

Απαλλακτική Εργασία – Διαδικτυακός Προγραμματισμός

Ενότητα 1: Απαιτήσεις Εφαρμογής

Οι στόχοι προς επίτευξη μέσω της υλοποίησης της εφαρμογής:

- Ενημέρωση χρηστών για τις επόμενες εκδηλώσεις , αλλά και εκείνων που έχουν ολοκληρωθεί
- Δυνατότητα περιήγησης στις ακόλουθες εκδηλώσεις και προβολής των λεπτομερειών τους
- Δυνατότητα αγοράς εισιτήριων για τις προβολές των εκδηλώσεις
- Δυνατότητα επιλογής θέσεων
- Δυνατότητα αναζήτησης εκδηλώσεων
- Δημιουργία σελίδας προσιτής προς τον χρήστη
- Δημιουργία σελίδας διαχειριστή που του επιτρέπει την εύκολη και γρήγορη διαχείριση της εφαρμογής

Καταγραφή φάσεων υλοποίησης της εργασίας:

Φάση 1^η : Καταγραφή γενικής ιδέας της εργασίας (Τέλη Μαΐου)

Φάση 2^η : Σύνδεση Εφαρμογής την βάση δεδομένων MongoDB (1^η Ιουνίου – 5^η Ιουνίου)

Φάση 3^η: Προσθήκη αρχικών σελίδων, homepage,login κλπ. (5^η Ιουνίου – 12^η Ιουνίου)

Φάση 4^η : Προσθήκη επιμέρους σελίδων και λειτουργιών (12^η Ιουνίου – 20 Ιουνίου)

Φάση 5^η : Τελειοποίηση λειτουργιών, επιδιόρθωση λαθών και δημιουργία αναφοράς (20^η Ιουνίου – 25^η Ιουνίου)

Ερωτηματολόγιο: <https://forms.gle/bwwiQVo8g1m3ehcr6>

Πίνακας απαντήσεων:

<https://docs.google.com/spreadsheets/d/1WcYVFnGXgaxxgbCNbbCDSX7VoRqn41gxJUb-GnMkes8/edit?resourcekey#gid=1563126204>

Για την δημιουργία του ερωτηματολογίου καθώς και τα αποτελέσματα του χρησιμοποιήθηκε το Google Forms και το google Doc

User Personas

1. Όνομα: Χάρης

Ηλικία: 32

Υπόβαθρο: Στον Χάρη αρέσει πολύ να ασχολείται με διαφορά games και άλλες ασχολίες περί αυτών στον ελεύθερο του χρόνο. Σαν επισκέπτης εκδηλώσεων εκτιμά πολύ την επιλογή να μπορεί να επιλεγεί την θέση κατά την διάρκεια των προβολών. Επίσης, έχει ένα αγαπημένο event στο οποίο θέλει να παραβρίσκεται οπότε διοργανώνεται.

Στόχοι: Ο κύριος στόχος του Χάρη είναι να μπορεί να βρει ευκολά και γρηγορά το αγαπημένο του event και να εξασφαλίσει την αγαπημένη του θέση. Θέλει μια φιλική προς τον χρήστη ιστοσελίδα ώστε να μπορεί τα κάνει όλα αυτά.

Συμπεριφορά: Ο Χάρης ελέγχει συχνά την ιστοσελίδα για ενημερώσεις σχετικά με το αγαπημένο του event αλλά και για αλλά. Επίσης του αρέσει να κοιτάει τις πληροφορίες για κάθε εκδήλωση ώστε αν υπάρχει κάποια που μοιάζει με την αγαπημένη του εκδήλωση να μπορεί να κλείσει εισιτήρια και για αυτήν.

2. Όνομα: Σοφία

Ηλικία: 25 ετών

Υπόβαθρο: Η Σοφία είναι μια δημοφιλής influencer στον χώρο των gaming events και όχι μόνο. Έχει μεγάλη ακολουθία από άτομα σε διάφορες κοινότητες που αφορούν το gaming. Δημιουργεί περιεχόμενο σχετικά με τις εκδηλώσεις στις οποίες παρευρίσκεται και από Live streaming τα οποία κάνει. Είναι πάντα πρόθυμη να πηγαίνει σε καινούργιες εκδηλώσεις για να μοιραστεί με το κοινό της.

Στόχοι: Η Σοφία θέλει να ανακαλύπτει συνέχεια νέες εκδηλώσεις και να έχει εύκολη καθοδήγηση προς αυτές. Επίσης της αρέσει να πηγαίνει στις εκδηλώσεις αυτές με τους φίλους της αλλά και τους συνεργάτες της αν χρειαστεί. Επιπλέον, θέλει να συνεργάζεται με τους διοργανωτές των εκδηλώσεων και να τις προωθεί στου ακολούθους της.

Συμπεριφορά: Η Σοφία δαπανά αρκετό χρόνο σε πλατφόρμες κοινωνικών μέσων, αναζητώντας πληροφορίες και αλληλοεπιδρώντας με την κοινότητα των gamers. Προτιμά ένα website με ελκυστική εμφάνιση οπού θα μπορεί να παρακολουθεί όλες τις επερχόμενες εκδηλώσεις, να βλέπει τις πληροφορίες για αυτές και να μπορεί να επιλεγεί ένα έξτρα είδος εισιτήριου για εκείνη και τους φίλους της.

3. Όνομα: Άλεξ

Ηλικία: 19 ετών

Υπόβαθρο: Ο Άλεξ είναι ένας παθιασμένος gamer που απολαμβάνει να παρευρίσκεται σε gaming events και να είναι ενήμερος για τις τελευταίες τάσεις στη βιομηχανία των video

games. Έχει παρακολουθήσει αρκετές εκδηλώσεις στο παρελθόν και πάντα αναζητά νέες για να παρακολουθήσει.

Στόχοι: Ο Άλεξ θέλει να ανακαλύψει προσεχή gaming events, να μπορεί να βλέπει τις πληροφορίες για αυτά τα events και να αγοράσει εισιτήρια για τα events που τον ενδιαφέρουν.

Συμπεριφορά: Ο Άλεξ αφιερώνει σημαντικό χρόνο στην αναζήτηση που αφορούν το gaming στο διαδίκτυο. Εκτιμά την ευκολία χρήσης και τις φιλικές προς τον χρήστη websites.

Απογοητεύσεις: Να χάνει events λόγω μη πληροφόρησης

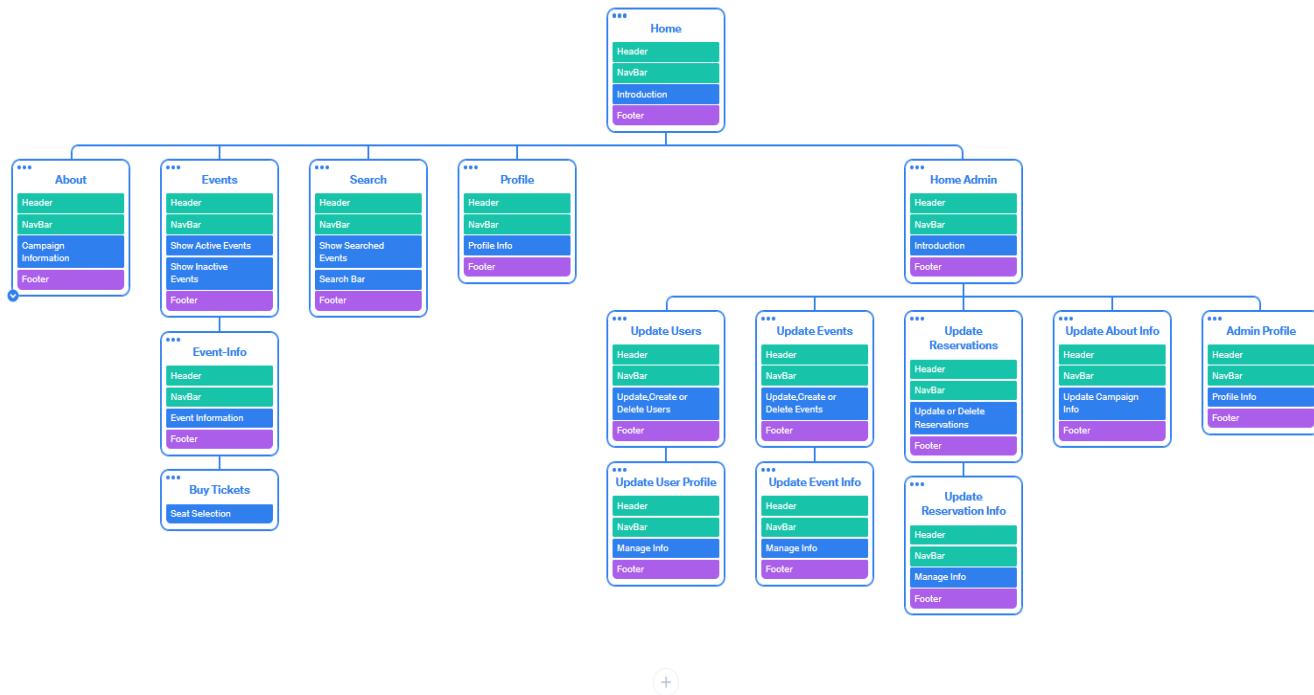
User Stories

1. Ως χρήστης θέλω να μπορώ να κάνω sign up/log in στην εφαρμογή
2. Ως χρήστης θέλω να περιηγούμαι στις σελίδες με ευκολία, τόσο στον υπολογιστή όσο και στο κινητό/tablet
3. Ως χρήστης θέλω να έχω πρόσβαση στο προφίλ μου και να μπορώ να τα επεξεργαστώ
4. Ως χρήστης θέλω να μπορώ να δω τις γενικές πληροφορίες της καμπάνιας
5. Ως χρήστης θέλω να βλέπω τις προγραμματισμένες εκδηλώσεις της καμπάνιας, όπου θα εμφανίζονται οι ολοκληρωμένες και οι μελλοντικές εκδηλώσεις
6. Ως χρήστης θέλω να έχω την δυνατότητα να επιλεγώ μια εκδήλωση και να βλέπω περισσότερες πληροφορίες σχετικά με αυτήν
7. Ως χρήστης θέλω να μπορώ να βλέπω την τοποθεσία της εκδήλωσης μέσα σε έναν χάρτη
8. Ως χρήστης θέλω να μπορώ να πραγματοποιήσω κράτηση για την προβολή της παρουσίασης στις εκδηλώσεις
9. Ως χρήστης θέλω να μπορώ να επιλεγώ τον τύπο του εισιτήριου αλλά και τον αριθμό των εισιτήριων
10. Ως χρήστης θέλω να μπορώ να αναζητήσω μια εκδήλωση με βάση την ονομασία της
11. Ως χρήστης θέλω να μπορώ να μεταφερθώ στα social media της καμπάνιας
12. Ως χρήστης θέλω να έχω την επιλογή να μπορώ να αλλάξω την εμφάνιση της σελίδας σε dark mode
13. Ως διαχειριστής θέλω να έχω πρόσβαση στην αρχική σελίδα διαχείρισης αλλά και στην αρχική σελίδα ενός χρήστη
14. Ως διαχειριστής θέλω να μπορώ να κάνω login και logout με τα στοιχεία που μου έχουν δοθεί από οποιαδήποτε σελίδα
15. Ως διαχειριστής θέλω να μπορώ να έχω πρόσβαση σε μια σελίδα όπου θα εμφανίζονται όλοι οι χρήστες και να μπορώ να του τροποποιώ τα στοιχεία, να δημιουργώ και νούμερα αλλά και να τους διαγραφώ
16. Ως διαχειριστής θέλω να έχω πρόσβαση σε μια σελίδα όπου θα εμφανίζονται οι βασικές πληροφορίες μια καμπάνιας, να τις τροποποιώ και να διαγραφώ πληροφορίες
17. Ως διαχειριστής θέλω να μπορώ να βλέπω όλες τις εκδηλώσεις, να βλέπω τα στοιχεία τους όπου θα μπορώ να προσθέτω, να τροποποιώ ή και να διαγραφώ τις πληροφορίες.

18. Ως διαχειριστής θέλω να μπορώ να βλέπω όλες τις κρατήσεις, τις πληροφορίες των κρατήσεων και να μπορώ να τις τρόποι αλλά και να διαγραφώ τις ιδίες τις κρατήσεις.

Visual Sitemap, Wireframe, mockup:

Visual sitemap:

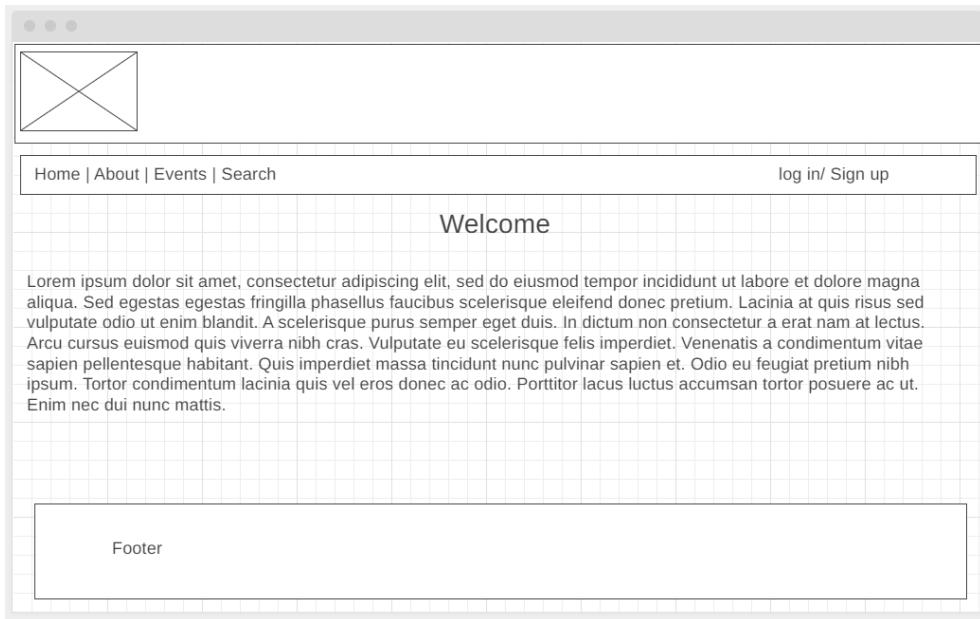


Για την δημιουργία του sitemap χρησιμοποιήθηκε η παρακάτω ιστοσελίδα:

