

Education For All

Πληροφοριακά Συστήματα Παγκόσμιου Ιστού 2020-2021

Μιχαήλ Έμιλη , 2988

Πεφτίσης Δημήτριος , 3040

Σαρρή Ευθυμία, 3124

Σωτηρίου Γεώργιος, 2414



Περιεχόμενα

1. Εισαγωγή	2
2. Αρχιτεκτονική προτεινόμενης λύσης	3
3. Περιγραφή της αρχιτεκτονικής του συστήματος.....	5
3.1 Front-End υλοποίηση	5
3.1.1 Home – Navigation bar - Footer.....	5
3.1.2 Information	6
3.1.3 Ways to Help	6
3.1.4 Learn Now	8
3.1.5 Contact us.....	9
3.1.6 Signup / Log In.....	9
3.1.7 Docs/Films/Series.....	10
3.1.8 Admin page	10
3.1.9 Μηνύματα λάθους - Validation.....	11
3.2 Back-End υλοποίηση	11
3.2.1 Η βάση δεδομένων.....	12
3.2.2 Sign Up.....	12
3.2.3 Login και dynamic navigation bar.....	13
3.2.4 Contact us	14
3.2.5 Comments	14
3.2.6 Admin page.....	14
4. Συγκρίσεις – Πειραματισμός	16
5. Μελλοντική επέκταση	16
6. Πηγές	16

1. Εισαγωγή



Στα πλαίσια του μαθήματος «Πληροφοριακά συστήματα παγκοσμίου ιστού» δημιουργήσαμε ιστοσελίδα με θέμα την ποιοτική εκπαίδευση (Quality Education) και τίτλο “Education For All”. Η εξασφάλιση δίκαιης και ποιοτικής εκπαίδευσης που δεν αποκλείει κανέναν αποτελεί έναν από τους 17 στόχους βιώσιμης ανάπτυξης (SDGs) του ΟΗΕ, οι οποίοι συμφωνήθηκαν από τους ηγέτες των κρατών - μελών του οργανισμού το 2015 για την καταπολέμηση κοινωνικών και περιβαλλοντικών προβλημάτων, όπως η φτώχεια, η κλιματική αλλαγή, η πρόσβαση στην υγεία και η δικαιοσύνη, μέχρι το 2030.

Η εκπαίδευση αποτελεί αναφαίρετο ανθρώπινο δικαίωμα και οδηγεί στην ανάπτυξη ολοκληρωμένων ανθρώπων, παίζει καθοριστικό ρόλο στην καταπολέμηση του

κοινωνικού αποκλεισμού και περιορίζει το χάσμα μεταξύ των δύο φύλων¹. Ωστόσο, σύμφωνα τον ΟΗΕ, αν και σημειώθηκε σημαντική πρόοδος όσον αφορά την πρόσβαση στην εκπαίδευση με την **φοίτηση στην πρωτοβάθμια εκπαίδευση** στις αναπτυσσόμενες χώρες να αγγίζει το 91%, 57 εκατομμύρια παιδιά εξακολουθούν να μην φοιτούν σε σχολεία² με περίπου το 50% αυτών να ζουν σε εμπόλεμες ζώνες. Παράλληλα, περισσότερα από τα μισά παιδιά που δεν φοιτούν σε σχολεία ζουν στην υποσαχάρια Αφρική και 103 εκατομμύρια νέοι παγκοσμίως παρουσιάζουν βασικές ελλείψεις γραμματισμού με το 60% των νέων να είναι γυναίκες.

Την πρόσβαση σε ποιοτική εκπαίδευση έρχεται να εμποδίσει και η έξαρση της πανδημίας του COVID-19, επηρεάζοντας 1.6 δισεκατομμύριο μαθητές σε παραπάνω από 190 χώρες³ και σε όλες τις βαθμίδες εκπαίδευσης. Σύμφωνα με δεδομένα της Unicef⁴, τουλάχιστον 463 εκατομμύρια παιδιά παγκοσμίως δεν είχαν πρόσβαση στην εξ αποστάσεως διδασκαλία κατά τη διάρκεια του lockdown του 2020, ενώ 2 στους 3 νέους δεν έχουν πρόσβαση στο διαδίκτυο από το σπίτι σύμφωνα με έρευνα της Unicef σε συνεργασία με την Διεθνή Ένωση Τηλεπικοινωνιών ITU⁵. Παράλληλα, προβλέπεται από την UNESCO ότι 23.8 εκατομμύρια επιπλέον παιδιά ίσως να εγκαταλείψουν το σχολείο ή να μην έχουν πρόσβαση στην εκπαίδευση λόγω των οικονομικών επιπτώσεων της πανδημίας⁶.

Σύμφωνα με τα παραπάνω αποφασίσαμε να ασχοληθούμε με το SDG4 – Quality Education ώστε να παρέχουμε ενημέρωση, να προτείνουμε τρόπους καταπολέμησης του προβλήματος και να κάνουμε προτάσεις για εκπαιδευτικό υλικό και υλικό για επιπλέον εμβάθυνση.

Πηγές:

1. <https://en.unesco.org/news/what-you-need-know-about-right-education>
2. <https://unric.org/el/%cf%83%cf%84%ce%bf%cf%87%ce%bf%cf%83-4-%cf%80%ce%bf%ce%b9%ce%bf%cf%84%ce%b9%ce%ba%ce%b7-%ce%b5%ce%ba%cf%80%ce%b1%ce%b9%ce%b4%ce%b5%cf%85%cf%83%ce%b7/>
3. Έκθεση ΟΗΕ για την εκπαίδευση κατά τη διάρκεια της πανδημίας του COVID-19 https://www.un.org/sites/un2.un.org/files/sg_policy_brief_covid-19_and_education_august_2020.pdf
4. Exacerbating the learning crisis: <https://data.unicef.org/covid-19-and-children/>
5. Internet access at home: <https://data.unicef.org/resources/children-and-young-people-internet-access-at-home-during-covid19/>
6. Σελ. 10: https://www.un.org/sites/un2.un.org/files/sg_policy_brief_covid-19_and_education_august_2020.pdf

