

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΨΗΦΙΑΚΟ ΜΑΡΚΕΤΙΝΓΚ (DIGITAL MARKETING)

ΤΕΧΝΙΚΗ ΕΚΘΕΣΗ ΕΡΓΑΣΙΑΣ

# WDI.P1 Σχεδιασμός διεπαφής website που περιλαμβάνει μια διαδικτυακή υπηρεσία

ΤΩΝ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΦΟΙΤΗΤΩΝ

ΔΗΜΗΤΡΑ ΓΥΜΝΟΠΟΥΛΟΥ  
ΡΟΖΑ ΜΑΡΙΑ ΜΑΣΣΑΡΟ

Καβάλα 2026

# Περιεχόμενα

Περιεχόμενα	2
Κατάλογος Σχημάτων	3
Κεφάλαιο 1 Ανάπτυξη Website	4
1.1. Γενικά	4
1.2. Δημιουργία Header-Footer	6
1.3. Μενού πλοήγησης τουλάχιστον δύο επιπέδων	9
1.4. Το κεντρικό μήνυμα που θέλει να περάσει η επιχείρηση	11
1.5. Άλλες σημαντικές πληροφορίες ή λειτουργίες που προσφέρονται παράλληλα	14
1.6. Slideshow	16
1.7. Ημερολόγιο	19
Κεφάλαιο 2 Παρουσίαση η διαχείριση δεδομένων	25

# Κατάλογος Σχημάτων

Σχήμα 3. 1: Η δομή των αρχείων φωτογραφίας του website	4
Σχήμα 3. 2: Τα αρχεία των σελίδων της ιστοσελίδας	5
Σχήμα 3. 3: Η βασική οργάνωση του main.html	5
Σχήμα 3. 4: Το header της ιστοσελίδας	6
Σχήμα 3. 5: Ο κώδικας του header	7
Σχήμα 3. 6: Ο HTML κώδικας των Admin & Guest buttons	7
Σχήμα 3. 7: Ο κώδικας Javascript για την εναλλαγή των admin/guest mode	8
Σχήμα 3. 8: Το footer της σελίδας	8
Σχήμα 3. 9: Το navigation bar της ιστοσελίδας	9
Σχήμα 3. 10: Η δομή του κώδικα HTML του navigation bar	10
Σχήμα 3. 11: Στιγμιότυπο λειτουργίας του dropdown menu	10
Σχήμα 3. 12: Ο κώδικας HTML του dropdown menu	11
Σχήμα 3. 13: Στιγμιότυπο της τοποθεσίας του slogan της επιχείρησης	12
Σχήμα 3. 14: Ο κώδικας HTML του slogan	12
Σχήμα 3. 15: Η CSS του slogan	13
Σχήμα 3. 16: Οι ρυθμίσεις διαδραστικότητας στη CSS της σελίδας	14
Σχήμα 3. 17: Στιγμιότυπο της σελίδας προϊόντων	15
Σχήμα 3. 18: Ο κώδικας της συνάρτησης renderProducts	15
Σχήμα 3. 19: Ο HTML κώδικας του slideshow	16
Σχήμα 3. 20: Στιγμιότυπο της σελίδας "Ποιοι είμαστε"	17
Σχήμα 3. 21: Η CSS του κειμένου της σελίδας "Ποιοι είμαστε"	17
Σχήμα 3. 22: Η CSS του slideshow	18
Σχήμα 3. 23: Η HTML του floating calendar	19
Σχήμα 3. 24: Στιγμιότυπο λειτουργίας του ημερολογίου	20
Σχήμα 3. 25: Η επιλογή των στοιχείων με Javascript	20
Σχήμα 3. 26: Η CSS του Calendar	24
Σχήμα 3. 27: Στιγμιότυπο λειτουργίας της σελίδας "Ραντεβού"	25
Σχήμα 3. 28: Παράδειγμα εισαγωγής δεδομένων μέσω Javascript	26
Σχήμα 3. 29: Ο κώδικας Javascript της renderBookings	27
Σχήμα 3. 30: Ο κώδικας Javascript των mock buttons	28
Σχήμα 3. 31: Παράδειγμα λειτουργίας του κουμπιού διαγραφής	28
Σχήμα 3. 32: Στιγμιότυπο της λειτουργίας του modal	28
Σχήμα 3. 33: Ο κώδικας HTML του modal	29



```
⟨ Athens.html
⟨ Companyinfo.html
⟨ Haircut.html
⟨ Hairdye.html
⟨ Hairstyling.html
# main.css
⟨ main.html
⟨ products.html
⟨ rendezvous.html
⟨ Thessaloniki.html
```

Για την εμφάνιση και τη μορφοποίηση του περιεχομένου χρησιμοποιήθηκε CSS, με όλες τις σελίδες να φορτώνουν το ίδιο αρχείο στυλ (main.css). Με αυτόν τον τρόπο εξασφαλίζεται ομοιομορφία στην αισθητική του site, καθώς κοινά στοιχεία όπως το header, το navigation bar και το footer παραμένουν ίδια σε όλες τις σελίδες. Επιπλέον, αξιοποιήθηκε το framework Bootstrap 5 για responsive σχεδίαση και έτοιμα UI components, όπως το navbar, το carousel και τα κουμπιά.

Για την προσθήκη λειτουργικότητας και διαδραστικών στοιχείων χρησιμοποιήθηκαν scripts σε JavaScript. Ενδεικτικά, υλοποιήθηκε δυναμικό carousel εικόνων καθώς και ένα αιωρούμενο ημερολόγιο το οποίο εμφανίζεται με το πάτημα κουμπιού και προβάλλει την τρέχουσα ημερομηνία. Τα scripts αυτά ενισχύουν την εμπειρία χρήστη και καθιστούν τον ιστότοπο πιο λειτουργικό και σύγχρονο.

```
<!DOCTYPE html>
<html>
  <!-- Ο εισαγωγικός κόμβος head επαναλαμβάνεται σε όλες τις σελίδες -->
  <head>...
  </head>
  <!-- Η κεφαλίδα επαναλαμβάνεται σε όλες τις σελίδες -->
  <header class="bg-light py-3">...
  </header>
  <!-- Το μενού δυο διαστάσεων επαναλαμβάνεται σε όλες τις σελίδες -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">...
  </nav>

  <!-- Το στοιχείο body είναι το μόνο που αλλάζει δυναμικά σε όλες τις σελίδες .html -->
  <body>...
  </body>

  <!-- Footer επαναλαμβάνεται σε όλες τις σελίδες -->
  <footer class="bg-dark text-light pt-4">...
  </footer>
</html>
```

Συνολικά, το project ακολουθεί μια οργανωμένη δομή αρχείων, συνδυάζοντας HTML, CSS και JavaScript, με στόχο τη δημιουργία ενός ολοκληρωμένου, καλαίσθητου και εύχρηστου επαγγελματικού website για κομμωτήρια.

Για την δημιουργία δυνατότητας διαχείρισης ραντεβού στον ιστότοπο, υλοποιήθηκε μια **εικονική (dummy) βάση δεδομένων** με τη μορφή δομής δεδομένων σε JavaScript. Η βάση

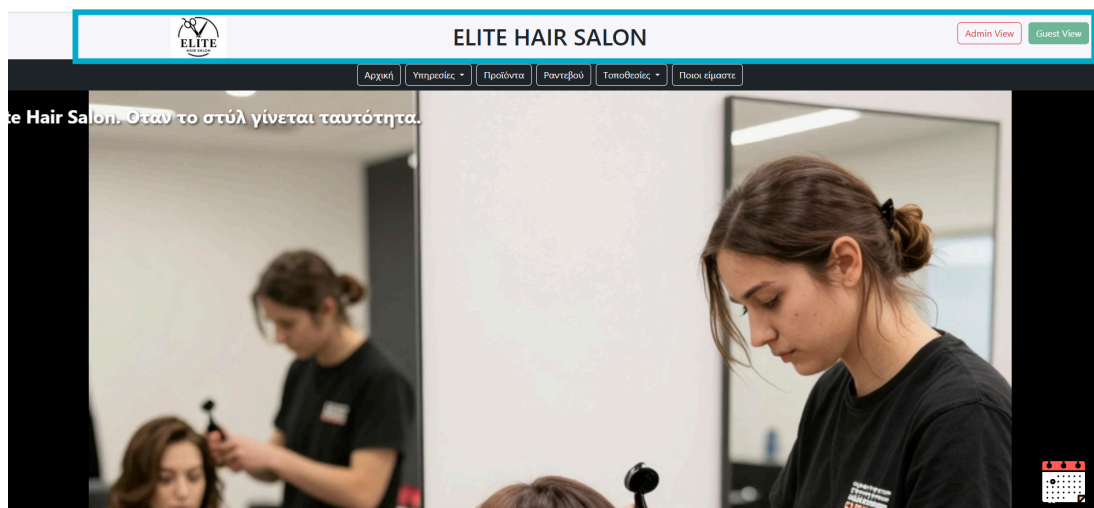
αυτή περιλαμβάνει προκαθορισμένες εγγραφές κρατήσεων, όπως όνομα πελάτη, υπηρεσία, ημερομηνία, ώρα, τηλέφωνο και κατάσταση ραντεβού. Η χρήση dummy δεδομένων επιτρέπει την προσομοίωση της πραγματικής λειτουργίας ενός συστήματος κρατήσεων χωρίς τη χρήση backend ή πραγματικής βάσης δεδομένων, γεγονός που είναι ιδανικό για εκπαιδευτικούς σκοπούς και για την επίδειξη της λειτουργικότητας του site.

Η παρουσίαση των ραντεβού προσαρμόζεται δυναμικά ανάλογα με το επιλεγμένο **view (Admin ή Guest)**, το οποίο αποθηκεύεται στο localStorage του browser. Στη λειτουργία Admin προβάλλονται όλες οι κρατήσεις πελατών, μαζί με επιπλέον δυνατότητες διαχείρισης, όπως προσθήκη, επεξεργασία και διαγραφή ραντεβού (σε mock μορφή). Αντίθετα, στη λειτουργία Guest εμφανίζονται μόνο τα ραντεβού που αντιστοιχούν στον συγκεκριμένο χρήστη, προσφέροντας μια πιο περιορισμένη και ρεαλιστική προβολή. Με αυτόν τον τρόπο προσομοιώνεται η διαφοροποίηση δικαιωμάτων χρήστη και αναδεικνύεται η δυναμική συμπεριφορά της εφαρμογής μέσω JavaScript.

Για τις ανάγκες της παρούσας εργασίας, μέρος του οπτικού υλικού που χρησιμοποιείται στον ιστότοπο (εικόνες και γραφικά) έχει παραχθεί με τη βοήθεια εργαλείων **Τεχνητής Νοημοσύνης (AI)**. Οι εικόνες αυτές χρησιμοποιούνται αποκλειστικά για εκπαιδευτικούς και ενδεικτικούς σκοπούς και δεν αντιπροσωπεύουν πραγματικά πρόσωπα ή πραγματικούς χώρους της επιχείρησης.

Η χρήση τους αποσκοπεί στην καλύτερη αισθητική παρουσίαση του project και στην πληρέστερη κατανόηση της λειτουργικότητας του ιστότοπου.