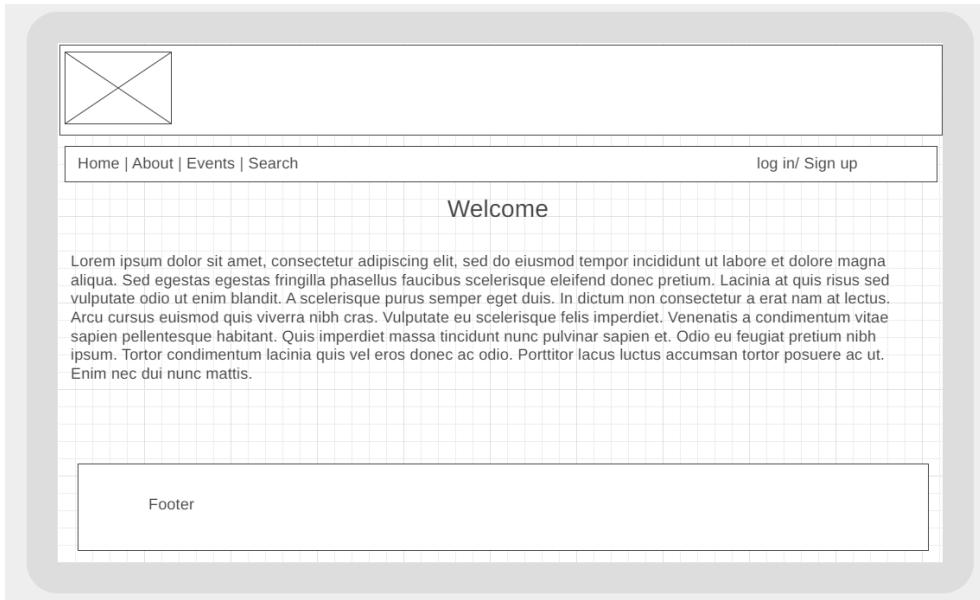
<https://octopus.do/l4mzl3wg4lo>

Wireframe:

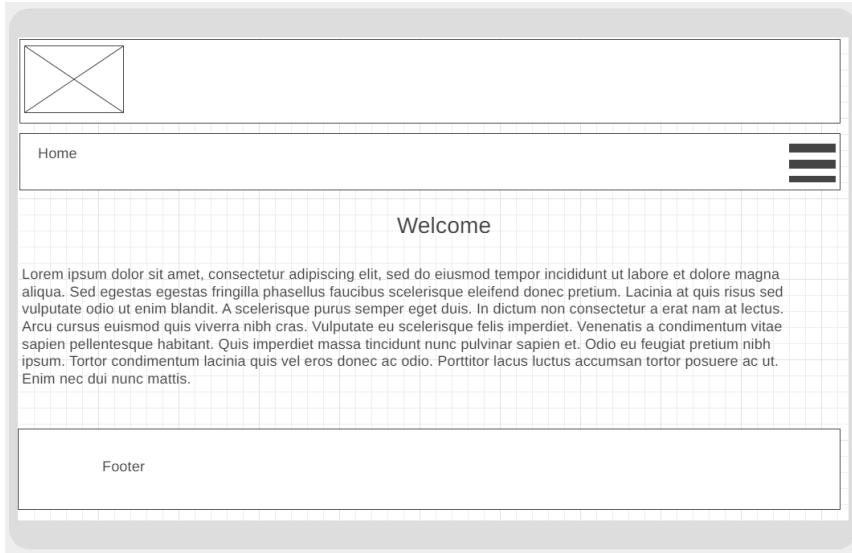
Desktop:



Tablet:



Smartphone:

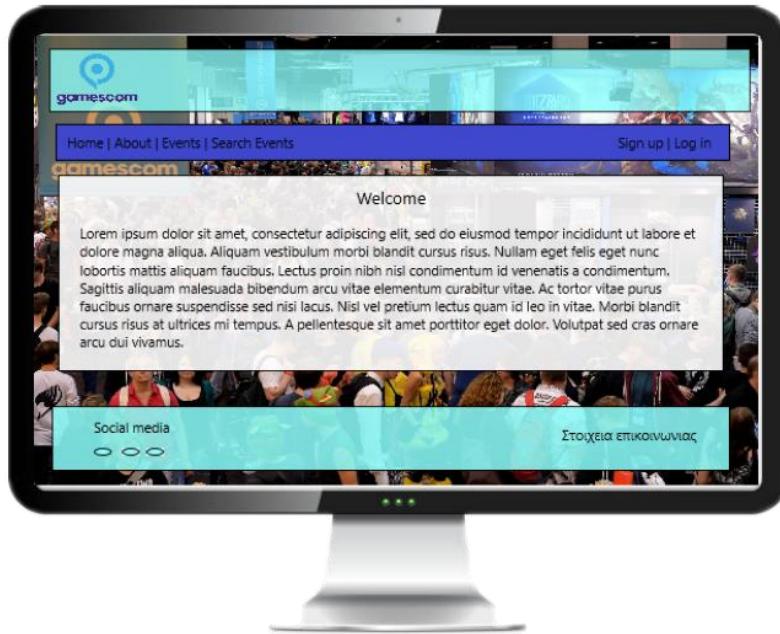


Με την επιλογή που θα ανοίγει το dropdown menu στο topnav bar



Mockup:

Desktop



Tablet



Smartphone:

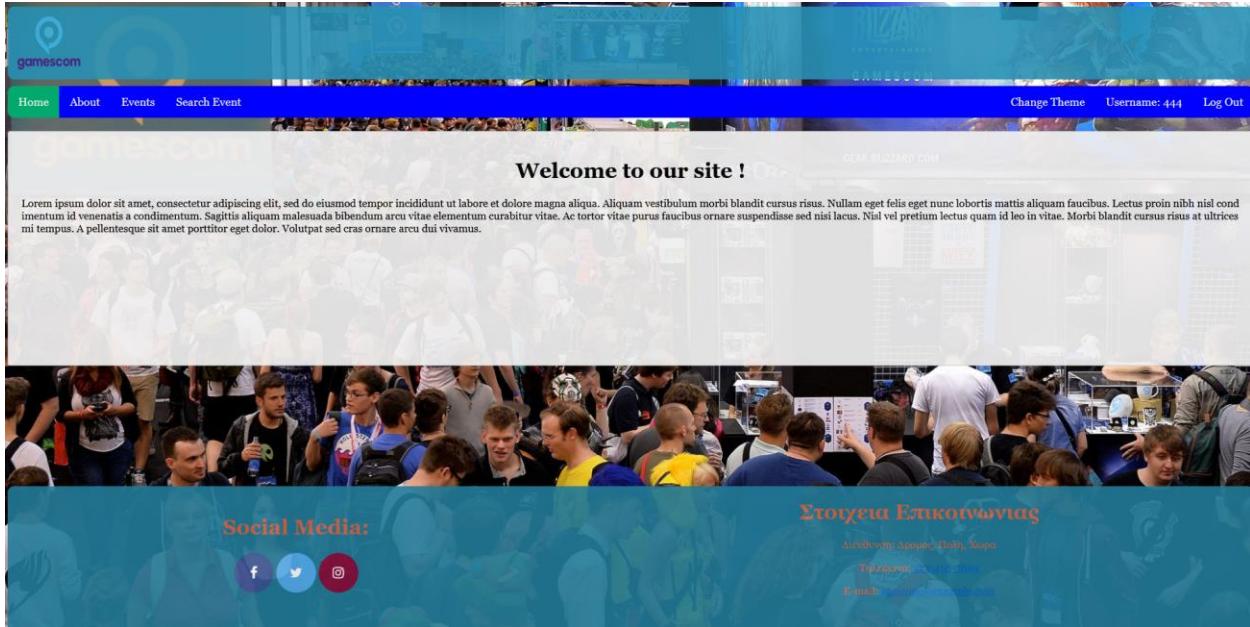


Ενότητα 2: Σύγχρονες Τάσεις Διαδικτυακών Εφαρμογών

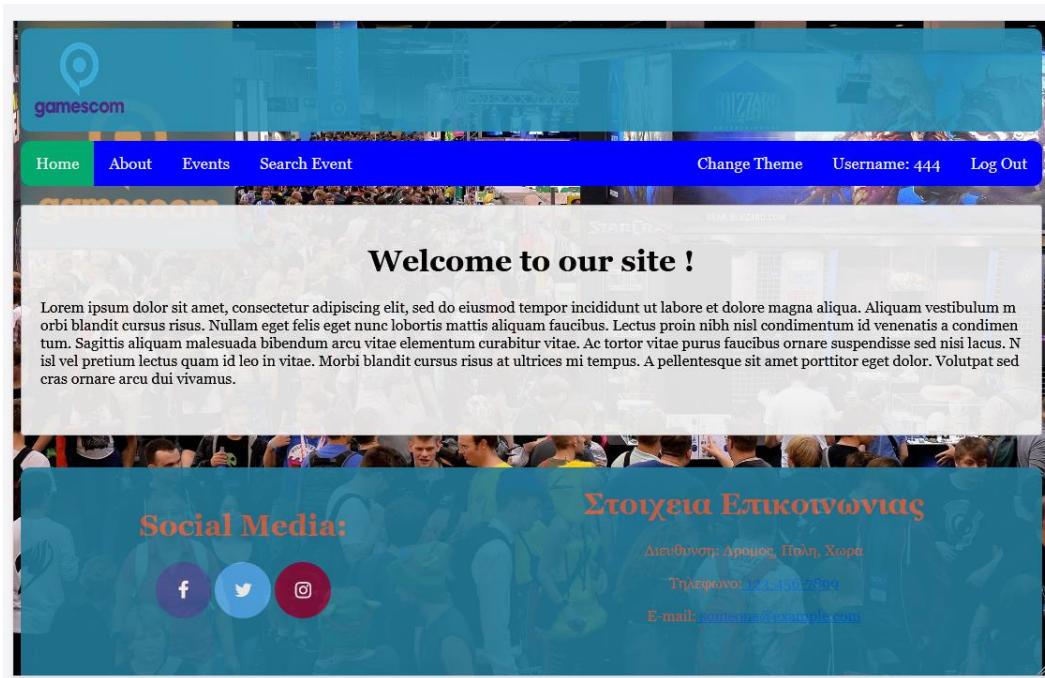
Στην εργασία μας χρησιμοποιήσαμε 3 Σύγχρονες Τάσεις Διαδικτυακών εφαρμογών του 2023. Αυτές είναι οι: Dark Mode, JavaScript Framework και Responsive Design.

Responsive Design: Στην αρχική σελίδα, με την αλλαγή των διαστάσεων λόγο ανοίγματος της εφαρμογής σε διαφορετικές συσκευές ή αναδιάταξής της σελίδας από desktop, τα στοιχεία της σελίδας θα αλλάξουν μέγεθος αντίστοιχα.

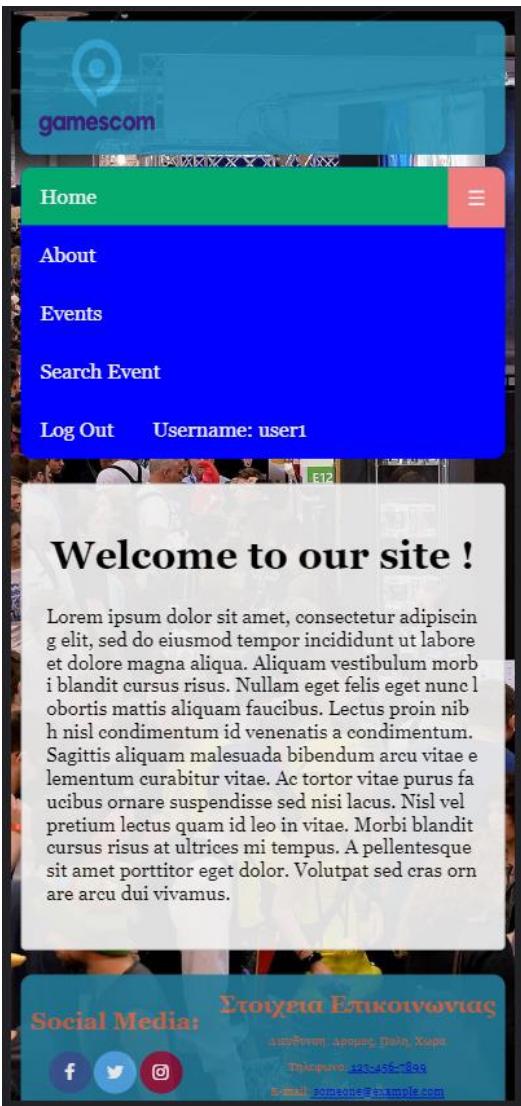
Desktop:



Tablet:



Smartphone:

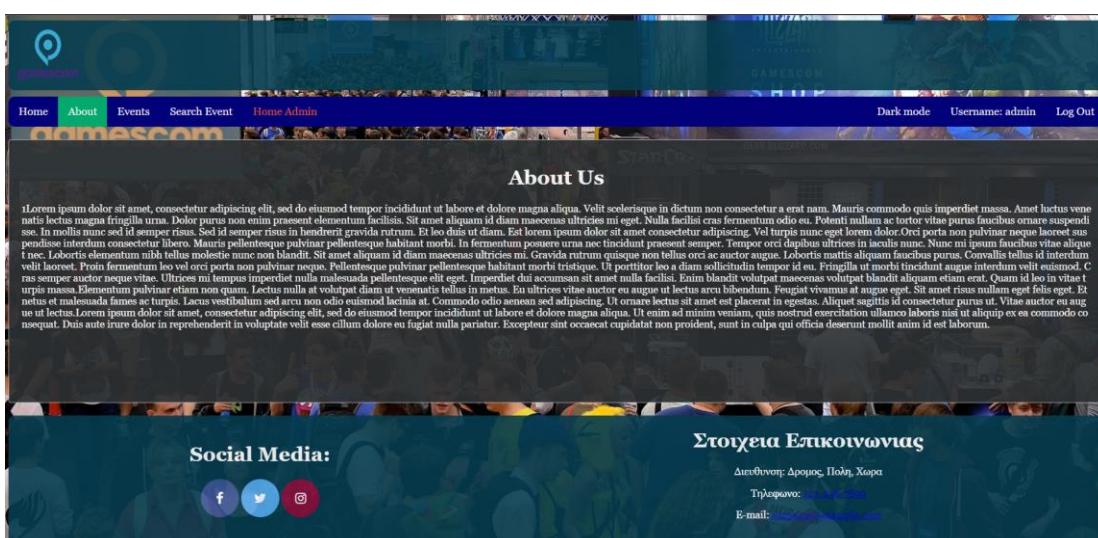


JavaScript Framework: Για την εφαρμογή μας χρησιμοποιήσαμε Node.js για την υλοποίηση του Server με Express Js.

```
JS index.js > ...
1  const express = require('express');
2  const { MongoClient } = require('mongodb');
3  const path = require('path');
4  const bodyParser = require('body-parser');
5  const session = require('express-session');
6  const { log } = require('console');
7  const { v4: uuidv4 } = require('uuid');
8
9
10 |
11
12
13 const app = express();
14 const port = 3000;
15
16 // Connection URL
17 const url = 'mongodb://127.0.0.1:27017';
18 const client = new MongoClient(url);
19
20 // Database Name
21 const dbName = 'test2';
22
23
24 //set the view engine to EJS
25 app.set('view engine', 'ejs');
26 app.engine('html', require('ejs').renderFile); //register both .html and .ejs files for rendering
27
28 // Middleware to parse the request body
29 app.use(bodyParser.urlencoded({ extended: true }));
30 app.use(bodyParser.json());
31
32
33 // Configure session middleware
34 app.use(
35   session({
36     secret: 'mySecret',
37     resave: false,
38     saveUninitialized: true,
39   })
40 );
41
42 // Copy static files from the "static" directory
```

Dark Mode: Πατώντας ένα κουμπί σε οποιαδήποτε σελίδα, η εφαρμογή αλλάζει σε dark mode ή light mode.