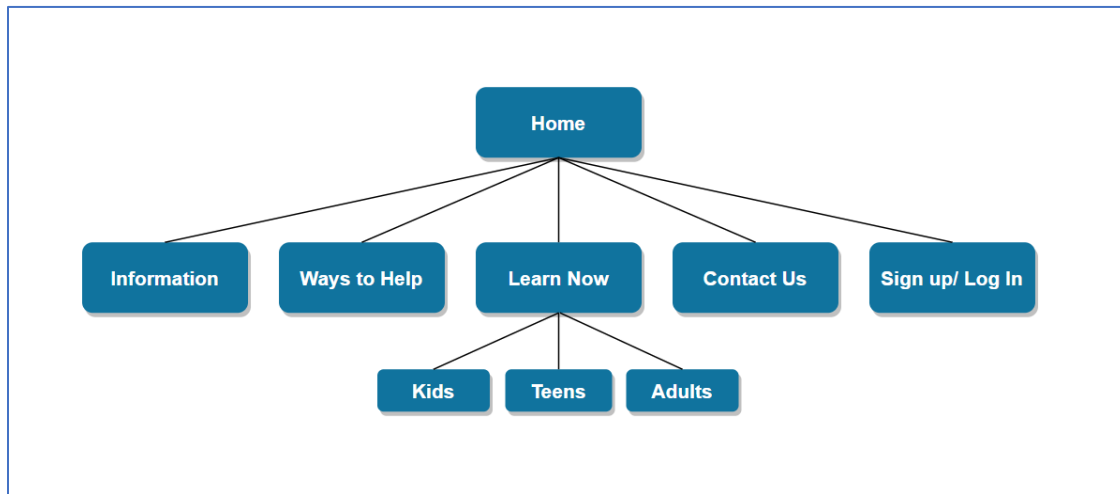
2. Αρχιτεκτονική προτεινόμενης λύσης

Η ιστοσελίδα μας αποτελείται από τις παρακάτω σελίδες:

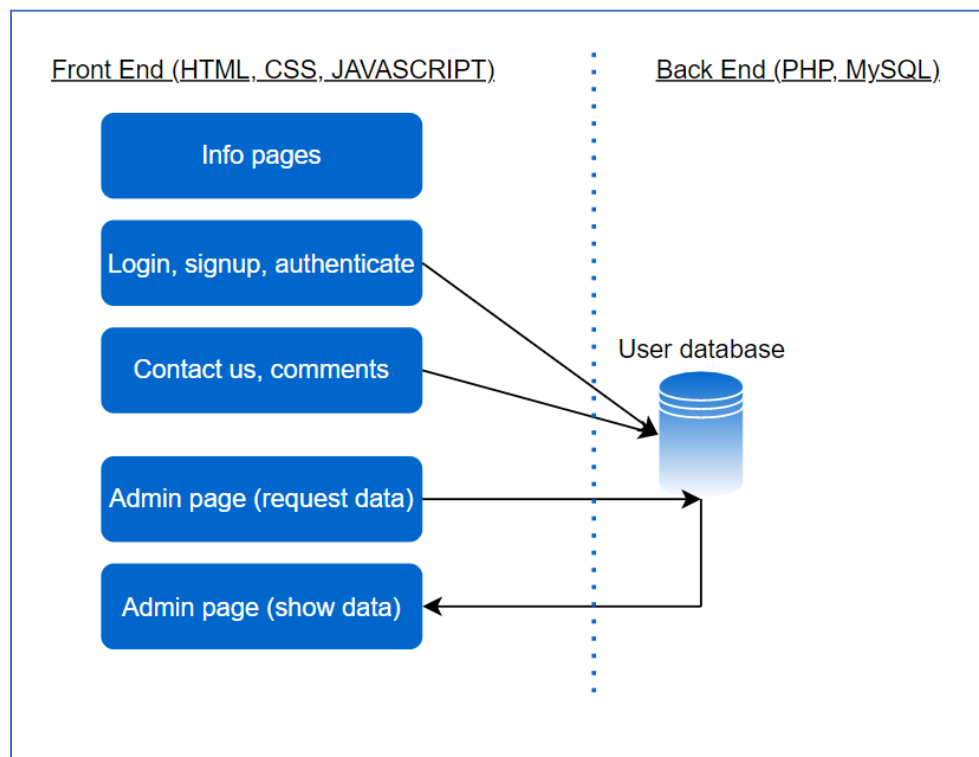
1. **Home**, όπου παρουσιάζονται βασικές πληροφορίες πάνω στο SDG 4 – Quality Education και μια παρουσίαση για το ποιοι είμαστε.
2. **Information**, όπου παρουσιάζονται πληροφορίες πάνω στην κατάσταση της εκπαίδευσης σε όλο τον κόσμο. Συγκεκριμένα, γίνεται αναφορά στα θέματα της ανισότητας των δυο φύλων, του αναλφαριθμητισμού, της συμπεριληψιμότητας (inclusivity) και τις χώρες χαμηλού εισοδήματος.
3. **Ways to Help**, όπου προτείνονται τρόποι με τους οποίους μπορεί να προωθήσει κανείς την πρόσβαση σε ποιοτική εκπαίδευση και στη διασφάλιση του δικαιώματος σε αυτή. Γίνεται διαχωρισμός σε τρία μέρη: στο **πρώτο** παρουσιάζονται τρόποι για να βοηθήσει κάποιος ανάλογα με την ιδιότητά του: εάν είναι εκπαιδευτικός, μαθητής κτλ., στο **δεύτερο** παρουσιάζονται πληροφορίες που αφορούν εθελοντική δράση ή οικονομική συνεισφορά και στο **τρίτο** γίνονται ενδεικτικές προτάσεις κάποιων ΜΚΟ.
4. **Learn Now**, όπου γίνονται προτάσεις για ιστοσελίδες που παρέχουν δωρεάν εκπαιδευτικό υλικό σε μορφή παιχνιδιών, youtube videos ή online courses. Γίνεται διάκριση σε τρεις κατηγορίες: προτάσεις για παιδιά, εφήβους και ενήλικες.
5. **Docs/Films/Series**, όπου γίνονται προτάσεις σε ταινίες, σειρές και ντοκιμαντέρ σχετικά με το θέμα της εκπαίδευσης.
6. **Contact us**, όπου ο επισκέπτης μπορεί να επικοινωνήσει με τους διαχειριστές της ιστοσελίδας εισάγοντας το όνομά του, το email του, το τηλέφωνό του και το μήνυμά του.
7. **Signup/Login**, όπου πραγματοποιείται η είσοδος του χρήστη στο λογαριασμό του (Log In) στην ιστοσελίδα μας ή η εγγραφή του (Sign Up) σε αυτή. Για το Log In απαιτείται είσοδος email και κωδικού πρόσβασης και για το Sign Up ζητούνται το Username, το email και ο κωδικός πρόσβασης.