## 1.2. Δημιουργία Header-Footer



Σχήμα 3. 4: Το header της ιστοσελίδας

Η κεφαλίδα (header) του ιστότοπου αποτελεί βασικό δομικό στοιχείο που επαναλαμβάνεται σε όλες τις σελίδες, εξασφαλίζοντας συνοχή, αναγνωρισιμότητα και σταθερή εμπειρία χρήστη. Η υλοποίησή της βασίζεται σε σύγχρονες τεχνικές HTML5 σε συνδυασμό με το framework Bootstrap 5, το οποίο προσφέρει έτοιμες κλάσεις για διάταξη, στοίχιση και responsive συμπεριφορά. Η χρήση της κλάσης bg-light προσδίδει ουδέτερο και επαγγελματικό φόντο, ενώ

η py-3 προσθέτει κατακόρυφο padding, βελτιώνοντας την οπτική άνεση και την ευαναγνωσιμότητα του περιεχομένου.

```
<!-- Η κεφαλίδα επαναλαμβάνεται σε όλες τις σελίδες -->
<header class="bg-light py-3">
  <div class="container d-flex align-items-center justify-content-center position-relative">
    <!-- Logo (top-left) -->
    
    <!-- Centered Title -->
    <h1 class="text-center m-0">ELITE HAIR SALON</h1>
  </div>

  <!-- admin/guest view -->
  <div class="view-toggle position-absolute top-0 end-0 p-3">...
</div>
<script>...
</script>
</header>
```

Στο εσωτερικό της κεφαλίδας χρησιμοποιείται η κλάση container, η οποία ορίζει ένα σταθερό πλάτος περιεχομένου και διασφαλίζει σωστή προσαρμογή σε διαφορετικά μεγέθη οθόνης. Παράλληλα, οι κλάσεις d-flex, align-items-center και justify-content-center αξιοποιούνται για τη δημιουργία ενός flexbox layout, επιτρέποντας την κάθετη και οριζόντια ευθυγράμμιση των στοιχείων. Ο συνδυασμός αυτός εξασφαλίζει ότι ο τίτλος του ιστότοπου παραμένει πάντα κεντραρισμένος, ανεξάρτητα από την παρουσία επιπλέον στοιχείων στην κεφαλίδα.

Το λογότυπο της επιχείρησης τοποθετείται στρατηγικά στην επάνω αριστερή πλευρά της κεφαλίδας μέσω της χρήσης της κλάσης position-absolute. Με αυτή την επιλογή αποφεύγεται η μετατόπιση του τίτλου και διατηρείται σταθερή η διάταξη του header. Το λογότυπο λειτουργεί ως βασικό στοιχείο branding, ενισχύοντας την ταυτότητα του Elite Hair Salon και προσφέροντας οπτική συνέχεια σε όλες τις σελίδες του site.

```
<!-- admin/guest view -->
<div class="view-toggle position-absolute top-0 end-0 p-3">
  <button id="adminBtn" class="btn btn-outline-danger me-2">
    Admin View
  </button>
  <button id="guestBtn" class="btn btn-outline-success">
    Guest View
  </button>
</div>
```

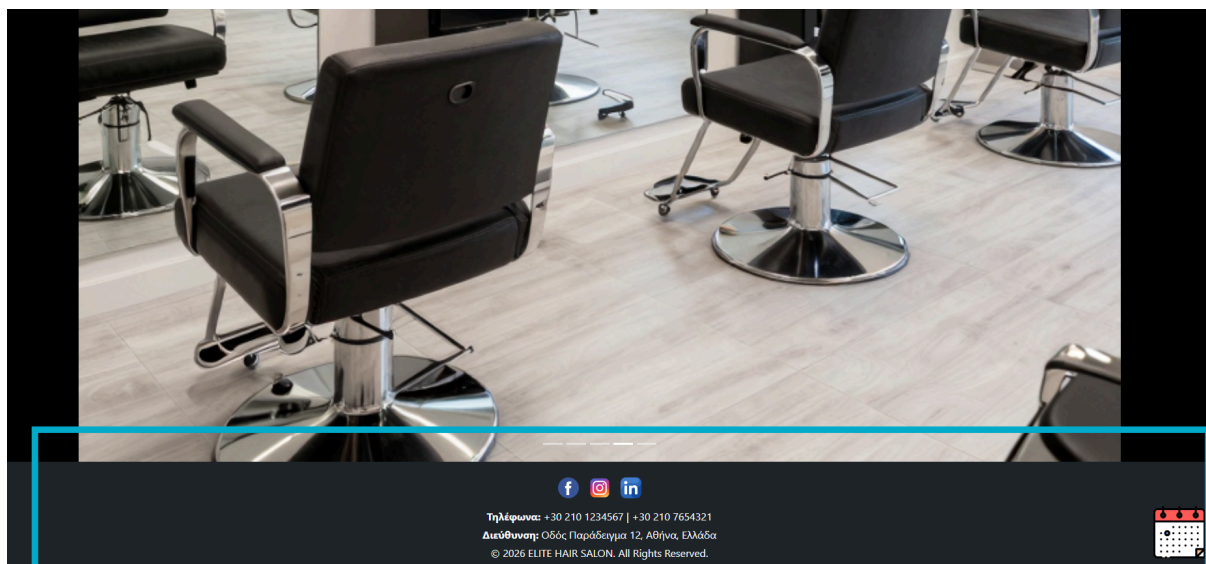
Ιδιαίτερη σημασία έχει ο μηχανισμός εναλλαγής προβολής μεταξύ Admin View και Guest View, ο οποίος είναι ενσωματωμένος στην κεφαλίδα ώστε να είναι άμεσα προσβάσιμος από οποιαδήποτε σελίδα. Τα κουμπιά αξιοποιούν Bootstrap κλάσεις όπως btn, btn-outline-danger και btn-outline-success, επιτρέποντας την ξεκάθαρη χρωματική διαφοροποίηση των ρόλων.

Μέσω JavaScript, η επιλογή του χρήστη αποθηκεύεται στο localStorage, γεγονός που επιτρέπει τη διατήρηση της επιλεγμένης προβολής ακόμα και μετά από ανανέωση ή αλλαγή σελίδας.

```
34 <script>
35 const adminBtn = document.getElementById('adminBtn'); // αναζήτηση του admin button μέσω Id.
36 const guestBtn = document.getElementById('guestBtn'); // αναζήτηση του guest button μέσω Id.
37
38 function applyStyles(mode) {
39     localStorage.setItem('viewMode', mode);
40
41     if (mode === 'admin') {
42         adminBtn.disabled = true; // η τιμή disabled καθορίζει αν το κουμπί είναι δυνατόν να δεχτεί click
43         guestBtn.disabled = false;
44
45         adminBtn.classList.add('btn-danger'); // εφαρμόζει την κλάση 'btn-danger' η οποία κοκκινίζει το κουμπί
46         adminBtn.classList.remove('btn-outline-danger'); // αφαιρεί την κλάση 'btn-outline-danger' η οποία αφαιρεί το κόκκινο περίγραμμα
47
48         guestBtn.classList.add('btn-outline-success'); // εφαρμόζει την κλάση 'btn-outline-success' η οποία πρασινίζει το περίγραμμα του κουμπιού
49         guestBtn.classList.remove('btn-success'); //πρασινίζει το κουμπί
50     } else { //αντίστοιχη η λειτουργία αν πατηθεί το κουμπί Guest View
51         guestBtn.disabled = true;
52         adminBtn.disabled = false;
53
54         guestBtn.classList.add('btn-success'); //πρασινίζει το κουμπί Guest
55         guestBtn.classList.remove('btn-outline-success'); // αφαιρείται η κλάση 'btn-outline-success' η οποία πρασινίζει το περίγραμμα του κουμπιού
56
57         adminBtn.classList.add('btn-outline-danger'); // εφαρμόζει την κλάση 'btn-outline-danger' η οποία αφαιρεί το κόκκινο περίγραμμα
58         adminBtn.classList.remove('btn-danger'); // αφαιρεί την κλάση 'btn-danger' η οποία κοκκινίζει το κουμπί
59     }
60 }
61
62 adminBtn.addEventListener('click', () => {applyStyles('admin');window.location.reload();});
63 guestBtn.addEventListener('click', () => {applyStyles('guest');window.location.reload();});
64
65 const savedMode = localStorage.getItem('viewMode') || 'guest'; // επιλέγει την τιμή του viewMode από το localStorage ή αν δεν υπάρχει πάρε την default τιμή "guest"
66 applyStyles(savedMode); // μορφοποιεί τα κουμπία ανάλογα με την τιμή
67 </script>
```

Σχήμα 3. 7: Ο κώδικας Javascript για την εναλλαγή των admin/guest mode

Η δυναμική συμπεριφορά των κουμπιών ενισχύεται με την προσθαφαίρεση Bootstrap κλάσεων, όπως btn-danger και btn-success, οι οποίες μεταβάλλουν την εμφάνιση και την ενεργή κατάσταση κάθε κουμπιού. Παράλληλα, η ιδιότητα disabled χρησιμοποιείται για να αποτρέπεται η επαναλαμβανόμενη επιλογή της ήδη ενεργής προβολής. Με αυτόν τον τρόπο προσομοιώνεται μια βασική μορφή διαχείρισης ρόλων χρήστη, χωρίς την ανάγκη backend συστήματος ή μηχανισμού αυθεντικοποίησης.

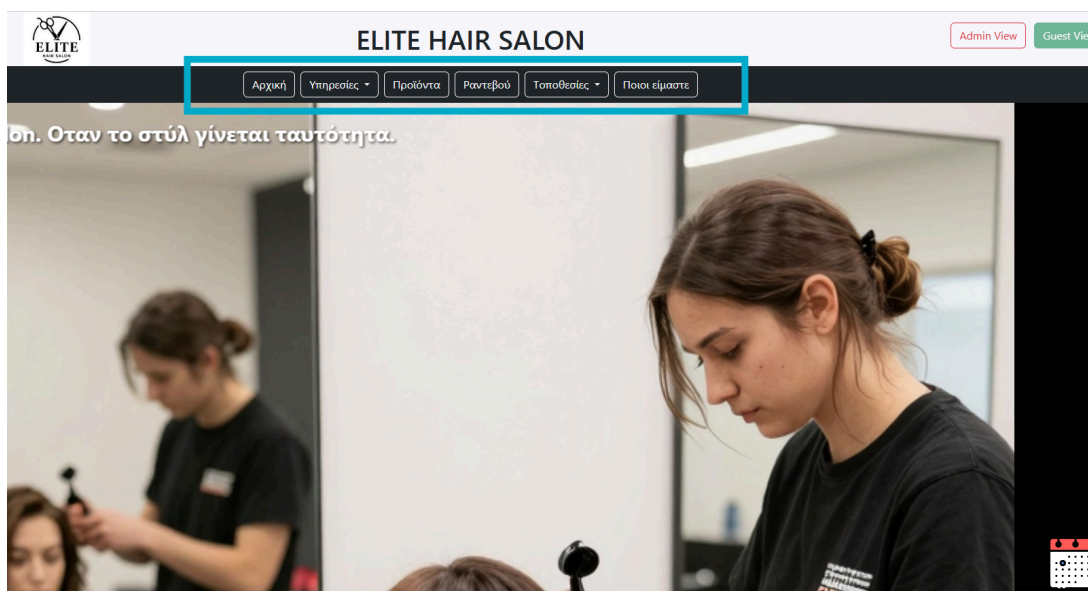


Το υποσέλιδο (footer) επαναλαμβάνεται επίσης σε όλες τις σελίδες και λειτουργεί ως σημείο ολοκλήρωσης της πλοήγησης, παρέχοντας πληροφορίες επικοινωνίας και επιπλέον λειτουργίες. Η χρήση των κλάσεων bg-dark και text-light δημιουργεί έντονη οπτική αντίθεση σε σχέση με το κύριο περιεχόμενο, βοηθώντας τον χρήστη να αναγνωρίζει εύκολα το τέλος της σελίδας. Η κλάση pt-4 προσθέτει επαρκή εσωτερικό χώρο, ενισχύοντας την καθαρή και επαγγελματική αισθητική του footer.

Στο εσωτερικό του footer χρησιμοποιούνται Bootstrap flex utilities (d-flex, justify-content-center, gap-3) για τη σωστή στοίχιση των εικονιδίων κοινωνικών δικτύων. Η συγκεκριμένη διάταξη επιτρέπει ίσες αποστάσεις μεταξύ των στοιχείων και διασφαλίζει ότι το layout παραμένει λειτουργικό και ευανάγνωστο σε κινητές και μεγαλύτερες οθόνες. Τα στοιχεία επικοινωνίας παρουσιάζονται κεντραρισμένα, προσφέροντας άμεση πρόσβαση σε βασικές πληροφορίες, όπως τηλέφωνα και διεύθυνση.