Dark mode



Ενότητα 3: Τεχνολογίες Εφαρμογής

Οι τεχνολογίες που χρησιμοποιήθηκαν είναι:

Nodejs

ExpressJs

EJS

JavaScript

Html

CSS

MongoDB

NodeJS – MongoDB : Χρησιμοποιήθηκαν για την αποθήκευση πληροφοριών στην εφαρμογή πχ. Λογαριασμούς χρηστών, Δεδομένα της καμπάνιας, Πληροφορίες των εκδηλώσεων και Αποθήκευση των Εισιτήριών.

(Χρησιμοποιήθηκαν στις /About.html, /sign_up, /log_in, /update_profile, /events.html, /event-info.html, /buy_tickets.html, /updateUsers.html, /submit_seats, /updateUserProfile.html, /updateUserProfile, /deleteUser, /eventSearch.html, /getSearchQuery, /updateEvent.html, /updateEventInfo.html, /updateEventInfo,

/deleteEvent, /createEvent, /updateEvent, /updateReservation.html, /deleteReservation, /updateReservationInfo.html, /updateReservationInfo, /updateAbout.html, /updateAboutInfo)

HTML/CSS : Χρησιμοποιήθηκαν για την αισθητική εμφάνιση της σελίδας. Όλες οι σελίδες είναι γραμμένες σε HTML και καλούν αρχεία CSS

(Χρησιμοποιήθηκαν στις /About.html, events.html, /event-info.html, /buy_tickets.html, /updateUsers.html, /updateUserProfile.html, /updateEvent.html, /updateEventInfo.html, /updateReservation.html, /updateReservationInfo.html, /updateAbout.html, /homeAdmin.html, /homeT.html, /profile.html, /eventSearch)

Javascript/EJS: Χρησιμοποιήθηκαν για την δια δραστικότητα του χρήστη με την σελίδα. Άλλαζαν τα δεδομένα που εμφανίζονται ανάλογα με τις επιλογές του χρήστη (πχ Dark Mode, Search Bar) καθώς και έδειχναν τις πληροφορίες που έπαιρναν από την βάση δεδομένων. Όλες οι σελίδες χρησιμοποίησαν την EJS/JavaScript για την αλλαγή πληροφοριών τους.

(Χρησιμοποιήθηκαν στις /About.html, events.html, /event-info.html, /buy_tickets.html, /updateUsers.html, /updateUserProfile.html, /updateEvent.html, /updateEventInfo.html, /updateReservation.html, /updateReservationInfo.html, /updateAbout.html, /homeAdmin.html, /homeT.html, /profile.html, /eventSearch)

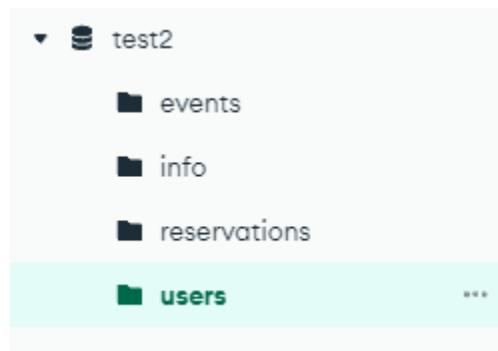
ExpressJs: ο βασικός παράγοντας για την εμφάνιση των σελίδων και το routing. Σε κάθε ενέργεια του χρήστη, Δήλωση φόρμας ή άνοιγμα ενός συνδέσμου, εμφάνιζε / έπαιρνε τα στοιχεία από την σελίδα.

Όλες οι σελίδες χρησιμοποίησαν την ExpressJs για την εμφάνιση τους.

Ενότητα 4: Εγχειρίδιο Χρήσης Εφαρμογής

Ανάλυση της βάσης δεδομένων:

Η βάση που δημιουργήθηκε έχει το όνομα “test2” και περιέχει 4 collections, τα “events”, “info”, “reservations” και “users”



Αρχικά, το collection “events” περιέχει τα documents που αφορούν τις εκδηλώσεις. Κάθε document αποτελείται απτά fields:

The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with 'My Queries' and 'Databases' sections. Under 'Databases', 'test2' is selected, and its 'events' collection is highlighted. A search bar is at the top. The main area is titled 'test2.events' and has tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. Below the tabs is a search bar with placeholder text 'Type a query: { field: 'value' }'. There are 'Filter' and 'ADD DATA' buttons. An 'EXPORT COLLECTION' button is also present. The 'Documents' section contains three documents, each with a red-bordered JSON representation:

```

_id: ObjectId('64977f4e5387f5564b6dc478')
name: "event1"
active: "1"
date: "25/08/2023"
info: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod..."
hours: "16:00-00:30"
url: "https://www.athensvoice.gr/images/1074x600/jpg/sites/default/files/art..."
map_url: "https://www.google.com/maps/embed?pb=!m14!1m8!1m3!1d12587.03294320619..."

_id: ObjectId('64977faa5387f5564b6dc479')
name: "event2"
active: "1"
date: "04/07/2023"
info: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod..."
hours: "12:00-23:00"
url: "https://www.sport24.gr/img/4790/8885955/121000/fb1200/1200/gamel.jpg"
map_url: "https://www.google.com/maps/embed?pb=!m14!1m8!1m3!1d12587.03294320619..."

_id: ObjectId('64977ff25387f5564b6dc47a')
name: "event3"
active: "0"
date: "23/05/2023"
info: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod..."
hours: "19:00-00:30"
url: "https://s.yimg.com/uu/api/res/1.2/l915AIHABnk4rR4pgeKBAG---B/Zmk9Zmlsb..."
map_url: "https://www.google.com/maps/d/embed?mid=1TAAr1uL9uMp8LuducGfczYLUR0&h..."

```

1. name: που κρατάει το όνομα του event
2. active: που είναι είτε 0 είτε 1 , αναλόγως αν το event είναι μελλοντικό η όχι, 1 σημαίνει ότι είναι μελλοντικό event
3. date: που κρατάει το date ενός event
4. info: οπού κρατάει τις σχετικές πληροφορίες
5. hours: οπού είναι οι ώρες που διοργανώνεται το event
6. url: οπού είναι το url της εικόνας που θα έχει το event στην σελίδα με τις πληροφορίες του
7. map_url: οπού είναι το url της τοποθεσίας της εκδήλωσης από το google maps

To collection "info" αποτελείται με documents οπού έχουν ως fields:

The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with a tree view of collections: 'local', 'test2' (expanded), 'events', 'info' (selected and highlighted in green), 'reservations', and 'users'. At the top right, there are buttons for 'ADD DATA' and 'EXPORT COLLECTION'. To the right of the sidebar, a preview pane displays a single document from the 'info' collection. The document has two fields: '_id' (with a value of 'ObjectId('649736277afafe1953f19a41')') and 'info' (with a value of 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod...').

1. info: Το πεδίο αυτό κρατάει τις γενικές πληροφορίες της καμπάνιας

To collection "reservations" αποτελείται με documents οπού έχουν ως fields:

The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with a tree view of collections: 'test2' (expanded), 'events', 'info', 'reservations' (selected and highlighted in green), and 'users'. At the top right, there are buttons for 'ADD DATA' and 'EXPORT COLLECTION'. To the right of the sidebar, a preview pane displays a single document from the 'reservations' collection. The document has several fields: '_id' (with a value of 'ObjectId('649780265387f5564b6dc47b')'), 'id' (with a value of 'e09504d4-31fd-4202-a3a6-0ec2587a7533'), 'seats' (an array), 'username' (with a value of 'user1'), 'eventname' (with a value of 'event1'), 'total_price' (with a value of 20), 'ticket_amount' (with a value of 2), and 'ticket_type' (with a value of 'normal').

1. id: είναι ένα τυχαίο μοναδικό id ώστε κάθε κράτηση να ξεχωρίζει
2. seats: είναι ένα array οπού έχει τον αριθμό των θέσεων που έχει επιλέξει ο user όταν αγόρασε τα εισιτήρια
3. username: είναι το username του χρήστη που έκανε την κράτηση
4. event name: είναι το όνομα του event για το οποίο έγινε η κράτηση
5. total_price: είναι η συνολική τιμή που πρέπει να πληρώσει
6. ticket_amount: είναι ο αριθμός των εισιτήριων που έχει αγοράσει
7. ticket_type: είναι ο τύπος των εισιτήριων που έχει αγοράσει ο user

To collection "users" αποτελείται με documents οπού έχουν ως fields:

```

{
  "username": "admin",
  "name": "admin",
  "email": "admin@test.com",
  "password": "admin",
  "phone": "12345"
}

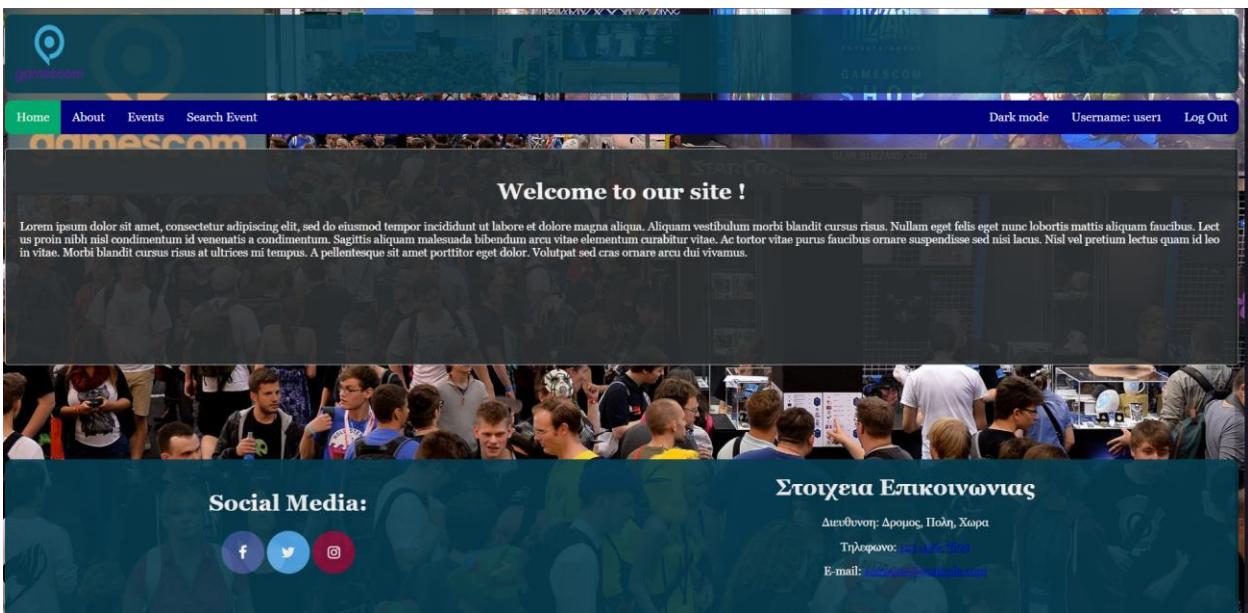
```

1. username: είναι ένα μοναδικό username που δίνεται στον user αυτόματα κατά την εγγραφή του
2. name: το όνομα του χρήστη (πάνω σε αυτό βασίζεται η δημιουργία του username)
3. email: το email του user
4. password: ο κωδικός του user
5. phone: το κινητό του user

Ανάλυση των media Queries:

Τα media που δημιουργήθηκαν έχουν τοποθετηθεί στο αρχείο “styles.css” στον φάκελο “other”.

Αρχικά, σε ένα desktop, η αρχική σελίδα μοιάζει έτσι:



Το πρώτο media query που χρησιμοποιούμαι αφορά τις συσκευές μεταξύ 472 και 768 Pixel.