Για κάθε μία από τις παραπάνω σελίδες υλοποιήθηκε ξεχωριστό αρχείο .php και .css . Επιπλέον, σε ξεχωριστά αρχεία υλοποιήθηκαν το navigation bar και το footer της ιστοσελίδας που είναι κοινά σε όλες τις σελίδες (navbar.php, footer.php και index.css) .

Στο back end κομμάτι υλοποιήθηκε βάση δεδομένων χρηστών όπου αποθηκεύονται δεδομένα σχετικά με τους χρήστες και την αλληλεπίδρασή τους με τον ιστότοπο (δεδομένα register, μηνύματα, σχόλια, πληροφορίες για τους διαχειριστές) και σελίδα διαχειριστή όπου παρουσιάζονται τα δεδομένα από την βάση. Μεταξύ της βάσης και του ιστοτόπου εγκαθιδρύσαμε σύνδεση με PHP ώστε να στέλνουμε τα δεδομένα στην βάση και από την βάση να παρουσιάζουμε τα δεδομένα στην σελίδα διαχείρισης.



Εικόνα 1: Διάγραμμα αρχιτεκτονικής front end του συστήματος



Εικόνα 2: Διάγραμμα αρχιτεκτονικής του συστήματος

3. Περιγραφή της αρχιτεκτονικής του συστήματος

3.1 Front-End υλοποίηση



Εικόνα 1: Το navigation bar και το header.

Για το front-end κομμάτι της ιστοσελίδας χρησιμοποιήθηκαν τεχνολογίες HTML, CSS, Javascript και Bootstrap. Επιπλέον, δόθηκε προσοχή και στο responsiveness της ιστοσελίδας με χρήση των css media queries.

Στο πρώτο μέρος της εργασίας δημιουργήθηκαν ξεχωριστά αρχεία .html και .css για κάθε κατηγορία. Στο δεύτερο μέρος της εργασίας τα αρχεία .html μετατράπηκαν σε αρχεία .php ώστε να συμπεριλάβουμε και λειτουργίες που υλοποιούνται με PHP. Αρχικά όλες οι σελίδες περιλάμβαναν κώδικα html για το navigation bar και τον footer. Πλέον ο κώδικας για τα παραπάνω βρίσκεται σε δύο ξεχωριστά αρχεία nav.php και

footer.php τα οποία κάνουμε include στην σελίδα με php.

Το στυλ των παραπάνω βρίσκεται στο αρχείο index.css, το οποίο κάνουμε link σε κάθε σελίδα. Τέλος, κάθε σελίδα περιλαμβάνει κώδικα για το κουμπί που οδηγεί στην αρχή της σελίδας.

Επιπλέον, οι σελίδες Home, Information, Ways to Help και Learn Now περιλαμβάνουν κώδικα για τον header, ο οποίος έχει την μορφή εικόνας με κείμενο και button(s).

3.1.1 Home – Navigation bar - Footer

Η αρχική σελίδα Home περιλαμβάνει το navigation bar, ένα slider τριών εικόνων, ένα quote, δύο rows με κείμενο και εικόνες και ένα κουμπί (top button) που οδηγεί στην κορυφή στην σελίδας. Το **navigation bar** περιλαμβάνει όλες τις σελίδες στις οποίες μπορεί να πλοηγηθεί κανείς στην ιστοσελίδα μας. Υλοποιήθηκε σε ένα section header και nav tags. Αριστερά υπάρχει το όνομα της σελίδας σε div tags και δεξιά οι διαθέσιμες σελίδες σε ul tags.

```

<!--IMAGE SLIDER-->
<section class="img-slider">
<div class="slide active">
</div>
<div class="slide">
</div>
<div class="slide">
</div>
<div class="navigation">
</div>
</section>

```

Το **slider** αποτελείται από ένα section (img-slider) όπου βρίσκονται οι 3 εικόνες και μια navigation κλάση με τρία κουμπιά για τη χειροκίνητη αλλαγή των εικόνων. Επίσης, οι εικόνες εναλλάσσονται αυτόματα κάθε δέκα (10) δευτερόλεπτα με τη χρήση της JavaScript.

Στο **κυρίως μέρος** της βρίσκεται μια κλάση (container) με δυο παραγράφους και δυο εικόνες και ένα container-fluid με μια σειρά και μια στήλη που περιλαμβάνει ένα quote σε εικόνα με parallax effect.

Στο τέλος της σελίδας είναι το **υποσέλιδο** (footer) με βασικές πληροφορίες και τους τρόπους επικοινωνίας. Αποτελείται από 3 rows, το **αριστερό τμήμα** περιέχει ένα header με τον τίτλο (About

Us), μια παράγραφο με τις πληροφορίες και μια κλάση social όπου θα βρούμε συνδέσμους με τη μορφή εικονιδίων που πήραμε με τη χρήση bootstrap και μας οδηγούν στα μέσα κοινωνικής δικτύωσης. Το **κεντρικό τμήμα** (Address) αποτελείται από την κλάση place όπου σύμφωνα με το bootstrap που χρησιμοποιήσαμε και στο προηγούμενο βήμα, μας δίνει τα εικονίδια map, email, phone τα οποία με τη σειρά τους μας οδηγούν στις αντίστοιχες διευθύνσεις. Στο **δεξί τμήμα** του υποσέλιδου βρίσκεται μια φόρμα επικοινωνίας (Contact Us) αποτελούμενη από 3 κλάσεις τύπου text και ένα κουμπί τύπου button για την αποστολή του μηνύματος.

3.1.2 Information

Για την δημιουργία του information χρησιμοποιήθηκε κυρίως το grid system του Bootstrap. Η εικόνα στο header είναι κάτω από ένα section μαζί με το navbar έτσι ώστε να παίρνει 100% της οθόνης (100vh) ανεξάρτητου μεγέθους, όταν κάποιος μπει στην σελίδα. Επίσης χρησιμοποιήθηκαν css properties στην εικόνα για να δημιουργηθεί parallax effect.

Για τον τίτλο πάνω στην εικόνα χρησιμοποιήθηκε Bootstrap flex container κεντράροντάς το με CSS (height 50%, top 17%). Το button κάτω από την εικόνα είναι επίσης bootstrap με την διαφορά ότι έχει γίνει αλλαγή στο hover effect και transition duration.