Τέλος, στο footer ενσωματώνεται και το κουμπί εμφάνισης ημερολογίου ως πρόσθετη διαδραστική λειτουργία, η οποία αναφέρεται συνοπτικά, καθώς αναλύεται σε ξεχωριστή ενότητα της εργασίας. Η τοποθέτησή του στο υποσέλιδο επιτρέπει την καθολική διαθεσιμότητά του χωρίς να επιβαρύνει την κύρια πλοήγηση. Συνολικά, ο σχεδιασμός του header και του footer συμβάλλει καθοριστικά στη δομική συνοχή, στη λειτουργικότητα και στην επαγγελματική εικόνα του ιστοτόπου, αξιοποιώντας αποτελεσματικά τις δυνατότητες του Bootstrap και της JavaScript.

### 1.3. Μενού πλοήγησης τουλάχιστον δύο επιπέδων



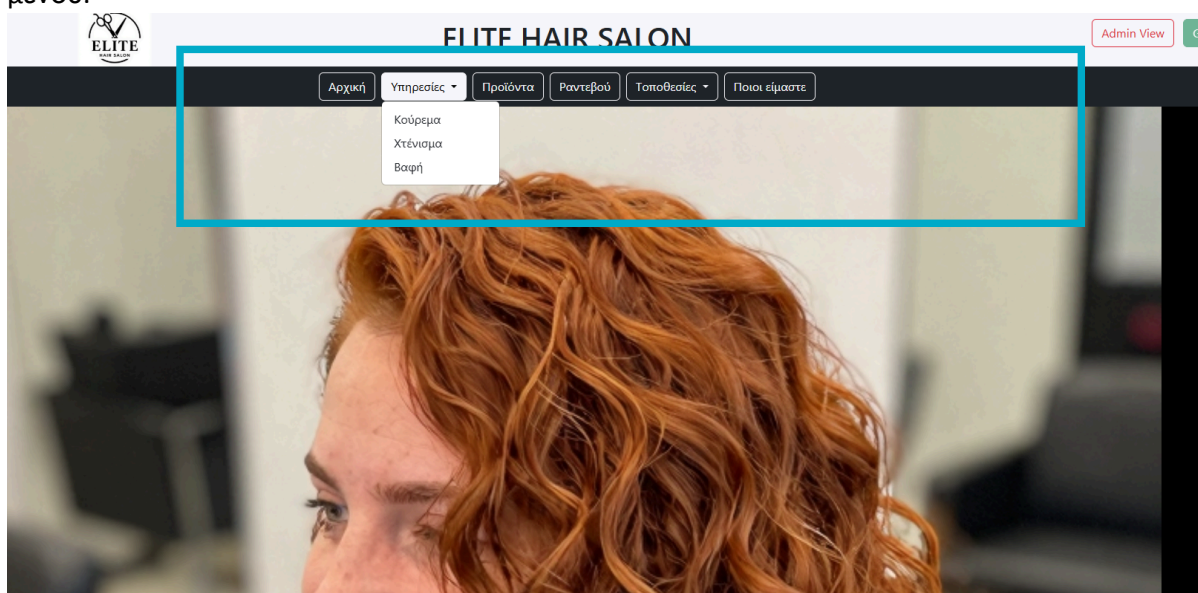
Το μενού πλοήγησης (navigation bar) αποτελεί κεντρικό στοιχείο λειτουργικότητας του ιστοτόπου, καθώς επαναλαμβάνεται σε όλες τις σελίδες και επιτρέπει τη γρήγορη και οργανωμένη μετακίνηση του χρήστη ανάμεσα στις βασικές ενότητες. Η υλοποίησή του βασίζεται στη Bootstrap κλάση navbar, η οποία προσφέρει έτοιμη δομή για responsive πλοήγηση και καθαρό διαχωρισμό του μενού από το υπόλοιπο περιεχόμενο της σελίδας. Η επιλογή των κλάσεων navbar-dark και bg-dark δημιουργεί υψηλή χρωματική αντίθεση, βελτιώνοντας την αναγνωσιμότητα των στοιχείων του μενού και διατηρώντας οπτική συνοχή με το υποσέλιδο του site.

Η διάταξη του μενού οργανώνεται μέσω ενός container, ο οποίος εξασφαλίζει ότι το περιεχόμενο παραμένει κεντραρισμένο και ευθυγραμμισμένο σε διαφορετικά μεγέθη οθόνης. Στο εσωτερικό του container χρησιμοποιείται λίστα ul με την κλάση navbar-nav, η οποία αποτελεί το βασικό δομικό στοιχείο πλοήγησης της Bootstrap. Η προσθήκη των κλάσεων

justify-content-center και w-100 επιτρέπει την οριζόντια κεντραρισμένη τοποθέτηση όλων των επιλογών, δίνοντας στο μενού συμμετρική και ισορροπημένη εμφάνιση.

```
<!-- Το μενού δυο διαστάσεων επαναλαμβάνεται σε όλες τις σελίδες -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container">
    <ul class="navbar-nav justify-content-center w-100 gap-2">
      <li class="nav-item">
        <a class="btn btn-outline-light" href="main.html">Αρχική</a>
      </li>
      <li class="nav-item dropdown">...
      </li>
      <li class="nav-item ">...
      </li>
      <li class="nav-item">...
      </li>
      <li class="nav-item dropdown">...
      </li>
      <li class="nav-item">...
      </li>
    </ul>
  </div>
</nav>
```

Κάθε επιλογή πλοήγησης υλοποιείται ως nav-item, ενώ τα links εμφανίζονται με τη μορφή κουμπιών μέσω των κλάσεων btn και btn-outline-light. Η συγκεκριμένη επιλογή σχεδίασης προσδίδει πιο έντονη διαδραστική αίσθηση σε σχέση με απλούς συνδέσμους κειμένου, καθιστώντας σαφές στον χρήστη ότι πρόκειται για ενεργά στοιχεία πλοήγησης. Παράλληλα, η outline μορφή των κουμπιών διατηρεί μια λιτή αισθητική, χωρίς να υπερφορτώνει οπτικά το μενού.



Για την ομαδοποίηση σχετικών σελίδων χρησιμοποιούνται dropdown μενού, τα οποία υλοποιούνται με την κλάση dropdown. Οι ενότητες «Υπηρεσίες» και «Τοποθεσίες» αξιοποιούν dropdown κουμπιά (dropdown-toggle), τα οποία αποκαλύπτουν δευτερεύοντα μενού επιλογών.

Με αυτόν τον τρόπο επιτυγχάνεται ιεραρχική οργάνωση της πληροφορίας, περιορίζοντας τον αριθμό των κύριων επιλογών στο μενού και βελτιώνοντας τη χρηστικότητα.

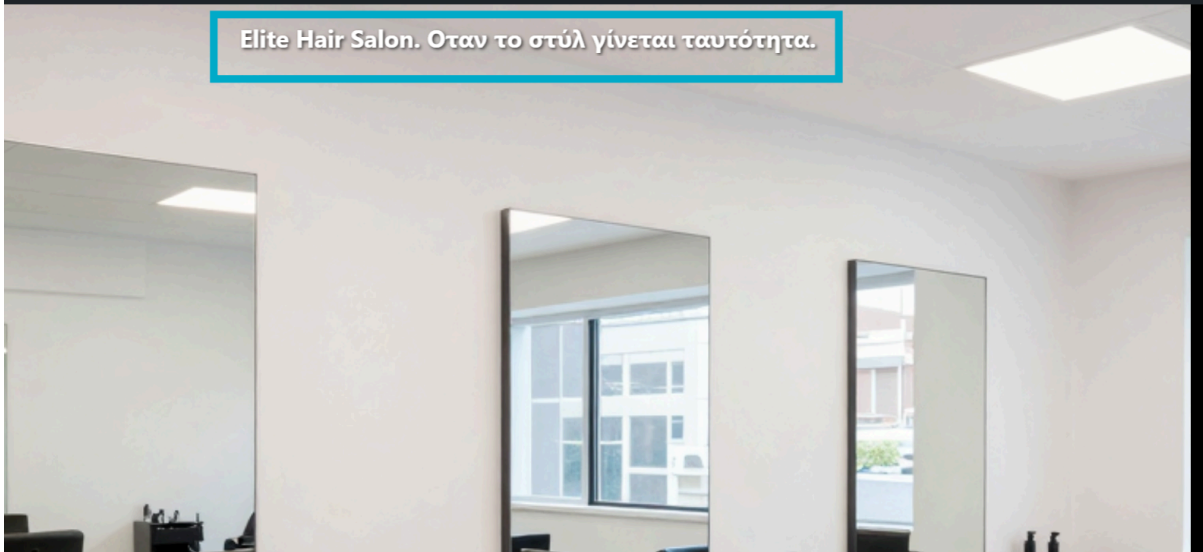
```
<!-- Το μενού δυο διαστάσεων επαναλαμβάνεται σε όλες τις σελίδες -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container">
    <ul class="navbar-nav justify-content-center w-100 gap-2">
      <li class="nav-item">...
    </li>
    <li class="nav-item dropdown">...
    </li>
    <li class="nav-item ">...
    </li>
    <li class="nav-item">...
    </li>
    <li class="nav-item dropdown">
      <button class="btn btn-outline-light dropdown-toggle">
        Τοποθεσίες
      </button>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="Athens.html">Αθήνα</a></li>
        <li><a class="dropdown-item" href="Thessaloniki.html">Θεσσαλονίκη</a></li>
      </ul>
    </li>
    <li class="nav-item">...
    </li>
  </ul>
</div>
</nav>
```

Τα περιεχόμενα των dropdown εμφανίζονται μέσω της κλάσης dropdown-menu, ενώ κάθε επιμέρους σύνδεσμος υλοποιείται με dropdown-item. Η χρήση αυτών των Bootstrap κλάσεων εξασφαλίζει ομοιόμορφη εμφάνιση, σωστή στοίχιση και συνεπή συμπεριφορά σε διαφορετικά περιβάλλοντα και συσκευές. Επιπλέον, η συγκεκριμένη δομή επιτρέπει εύκολη επέκταση του μενού στο μέλλον, προσθέτοντας νέες υπηρεσίες ή τοποθεσίες χωρίς αλλαγές στη βασική λογική σχεδίασης.

Συνολικά, το navigation bar έχει σχεδιαστεί με έμφαση στη σαφήνεια, την ευχρηστία και την επεκτασιμότητα. Η συνδυασμένη χρήση Bootstrap κλάσεων για διάταξη, χρωματισμό και αλληλεπίδραση συμβάλλει στη δημιουργία ενός σταθερού και επαγγελματικού συστήματος πλοήγησης, το οποίο υποστηρίζει αποτελεσματικά τη δομή του ιστότοπου και καθοδηγεί τον χρήστη με φυσικό και κατανοητό τρόπο.

## 1.4. Το κεντρικό μήνυμα που θέλει να περάσει η επιχείρηση

Elite Hair Salon. Όταν το στυλ γίνεται ταυτότητα.



Το κεντρικό μήνυμα της επιχείρησης αποτυπώνεται μέσω ενός ξεχωριστού οπτικού στοιχείου, το οποίο περιλαμβάνει το σύνθημα «Elite Hair Salon. Όταν το στυλ γίνεται ταυτότητα.». Το συγκεκριμένο μήνυμα έχει τοποθετηθεί σε ειδικά διαμορφωμένο container, ώστε να ξεχωρίζει από το υπόλοιπο περιεχόμενο της σελίδας και να γίνεται άμεσα αντιληπτό από τον χρήστη. Η επιλογή αυτή ενισχύει την εταιρική ταυτότητα και βοηθά τον επισκέπτη να κατανοήσει από την πρώτη στιγμή τη φιλοσοφία και τον χαρακτήρα της επιχείρησης.

```
<div class="slogan-container">
  <div class="slogan-text">
    Elite Hair Salon. Όταν το στυλ γίνεται ταυτότητα.
  </div>
</div>
```

Από πλευράς εμπειρίας χρήστη (UX), η παρουσία του κεντρικού μηνύματος λειτουργεί ως σημείο προσανατολισμού, καθοδηγώντας τον χρήστη και δημιουργώντας μια συναισθηματική σύνδεση με το brand. Η λιτή διατύπωση του slogan, σε συνδυασμό με την καθαρή οπτική παρουσίαση, επιτρέπει στον χρήστη να αντιληφθεί το μήνυμα χωρίς γνωστική επιβάρυνση. Έτσι, ο επισκέπτης δεν χρειάζεται να «αναζητήσει» πληροφορίες, αλλά τις λαμβάνει άμεσα και με φυσικό τρόπο.