The screenshot shows a code editor with two panes. The left pane displays a block of CSS code with line numbers from 506 to 552. The right pane shows a continuation of the CSS code, likely from a different file or a larger context, starting with a closing brace and ending with a closing brace for a 'fa' class definition. The code includes various CSS properties like 'display', 'float', 'position', 'width', 'padding', and 'font-size'.

```
506 /* Specific styles for phone devices up to 768px */
507 @media (max-width: 768px) and (min-width: 472px) {
508   .topnav a:not(:first-child), .dropdown .dropbtn {
509     display: none;
510   }
511   .topnav a.icon {
512     float: right;
513     display: block;
514   }
515
516   .topnav.responsive {position: relative;}
517   .topnav.responsive a.icon {
518     position: absolute;
519     right: 0;
520     top: 0;
521   }
522   #signup{
523     float: left;
524   }
525   .topnav.responsive a {
526     float: none;
527     display: block;
528     text-align: left;
529   }
530   .topnav.responsive .dropdown {float: none;}
531   .topnav.responsive .dropdown-content {position: relative;}
532   .topnav.responsive .dropdown .dropbtn {
533     display: block;
534     width: 100%;
535     text-align: left;
536   }
537
538   header{
539     padding: 0px;
540     font-size: 24px;
541     height: 100px;
542   }
543
544   .info-box{
545     font-size: 15px;
546   }
547
548   footer{
549     max-height: 150px;
550     padding: 0px;
551   }
552 }
```

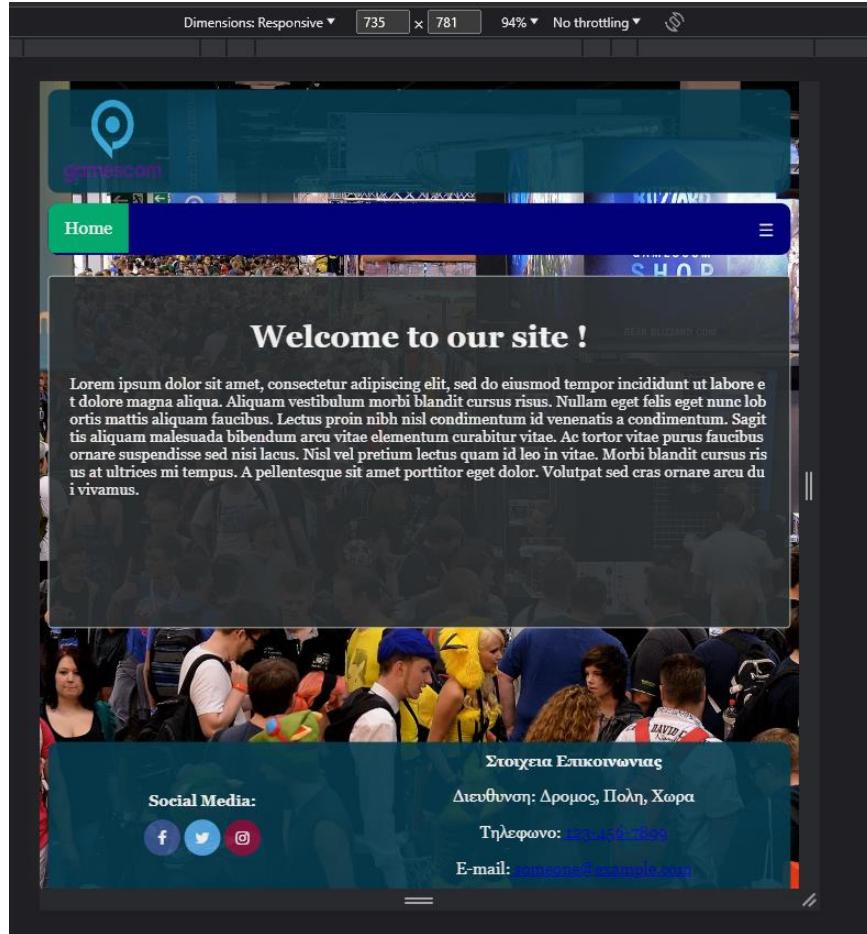
```
}
```

```
.fa{
  width: 35px;
  padding: 10px;
  font-size: 15px;
}
```

```
footer>div>div>h1{
  font-size: 15px;
}
.child>img{
  width: 250px;
}
```

Το πρώτο child της topnav bar θα παραμείνει ορατό ενώ όλα τα αλλά θα γίνουν επιλογές στο dropdown menu. Επίσης τοποθετείται αριστερά και ορίζεται ότι το position της κλάσης "topnav" θα είναι "responsive". Στην συνέχεια αλλάζει λίγο ο header καθώς και το "info-box" που κρατάει τις βασικές πληροφορίες του body που παρουσιάζονται στον χρήστη. Εκτός από αυτά αλλάζει και το μέγεθος του "footer" μαζί με το μέγεθος της γραμματοσειράς του. Τέλος αλλάζει μέγεθος και η εικόνα του logo πάνω στο header.

Έτσι η αρχική σελίδα θα μοιάζει κάπως έτσι όταν την βλέπει κάνεις σε συσκευές με τις συγκεκριμένες διαστάσεις:

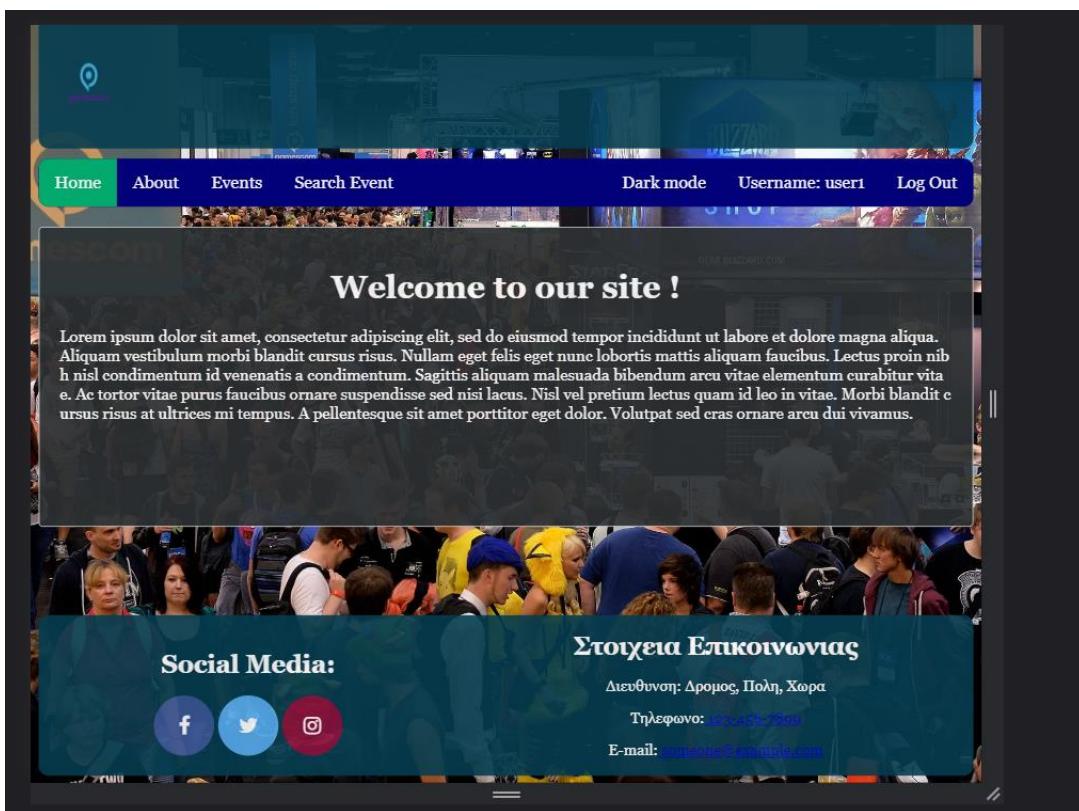


Το επόμενο media query που δημιουργήθηκε αφορά τις συσκευές μεταξύ 992 και 768 pixel. Οι συγκεκριμένες αλλαγές αφορούν κυρίως τα tablet και συσκευές με σχετικά μεγάλη οθόνη επομένως δεν αλλάζουν πάρα πολλά πράγματα. Αυτά που αλλάζουν είναι το μέγεθος του header, του footer και της εικόνας του logo.

```
/* Specific styles for tablet devices */
@media (max-width: 992px) and (min-width:768px) {

    header{
        padding: 0px;
        font-size: 24px;
        height: 130px;
    }
    header>img{
        width: 100px;
        height:100px;
        align-self: center;
        padding: 30px;
    }

    footer{
        height: 160px;
        padding: 0px;
        font-size: 15px;
    }
    footer>div>div>h1{
        font-size: 25px;
    }
    .child>img{
        width: 400px;
    }
}
```



Τέλος έχουμε το media query που αφορά τα κινητά και συσκευές με πολύ μικρή οθόνη

```
@media (max-width: 472px) {
    .topnav a:not(:first-child), .dropdown .dropbtn {
        display: none;
    }
    .topnav a.icon {
        float: right;
        display: block;
    }

    .topnav.responsive {position: relative;}
    .topnav.responsive a.icon {
        position: absolute;
        right: 0;
        top: 0;
    }
    #signup{
        float: left;
    }
    .topnav.responsive a {
        float: none;
        display: block;
        text-align: left;
    }
    .topnav.responsive .dropdown {float: none;}
    .topnav.responsive .dropdown-content {position: relative;}
    .topnav.responsive .dropdown .dropbtn {
        display: block;
        width: 100%;
        text-align: left;
    }

    .fa{
        width: 35px;
        padding: 10px;
        font-size: 15px;
    }

    footer{
        font-size: 10px;
        max-height: 150px;
    }
    .child>img{
        width: 100px;
    }
}
```

Εδώ οι αλλαγές είναι παρόμοιες με τις αλλαγές που έγιναν στο 1^o media query για τις συσκευές μεταξύ 472 και 768 pixel, ωστόσο εδώ τα αντικείμενα στην οθόνη μικραίνουν ακόμη περισσότερο ώστε να χωρέσουν στην οθόνη με έναν ωραίο τρόπο.



Εγχειρίδιο εγκατάστασης της εφαρμογής:

Η διαδικτυακή εφαρμογή χρησιμοποιεί την MongoDB για βάση δεδομένων, NodeJS με ExpressJS για backend και html με ejs για frontend.

Απαιτήσεις Συστήματος:

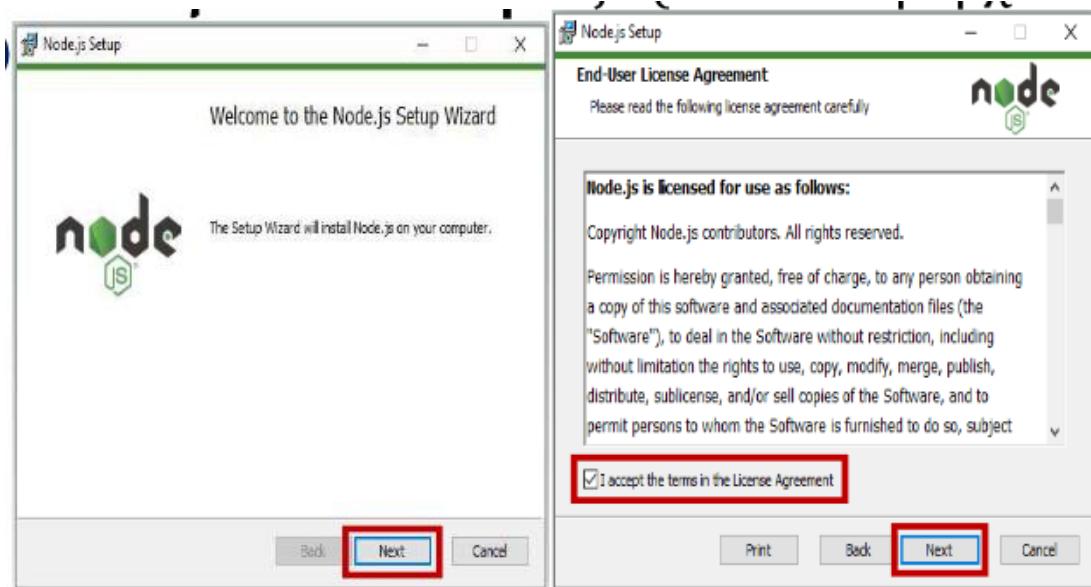
Λειτουργικό σύστημα: Windows

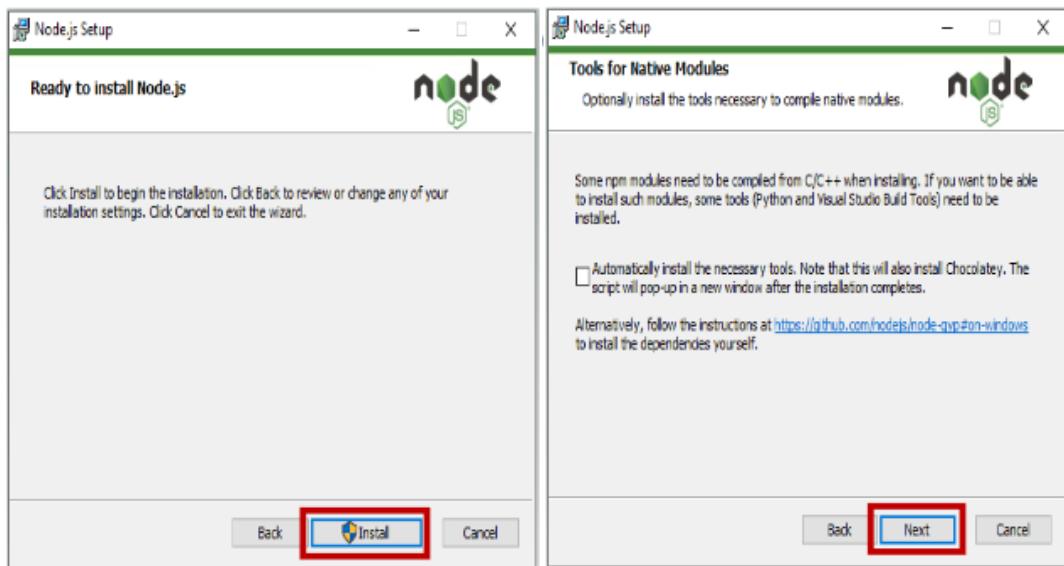
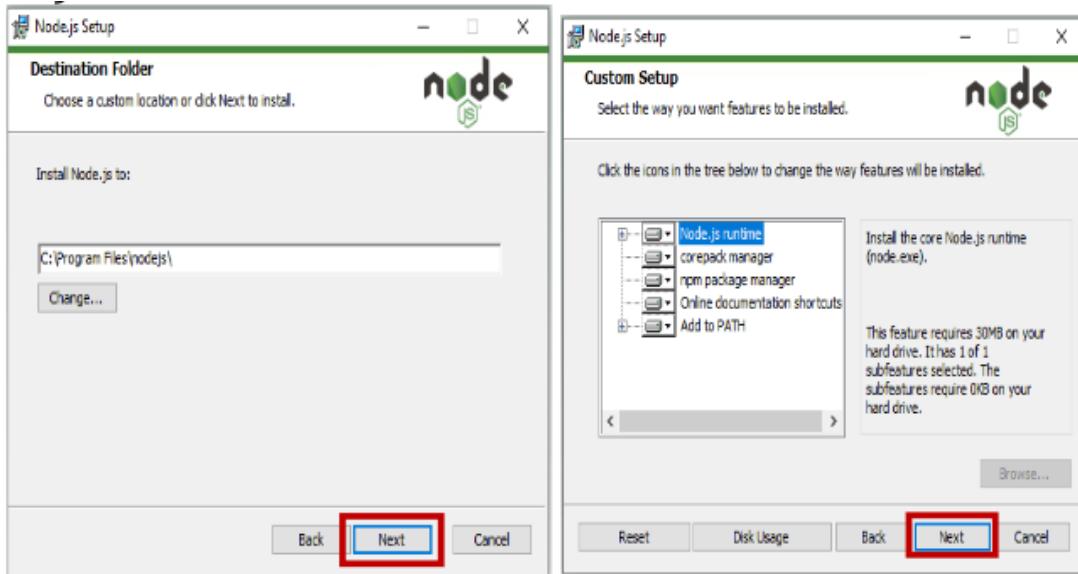
MongoDB: Version 6.0.6 ή νεότερη