Για το content της σελίδας χρησιμοποιήθηκαν containers (rows και columns) bootstrap.

3.1.3 Ways to Help

Η σελίδα Ways to Help, εκτός από τον header και τον footer, αποτελείται από τρία βασικά μέρη. Το **πρώτο** είναι το ways-to-help και περιλαμβάνει πληροφορίες σε text boxes με εικόνες, το **δεύτερο** είναι το take-action-now και περιλαμβάνει μερικά στατιστικά και πληροφορίες σε κείμενο και το **τρίτο** το recommendations που περιλαμβάνει ένα carousel με εικόνες με κουμπί.

```

<!--INTRO-->
<section class="header">
</section>
<div class="text-box">
</div>
<div class="intro">
</div>
<!--INTRO-->

```

box μία εικόνα `img` και κείμενο σε παράγραφο.

Το δεύτερο μέρος αποτελείται από ένα `section` με `class="did-you-know"` και ένα bootstrap container. Το `section` περιλαμβάνει ένα container με τρεις στήλες και κάθε στήλη περιλαμβάνει εικόνα και κείμενο .

```

<!--INTRO-->
<div id="recommendations-carousel"
class="carousel slide"
data-interval="false">
<div class="carousel-inner">
<div class="carousel item active">
<section class="recommendations">
<div class="container-fluid">
<div class="row">
<div class="col-xs-4">
<div class="container">
<img../>

```

```

<div class="middle">
<div class="button-text">
<a..> Text </a>

```

```

</div>
</div>
</div>
</div>

```

```

* * *

```

Εισαγωγικά, η σελίδα `Ways to Help` αποτελείται από το `header`, ένα `div` με `class="text-box"`, το οποίο περιλαμβάνει κείμενο με εικόνα στο `background` και ένα `div` με `class="intro"` που περιλαμβάνει μια εισαγωγική παράγραφο.

Ακολουθεί το **πρώτο μέρος**, το οποίο υλοποιήθηκε σε ένα `<section class="ways-to-help">` . Αποτελείται από ένα bootstrap container το οποίο έχει δύο σειρές (`<div class="rows">`) με δύο στήλες σε κάθε σειρά αντίστοιχα. Κάθε σειρά περιλαμβάνει δύο ορθογώνια `boxes (boxbox)` και κάθε

Το τρίτο μέρος αποτελείται από ένα bootstrap carousel slide με `id recommendations-carousel` το οποίο περιλαμβάνει δυο `carousel-items` και δύο βελάκια (κουμπιά) **previous** και **next** . Κάθε carousel-item περιλαμβάνει ένα bootstrap container με μία σειρά και τέσσερις στήλες. Κάθε στήλη περιλαμβάνει ένα bootstrap container με εικόνα στην οποία κατά το `hover` εμφανίζεται κουμπί το οποίο οδηγεί σε εξωτερική ιστοσελίδα. Ενδεικτικά, στην διπλανή εικόνα φαίνεται ένα carousel item, το **carousel item active**, το **container-fluid** που περιέχει ένα **row**, μία από τις τέσσερις στήλες **col-xs-4** και τα περιεχόμενα αυτής (ένα **container με εικόνα** και ένα `div middle` που περιλαμβάνει το **κουμπί** και το **κείμενο** πάνω σε αυτό.

3.1.4 Learn Now

Η σελίδα Learn Now διακρίνεται σε τρία κομμάτια (προτάσεις για παιδιά, για εφήβους και για ενήλικες) και περιλαμβάνει τον header με μία εικόνα με κείμενο και τρία κουμπιά που οδηγούν στις αντίστοιχες κατηγορίες στην ιστοσελίδα.

```
<!--FIRST SECTION-ABOUT THE TOPLIST -->
<section class="toplist-wrapper">
  <div class="text-container">
    <div class="main-text">
      <h2>About the Toplist.</h2>
      <p>People in 2021...</p>
    </div></div>
  </div>
</section>
```

Το **κυρίως τμήμα** της σελίδας υλοποιήθηκε σε main tags και ξεκινάει με ένα **wrapper** (toplist-wrapper) με λευκό background. Μέσα σε αυτόν έχουμε έναν **τίτλο** (h2-about the toplist) και μια μικρή παράγραφο.

```
<!--CARDS AREA-->
<section class="cards-area">
  <section class="cards-area" id="kids-section">
    <div class="wrapper">
      </div>
    <div class="cards-area-container">
      <!-- CARD 1 -->
      <div class="card">
        <div class="card-header">
          </div>
        <div class="card-body">
          </div>
        <div class="card-body">
          </div>
        <div class="card-footer">
          </div>
        </div>
      <!-- CARD 2 -->
      <div class="card">
        </div>
      <!-- CARD 3 -->
      <div class="card">
        </div>
    </div>
  </section>
  ...
</section>
```

Ακολουθούν οι προτάσεις οι οποίες υλοποιήθηκαν σε ένα **section cards-area** που περιλαμβάνει τρία cards-area sections με id **kids-section**, **teenagers section** και **adults section**. Καθένα από τα sections είναι ένα **container** που περιέχει τον τίτλο και τρεις κάρτες. Κάθε κάρτα υλοποιήθηκε σε ένα div card και περιλαμβάνει το card header, το card body και το card footer. Σε κάθε κάρτα υπάρχει μια προτεινόμενη **ιστοσελίδα με link** σε αυτήν και μια σύντομη περιγραφή της. Στην κορυφή κάθε κάρτας αποφεύχθηκε η χρήση εικόνων των αντίστοιχων σελίδων λόγω copyrights και έτσι χρησιμοποιήθηκαν γενικευμένες εικόνες. Ενδεικτικά, ο κώδικας αφορά την κατηγορία kids και φαίνονται οι τρεις κάρτες και τα components τους.

3.1.5 Contact us

```
<!--CONTACT US SECTION -->
<div class="contact-section">
  <div class="contact-wrap">
    <h1>Contact Us</h1>
    <form class="contact-form" action="index.html" method="post">
      <input type="text" class="contact-form-text" placeholder="Your name">
      <input type="email" class="contact-form-text" placeholder="Your email">
      <input type="text" class="contact-form-text" placeholder="Your phone">
      <textarea class="contact-form-text" placeholder="Your message"></textarea>
      <button class="btn" type="submit">Submit</button>
    </form>
  </div>
</div>
```

Η σελίδα Contact us αποτελείται από μία εικόνα στο background και μια κάρτα contact-section. Η κάρτα ανήκει σε ένα container (contact-wrap) και διαθέτει τίτλο και την φόρμα σε form tags. Η φόρμα (contact-form) διαθέτει inputs (για όνομα, email, τηλέφωνο ,μήνυμα) και κουμπί για υποβολή με τα οποία ο χρήστης μπορεί να επικοινωνήσει με την ομάδα του Education for All.