Ιδιαίτερη έμφαση έχει δοθεί και στη χρήση οπτικού υλικού, καθώς ο ιστότοπος περιλαμβάνει πλήθος εικόνων υψηλής αισθητικής, οι οποίες πλαισιώνουν και ενισχύουν το κεντρικό μήνυμα. Οι εικόνες λειτουργούν συμπληρωματικά προς το κείμενο, προσφέροντας οπτικά ερεθίσματα που βοηθούν τον χρήστη να σχηματίσει ξεκάθαρη εικόνα για τις υπηρεσίες και το ύφος του κομμωτηρίου. Η συγκεκριμένη επιλογή ευθυγραμμίζεται με βασικές αρχές UX, σύμφωνα με τις οποίες το οπτικό περιεχόμενο βελτιώνει την κατανόηση και αυξάνει την αλληλεπίδραση.

Η υλοποίηση του κινούμενου κεντρικού μηνύματος πραγματοποιήθηκε αποκλειστικά με τη χρήση CSS, χωρίς την ανάγκη JavaScript, γεγονός που συμβάλλει σε καλύτερη απόδοση και απλούστερη συντήρηση του κώδικα. Το στοιχείο .slogan-container τοποθετείται με position: absolute πάνω από το carousel, επιτρέποντας στο μήνυμα να εμφανίζεται σε σταθερή θέση

ανεξάρτητα από την εναλλαγή των εικόνων. Η ιδιότητα `z-index` διασφαλίζει ότι το κείμενο παραμένει ορατό πάνω από τα υπόλοιπα οπτικά στοιχεία, ενώ το `pointer-events: none` επιτρέπει την κανονική αλληλεπίδραση του χρήστη με το carousel, χωρίς παρεμβολές.

```
/* Floating slogan */
.slogan-container {
  position: absolute;
  top: 20px;
  left: 0;
  width: 100%;
  overflow: hidden;
  z-index: 2000;
  pointer-events: none;
}

.slogan-text {
  display: inline-block;
  white-space: nowrap;
  font-size: 2rem;
  font-weight: bold;
  color: #fff;
  text-shadow: 2px 2px 4px rgba(0,0,0,0.7);
  animation: scroll-left 25s linear infinite;
}
```

Το ίδιο το κείμενο του συνθήματος υλοποιείται μέσω της κλάσης `.slogan-text`, η οποία αξιοποιεί CSS animation για τη δημιουργία ομαλής, συνεχούς κίνησης από τα δεξιά προς τα αριστερά. Η χρήση του `white-space: nowrap` αποτρέπει την αλλαγή γραμμής, διατηρώντας το μήνυμα ενιαίο καθ' όλη τη διάρκεια της κίνησης. Παράλληλα, το `text-shadow` ενισχύει την αναγνωσιμότητα του κειμένου πάνω σε εναλλασσόμενα φόντα εικόνων, εξασφαλίζοντας ότι το μήνυμα παραμένει ευδιάκριτο ανεξαρτήτως χρωματικών αντιθέσεων.

```

@keyframes scroll-left {
  0% { transform: translateX(100%); }
  100% { transform: translateX(-100%); }
}
/* Responsive adjustments */
@media (max-width: 768px) {
  .slogan-text {
    font-size: 1.5rem;
  }
}
@media (max-width: 480px) {
  .slogan-text {
    font-size: 1.2rem;
  }
}

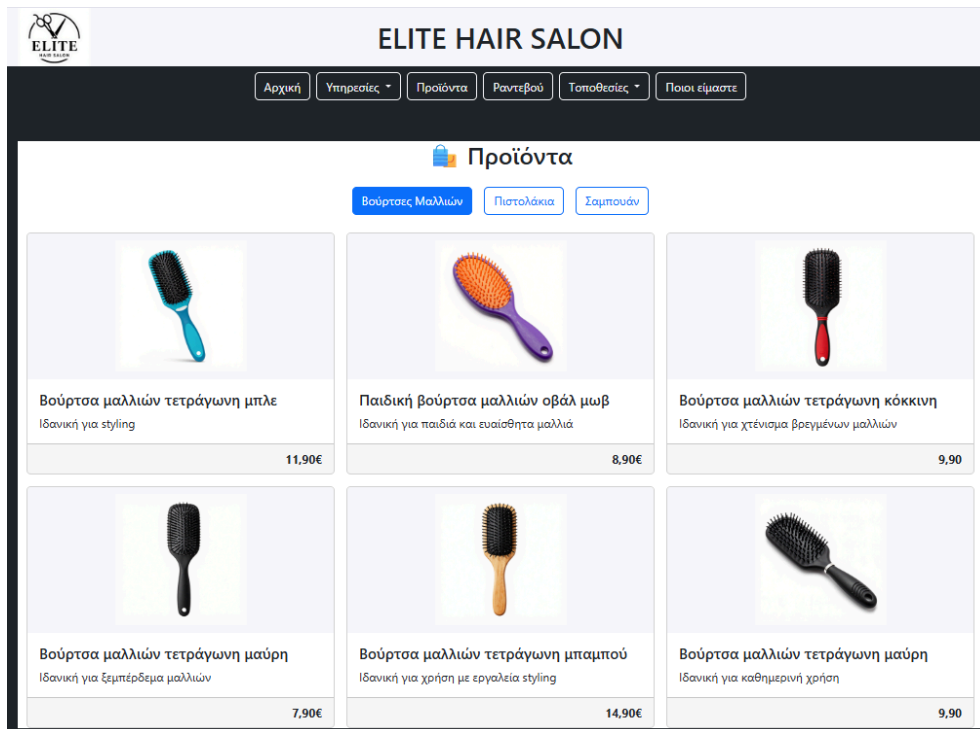
```

Τέλος, δόθηκε ιδιαίτερη έμφαση στη responsive συμπεριφορά του στοιχείου, μέσω media queries που προσαρμόζουν το μέγεθος της γραμματοσειράς σε μικρότερες οθόνες. Με αυτόν τον τρόπο διατηρείται η ισορροπία μεταξύ ορατότητας και αισθητικής σε κινητές συσκευές και tablets, βελτιώνοντας συνολικά την εμπειρία χρήστη. Η συγκεκριμένη υλοποίηση συνδυάζει απλότητα, απόδοση και καθαρό οπτικό αποτέλεσμα, ενισχύοντας αποτελεσματικά το κεντρικό μήνυμα της επιχείρησης.

Συνολικά, ο συνδυασμός του κεντρικού μηνύματος με καθαρή διάταξη, απλή γλώσσα και πλούσιο εικονικό υλικό συμβάλλει στη δημιουργία ενός φιλικού και εύχρηστου περιβάλλοντος πλοήγησης. Ο χρήστης αισθάνεται άνετα και καθοδηγείται ομαλά μέσα στον ιστότοπο, χωρίς σύγχυση ή περιττή πληροφορία. Με αυτόν τον τρόπο, το slogan δεν λειτουργεί απλώς ως διακοσμητικό στοιχείο, αλλά ως ουσιαστικό μέρος της συνολικής εμπειρίας χρήστη και της ταυτότητας της επιχείρησης.

## 1.5. Άλλες σημαντικές πληροφορίες ή λειτουργίες που προσφέρονται παράλληλα

Πρέπει να αναφερθούμε στην υλοποίηση της σελίδας που προβάλλει τα προϊόντα της επιχείρησης. Τα προϊόντα που επιλέχθηκαν χωρίζονται σε 3 κατηγορίες: 1) Βούρτσες μαλλιών, 2) Πιστολάκια μαλλιών και 3) Σαμπουάν.



Σχήμα 3. 17: Στιγμιότυπο της σελίδας προϊόντων

Για την HTML χρησιμοποιήθηκε ένα απλό section container, με τον τίτλο, τα 3 κουμπιά που αντιπροσωπεύουν τις κατηγορίες και ένα container με cards προϊόντων. Ο χρήστης μπορεί εύκολα και άμεσα να μεταφερθεί από τη μία κατηγορία στην άλλη, κάνοντας click στο κουμπί της κατηγορίας που επιθυμεί. Τότε ο πίνακας προϊόντων θα ξαναφορτωθεί με τα προϊόντα της επιλεγμένης κατηγορίας.

Λόγω έλλειψης υλοποίησης συστήματος backend, οι πληροφορίες σχετικά με τα προϊόντα της επιχείρησης, γράφτηκαν σε πίνακες Javascript και φορτώθηκαν δυναμικά μέσω της συνάρτησης renderProducts.

```

180 function renderProducts(category) {
181   const container = document.getElementById('productsContainer');
182   container.innerHTML = '';
183
184   productsData[category].forEach(prod => {
185     const card = document.createElement('div');
186     card.className = 'col-md-4';
187
188     card.innerHTML = `
189       <div class="card h-180">
190         
191         <div class="card-body">
192           <h5 class="card-title">${prod.name}</h5>
193           <p class="card-text">${prod.description}</p>
194         </div>
195         <div class="card-footer text-end">
196           <strong>${prod.price}</strong>
197         </div>
198       </div>
199     `;
200     container.appendChild(card);
201   });
202 }
203
204 // κλήση της συνάρτησης renderProducts
205 document.querySelectorAll('.category-btn').forEach(btn => { //επίλεξε όλα τα κουμπιά της σελίδας που έχουν category "category-btn"
206   btn.addEventListener('click', () => { //φτιάξε αυτόματα listener για όλα
207     document.querySelectorAll('.category-btn').forEach(b => b.classList.remove('btn-primary')); //αφαίρεσε για κάθε κουμπί το class primary (μπλε κουμπί)
208     btn.classList.add('btn-primary'); //πρόσθεσε την κλάση "btn-primary" για το επιλεγμένο κουμπί
209     btn.classList.remove('btn-outline-primary'); //αφαίρεσε την κλάση "btn-outline-primary" για το επιλεγμένο κουμπί
210
211     renderProducts(btn.dataset.category); // εμφάνισε τα προϊόντα της κατηγορίας
212   });
213 });
214
215 document.querySelector('.category-btn').click(); // επέλεξε το 1ο κουμπί κατηγορίας και κάνε click
216 </script>

```

Σχήμα 3. 18: Ο κώδικας της συνάρτησης renderProducts

Η συνάρτηση λαμβάνει ως δεδομένο την κατηγορία των προϊόντων, και αφού καθαρίσει τη `innerHTML` του `container` των προϊόντων, για κάθε προϊόν της κατηγορίας δημιουργεί ένα `div` στοιχείο με class `"card h-180"`. Η κάρτα του προϊόντος αποτελείται από την εικόνα του (η οποία περιλαμβάνει στον αντίστοιχο φάκελο φωτογραφιών), το όνομα και την περιγραφή του στο κύριο `body` του, καθώς και την τιμή του κάτω δεξιά στο `footer` της.

Στη συνέχεια για κάθε κουμπί με την κλάση `"category-btn"`, δηλαδή για κάθε ένα από τα κουμπιά των κατηγοριών, δημιουργείται ένας `EventListener` ο οποίος: α) αφαιρεί από όλα τα κουμπιά (αλλά κυρίως μας ενδιαφέρει το κουμπί που έχει ήδη πατηθεί) την κλάση `"primary"`, προσθέτει μόνο στο επιλεγμένο κουμπί την κλάση `"btn-primary"`, αφαιρεί από το ίδιο κουμπί το `"btn-outline-primary"` και β) καλεί την `renderProducts` για την επιλεγμένη κατηγορία. Εξασφαλίζεται η επέκταση της επιχείρησης σε περισσότερες κατηγορίες, καθώς οι δημιουργία των `listeners` και η κλήση της συνάρτησης γίνεται δυναμικά.