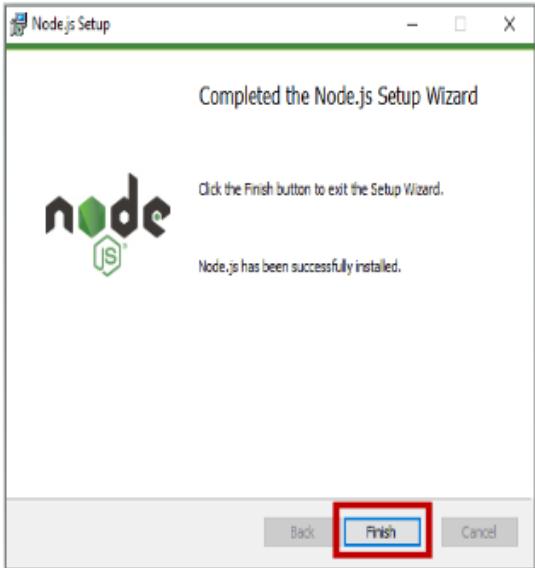
NodeJS: Version 18.16.0 ή νεότερη

Βήμα 1^ο : Εγκατάσταση της NodeJS

- 1) Επίσκεψη στο site της NodeJS <https://nodejs.org/en/#download> και εγκατάσταση της προτεινόμενης έκδοσης.
- 2) Ακολουθούμε τα βήματα από τις εικόνες για την σωστή εγκατάσταση της NodeJS







- 3) Μετά από επανεκκίνηση ελέγχουμε αν η εγκατάσταση έγινε σωστά.

```
C:\Users\User\Desktop\dprog\ergasiaDProgrammatismos>node -v  
v18.16.0
```

Βήμα 2^ο : Εγκατάσταση της MongoDB

- 1) Επίσκεψη στο site της MongoDB και εγκατάσταση της MongoDB Community Server
<https://www.mongodb.com/try/download/community>

MONGODB COMMUNITY SERVER

MongoDB Community Server Download

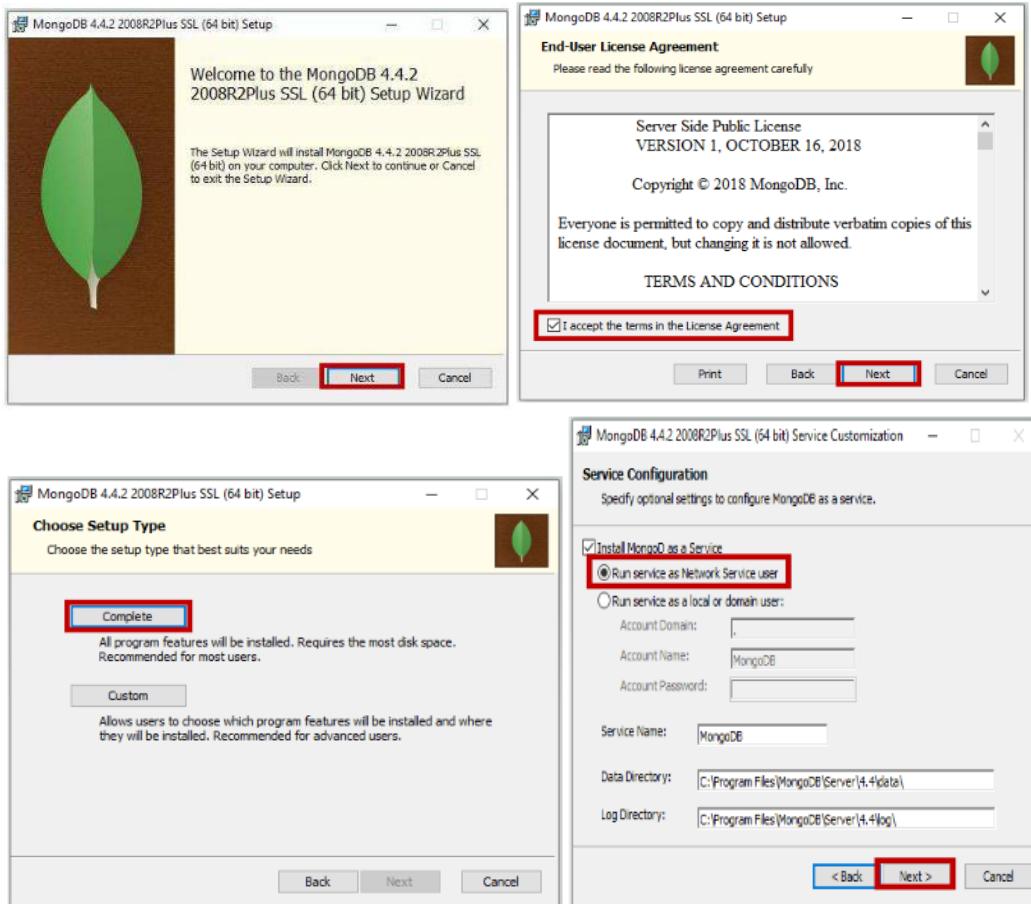
Version
6.0.6 (current)

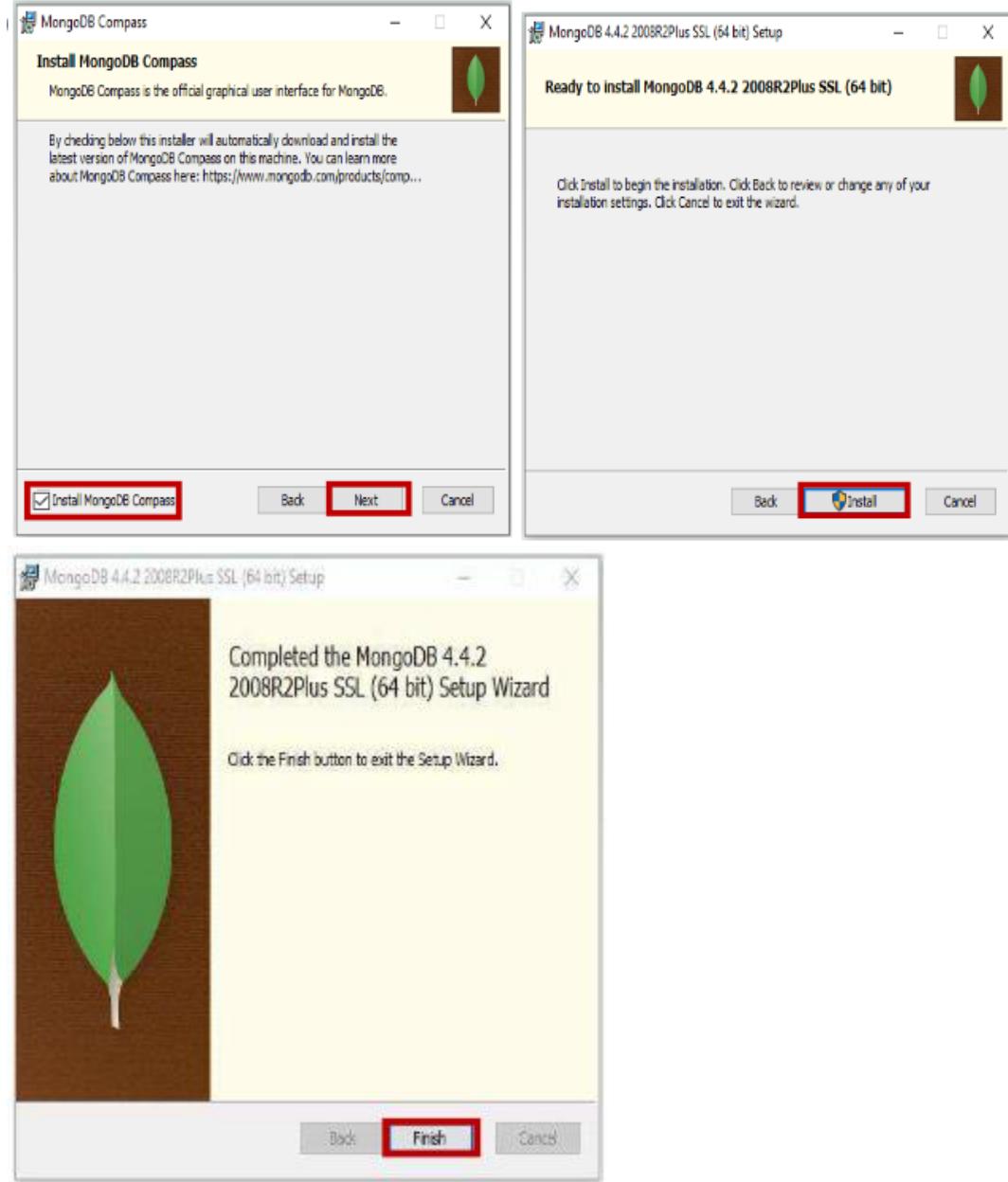
Platform
Windows x64

Package
msi

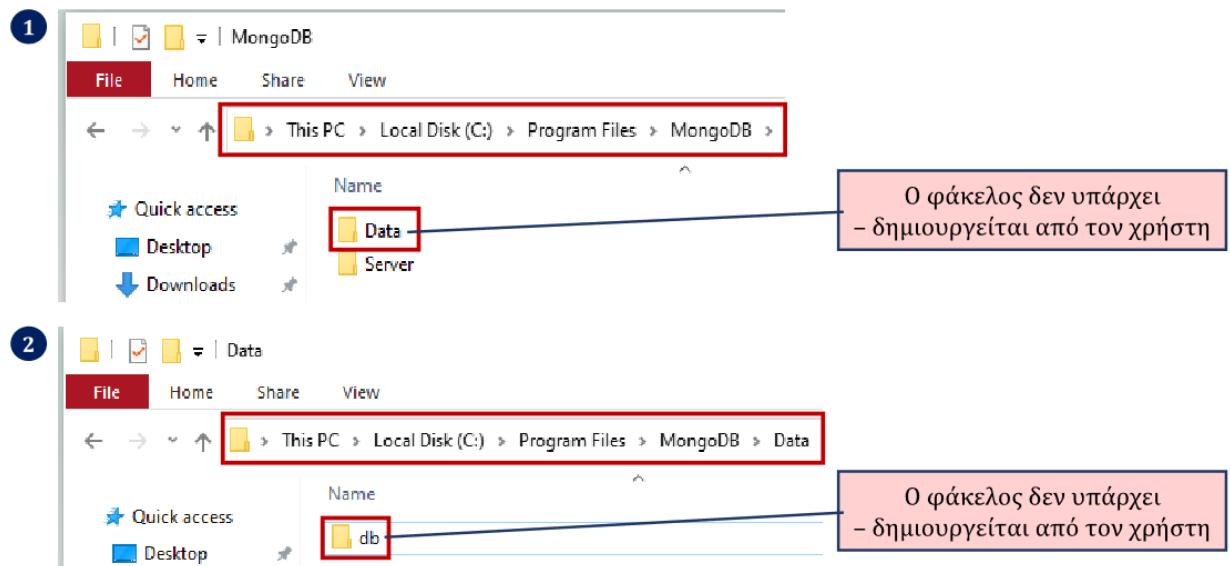
[Download](#)
 Copy link
More Options

2) Ακολουθούμε τα βήματα στις εικόνες για την σωστή εγκατάσταση





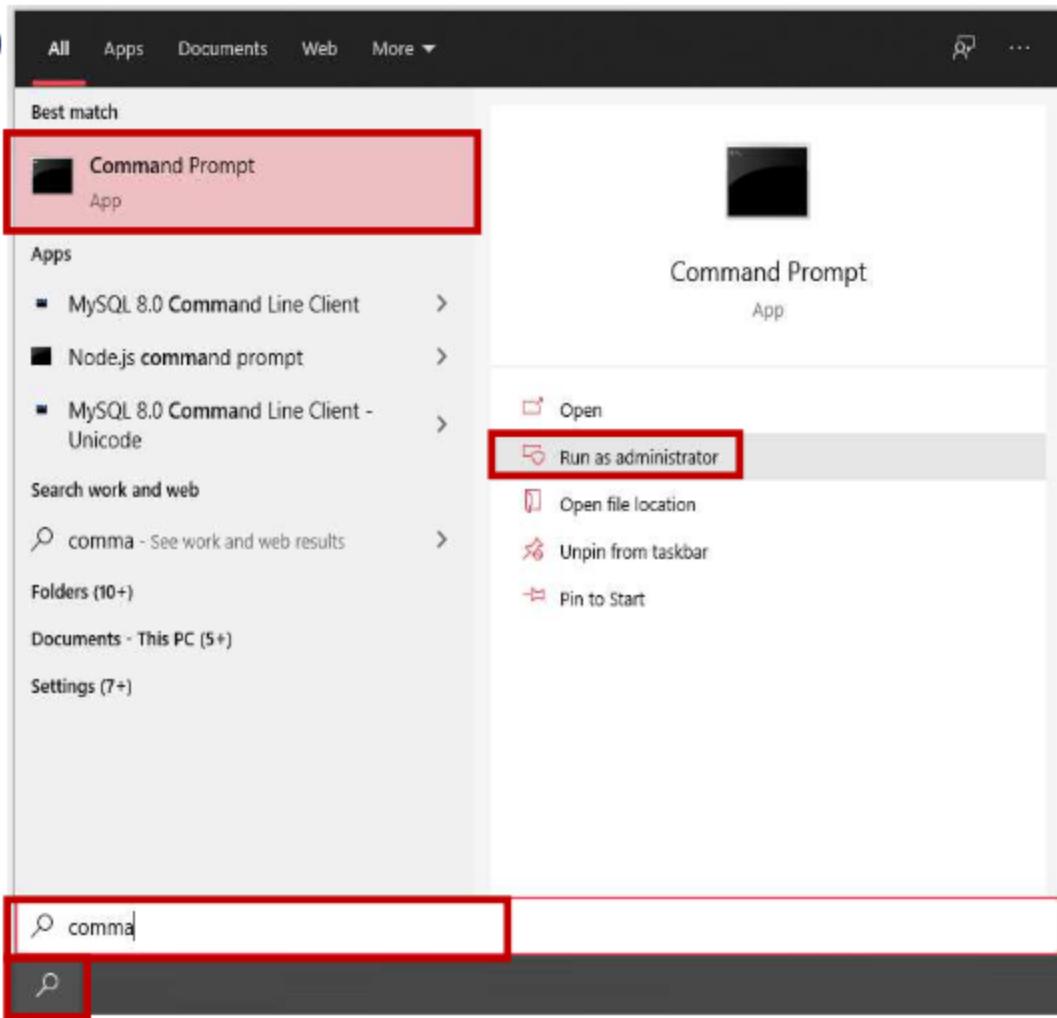
3) Για την ομαλή εκκίνηση του MongoDB Server και την λειτουργία της βάσης δημιουργούμε φάκελο για αποθήκευση βάσεων δεδομένων.