3.1.6 Signup / Log In

```
<!--CONTACT US SECTION -->
<section class="register">
  <div class="container">
    <div class="card">
      <div class="inner-box" id="card">
        <div class="card-front">
        </div>
        <div class="card-back">
        </div>
      </div>
    </div>
  </div>
</div>
<script> ... </script>
</section>
```

Η σελίδα Signup/Log In υλοποιήθηκε σε αρχείο με όνομα register και περιλαμβάνει ένα section register μέσα στο οποίο υπάρχει ένα bootstrap container με ένα div card που αποτελεί την φόρμα log in/ sign up και περιλαμβάνει δύο cards: τη **card-front** και τη **card-back**. Η **front** είναι η φόρμα για **Log In** και περιλαμβάνει δύο inputs (email και password), remember me checkbox, κουμπί Log in και Create a new account, "Forgot Password" και "Go to Home Page". Στην επιλογή create new account η κάρτα γυρνάει με χρήση Javascript και εμφανίζεται η φόρμα για εγγραφή με την είσοδο ονόματος χρήστη (username), email και κωδικού (password) και κουμπί "Create your account" και "I have an account".

3.1.7 Docs/Films/Series

Η σελίδα Docs/Films/Series υλοποιήθηκε στο αρχείο films.php. Αποτελείται από ένα container με id = "list" το οποίο περιλαμβάνει δέκα σειρές (rows). Κάθε σειρά αποτελείται από ένα τίτλο, ένα div που περιλαμβάνει iframe και κείμενο.

3.1.8 Admin page

Το front end κομμάτι της σελίδας διαχείρισης (admin page) αποτελείται από τέσσερα sections.

Το πρώτο είναι το stats section. Υλοποιήθηκε στο αρχείο stats.php και περιλαμβάνει ένα container fluid με ένα row και τρία columns. Κάθε column περιλαμβάνει ένα box, καθένα από τα οποία περιλαμβάνει κείμενο και ένα icon.

```
//admin-page.php
<table class='content-table' id="comment-
table" style='margin-left: 50px; margin-
bottom: 100px;'>
  <thead>
    <tr>
      <th>ID</th>
      <th>Page ID</th>
      <th>Username</th>
      <th>Comment</th>
      <th>Manage</th>
    </tr>
  </thead><tbody>
    <?php include("../table-comments.php") ?>
  </tbody>
</table>
```

Τα υπόλοιπα τρία sections είναι τα userTable, commentTable και messageTable και περιλαμβάνουν το καθένα κείμενο σε h3 tags, φόρμα αναζήτησης και πίνακα όπου παρουσιάζονται αντίστοιχα δεδομένα από την βάση δεδομένων χρηστών.

Κάθε φόρμα αναζήτησης υλοποιήθηκε σε input tags με class searchBox και διαφορετικό id. Για την αναζήτηση χρησιμοποιείται η συνάρτηση searchTable η οποία δέχεται ως ορίσματα το id του search bar, το id του πίνακα προς αναζήτηση και 3 παραμέτρους που είναι οι δείκτες των στηλών που θέλουμε να γίνεται αναζήτηση.

Κάθε πίνακας υλοποιήθηκε σε table tags, όπου οι header βρίσκονται σε thead tags και τα δεδομένα σε tbody tags. Τα περιεχόμενα

κάθε σειράς βρίσκονται σε td tags και υλοποιήθηκαν σε ξεχωριστά αρχεία (table-users.php, table-messages.php, table comments.php) όπου συνδυάζονται με το back end κομμάτι.

Ενδεικτικά παρατίθεται ο κώδικας για το table με τα σχόλια των χρηστών.

3.1.9 Μηνύματα λάθους - Validation

```
//register-validation.php
$(document).ready(function(){
$("#register-form").validate({
  rules:
  {
    username:{
      required: true,
      minlength: 3,
      maxlength: 50,
      remote: {
        url: 'Server.php',
        type: 'post'
      }
    }, ...
  }, ...
});
```

Σε περίπτωση που το login ή το signup δεν γίνουν με επιτυχία λόγω λανθασμένων credentials ή επειδή τα στοιχεία που έδωσε ο χρήστης κατά την εγγραφή υπάρχουν ήδη στην βάση, εμφανίζονται αντίστοιχα μηνύματα λάθους. Για τον σκοπό αυτό χρησιμοποιήθηκε plugin της jquery για να γίνει το validate στις παραπάνω φόρμες. (περισσότερα <https://jqueryvalidation.org/>)

Συγκεκριμένα, χρησιμοποιήθηκε η μέθοδος validate η οποία χρησιμοποιεί το **αντικείμενο rules** για να δημιουργήσει **κανόνες** για το κάθε πεδίο (πχ username, email, password) και το **αντικείμενο messages** για να παρέχει **custom μηνύματα σφαλμάτων** για το κάθε πεδίο αντίστοιχα. Ενδεικτικά, η τιμή true

σημαίνει ότι το πεδίο πρέπει να είναι υποχρεωτικά συμπληρωμένο για να περάσει ένα input το validate. Εδώ αξίζει επίσης να αναφερθεί η μέθοδος **remote**, η οποία στέλνει jquery.ajax request (XMLHttpRequest) στον server. Ο server, σκοπός του οποίου είναι να απαντήσει κάνοντας echo ένα json string με τιμή true ή false, υλοποιήθηκε σε αρχείο Server.php και ελέγχει εάν το input υπάρχει στη βάση ώστε να στείλει την αντίστοιχη τιμή στη μέθοδο remote.

Ενδεικτικά, παρατίθεται κώδικας από το αρχείο register-validation.js, όπου φαίνεται ένα αντικείμενο rules με τους κανόνες που ορίσαμε για το username και το ajax request που κάνουμε στον server, το οποίο θα επιστρέψει true ή false.

3.2 Back-End υλοποίηση

Το back-end κομμάτι της εργασίας υλοποιήθηκε στο περιβάλλον XAMPP με τεχνολογίες PHP, MYSQL, ajax jquery και javascript. Συγκεκριμένα, δημιουργήσαμε φάκελο με όνομα PSPI τον οποίο βάλαμε στον φάκελο \xampp\htdocs.