## 1.6. Slideshow

Η ενότητα `Slideshow` αποτελεί ένα από τα βασικά δομικά και οπτικά στοιχεία του ιστότοπου και υλοποιείται με τη χρήση του `Carousel component` του `Bootstrap 5`. Η συγκεκριμένη επιλογή έγινε λόγω της σταθερότητας, της ευρείας υποστήριξης και της ενσωματωμένης λειτουργικότητας που προσφέρει το `Bootstrap`, χωρίς την ανάγκη ανάπτυξης πρόσθετου `JavaScript` κώδικα. Η βασική κλάση `carousel slide` ενεργοποιεί τη συμπεριφορά εναλλαγής των `slides`, ενώ το attribute `data-bs-ride="carousel"` επιτρέπει την αυτόματη εκκίνηση του `slideshow` κατά τη φόρτωση της σελίδας, βελτιώνοντας την εμπειρία του χρήστη με δυναμική παρουσίαση περιεχομένου.

```

122 <div class="carousel-container">
123
124 <div id="carouselExample" class="carousel slide" data-bs-ride="carousel"> <!-- δηλώνουμε ότι πρόκειται για carousel -->
125
126 <!-- Κουμπιά στο κάτω μέρος του carousel για επιλογή εικόνας -->
127 <div class="carousel-indicators">
128 <button type="button" data-bs-target="#carouselExample" data-bs-slide-to="0" class="active"><!-- active γιατί η 0 είναι η default σελίδα -->
129 <button type="button" data-bs-target="#carouselExample" data-bs-slide-to="1">
130 <button type="button" data-bs-target="#carouselExample" data-bs-slide-to="2">
131 <button type="button" data-bs-target="#carouselExample" data-bs-slide-to="3">
132 <button type="button" data-bs-target="#carouselExample" data-bs-slide-to="4">
133 </div>
134
135 <!-- Slides / εικόνες -->
136 <div class="carousel-inner">
137 <div class="carousel-item active">
138 
139 </div>
140 <div class="carousel-item">
141 
142 </div>
143 <div class="carousel-item">
144 
145 </div>
146 <div class="carousel-item">
147 
148 </div>
149 <div class="carousel-item">
150 
151 </div>
152 </div>
153
154 <!-- Κουμπιά πλοήγησης (βέλη αριστερά/δεξιά) -->
155
156 <button class="carousel-control-prev" type="button" data-bs-target="#carouselExample" data-bs-slide="prev">
157 <span class="carousel-control-prev-icon"></span> <!-- καλούμε το βελόκι -->
158 <span class="visually-hidden">Previous</span> <!-- για να γνωρίζουμε τη λειτουργία του κουμπιού -->
159 </button>
160
161 <button class="carousel-control-next" type="button" data-bs-target="#carouselExample" data-bs-slide="next">
162 <span class="carousel-control-next-icon"></span> <!-- καλούμε το βελόκι -->
163 <span class="visually-hidden">Next</span> <!-- για να γνωρίζουμε τη λειτουργία του κουμπιού -->
164 </button>
165

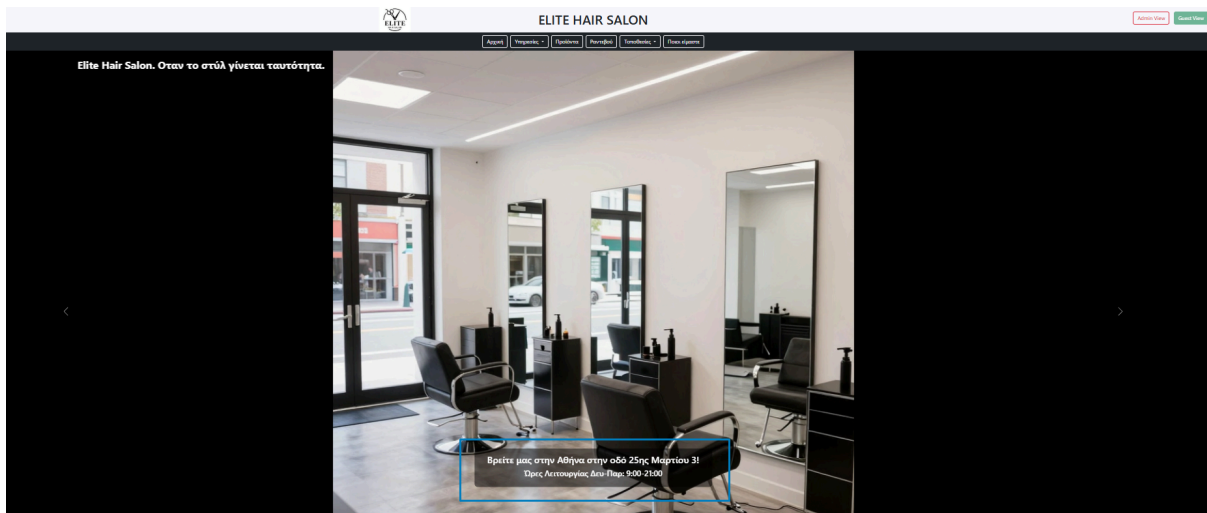
```

Σχήμα 3. 19:Ο HTML κώδικας του slideshow

Η εσωτερική δομή του carousel βασίζεται σε καθιερωμένες Bootstrap κλάσεις, όπως carousel-inner και carousel-item, οι οποίες διασφαλίζουν τη σωστή οργάνωση και εναλλαγή των εικόνων. Η κλάση carousel-item active ορίζει το slide που εμφανίζεται πρώτο, ενώ τα υπόλοιπα slides φορτώνονται δυναμικά στο παρασκήνιο. Οι δείκτες πλοήγησης (carousel-indicators) χρησιμοποιούνται για να παρέχουν οπτική ανατροφοδότηση σχετικά με τη θέση του χρήστη μέσα στο slideshow, αλλά και για άμεση μετάβαση σε συγκεκριμένο slide, ενισχύοντας τη χρηστικότητα και την προσβασιμότητα.

Παράλληλα, τα κουμπιά πλοήγησης carousel-control-prev και carousel-control-next αξιοποιούν τις ενσωματωμένες Bootstrap κλάσεις και icons, προσφέροντας συνεπή εμφάνιση και λειτουργικότητα σε όλες τις συσκευές. Η χρήση της κλάσης visually-hidden για τα βοηθητικά κείμενα διασφαλίζει τη συμμόρφωση με βασικές αρχές προσβασιμότητας (accessibility), επιτρέποντας τη σωστή ερμηνεία των στοιχείων από screen readers χωρίς να επιβαρύνεται ο οπτικός σχεδιασμός.

Σε επίπεδο αισθητικής και περιεχομένου, σε επιλεγμένες σελίδες το carousel εμπλουτίζεται με overlay κείμενο, το οποίο τοποθετείται πάνω από τις εικόνες μέσω custom κλάσης .carousel-text-overlay. Το στοιχείο αυτό επιτρέπει την παρουσίαση εισαγωγικών ή περιγραφικών μηνυμάτων που σχετίζονται με τη συγκεκριμένη υπηρεσία ή τοποθεσία. Η επιλογή ημιδιαφανούς φόντου (rgba(0,0,0,0.5)) ενισχύει την αναγνωσιμότητα, ενώ η απόλυτη τοποθέτηση σε σχέση με το .carousel-container παρέχει πλήρη έλεγχο της θέσης του περιεχομένου.



Σχήμα 3. 20: Στιγμιότυπο της σελίδας "Ποιοι είμαστε"

```

/* About Us Page */
.companyinfo-text-overlay {
  position: absolute;
  top: 97%;           /* Στο χαμηλό μέρος της σελίδας */
  left: 50%;         /* Στο κέντρο οριζόντια */
  transform: translate(-50%, -50%); /* απαραίτητο για να κεντραριστεί το κείμενο */
  color: white;
  background-color: rgba(0, 0, 0, 0.5); /* ημι-διάφανο μαύρο, 0,5 = 50% διαφάνεια */
  padding: 10px 10px;
  border-radius: 10px;
}

```

Σχήμα 3. 21: Η CSS του κειμένου της σελίδας "Ποιοι είμαστε"

Η συμπεριφορά και η εμφάνιση του carousel ενισχύονται περαιτέρω μέσω custom CSS, το οποίο προσαρμόζει το Bootstrap component στις ανάγκες του project. Το στοιχείο #carouselExample ορίζεται με συγκεκριμένο ύψος, ώστε το slideshow να λειτουργεί ως κυρίαρχο οπτικό τμήμα της σελίδας. Οι κλάσεις .carousel-inner και .carousel-item λαμβάνουν height: 100%, εξασφαλίζοντας ότι κάθε slide καταλαμβάνει πλήρως το διαθέσιμο ύψος. Οι εικόνες ρυθμίζονται με object-fit: contain, επιλογή που διασφαλίζει ότι ολόκληρο το περιεχόμενο της εικόνας παραμένει ορατό, χωρίς παραμορφώσεις ή απώλειες σημαντικών λεπτομερειών.

```
# main.css ×
# main.css > .carousel-item img
1  /* Carousel */
2  #carouselExample {
3    width: 100%;
4    height: 1600px; /* σταθερό ύψος */
5    background-color: black;
6  }
7
8  .carousel-inner {
9    height: 100%; /* τα slides έχουν το 100% του ύψους του καρουζελ */
10 }
11
12 .carousel-item {
13   height: 100%; /* το κάθε slide έχει το 100% του ύψους του carousel */
14 }
15
16 .carousel-item img {
17   height: 100%; /* η εικόνα λαμβάνει το 100% του ύψους */
18   width: auto; /* κρατάμε το πλάτος της αρχικής εικόνας, ώστε να μην παραμορφωθεί */
19   max-width: 100%; /* σε περίπτωση που το πλάτος της εικόνας ξεπεράσει το πλάτος της οθόνης, μην ξεφύγει η εικόνα από το καρουζέλ */
20   display: block; /* για να κεντράρουμε τις εικόνες με auto margins */
21   margin-left: auto;
22   margin-right: auto;
23   object-fit: contain; /* Φαίνεται ολόκληρη η εικόνα, δεν κόβεται */
24 }
25
26 .carousel-text-overlay {
27   position: absolute;
28   top: 80%; /* Στο χαμηλό μέρος της σελίδας */
29   left: 50%; /* Στο κέντρο οριζόντια */
30   transform: translate(-50%, -50%); /* απαραίτητο για να κεντραριστεί το κείμενο */
31   color: white;
32   background-color: rgba(0, 0, 0, 0.5); /* ημι-διάφανο μαύρο, 0,5 = 50% διαφάνεια */
33   padding: 20px 40px;
34   border-radius: 10px;
35 }
36
37 .carousel-container {
38   position: relative; /* δηλώνουμε τη θέση ως relative, έτσι ώστε οι εικόνες να μην ξεφύγουν από το carousel-container */
39 }
```

Σχήμα 3. 22: Η CSS του slideshow

Σημαντικό στοιχείο της υλοποίησης αποτελεί και ο συνδυασμός του carousel με το floating slogan, το οποίο παραμένει σταθερό πάνω από τις εναλλασσόμενες εικόνες. Η χρήση position: relative στο .carousel-container επιτρέπει τη σωστή στοίχιση τόσο του slogan όσο και του overlay κειμένου, χωρίς να επηρεάζεται η βασική λειτουργία του Bootstrap carousel. Ο συνδυασμός αυτός δημιουργεί μια συνεκτική και δυναμική παρουσίαση, η οποία ενισχύει την εταιρική ταυτότητα και κρατά το ενδιαφέρον του χρήστη.

Τέλος, αξίζει να σημειωθεί ότι παρόμοια δομή και λογική carousel έχει υλοποιηθεί και στις σελίδες Haircut.html, Hairdye.html, Hairstyling.html, Athens.html και Thessaloniki.html, με προσαρμοσμένο οπτικό υλικό και περιεχόμενο ανά τοποθεσία. Η επαναχρησιμοποίηση της ίδιας αρχιτεκτονικής βασίζεται στις αρχές της συνέπειας (consistency) και της επεκτασιμότητας, προσφέροντας ενιαία εμπειρία πλοήγησης σε όλο τον ιστότοπο και διευκολύνοντας τη μελλοντική συντήρηση και εξέλιξη του project.