(Σημείωση: στην αρχή της εργασίας είχαμε ένα πρόβλημα με την εκκίνηση του mongodb server και με τον συγκεκριμένο τρόπο, τον οποίο βρήκαμε στα forum της mongo λύθηκε)

4) Παραμετροποίηση MongoDB

1



2

```
C:\ Administrator: Command Prompt
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Program Files\MongoDB\Server\4.4\bin
C:\Program Files\MongoDB\Server\4.4\bin>
```

3

```
C:\ Administrator: Command Prompt
C:\Program Files\MongoDB\Server\4.4\bin>mongod
{"t":{"$date":"2020-11-26T11:50:36.398+02:00"}, "s": "I", "c": "CONTROL", "id": 2
3285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to force-enable TL
S 1.0 specify --sslDisabledProtocols 'none'"}
 {"t":{"$date":"2020-11-26T11:50:36.716+02:00"}, "s": "W", "c": "ASIO", "id": 2
```

4

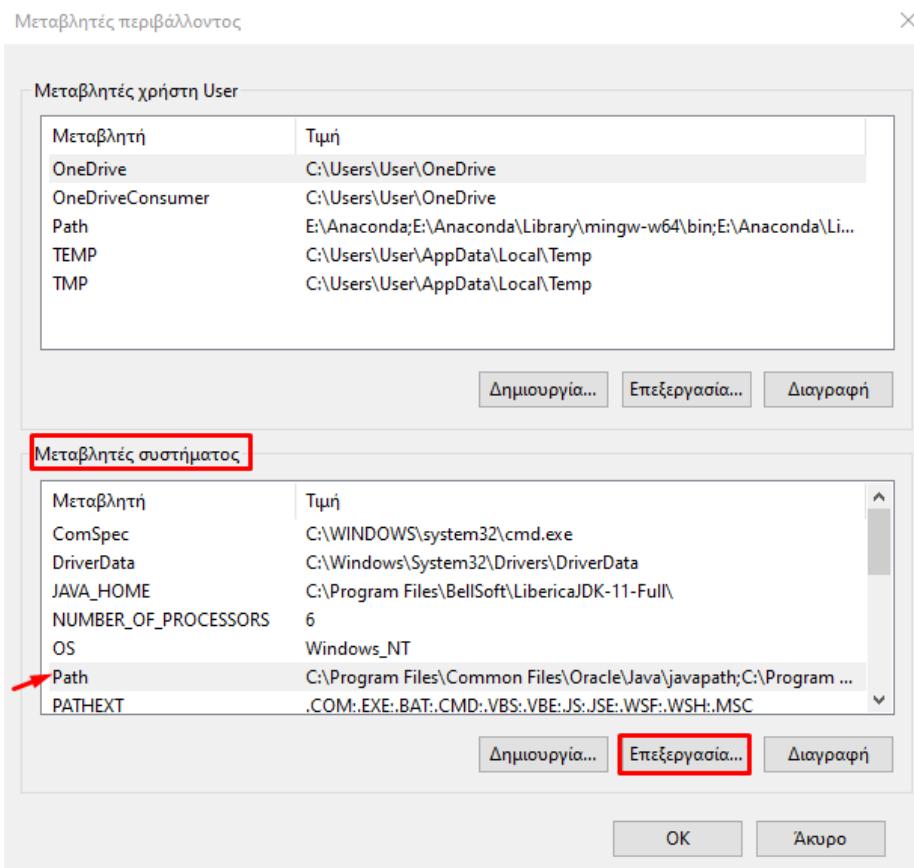
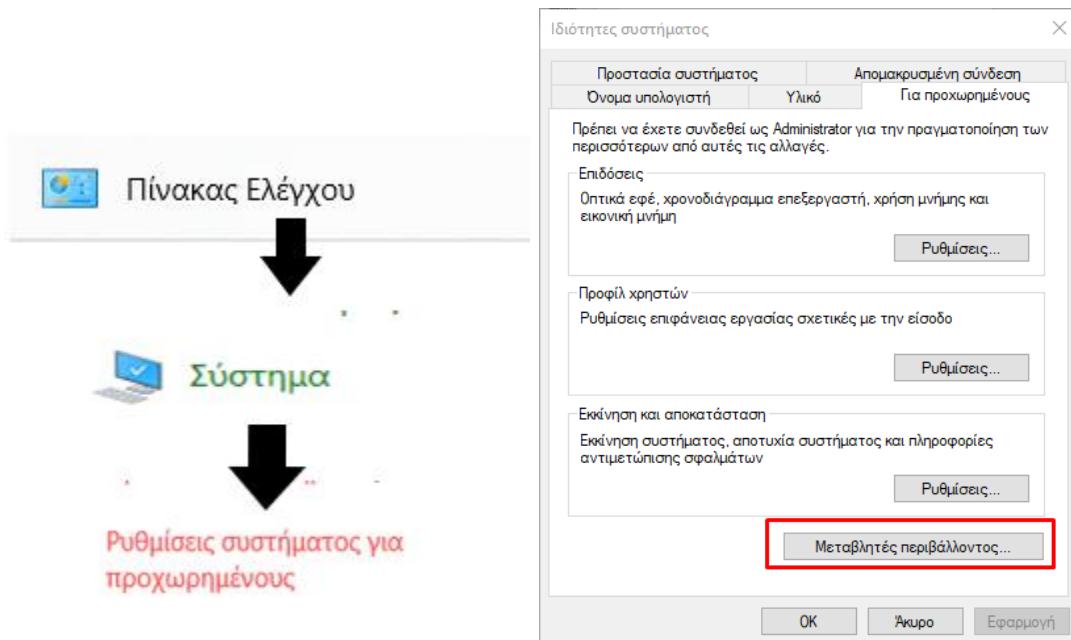
```
C:\ Administrator: Command Prompt - mongo
C:\Program Files\MongoDB\Server\4.4\bin>mongo
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceNam
e=mongodb
Implicit session: session { "id" : UUID("eb65f569-30b4-4a60-891b-afa997863531")
}
MongoDB server version: 4.4.2
```

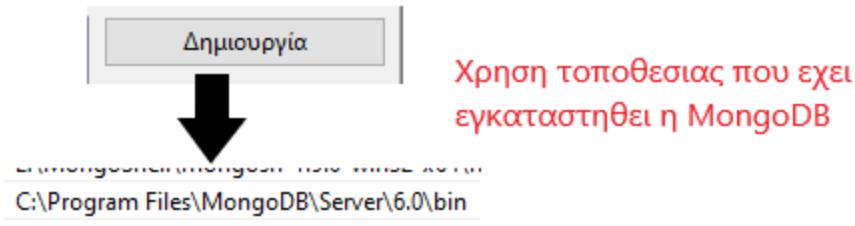
5

```
C:\ Administrator: Command Prompt
C:\Program Files\MongoDB\Server\4.4\bin>mongod --directoryperdb --dbpath "C:\Pr
ogram Files\MongoDB\Data\db" --install
{"t":{"$date":"2020-11-26T11:55:12.997+02:00"}, "s": "I", "c": "CONTROL", "id": 2
3285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to force-enable TL
S 1.0 specify --sslDisabledProtocols 'none'"}
```

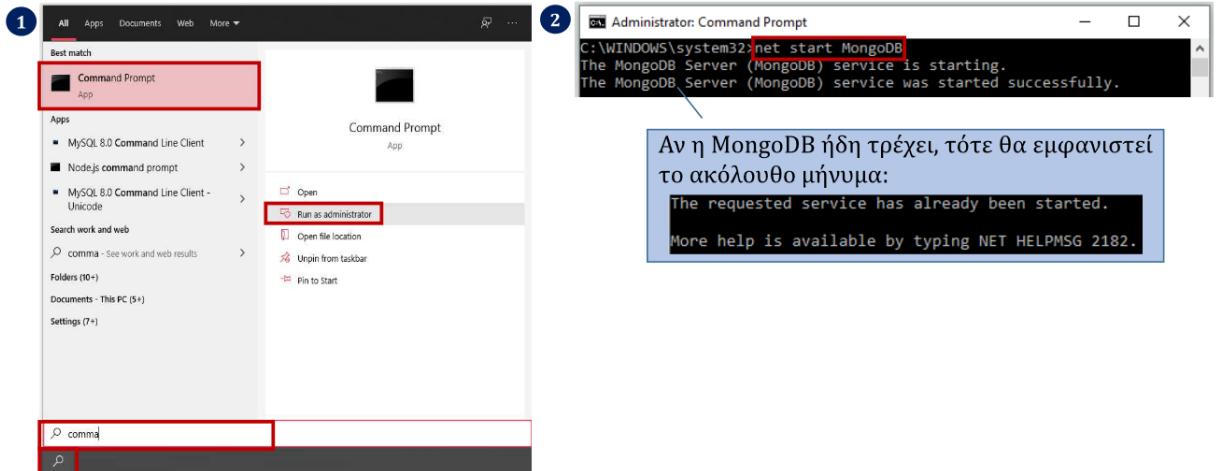
To path στον φάκελο που φτιάχτηκε στο 3^o βήμα της εγκατάστασης

5) Ενημέρωση μεταβλητής συστήματος (System Variable) Path

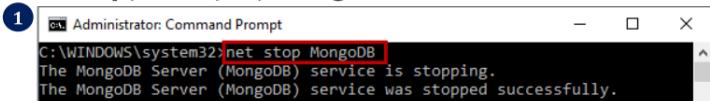




6) Εκκίνηση MongoDB



□ Τερματισμός MongoDB:



- 7) Εκκίνηση Mongo Shell ή Εγκατάσταση Mongo Compass και εκκίνηση για την προβολή των βάσεων δεδομένων.
- 8) Έλεγχος της Mongo

```
mongo --version
```

- 9) Εγκατάσταση MongoDB node module μέσω cmd, ακολουθώντας τα βήματα από το site <https://www.npmjs.com/package/mongodb>

```
npm init -y
```

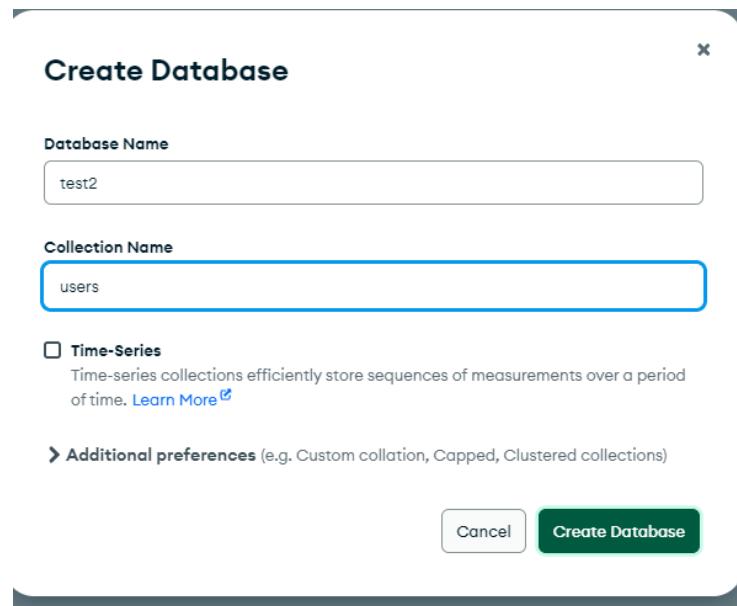
Next, install the driver as a dependency.

```
npm install mongodb
```

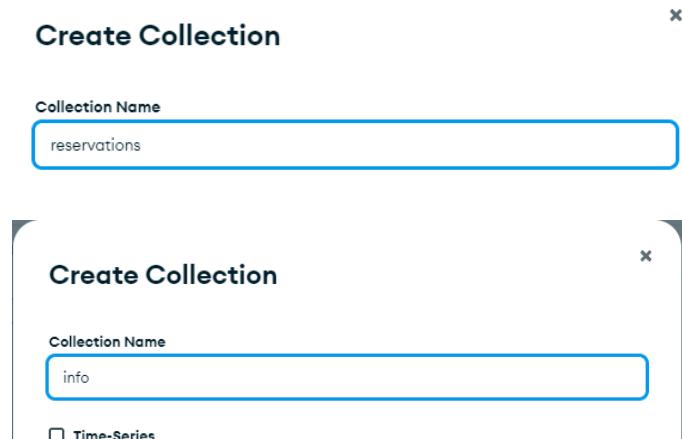
(Σημείωση το ‘npm init -y’ δημιουργεί ένα json αρχείο με το όνομα package.json το οποίο θα γραφεί μέσα τα στοιχεία της εφαρμογής. Η εφαρμογή δεν το χρειάζεται επειδή ο φάκελος το περιέχει ήδη.)

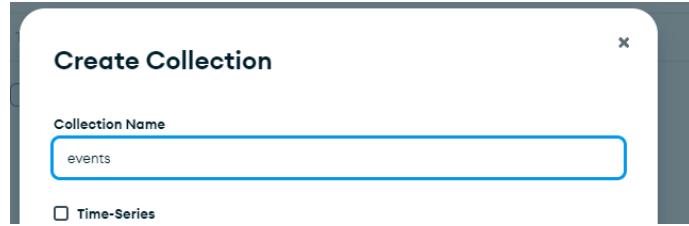
10) Δημιουργία mongodb database και collections:

Για να δημιουργηθεί η βάση δεδομένων με την συγκεκριμένη ονομασία που χρειάζεται πρέπει να δημιουργήσουμε μια βάση με το όνομα “test2” μια αρχική collection με το όνομα “users”.



Στην συνέχεια δημιουργούμε τα υπόλοιπα collections που αντιστοιχούν στα αρχεία json που θα κάνουμε import.





Τέλος πηγαίνουμε σε κάθε collection και κάνουμε Import το αντίστοιχο αρχείο json

Ομοίως κάνουμε για τα αλλά collections.

Βήμα 3^ο : Κανουμε αποσυμπιεση στον φακελο με την εφαρμογη και τρεχουμε την εντολη ‘npm install’ για την ενημερωση ή εγκατασταση ολων των εξαρτομενων αρχειων από το node_modules φακελο.

```
npm install
```

Βήμα 4^ο : Ανοίγουμε την mongo για αποθήκευση των logs.

```
C:\Users\User\Desktop\dprog\ergasiaDProgrammatismos>mongod
{"t": {"$date": "2023-06-25T15:42:32.871+03:00"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "thread1", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "outgoing": {"minWireVersion": 6, "maxWireVersion": 17}, "isInternalClient": true}}}
{"t": {"$date": "2023-06-25T15:42:32.872+03:00"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "thread1", "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t": {"$date": "2023-06-25T15:42:34.155+03:00"}, "s": "I", "c": "NETWORK", "id": 4648602, "ctx": "thread1", "msg": "Implicit TCP FastOpen in use."}
{"t": {"$date": "2023-06-25T15:42:34.157+03:00"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "thread1", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationDonorService", "namespace": "config.tenantMigrationDonors"}}
{"t": {"$date": "2023-06-25T15:42:34.157+03:00"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "thread1", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationRecipientService", "namespace": "config.tenantMigratio
```

Βήμα 5^ο : Εκτελούμε την εντολή για να ξεκινήσει το σύστημα και μεταφερόμαστε στην σελίδα ‘localhost:3000’ για χρήση της εφαρμογής.