Ολοκληρώθηκαν οι λειτουργίες που αφορούν τη δημιουργία λογαριασμού χρήστη (**signup**), την είσοδο του χρήστη στο λογαριασμό του (**login**) και την επικοινωνία με την σελίδα (**contact us**). Επιπλέον, υλοποιήθηκε λειτουργία που δίνει στο χρήστη τη δυνατότητα να αφήσει **σχόλια** σε κάθε μια από τις σελίδες της ιστοσελίδας, σελίδα για την **διαχείριση** των χρηστών, των σχολίων και των μηνυμάτων από χρήστες οι οποίοι έχουν οριστεί **διαχειριστές**.

3.2.1 Η βάση δεδομένων

Για την υλοποίηση των παραπάνω λειτουργιών δημιουργήθηκε **βάση δεδομένων** χρηστών με όνομα **'users'** η οποία περιέχει τα **tables admin, user, message και comments** που θα αναλυθούν παρακάτω. Για την πρόσβαση στην βάση δημιουργήθηκε χρήστης με username psri και password psri2021 με όλα τα privileges.

- Το **table 'user'** αποτελείται από τα πεδία id (κύριο κλειδί), username (UNIQUE), email(UNIQUE), password και reg_date (ημερομηνία εγγραφής). Όλα τα πεδία είναι not null .
- Το **table 'admin'** αποτελείται από τα πεδία id (κύριο κλειδί), userID (ξένο κλειδί), που είναι το id του user που έχουμε ορίσει διαχειριστή.
- Το **table 'comments'** αποτελείται από τα πεδία
 - id (κύριο κλειδί)
 - page_id, το id της κάθε σελίδας όπως το έχουμε ορίσει εμείς
 - parent_id, το id του σχολίου στο οποίο το συγκεκριμένο σχόλιο ενδεχομένως να είναι reply. Εάν δεν είναι, το parent_id έχει default τιμή -1.
 - name, το όνομα του χρήστη
 - content, το περιεχόμενο του σχολίου
 - submit_date, το timestamp στο οποίο έγινε η δημοσίευση του σχολίου.
- Τέλος, το **table 'message'** αποτελείται από τα πεδία id, name, email, phone, message και reg_date. Δεν έγινε σύνδεση μέσω ξένου κλειδιού με κάποιον άλλο πίνακα διότι μπορεί κάποιο μήνυμα να είναι από μη εγγεγραμμένο χρήστη.

3.2.2 Sign Up

```
$password_hash = password_hash($password,
PASSWORD_BCRYPT);
$sqlQuery = "INSERT INTO user (username,
email, password, reg_date)
VALUES ('$username', '$email',
'$password_hash', CURRENT_TIMESTAMP)";
if (mysqli_query($connection, $sqlQuery))
{
    $_SESSION['isLoggedIn'] = true;
    header('Location: ../home.php');
} else {
    echo "Error: " . $sqlQuery . "<br>" .
mysqli_error($connection);
}
}
```

Η λειτουργία για το signup υλοποιήθηκε στο αρχείο register.php με χρήση της μεθόδου POST στο αρχείο όπου βρίσκεται το front end κομμάτι ώστε να στέλνονται τα δεδομένα που έδωσε ο χρήστης. Αρχικά, ξεκινάει ένα session και γίνεται η σύνδεση στην βάση. Στην global μεταβλητή \$_SESSION αποθηκεύονται το username του χρήστη και μια boolean μεταβλητή isLoggedIn η οποία δείχνει αν ο χρήστης είναι συνδεδεμένος στον λογαριασμό του. Από την μέθοδο POST παίρνουμε τα δεδομένα που έδωσε ο χρήστης στη φόρμα και τα καθαρίζουμε από whitespaces, lashes και html special characters. Τέλος, κρυπτογραφούμε τον κωδικό με χρήση του PASSWORD_BCRYPT και στέλνουμε τα δεδομένα στη βάση με ένα sql query. Εάν το username ή το email υπάρχουν ήδη στη βάση, εμφανίζεται αντίστοιχο μήνυμα στον χρήστη και δεν επιτρέπεται η εγγραφή.

3.2.3 Login και dynamic navigation bar

```
$checkPass =
"SELECT * FROM `user` WHERE email='$email' LIMIT 1";
$checkPassQuery =
mysqli_query($connection, $checkPass);
if(mysqli_num_rows($checkPassQuery) == 1){
$result = mysqli_fetch_assoc($checkPassQuery);
$hashedPassword = $result['password'];
$verify =
password_verify($password, $hashedPassword);
if($verify){
    $_SESSION['isLogged'] = true;
    $_SESSION['username'] = $result['username'];
    $_SESSION['id'] = $result['id'];
    header('Location: ../home.php');
}
```

Στο login η διαδικασία είναι παρόμοια με το register. Ξεκινάμε το session και αποθηκεύουμε τις μεταβλητές isLogged, isValid, username και id στο global variable ώστε να τις χρησιμοποιήσουμε αργότερα στον κώδικα. Γράφουμε και **εκτελούμε το query στην βάση** και στη συνέχεια **ελέγχουμε αρχικά εάν υπάρχει στην βάση το email** που δόθηκε και, αν υπάρχει, **ελέγχουμε αν είναι σωστός και ο κωδικός** με την συνάρτηση password_verify, η οποία συγκρίνει τον κωδικό που δόθηκε με τον κρυπτογραφημένο κωδικό που είναι αποθηκευμένος στην βάση. Εάν το Login γίνει με επιτυχία ο χρήστης ανακατευθύνεται στην σελίδα home και αλλάζει το navigation bar. Σε κάθε άλλη περίπτωση εμφανίζεται μήνυμα λάθους.

```
//navbar.php
<li><?php include('nav/logged-nav.php');?></li>
//nav/Logged-nav.php
if(isset($_SESSION['isLogged'])){
    $id = $_SESSION['id'];
    $adminQuery =
"SELECT * FROM `admin` WHERE userID='$id' LIMIT 1";
    $checkAdminQuery =
mysqli_query($connection, $adminQuery);
    if($_SESSION['isLogged'] === true){
        echo " //echo html code for user options
...<li>"
        . ( mysqli_num_rows($checkAdminQuery) == 1 ?
        "<li><a href='../admin-page.php'>Admin</a> </li>"
        : "" ) . " //echo html code ";
    }
    else{
        echo "<a href='Register.php'> . 'Sign Up/
Log In' . "</a>";
    }
}
```

Για την περίπτωση σύνδεσης του χρήστη στον λογαριασμό του υλοποιήθηκε δυναμικό navigation bar το οποίο εμφανίζει στον χρήστη το username του με επιλογές να δει το προφίλ του και να αποσυνδεθεί από τον λογαριασμό του. Εάν ο χρήστης είναι διαχειριστής μπορεί επιπλέον να δει και την σελίδα διαχειριστή (admin page). Η λειτουργία αυτή υλοποιήθηκε στο αρχείο logged-nav.php το οποίο κάνουμε include στο αρχείο navbar.php που περιέχει τον κώδικα HTML του navigation bar.