## 1.7. Ημερολόγιο

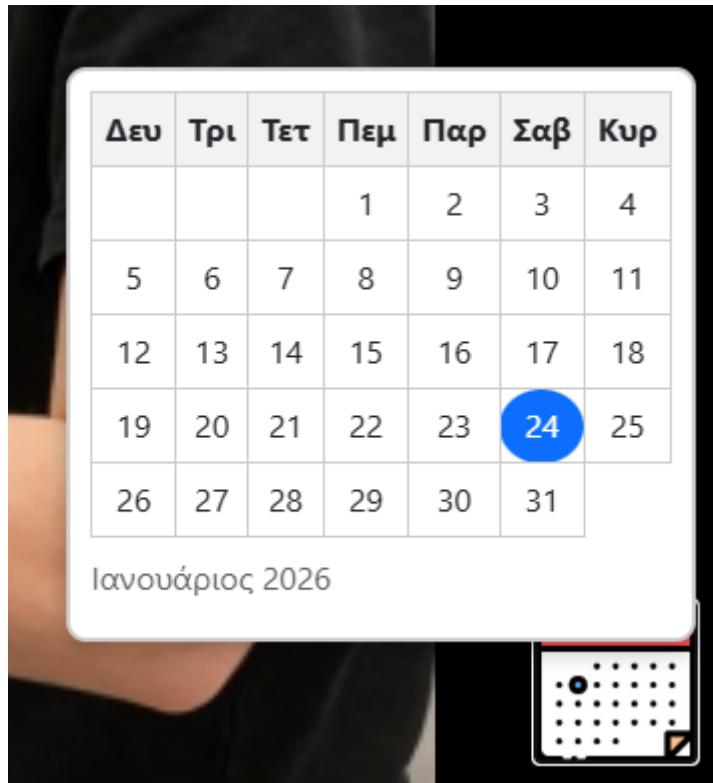
Η ενότητα του floating ημερολογίου σχεδιάστηκε ως ένα βοηθητικό, μη παρεμβατικό εργαλείο πλοήγησης, το οποίο είναι διαθέσιμο σε όλες τις σελίδες του ιστότοπου. Ο βασικός στόχος του είναι να προσφέρει στον χρήστη άμεση χρονική αναφορά (τρέχουσα ημερομηνία και μήνα) χωρίς να απαιτείται μετάβαση σε ξεχωριστή σελίδα ή άνοιγμα εξωτερικού ημερολογίου. Η επιλογή της floating μορφής έγινε με γνώμονα τις αρχές του UX, ώστε το στοιχείο να είναι πάντοτε προσβάσιμο αλλά ταυτόχρονα διακριτικό.

Η υλοποίηση βασίζεται αποκλειστικά σε HTML, CSS και JavaScript, χωρίς τη χρήση έτοιμων βιβλιοθηκών ημερολογίου. Αυτό επιτρέπει πλήρη έλεγχο της συμπεριφοράς, της εμφάνισης και της λογικής λειτουργίας του ημερολογίου. Το ημερολόγιο ενεργοποιείται μέσω ενός εικονιδίου (calendar button) τοποθετημένου σταθερά στο κάτω δεξί μέρος της οθόνης, ακολουθώντας σύγχρονες πρακτικές UI για βοηθητικά εργαλεία (floating action buttons).

Η δομή του ημερολογίου βασίζεται σε έναν απλό HTML πίνακα (<table>), ο οποίος δημιουργείται και ενημερώνεται δυναμικά μέσω JavaScript. Με αυτόν τον τρόπο, το ημερολόγιο είναι πάντα συγχρονισμένο με την τρέχουσα ημερομηνία του συστήματος του χρήστη, χωρίς χειροκίνητες παρεμβάσεις ή hardcoded δεδομένα.

```
177 <!-- Calendar Icon Button -->
178 <button class="date-btn" id="dateButton">
179 | 
180 </button>
181
182 <!-- Floating Calendar -->
183 <div id="calendarPopup">
184 | <table>
185 | | <caption id="monthYear"></caption>
186 | | <thead>
187 | | | <tr>
188 | | | | <th>Δευ</th>
189 | | | | <th>Τρι</th>
190 | | | | <th>Τετ</th>
191 | | | | <th>Πεμ</th>
192 | | | | <th>Παρ</th>
193 | | | | <th>Σαβ</th>
194 | | | | <th>Κυρ</th>
195 | | | </tr>
196 | | </thead>
197 | | <tbody id="calendarBody"></tbody>
198 | </table>
199 </div>
200
```

Σχήμα 3. 23: Η HTML του floating calendar



Σχήμα 3. 24: Στιγμιότυπο λειτουργίας του ημερολογίου

Αρχικά, γίνεται η επιλογή και αποθήκευση βασικών DOM στοιχείων σε μεταβλητές (`dateButton`, `calendarPopup`, `calendarBody`, `monthYear`). Αυτή η πρακτική ακολουθείται για λόγους καθαρότητας κώδικα, απόδοσης και επαναχρησιμοποίησης, αποφεύγοντας επαναλαμβανόμενες κλήσεις στο DOM. Παράλληλα, δημιουργείται ένα αντικείμενο `Date`, το οποίο αποθηκεύει την τρέχουσα ημερομηνία του χρήστη και λειτουργεί ως σημείο αναφοράς για όλη τη λογική του ημερολογίου.

```

201 <script>
202   const dateButton = document.getElementById('dateButton');
203   const calendarPopup = document.getElementById('calendarPopup');
204   const calendarBody = document.getElementById('calendarBody');
205   const monthYear = document.getElementById('monthYear');
206
207   const today = new Date(); // όταν δημιουργούμε ένα καινούριο Date, έχει τη σημερινή ημερομηνία
208
209

```

Σχήμα 3. 25: Η επιλογή των στοιχείων με Javascript

Η καρδιά της λειτουργίας βρίσκεται στη συνάρτηση `generateCalendar(year, month)`. Η συνάρτηση αυτή αναλαμβάνει να δημιουργήσει πλήρως το ημερολόγιο για ένα συγκεκριμένο μήνα και έτος. Αρχικά, ορίζεται ένας πίνακας με τα ονόματα των μηνών στα ελληνικά, ώστε η εμφάνιση να είναι πλήρως προσαρμοσμένη στη γλώσσα του ιστότοπου. Στη συνέχεια, γίνεται εκκαθάριση του `<tbody>` του πίνακα, ώστε κάθε νέα δημιουργία ημερολογίου να ξεκινά από καθαρή βάση.

Η JavaScript υπολογίζει με ακρίβεια την πρώτη ημέρα του μήνα (`firstDay`) και τον συνολικό αριθμό ημερών (`lastDate`) χρησιμοποιώντας το αντικείμενο `Date`. Επειδή η JavaScript ξεκινά την εβδομάδα από την Κυριακή, εφαρμόζεται μετατροπή ώστε η εβδομάδα να ξεκινά από τη Δευτέρα, κάτι που είναι σύμφωνο με την ευρωπαϊκή μορφή ημερολογίου. Στη συνέχεια, δημιουργούνται κενά κελιά για τις ημέρες του προηγούμενου μήνα, ώστε το ημερολόγιο να ευθυγραμμίζεται σωστά.

Για κάθε ημέρα του μήνα δημιουργείται δυναμικά ένα `<td>`, στο οποίο τοποθετείται ο αριθμός της ημέρας. Αν η ημέρα συμπίπτει με τη σημερινή ημερομηνία, προστίθεται η κλάση `today`, ώστε να επισημαίνεται οπτικά. Οι γραμμές (`<tr>`) ολοκληρώνονται και προστίθενται στο DOM όταν συμπληρωθεί πλήρης εβδομάδα ή όταν φτάσουμε στο τέλος του μήνα. Τέλος, ενημερώνεται το `<caption>` του πίνακα με τον μήνα και το έτος.

Η εμφάνιση και απόκρυψη του ημερολογίου ελέγχεται μέσω `event listeners`. Με κλικ στο κουμπί ημερολογίου, γίνεται εναλλαγή της κλάσης `show`, η οποία ενεργοποιεί την εμφάνιση του `popup`. Παράλληλα, υπάρχει `global listener` στο `document`, ο οποίος ανιχνεύει `clicks` εκτός του ημερολογίου και του κουμπιού και κλείνει αυτόματα το `popup`. Με αυτόν τον μηχανισμό αποφεύγονται ανεπιθύμητες επικαλύψεις και διασφαλίζεται φυσική και αναμενόμενη συμπεριφορά για τον χρήστη. Τέλος, κατά τη φόρτωση της σελίδας, η συνάρτηση `generateCalendar` καλείται άμεσα με τον τρέχοντα μήνα και έτος, εξασφαλίζοντας ότι το ημερολόγιο είναι έτοιμο προς χρήση.

### Αναλυτική επεξήγηση της συνάρτησης `generateCalendar(year, month)`

#### `function generateCalendar(year, month)`

Η συνάρτηση `generateCalendar` δέχεται δύο παραμέτρους:

- `year`: το έτος που θέλουμε να εμφανιστεί (π.χ. 2026)
- `month`: τον μήνα σε μορφή `index` (0–11), όπως ορίζεται από το αντικείμενο `Date` της JavaScript

Με αυτόν τον τρόπο η συνάρτηση είναι επαναχρησιμοποιήσιμη και μπορεί να εμφανίσει οποιοδήποτε μήνα και έτος.

```
const monthsFull = ['Ιανουάριος', 'Φεβρουάριος', 'Μάρτιος', 'Απρίλιος', 'Μάιος', 'Ιούνιος',  
                    'Ιούλιος', 'Αύγουστος', 'Σεπτέμβριος', 'Οκτώβριος', 'Νοέμβριος', 'Δεκέμβριος'];
```

Δημιουργείται ένας πίνακας που περιέχει τα ονόματα των μηνών στα ελληνικά.

Ο πίνακας αυτός χρησιμοποιείται αργότερα για την εμφάνιση του τίτλου του ημερολογίου (μήνας + έτος), εξασφαλίζοντας σωστή γλωσσική προσαρμογή του UI.

```
calendarBody.innerHTML = "";
```

Καθαρίζεται πλήρως το `<tbody>` του πίνακα.

Αυτό είναι κρίσιμο βήμα, διότι:

- αποφεύγεται η συσσώρευση παλιών ημερών
- κάθε νέα δημιουργία ημερολογίου ξεκινά από καθαρή βάση
- εξασφαλίζεται σωστή ανανέωση του DOM

```
const firstDay = new Date(year, month, 1).getDay();
```

Υπολογίζεται η ημέρα της εβδομάδας στην οποία πέφτει η 1η ημέρα του μήνα.

- Η `getDay()` επιστρέφει τιμές από 0 έως 6
- 0 = Κυριακή, 1 = Δευτέρα, ..., 6 = Σάββατο

Αυτή η πληροφορία χρησιμοποιείται για τη σωστή στοίχιση του ημερολογίου.

```
const lastDate = new Date(year, month + 1, 0).getDate();
```

Εδώ υπολογίζεται πόσες ημέρες έχει ο συγκεκριμένος μήνας.

- Με το `month + 1` πηγαίνουμε στον επόμενο μήνα
- Με το 0 ως ημέρα, επιστρέφεται η τελευταία ημέρα του προηγούμενου μήνα
- Το `getDate()` επιστρέφει τον αριθμό ημέρας (28–31)

Αυτός ο τρόπος είναι αξιόπιστος και ανεξάρτητος από `leap years`.

```
let row = document.createElement('tr');
```

Δημιουργείται δυναμικά ένα στοιχείο <tr>, το οποίο αντιπροσωπεύει μία εβδομάδα στο ημερολόγιο.

Όλες οι ημέρες (κελιά <td>) προστίθενται αρχικά σε αυτή τη γραμμή.

```
let last_month_days_blank = (firstDay + 6) % 7;
```

Μετατροπή της εβδομάδας ώστε να ξεκινά από Δευτέρα αντί για Κυριακή.

