πρώτη θέση του JSON object που μας επέστρεψε ο διακομιστής και περιέχει το ρόλο του χρήστη, στη session μνήμη της HTML5.

Η session μνήμη παραμένει στον υπολογιστή μέχρι να κλείσει ο browser. Στη γραμμή 67 μεταφέρουμε το χρήστη σε ένα νέο αρχείο html, στην περίπτωση που η σύνδεση ήταν επιτυχής. Τότε θα εμφανιστεί και στον browser μας στο στοιχείο role στη τοπική μνήμη session storage:



Πάμε να δημιουργήσουμε το αρχείο validation.php μέσα στο φάκελο php του project μας. Σε αυτό το σημείο είναι σημαντικό να έχουμε ετοιμάσει τη βάση δεδομένων. Στο συγκεκριμένο παράδειγμα θα χρησιμοποιήσουμε ένα πίνακα με όνομα users και με πεδία email, password και role.

Έστω:

_id	email	password	role
1	uesr1@mail.com	12345	editor
2	uesr2@mail.com	98765	administrator

Σημείωση: Καλό είναι όταν σχεδιάζουμε μια βάση δεδομένων να κρυπτογραφούνται τα δεδομένα, έτσι ώστε ακόμη και ο διαχειριστής της βάσης να μην έχει πρόσβαση σε αυτά. Σε αυτό το παράδειγμα δεν θα ασχοληθούμε με την ασφάλεια.

Δημιουργούμε λοιπόν το εξής αρχείο validation.php:

```
1 <?php
2 header("Content-Type: application/json; charset=UTF-8");
3 $obj = json_decode($_GET["x"], false);
4
5 $conn = mysqli_connect("localhost","my_user","my_password","test");
6 $result = $conn->query("SELECT role FROM users WHERE email = ".$obj->email." AND
7 password = ".$obj->password);
8 $outp = array();
9 $outp = $result->fetch_all(MYSQLI_ASSOC);
10 if ($outp) {
11     echo json_encode($outp);
12 } else {
13     echo json_encode("WRONG USER CREDENTIALS!");
14 }
15 mysqli_close($conn);
16
17 ?>
```

Το αρχείο αυτό τρέχει στο διακομιστή και δεν μπορεί να επηρεαστεί εύκολα από το χρήστη.

Η γραμμή 2 απλά ορίζει το περιεχόμενο και το είδος των χαρακτήρων. Στη σειρά 3 ο server αποκωδικοποιεί το JSON object που του στείλαμε και μετά στη σειρά 5 συνδέεται στη βάση δεδομένων.

Σημείωση: Οι μεταβλητές *my_user*, *my_password* και *my_db* πρέπει να αντικατασταθούν με τα δικά σας δεδομένα του χρήστη και της βάσης.

Στη σειρά 6 έχουμε ένα απλό SQL query και στις σειρές 7 και 8 επιστρέφονται τα αποτελέσματα από τη βάση, στη μεταβλητή *\$outp*. Αφού γίνει ο έλεγχος εάν επέστρεψε κάποιο αποτέλεσμα (γραμμές 9-13), ο διακομιστής επιστρέφει σε JSON object τα δεδομένα ή μήνυμα λάθους στο χρήστη.

3.4 Αποθήκευση στο local & session storage

Με τον τρόπο που είδαμε πιο πάνω μπορούμε πλέον να επικοινωνούμε επιτυχώς με τη βάση δεδομένων και με τον διακομιστή μας. Στη περίπτωση του login αποθηκεύσαμε μόνο το είδος (role) του χρήστη στο session storage, θα μπορούσαμε όμως να είχαμε κάνει ένα μεγαλύτερο query που θα επέστρεφε περισσότερα αποτελέσματα (π.χ. προϊόντα και τιμές). Σε αυτή τη περίπτωση θα μπορούσαμε να τα είχαμε αποθηκεύσει στο local storage π.χ.

```
// Store  
localStorage.setItem("price", myObj[i].price);
```

Ο αποθηκευτικός χώρος local storage είναι περιορισμένος στα 5 MB και δεν διαγράφεται από το browser του χρήστη χωρίς εντολή. Είναι σημαντικό να μπορούμε να επικοινωνούμε με το διακομιστή επειδή οι ιστοσελίδες τρέχουν στους υπολογιστές των χρηστών. Η Javascript είναι γλώσσα που μπορεί να μεταποιηθεί από τον κάθε χρήστη, έτσι όταν αποθηκεύουμε δεδομένα στο client πρέπει να φροντίζουμε να τα επαληθεύουμε μέσω του server.

Συμπεράσματα και πληροφορίες

4.1 Συμπεράσματα

Στο συγκεκριμένο εγχειρίδιο αναλύσαμε τις έννοιες HTML5, jQuery, Bootstrap. Είδαμε πως λειτουργεί η επικοινωνία client - server και μάθαμε πως να χρησιμοποιούμε αποτελεσματικά τα παραπάνω εργαλεία για να αναπτύξουμε μια ιστοσελίδα.

Όλοι οι κώδικες που χρησιμοποιήθηκαν και το παρόν εγχειρίδιο, βρίσκονται διαδικτυακά στο github σε public repository.

GitHub: <https://github.com/SpyrosKoustas/WebDevelopment-Bootstrap-jQuery>

4.2 Χρήσιμες Πηγές

W3schools HTML: <https://www.w3schools.com/html/>

W3school Bootstrap: <https://www.w3schools.com/bootstrap/default.asp>

Bootstrap: <https://getbootstrap.com/>

Bootstrap Grid System: <https://getbootstrap.com/docs/3.3/css/>

jQuery: <https://jquery.com/>

HTML5 Storage: https://www.w3schools.com/html/html5_webstorage.asp