Στο logged-nav.php αρχικά ελέγχουμε αν η μεταβλητή isLogged που έχουμε ορίσει στο login/register του χρήστη έχει τιμή. Στη συνέχεια παίρνουμε μέσω του session και το id του χρήστη ώστε να ελέγξουμε στον πίνακα admin αν ο χρήστης είναι διαχειριστής και με χρήση του τριαδικού τελεστή '?:' ελέγχουμε αν θα εμφανίζεται το admin page.

3.2.4 Contact us

Η υλοποίηση του συστήματος contact us έγινε σε αρχείο με αντίστοιχο όνομα. Αρχικά, αφού συνδεθούμε στη βάση, ελέγχουμε εάν έχει οριστεί η μεταβλητή POST, δηλαδή αν έχει πατηθεί το κουμπί submit της φόρμας. Στη συνέχεια κάνουμε validate το input του χρήστη και στέλνουμε τα δεδομένα στη βάση με χρήση της συνάρτησης INSERT INTO. Εάν το μήνυμα στάλθηκε επιτυχώς, εμφανίζεται αντίστοιχο μήνυμα στον χρήστη. Για την εμφάνιση του μηνύματος χρησιμοποιήσαμε την βιβλιοθήκη sweetalert.

3.2.5 Comments

Η υλοποίηση του συστήματος comments έγινε στο αρχείο comments.php με τεχνολογίες PHP, MySQL και AJAX. Αρχικά γίνεται σύνδεση στη βάση και αρχικοποιούνται 2 μεταβλητές: η \$loggedIn και \$name όπου στην περίπτωση που ο χρήστης είναι συνδεδεμένος αποθηκεύεται true και το όνομα του χρήστη. Στη συνέχεια υλοποιήθηκε η μέθοδος show_comments, η οποία αποθηκεύει τα σχόλια και τα replies σε μεταβλητή html για την εμφάνισή τους με την χρήση loop, και η μέθοδος show_write_comment_form η οποία τυπώνει σε κώδικα html τα σχόλια. Επίσης, γίνεται έλεγχος για να βρούμε σε ποια από τις σελίδες ανήκει το κάθε σχόλιο.

3.2.6 Admin page

Στο admin page οι διαχειριστές μπορούν να έχουν μία ολοκληρωμένη εικόνα για τους χρήστες και ό,τι αφορά την διάδρασή τους με τον ιστότοπο. Συγκεκριμένα παρουσιάζονται:

1. ο συνολικός αριθμός εγγεγραμμένων χρηστών, των σχολίων και των μηνυμάτων,
 2. πίνακας με τα στοιχεία των εγγεγραμμένων χρηστών, με δυνατότητα διαγραφής χρήστη και ορισμού χρήστη ως διαχειριστή
 3. πίνακας με όλα τα σχόλια με δυνατότητα διαγραφής σχολίων
 4. πίνακας με όλα τα εισερχόμενα μηνύματα
-
1. Ο **συνολικός αριθμός των δεδομένων** από τους πίνακες user, comments και message υπολογίζεται στο αρχείο **stats.php**. Ο υπολογισμός γίνεται με ένα query στον κάθε πίνακα με χρήση της συνάρτησης COUNT() και στη συνέχεια τα αποτελέσματα αποθηκεύονται σε arrays και τυπώνονται μέσα σε HTML κώδικ με την εντολή echo.

```
//user-table.php
<a id='setAdmin' href=
"user-interaction\set-admin.php?userID =
<?php echo $result[0]; ?>">
Make Admin </a> |
<a id='deleteUser' href="
user-interaction\delete-user.php?userID =
<?php echo $result[0]; ?>">
Delete</a>
//set-admin.php
$IDUser=$_GET['userID'];
$query = "INSERT INTO admin (userID)
VALUES ('$IDUser')";
$querySetAdmin= mysqli_query($connection, $query);
//delete-user.php
$IDUser=$_GET['userID'];
$query = "DELETE FROM `user` WHERE `id` = $IDU
ser";
$setAdmin = mysqli_query($connection, $query);
```

```
//delete-user.php
function deleteComments($commentID, $connection){
$parentIdQuery = "SELECT `id` FROM `comments`
WHERE `parent_id` IN ($commentID)";
$parentId=mysqli_query($connection, $parentIdQuery)
;
if( mysqli_num_rows($parentId) == 0){
} else {
$result = mysqli_fetch_array($parentId);
foreach ($result as &$value){
deleteComments($value, $connection);
}
$del = mysqli_query($connection,
"DELETE FROM `comments` WHERE `id` = $commentID " )
;
echo "deleted comment with id '$commentID'<br>";
echo "</br> Exiting...";
}
```

2. Ο πίνακας που παρουσιάζει τα δεδομένα των χρηστών υλοποιήθηκε στο αρχείο **user-table.php**. Επιπλέον, στα αρχεία **set-admin.php** και **delete-user.php** υλοποιήθηκαν ο ορισμός διαχειριστή και η διαγραφή χρήστη αντίστοιχα.

Στο αρχείο user-table.php υπάρχει ο κώδικας HTML για το front end κομμάτι του πίνακα και κώδικας php σε php tags για το back end κομμάτι. Σε αυτό το κομμάτι εκτελούμε ένα query στον πίνακα των χρηστών και παίρνουμε τα id, username και email του χρήστη τα οποία αποθηκεύουμε σε array και τυπώνουμε μέσα κατάλληλα HTML tags. Για την διαχείριση των χρηστών χρησιμοποιούμε anchor tags μέσα στα οποία έχουμε references στα αντίστοιχα php αρχεία όπου κάνουμε echo με php το id του χρήστη.