- Η JavaScript ξεκινά από Κυριακή
- Το ημερολόγιο ακολουθεί ευρωπαϊκό πρότυπο (Δευτέρα → Κυριακή)
- Ο τύπος  $(\text{firstDay} + 6) \% 7$  μετατρέπει σωστά τη σειρά

Αυτό εξασφαλίζει ορθή στοίχιση κάτω από τις επικεφαλίδες Δευ–Κυρ.

```
for (let i = 0; i < last_month_days_blank; i++)
```

```
  row.appendChild(document.createElement('td'));
```

Δημιουργούνται κενά κελιά για τις ημέρες του προηγούμενου μήνα.

- Αυτά τα κελιά δεν περιέχουν αριθμό
- Χρησιμοποιούνται μόνο για να μετατοπιστεί σωστά η 1η ημέρα
- Είναι απαραίτητα για σωστή διάταξη του πίνακα

```
  for (let day = 1; day <= lastDate; day++) {
```

Ξεκινά επανάληψη για κάθε ημέρα του μήνα (από 1 έως τελευταία ημέρα).

Κάθε επανάληψη αντιστοιχεί σε ένα ημερολογιακό κελί.

```
    const cell = document.createElement('td');
```

```
    cell.textContent = day;
```

Δημιουργείται ένα <td> και μέσα του τοποθετείται ο αριθμός της ημέρας.

Αυτό είναι το βασικό οπτικό στοιχείο κάθε ημέρας στο ημερολόγιο.

```
    if (day === today.getDate() && month === today.getMonth() && year === today.getFullYear()) {
```

```
      cell.classList.add('today');
```

```
    }
```

Έλεγχος αν η τρέχουσα ημέρα του loop είναι η σημερινή ημερομηνία.

Αν ισχύει:

- προστίθεται η κλάση today
- η ημέρα επισημαίνεται οπτικά μέσω CSS

Αυτό βελτιώνει σημαντικά την αναγνωσιμότητα και χρηστικότητα.

```
    row.appendChild(cell);
```

Το κελί της ημέρας προστίθεται στη τρέχουσα γραμμή (<tr>).

```
  if ((day + last_month_days_blank) % 7 === 0 || day === lastDate) {
```

```
    calendarBody.appendChild(row);
```

```
    row = document.createElement('tr');
```

```
  }
```

Έλεγχος για το πότε πρέπει να κλείσει μια εβδομάδα:

- Αν συμπληρώθηκαν 7 ημέρες
- Ή αν φτάσαμε στην τελευταία ημέρα του μήνα

Τότε:

- η γραμμή προστίθεται στο <tbody>

- δημιουργείται νέα γραμμή για την επόμενη εβδομάδα

```
monthYear.textContent = monthsFull[month] + ' ' + year;
```

Τέλος, ενημερώνεται το <caption> του πίνακα με:

- το όνομα του μήνα
- το έτος

Αυτό δίνει σαφή χρονικό προσδιορισμό στο ημερολόγιο.

### **Ανάλυση CSS και οπτικής συμπεριφοράς (Floating Calendar)**

Το CSS της ενότητας έχει σχεδιαστεί ώστε το ημερολόγιο να λειτουργεί ως floating UI element, ανεξάρτητο από τη ροή του περιεχομένου της σελίδας. Το κουμπί ημερολογίου (.date-btn) χρησιμοποιεί position: fixed, γεγονός που εξασφαλίζει ότι παραμένει πάντα ορατό στο κάτω δεξί μέρος της οθόνης, ανεξαρτήτως scroll. Το υψηλό z-index εγγυάται ότι το στοιχείο δεν καλύπτεται από άλλα components, ενώ η απουσία background και border διατηρεί καθαρό και εικονιδιακό χαρακτήρα.

Το popup του ημερολογίου (#calendarPopup) είναι επίσης fixed-positioned και τοποθετείται ακριβώς πάνω από το κουμπί, δημιουργώντας σαφή οπτική συσχέτιση μεταξύ των δύο στοιχείων. Αρχικά είναι κρυφό (display: none και opacity: 0), ώστε να μην επιβαρύνει τη σελίδα. Η κλάση show ελέγχει την εμφάνιση μέσω συνδυασμού display και opacity, προσφέροντας ομαλή μετάβαση και αποφεύγοντας απότομα animations που θα μπορούσαν να αποσπάσουν τον χρήστη.

Η εμφάνιση του ίδιου του ημερολογίου είναι λιτή και ευανάγνωστη. Το λευκό φόντο, το ελαφρύ border και το box-shadow δημιουργούν αίσθηση “αιώρησης” πάνω από τη σελίδα, ενώ η χρήση border-radius μαλακώνει τις γωνίες, εναρμονίζοντας το στοιχείο με τη συνολική αισθητική του site. Ο πίνακας είναι πλήρως responsive, με ίσο πλάτος κελιών και κεντραρισμένο κείμενο, ώστε να διαβάζεται εύκολα σε όλες τις αναλύσεις.

Ιδιαίτερη έμφαση δίνεται στην οπτική ανάδειξη της τρέχουσας ημέρας μέσω της κλάσης .today. Το μπλε χρώμα, σε συνδυασμό με το κυκλικό σχήμα, καθιστά άμεσα αναγνωρίσιμη τη σημερινή ημερομηνία, βελτιώνοντας τη χρηστικότητα χωρίς να απαιτείται επιπλέον επεξήγηση. Συνολικά, το CSS υποστηρίζει πλήρως τη λειτουργικότητα του JavaScript, μετατρέποντας το ημερολόγιο σε ένα ολοκληρωμένο, λειτουργικό και αισθητικά συνεπές στοιχείο του ιστότοπου.

```

74  /* Floating calendar */
75
76  #calendarPopup {
77      display: none;           /* αρχικά το popup είναι κρυμμένο */
78      position: fixed;
79      bottom: 80px;           /* για να βρίσκεται πάνω από το date button */
80      right: 20px;
81      z-index: 10001;        /* για να καλύπτει το date button */
82      background-color: #white;
83      border: 2px solid #ccc;
84      border-radius: 10px;
85      padding: 10px;
86      box-shadow: 0 4px 12px #rgba(0,0,0,0.3);
87      min-width: 280px;
88      opacity: 0;
89      transition: opacity 0.3s ease;
90  }
91
92  #calendarPopup.show { /* προστίθεται στο calendar κάθε φορά που κάνουμε κλικ στο date button */
93      display: block;
94      opacity: 1;
95  }
96
97  #calendarPopup table {
98      border-collapse: collapse; /* για μικρότερα borders */
99      width: 100%;
100     text-align: center;
101  }
102
103  #calendarPopup th, #calendarPopup td {
104     border: 1px solid #ccc;
105     padding: 6px;
106  }
107
108  #calendarPopup th {
109     background-color: #f2f2f2;
110  }
111
112  #calendarPopup td.today {
113     background-color: #0d6efd;
114     color: #white;
115     border-radius: 50%; /* πόσο ανοιχτός είναι ο κύκλος */
116  }
117

```

Σχήμα 3. 26: Η CSS του Calendar

## Κεφάλαιο 2

# Παρουσίαση η διαχείριση δεδομένων

Η ενότητα των ραντεβού αποτελεί ένα από τα βασικότερα λειτουργικά κομμάτια της εφαρμογής, καθώς προσομοιώνει τη διαχείριση κρατήσεων ενός πραγματικού κομμωτηρίου. Ο σχεδιασμός της βασίστηκε στη λογική ενός Dummy/Mock database, με στόχο την παρουσίαση της ροής δεδομένων, της αλληλεπίδρασης χρήστη-συστήματος και της διαφοροποίησης ρόλων (admin – guest), χωρίς τη χρήση πραγματικού backend.

**ELITE HAIR SALON**

Αρχική Υπηρεσίες Προϊόντα Ραντεβού Τοποθεσίες Ποιο είμαστε

### Οι Κρατήσεις μου + Νέα Κράτηση

#	Όνομα	Υπηρεσία	Ημερομηνία	Ώρα	Τηλέφωνο	Κατάσταση	Ενέργειες
1	Μαρία Παπαδοπούλου	Κούρεμα	12/01/2026	11:00	6901234567	Επιβεβαιωμένο	
2	Μαρία Παπαδοπούλου	Κούρεμα	18/01/2026	13:15	6981234567	Επιβεβαιωμένο	
3	Μαρία Παπαδοπούλου	Βαφή	19/01/2026	10:30	6971234567	Ακυρωμένη	
4	Μαρία Παπαδοπούλου	Κούρεμα	21/01/2026	16:00	6909876543	Ακυρωμένη	

Τηλέφωνο: +30 210 1234567 | +30 210 7654321  
Διεύθυνση: Οδός Παράδειγμα 12, Αθήνα, Ελλάδα  
© 2026 ELITE HAIR SALON. All Rights Reserved.

Σχήμα 3. 27: Στιγμιότυπο λειτουργίας της σελίδας "Ραντεβού"

Τα ραντεβού υλοποιούνται μέσω ενός πίνακα αντικειμένων JavaScript (bookings), ο οποίος λειτουργεί ως εικονική βάση δεδομένων. Κάθε αντικείμενο αναπαριστά μία κράτηση και περιλαμβάνει πεδία όπως:

- Όνομα πελάτη
- Υπηρεσία
- Ημερομηνία
- Ώρα ραντεβού
- Τηλέφωνο
- Κατάσταση κράτησης (Επιβεβαιωμένη / Σε Αναμονή / Ακυρωμένη)

```

217  /* MOCK DATA */
218  const bookings = [
219    {
220      name: "Μαρία Παπαδοπούλου",
221      service: "Κούρεμα",
222      date: "12/01/2026",
223      time: "11:00",
224      phone: "6901234567",
225      status: "Επιβεβαιωμένο"
226    },
227    {
228      name: "Γιώργος Νικολάου",
229      service: "Βαφή",
230      date: "13/01/2026",
231      time: "14:30",
232      phone: "6987654321",
233      status: "Σε αναμονή"
234    },
235    {
236      name: "Ελένη Καραγιάννη",
237      service: "Χτένισμα",
238      date: "14/01/2026",
239      time: "10:00",
240      phone: "6941234567",
241      status: "Επιβεβαιωμένο"
242    },
243    {
244      name: "Νίκος Παπαδόπουλος",
245      service: "Κούρεμα",
246      date: "15/01/2026",
247      time: "12:30",
248      phone: "6976543210",
249      status: "Σε αναμονή"
250    },

```

Σχήμα 3. 28: Παράδειγμα εισαγωγής δεδομένων μέσω Javascript

Η επιλογή αυτής της προσέγγισης επιτρέπει την εύκολη διαχείριση και προβολή δεδομένων, καθώς και τη δοκιμή λειτουργιών (προσθήκη, επεξεργασία, διαγραφή), χωρίς να απαιτείται σύνδεση με server ή σύστημα βάσης δεδομένων. Με αυτόν τον τρόπο προσομοιώνεται ρεαλιστικά η λειτουργία ενός συστήματος κρατήσεων.

### Διαχωρισμός Ρόλων: Admin View & Guest View

Κεντρικό ρόλο στη λειτουργία των ραντεβού παίζουν τα κουμπιά Admin View και Guest View, τα οποία επιτρέπουν την εναλλαγή ρόλων χρήστη. Η επιλογή ρόλου αποθηκεύεται στο localStorage, διασφαλίζοντας ότι η κατάσταση παραμένει ενεργή ακόμα και μετά από ανανέωση της σελίδας.

Ο διαχειριστής έχει πλήρη πρόσβαση σε όλες τις κρατήσεις της dummy βάσης δεδομένων. Μπορεί να δει όλα τα ραντεβού, ανεξαρτήτως πελάτη, καθώς και να χρησιμοποιήσει λειτουργίες όπως επεξεργασία και διαγραφή.

Ο απλός χρήστης βλέπει μόνο τις κρατήσεις που αντιστοιχούν στο δικό του όνομα. Αυτό επιτυγχάνεται με φιλτράρισμα των δεδομένων πριν την απόδοση στον πίνακα, προσομοιώνοντας βασικές αρχές ιδιωτικότητας και προσωποποιημένης εμπειρίας χρήστη.

Ο διαχωρισμός αυτός ενισχύει την κατανόηση της λογικής role-based access, η οποία αποτελεί βασικό στοιχείο σύγχρονων web εφαρμογών.