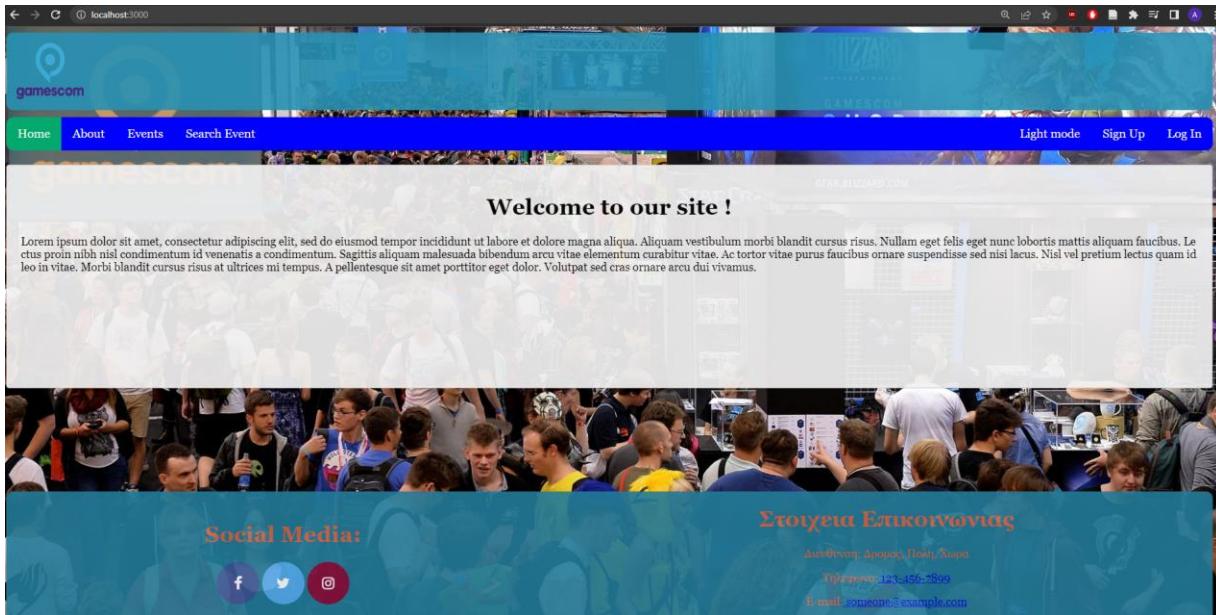
```
C:\Users\User\Desktop\dprog\ergasiaDProgrammatismos>node index.js
Server is listening on http://localhost:3000
```

Ανάλυση της εφαρμογής:

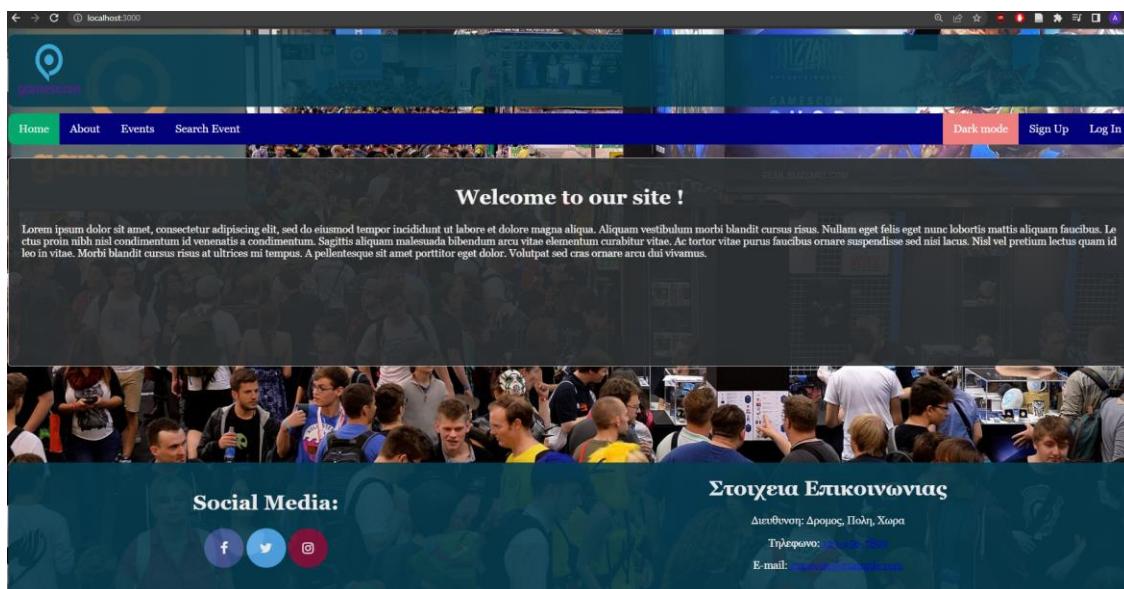
Αρχικά θα αναλύσουμε την εφαρμογή από την μεριά του χρήστη και υστέρα από την μεριά του διαχειριστή της σελίδας ώστε να παρουσιαστούν όλες οι λειτουργικότητες.

(1) Αρχική σελίδα:

Ο χρήστης μπαίνοντας στην ιστοσελίδα θα βρεθεί αρχικά στην κεντρική σελίδα:

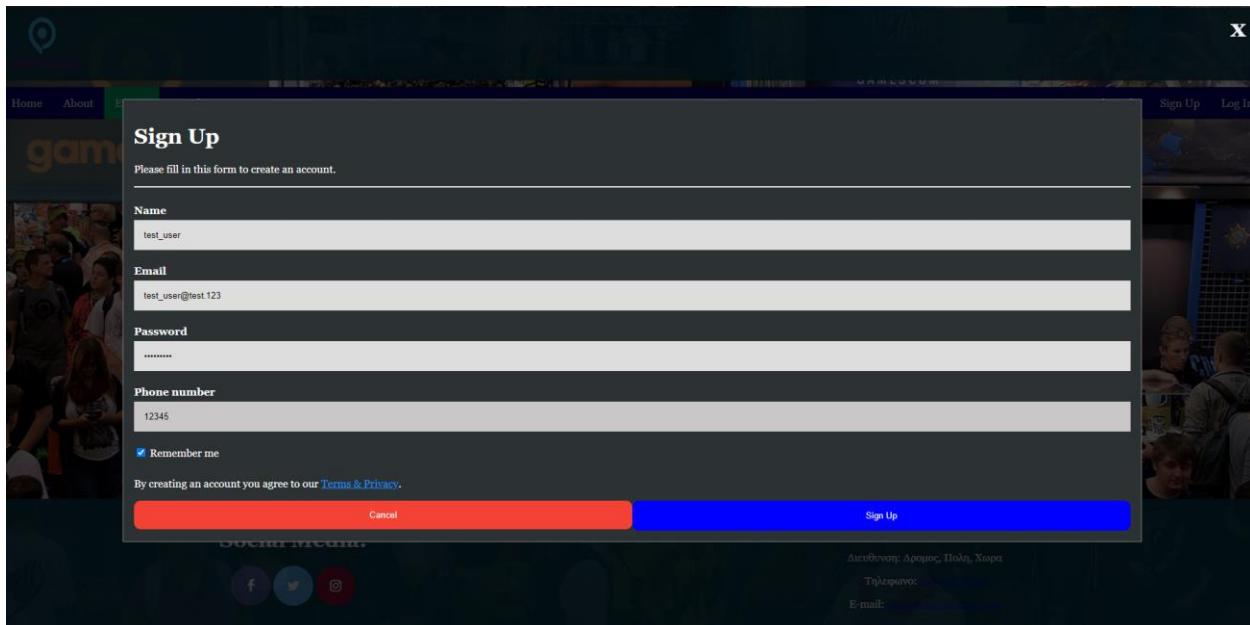


Από εκεί μπορεί να προηγηθεί αλλού πατώντας μια απτής επιλογές που βρίσκονται στο navbar, μπορεί να αλλάξει το theme των σελίδων σε dark/light mode και να κάνει sign up ή log in

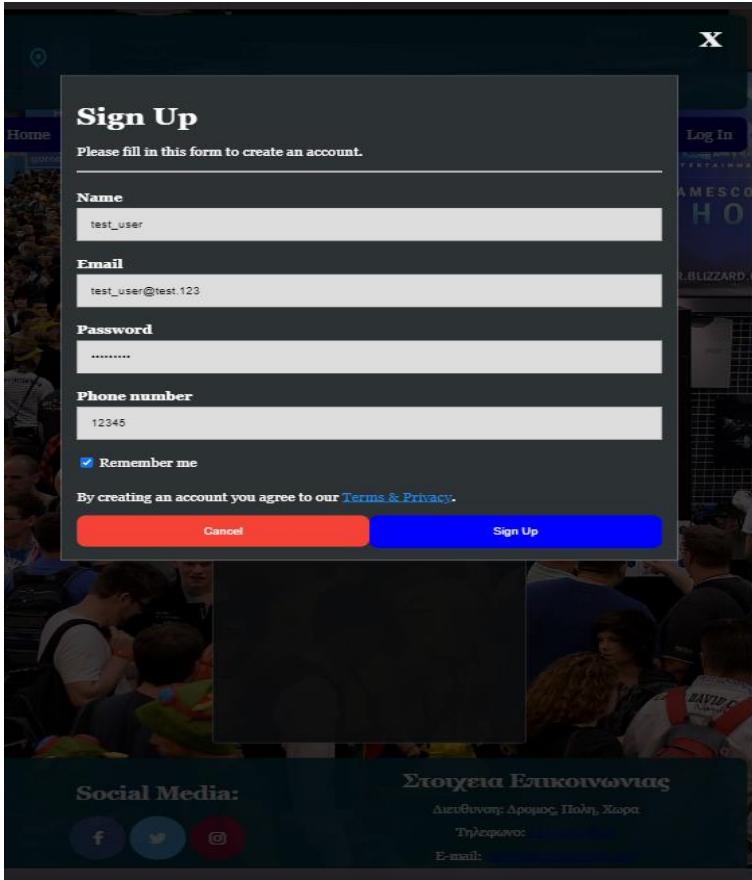


(2) Εγγραφή χρήστη:

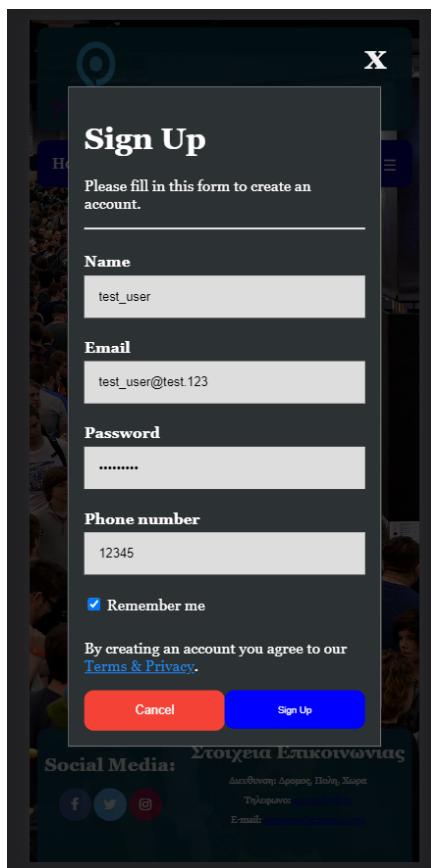
Τώρα ας δούμε την διαδικασία εγγραφής του χρήστη. Ο χρήστης μπορεί από οποιαδήποτε σελίδα να κάνει Sign up πατώντας το πάνω στο topnav bar. Πατώντας το θα εμφανιστεί ένα modal form με πεδία που πρέπει να συμπληρώσει, το "name", "email", "password" και "phone number"



Sign up στο Desktop



Tablet Sign up



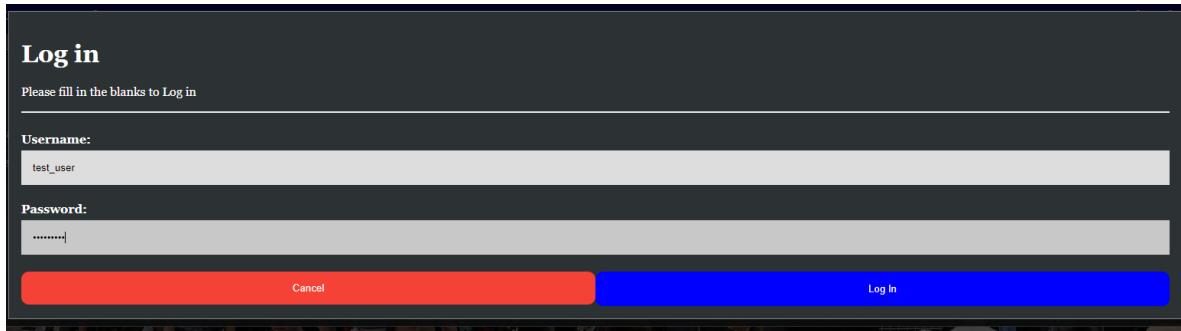
Smartphone Sign up

Πατώντας το "x" η οπουδήποτε αλλού έξω από το form θα το κάνει να κλείσει. Πατώντας το "Sign Up" στέλνονται τα inputs στην βάση δημιουργώντας ένα καινούργιο document. Αφού βάλει το όνομα που θέλει ο user του δίνεται ένα μοναδικό username με βάση το όνομα του.

```
_id: ObjectId('649829a53d0d7dd4e5e7e5e0')
username: "test_user"
name: "test_user"
email: "test_user@test.123"
password: "test_user"
phone: "12345"
```

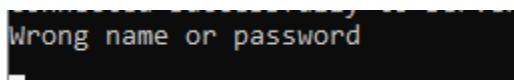
(3) Σύνδεση χρήστη:

Αφού ο χρήστης έχει κάνει Sign up τότε μπορεί και να κάνει Log in. Όταν κάνει Sign up τότε κάνει και αυτομάτως Log in αλλά σε οποιαδήποτε άλλη περίπτωση θα χρειαστεί να βάλει το username του και το password ώστε να μπορέσει να συνδεθεί ξανά.