Στη συνέχεια παίρνουμε αυτό το id μέσω της global μεταβλητής \$_GET και το χρησιμοποιούμε για να κάνουμε το query στην βάση.

3. Παρόμοια με το user-table υλοποιήθηκε και ο πίνακας με τα σχόλια στο αρχείο comment-table.php. Με query στην βάση παίρνουμε το id, το page id, το name και το content κάθε σχολίου, τα αποθηκεύουμε σε array και τα κάνουμε echo με μία δομή επανάληψης while μέσα σε καταλληλά HTML tags ώστε να εμφανίζονται ως σειρές σε πίνακα.

Η διαγραφή σχολίου υλοποιήθηκε στο αρχείο delete-comment.php με μία συνάρτηση deleteComments η οποία δέχεται ως ορίσματα το id του σχολίου (\$commentID) προς διαγραφή και την σύνδεση στη βάση (\$connection). Ξεκινώντας από το id του σχολίου που θέλουμε να διαγράψουμε, ψάχνουμε αν το σχόλιο έχει replies, δηλαδή αν το id του είναι parent_id για κάποιο άλλο σχόλιο στη βάση. Αναδρομικά ψάχνουμε και για τα replies. Όταν βρούμε σχόλιο που δεν έχει replies, μπορούμε να το διαγράψουμε. Αναδρομικά διαγράφουμε και τα υπόλοιπα σχόλια.

4. Συγκρίσεις – Πειραματισμός

Η ιστοσελίδα Education For All δίνει μία πιο σφαιρική και σύντομη ενημέρωση πάνω στο SDG που αφορά την ποιοτική εκπαίδευση. Παρέχει γενικές πληροφορίες σχετικά με το SDG ενώ η σελίδα Information παρέχει πιο ειδικές πληροφορίες πάνω στην κατάσταση της εκπαίδευσης σε όλο τον κόσμο σχετικά με διάφορα θέματα. Παράλληλα, η σελίδα Ways To Help έρχεται να δώσει μια πιο ολοκληρωμένη εικόνα σχετικά με τους τρόπους που μπορεί να στηρίξει κανείς την πρόσβαση στην εκπαίδευση συγκεντρώνοντας υλικό από διάφορες και έγκυρες πηγές και δίνοντας θεωρητικές και πρακτικές συμβουλές και προτάσεις. Η συγκεκριμένη σελίδα εμπνεύστηκε από παρόμοιες σελίδες άλλων ιστοσελίδων* και αποτελεί βελτίωση αυτών καθώς περιέχει εμπλουτισμένο υλικό και επιπλέον προτάσεις/πηγές για ενημέρωση. Τέλος, η σελίδα Learn Now έρχεται να ενθαρρύνει την online μάθηση ως ένα μέσο σε ποιοτική και προσβάσιμη εκπαίδευση κάνοντας προτάσεις έγκυρων ιστοσελίδων με δωρεάν υλικό σε παιδιά, εφήβους και ενήλικες.

* <https://www.savethechildren.org/us/ways-to-help>, <https://en.unesco.org/news/what-can-i-do-help-advance-right-education>

5. Μελλοντική επέκταση

Μελλοντικά θα μπορούσε να υλοποιηθεί σελίδα profile για τους χρήστες, όπου θα μπορούν να επεξεργάζονται τα στοιχεία τους και να έχουν πρόσβαση σε προσωποποιημένο υλικό. Επιπλέον, θα θέλαμε να εξασφαλίσουμε περισσότερη ασφάλεια στη σύνδεση στη σελίδα διαχείρισης με χρήση json web tokens, καθώς μέχρι τώρα η σύνδεση στη συγκεκριμένη σελίδα γίνεται μόνο μέσω του session, και στη εγγραφή των χρηστών με ενσωμάτωση validation του email. Κατ' επέκταση θα μπορούσαμε να υλοποιήσουμε έναν πιο έγκυρο και ασφαλή τρόπο αλλαγής κωδικού πρόσβασης και email/username από τον χρήστη.

6. Πηγές

Information:

1. <https://ourworldindata.org/global-education>
2. <https://www.unicef.org/education>
3. <https://restavekfreedom.org/2018/06/14/schools-around-the-world-how-does-education-differ-from-country-to-country/>
4. <https://www.globalpartnership.org/results/education-data-highlights>
5. <https://www.humanium.org/en/right-to-education/>

Ways to Help:

1. <https://gem-report-2020.unesco.org/poll/>
2. <https://gem-report-2020.unesco.org/recommendations/>
3. <https://www.right-to-education.org/news/covid-19-and-right-education-collated-resources>
4. <https://soeonline.american.edu/blog/5-ways-policy-makers-can-improve-the-quality-of-education>
5. <https://link.springer.com/article/10.1007/s10972-009-9180-5>
6. <https://tomprof.stanford.edu/posting/418#:~:text=Peer%20learning%20should%20be%20mutually,and%20experience%20between%20the%20participants.&text=They%20develop%20skills%20in%20organizing,and%20evaluating%20their%20own%20learning>
7. <https://www.savethechildren.org/us/ways-to-help>

MKO:

1. <https://www.savethechildren.net/what-we-do/learning#>
2. <https://www.unicef.org/take-action>
3. <https://www.unv.org/>
4. <https://www.volunteerworld.com/en/volunteer-abroad/youth-teaching>
5. https://aiesec.org/search?sdg_goals=11104&earliest_start_date=2021-05-18
6. <https://www.goeco.org/tags/volunteer-for-education>
7. <https://donate.street-child.org/gb-en/a-future-for-every-child>
8. <https://www.children.org/>

LEARN NOW:

1. Funbrain: <https://en.wikipedia.org/wiki/FunBrain>
2. Starfall: [https://en.wikipedia.org/wiki/Starfall_\(website\)](https://en.wikipedia.org/wiki/Starfall_(website))
3. Coolmath4kids: <https://www.coolmath4kids.com/>
4. Khan Academy: https://en.wikipedia.org/wiki/Khan_Academy
5. Hippocampus: <https://www.hippocampus.org/>
6. CK-12 Foundation: https://en.wikipedia.org/wiki/CK-12_Foundation
7. Coursera: <https://en.wikipedia.org/wiki/Coursera>
8. Udemy: <https://en.wikipedia.org/wiki/Udemy>
9. edX: <https://en.wikipedia.org/wiki/EdX>

EIKONEΣ - ICONS:

1. <https://unsplash.com/s/photos/education>
2. <https://freeicons.io/>
3. <https://images.google.com/>