### Προβολή & Απόδοση Ραντεβού (Rendering)

Η προβολή των ραντεβού γίνεται δυναμικά μέσω της συνάρτησης renderBookings().

```

305  /* RENDER TABLE */
306  function renderBookings() {
307    const guestName = "Μαρία Παπαδοπούλου"; // μόνο τα bookings της Μαρίας θα εμφανίζονται στο Guest View
308
309    bookingsBody.innerHTML = '';
310
311    const mode = localStorage.getItem('viewMode') || 'guest'; // πάρε είτε από το local storage το viewMode ΕΙΤΕ αν είναι κενό, το 'guest'
312
313    if(mode==='admin'){
314      bookingstext.innerHTML='Κρατήσεις Πελατών'
315    }
316    else{
317      bookingstext.innerHTML='👤 Κρατήσεις μου'
318    }
319    // φιλτράρουμε τις κρατήσεις αναλόγως το mode
320    const rowsToRender = (mode === 'admin') //αν το mode = 'admin'
321      ? bookings // πάρε όλα τα bookings
322      : bookings.filter(b => b.name === guestName); //αλλιώς φίλτραρε τα, και πάρε μόνο όσα έχουν name = 'Μαρία Παπαδοπούλου'
323
324    console.log('the mode is '+mode) //για debug
325
326    rowsToRender.forEach((b, index) => {
327      bookingsBody.innerHTML +=
328        <tr class="text-center">
329          <td>${index + 1}</td>
330          <td>${b.name}</td>
331          <td>${b.service}</td>
332          <td>${b.date}</td>
333          <td>${b.time}</td>
334          <td>${b.phone}</td>
335          <td>
336            <span class="badge ${ //διαλέγουμε με βάση το status της κράτησης, το χρώμα του span
337              b.status === 'Επιβεβαιωμένο' ? 'bg-success' :
338              b.status === 'Σε αναμονή' ? 'bg-warning text-dark' :
339              'bg-danger'
340            }">${b.status}</span>
341          </td>
342          <td>
343            <button class="btn btn-sm btn-warning me-1" onclick="editBooking(${index})">
344              ✎
345            </button>
346            <button class="btn btn-sm btn-danger" onclick="deleteBooking(${index})">
347              🗑️
348            </button>
349          </td>
350        </tr>
351      );
352    });
353  }

```

Σχήμα 3. 29: Ο κώδικας Javascript της renderBookings

Για τις ανάγκες επίδειξης της λειτουργικότητας του Guest View, θα χρησιμοποιηθεί ως guest χρήστης η Μαρία Παπαδοπούλου, συνεπώς στην αρχή επιλέγονται μονάχα οι δικές της κρατήσεις.

Έπειτα “καθαρίζεται” η innerHTML του bookingsBody, δηλαδή του body του πίνακα με τις κρατήσεις, ο οποίος στην HTML της σελίδας των κρατήσεων είναι άδειος, αλλά γεμίζει με δεδομένα δυναμικά μέσω της συνάρτησης.

Στη συνέχεια η συνάρτηση ελέγχει τον τρέχοντα ρόλο χρήστη και προσαρμόζει τον τίτλο της σελίδας. Εμφανίζεται ως “Οι κρατήσεις μου” αν βρισκόμαστε στο guest view ή ως “Κρατήσεις πελατών” αν βρισκόμαστε σε admin view. Έπειτα η συνάρτηση φιλτράρει τα δεδομένα αναλόγως το mode στο οποίο βρισκόμαστε. Αν δεν έχουμε επιλέξει το admin mode, επιλέγονται οι κρατήσεις με name ίσο με τη μεταβλητή guestName, την οποία ορίσαμε εξ αρχής ως “Μαρία Παπαδοπούλου”.

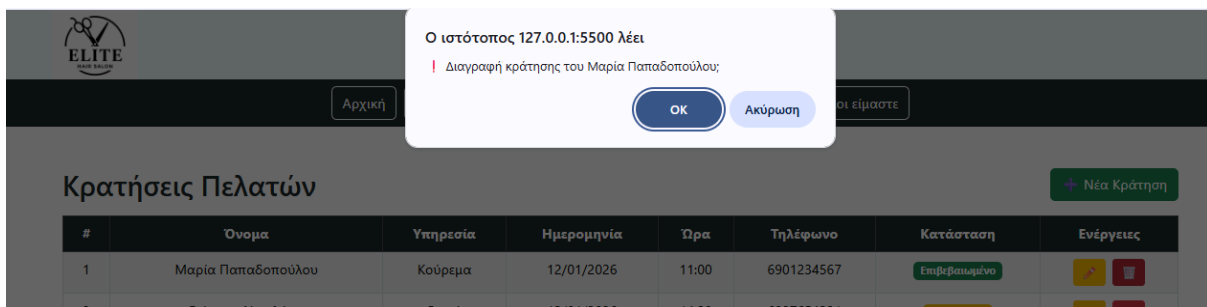
Για κάθε κράτηση δημιουργείται δυναμικά μία γραμμή πίνακα <tr> με όλα τα απαραίτητα στοιχεία. Ιδιαίτερο ενδιαφέρον παρουσιάζει η οπτική αναπαράσταση του πίνακα, καθώς χρησιμοποιούνται ειδικές κλάσεις Bootstrap όπως: table, table-striped, table-bordered για καθαρή και ευανάγνωστη παρουσίαση. Επιπλέον, είναι διαθέσιμη η αλλαγή χρώματος του περιγράμματος της Κατάστασης της κράτησης, ανάλογα με την τιμή της. Πράσινο για επιβεβαιωμένη, κίτρινο για αναμονή και κόκκινο για ακυρωμένη. Η χρήση αυτών των κλάσεων

βελτιώνει σημαντικά το UX, καθώς ο χρήστης μπορεί με μια ματιά να κατανοήσει την κατάσταση κάθε ραντεβού.

Τέλος, δημιουργούνται 2 κουμπιά σε κάθε γραμμή που προσομοιώνουν τις λειτουργίες της επεξεργασίας και της διαγραφής, για λόγους επίδειξης και εκπαιδευτικής χρήσης. Το κουμπί της επεξεργασίας μόλις ενεργοποιηθεί εμφανίζει ένα alert μήνυμα στο browser, ενώ το κουμπί της διαγραφής εκτός από alert, προτρέπει πρώτα το χρήστη να επιβεβαιώσει ότι συμφωνεί με τη διαγραφή της συγκεκριμένης κράτησης.

```
355  /* Mock αλληλεπιδράσεις */
356  function editBooking(index) {
357    alert('🔧 Mock: Επεξεργασία κράτησης του ${bookings[index].name}'); //εμφάνιση alert στον browser με το αντίστοιχο μήνυμα
358  }
359
360  function deleteBooking(index) {
361    if (confirm('! Διαγραφή κράτησης του ${bookings[index].name};')) { //εμφάνιση ερώτησης αν επιθυμεί ο χρήστης να προχωρήσει στη διαγραφή
362      alert('🗑️ Mock: 🗑️ κράτηση διαγράφηκε'); //εμφάνιση alert στον browser με το αντίστοιχο μήνυμα αν πατήσει ο χρήστης OK
363    }
364  }
```

Σχήμα 3. 30: Ο κώδικας Javascript των mock buttons



Σχήμα 3. 31: Παράδειγμα λειτουργίας του κουμπιού διαγραφής

Παράλληλα, η προσθήκη νέου ραντεβού γίνεται μέσω Bootstrap Modal, το οποίο περιέχει φόρμα με υποχρεωτικά πεδία. Αυτή η προσέγγιση διατηρεί τη σελίδα καθαρή, αποτρέπει άσκοπες μεταφορτώσεις και προσφέρει σύγχρονη και φιλική εμπειρία χρήστη.

Νέα Κράτηση

Όνομα

Υπηρεσία

Επιλέξτε υπηρεσία

Ημερομηνία

ηη/μμ/εεεε

Ώρα

--:-- --

Τηλέφωνο

Κατάσταση

Επιβεβαιωμένο

Ακύρωση Αποθήκευση

Κούρεμα 21/01/2026 16:00 6909876543

Σχήμα 3. 32: Στιγμιότυπο της λειτουργίας του modal

Η Bootstrap παρέχει τις απαιτούμενες κλάσεις για τη δημιουργία του modal, το οποίο αποτελείται από ένα form με πεδία εισαγωγής του ονόματος, της υπηρεσίας, της ημερομηνίας και ώρας του ραντεβού, το τηλέφωνο επικοινωνίας καθώς και την κατάσταση της κράτησης. Ακόμη περιλαμβάνονται και δύο κουμπιά: Ακύρωσης και Αποθήκευσης αντίστοιχα, τα οποία συμπεριλήφθηκαν για εκπαιδευτικούς λόγους και δεν είναι λειτουργικά.

```

161 <!-- Παράθυρο νέας κράτησης -->
162 <div class="modal fade" id="newBookingModal" tabindex="-1" aria-labelledby="newBookingLabel" aria-hidden="true">
163 <div class="modal-dialog">
164 <div class="modal-content">
165 <div class="modal-header">
166 <h5 class="modal-title" id="newBookingLabel"> Νέα Κράτηση</h5>
167 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Κλείσιμο"></button>
168 </div>
169 <div class="modal-body">
170 <form id="newBookingForm">
171 <div class="mb-3">
172 <label for="bookingName" class="form-label">Όνομα</label>
173 <input type="text" class="form-control" id="bookingName" required>
174 </div>
175 <div class="mb-3">
176 <label for="bookingService" class="form-label">Υπηρεσία</label>
177 <select class="form-select" id="bookingService" required>
178 <option value="">Επιλέξτε υπηρεσία</option>
179 <option>Κούρεμα</option>
180 <option>Χτένισμα</option>
181 <option>Βαφή</option>
182 </select>
183 </div>
184 <div class="mb-3">
185 <label for="bookingDate" class="form-label">Ημερομηνία</label>
186 <input type="date" class="form-control" id="bookingDate" required>
187 </div>
188 <div class="mb-3">
189 <label for="bookingTime" class="form-label">Ώρα</label>
190 <input type="time" class="form-control" id="bookingTime" required>
191 </div>
192 <div class="mb-3">
193 <label for="bookingPhone" class="form-label">Τηλέφωνο</label>
194 <input type="tel" class="form-control" id="bookingPhone" required>
195 </div>
196 <div class="mb-3">
197 <label for="bookingStatus" class="form-label">Κατάσταση</label>
198 <select class="form-select" id="bookingStatus" required>
199 <option value="Επιβεβαιωμένο">Επιβεβαιωμένο</option>
200 <option value="Σε αναμονή">Σε αναμονή</option>
201 <option value="Ακυρωμένη">Ακυρωμένη</option>
202 </select>
203 </div>
204 </form>
205 </div>
206 <div class="modal-footer">
207 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Ακύρωση</button>
208 <button type="button" class="btn btn-primary" id="saveBookingBtn">Αποθήκευση</button>
209 </div>
210 </div>
211 </div>
212 </div>

```

Σχήμα 3. 33: Ο κώδικας HTML του modal

Η φόρμα αδειάζει μέσω κώδικα Javascript κάθε φορά που ανοίγεται.

Η ενότητα των ραντεβού συνδυάζεται λειτουργικά με το floating ημερολόγιο, το οποίο επιτρέπει στον χρήστη να έχει άμεση χρονική αναφορά. Παρότι το ημερολόγιο δεν συνδέεται απευθείας με επιλογή ραντεβού, ενισχύει την αίσθηση οργάνωσης και προσομοιώνει την εμπειρία πραγματικής εφαρμογής κρατήσεων.

Η υλοποίηση της ενότητας ραντεβού καλύπτει πλήρως τις ανάγκες μιας dummy εφαρμογής, παρουσιάζοντας διαχείριση δεδομένων χωρίς backend υλοποίηση, διαχωρισμό ρόλων χρηστών, δυναμική απόδοση περιεχομένου και χρήση Bootstrap για βελτιωμένο UI/UX. Αποτελεί ένα ολοκληρωμένο παράδειγμα front-end προσέγγισης συστήματος κρατήσεων, έτοιμο να επεκταθεί με πραγματική βάση δεδομένων στο μέλλον.