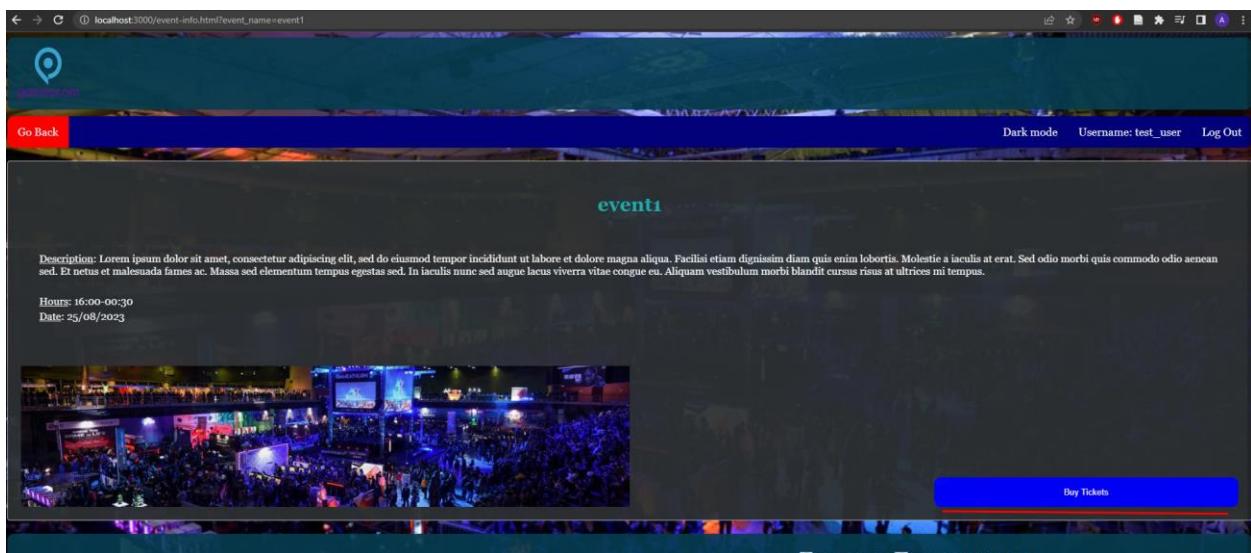


The screenshot shows a dark-themed login interface. At the top, it says "Log in" and asks for a username and password. The username field contains "test_user". The password field is redacted with dots. Below the fields are two buttons: "Cancel" on the left and "Log In" on the right, both in white text.

Σε περίπτωση που ένα από τα 2 είναι λάθος τότε δεν θα μπορεί να συνδεθεί και θα εμφανιστεί το ακόλουθο μήνυμα στο console:

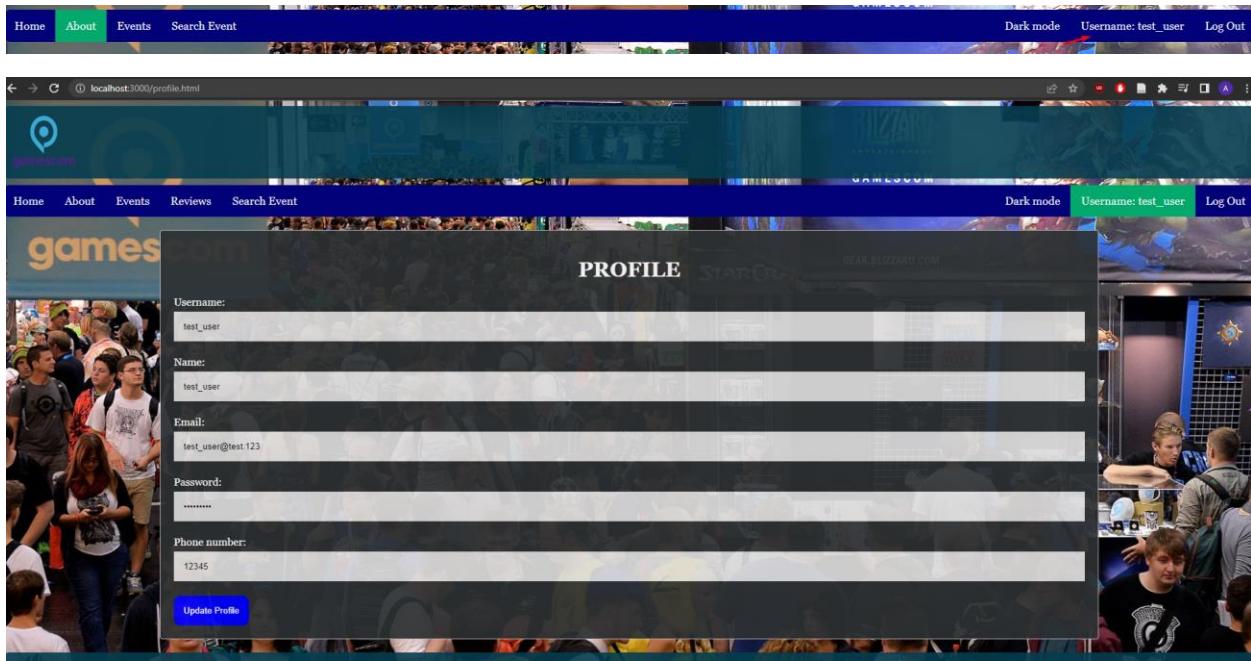


Εφόσον ο χρήστης έχει συνδεθεί κανονικά τότε θα μπορεί να κάνει ότι και πριν αλλά και να μπορεί να πατήσει στο κουμπί "Buy Tickets" που θα εμφανιστεί στην σελίδα με τις πληροφορίες για ένα συγκεκριμένο event ώστε να αγοράσει εισιτήρια για αυτό.

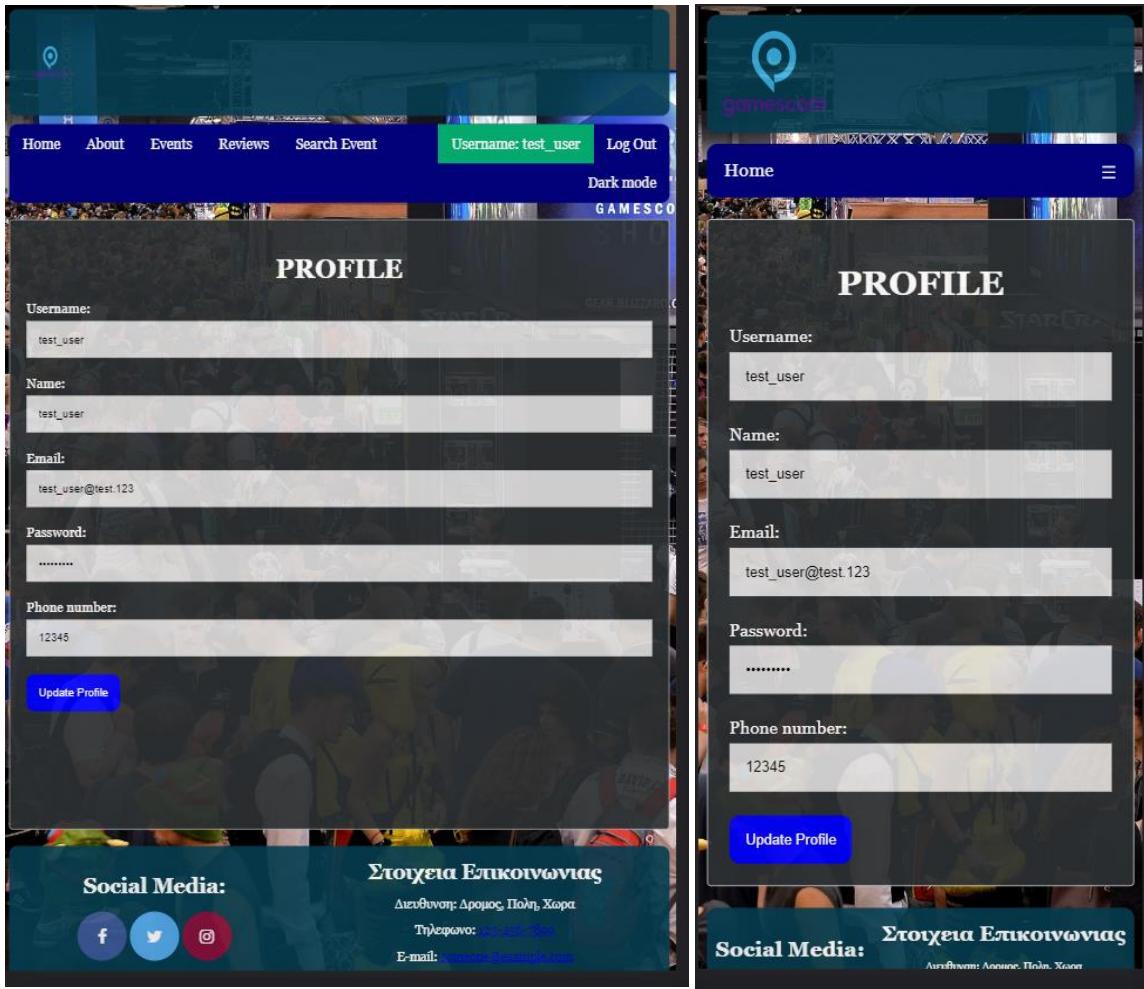


(4) Προφίλ Χρήστη:

Προφίλ Χρήστη: Εφόσον ο χρήστης έχει κάνει κανονικά log in τότε θα εμφανίζεται και το username στο topnav bar. Εκεί θα μπορεί να πατήσει πάνω του και να πάει στο προφίλ του.



Desktop



Tablet

Smartphone

Εκεί θα του εμφανίζονται όλες οι πληροφορίες για αυτόν μέσα σε ένα form οπού μπορεί να τις επεξεργαστεί ολκές εκτός από το "username". Για παράδειγμα αλλάζοντας το "email" από "test_user@test.123" σε "123@123.com" μπορούμε να δούμε την αλλαγή και στην βάση δεδομένων:

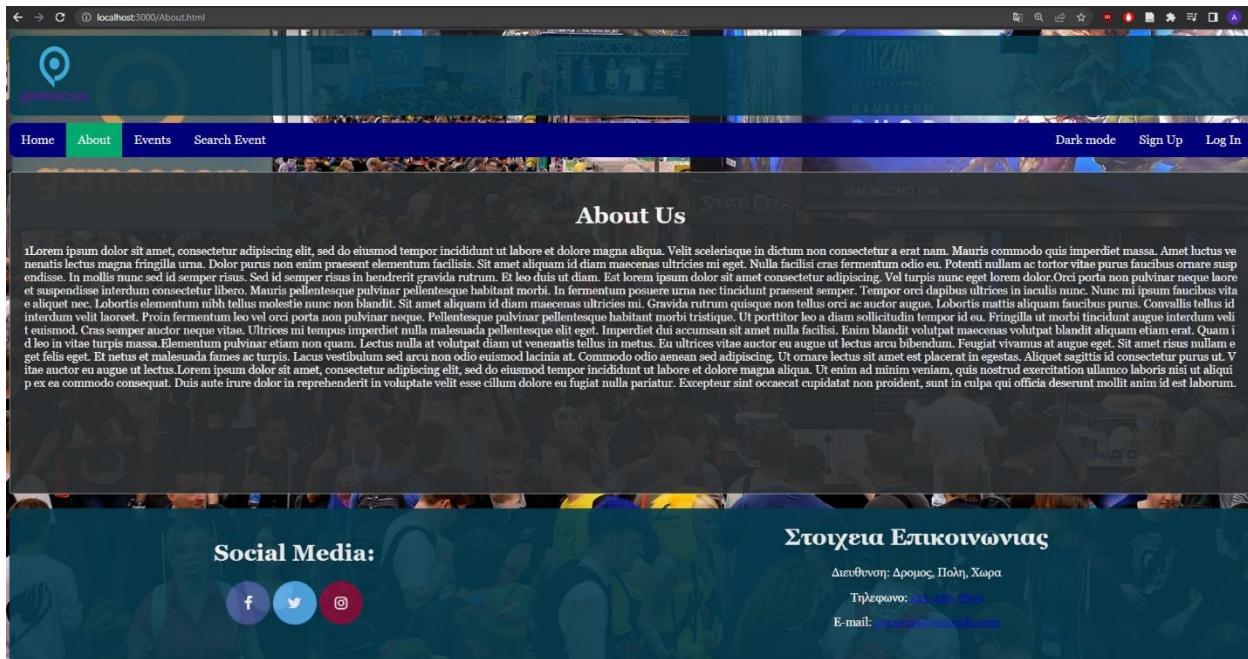


```
_id: ObjectId('649829a53d0d7dd4e5e7e5e0')
username: "test_user"
name: "test_user"
email: "123@123.com"
password: "test_user"
phone: "12345"
```

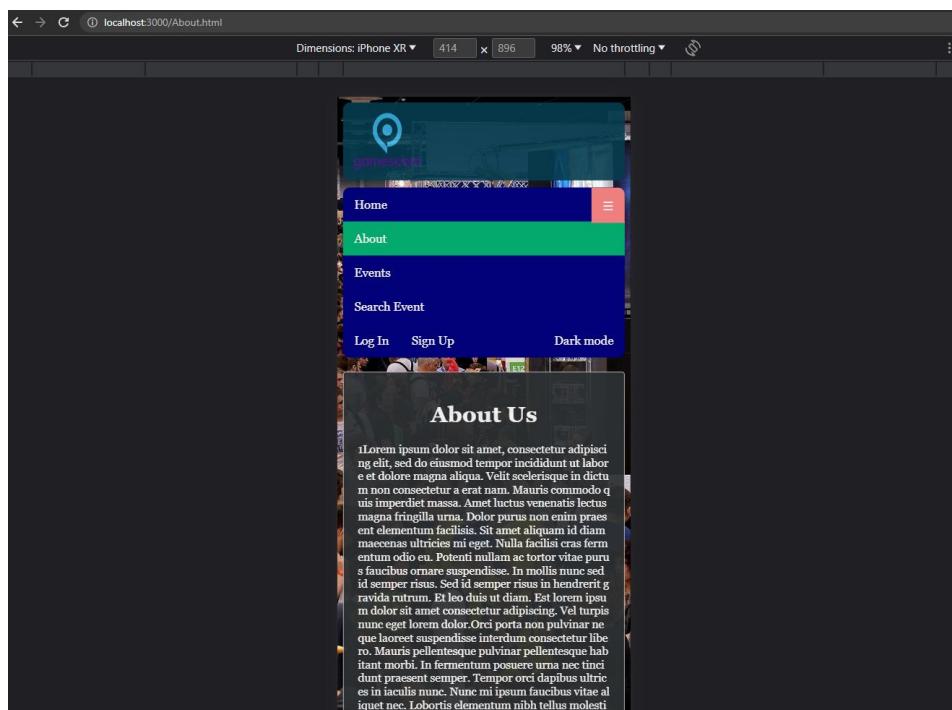
(5) Σελίδα Πληροφοριών:

Ας υποθέσουμε ότι ο χρήστης θα προηγηθεί στις σελίδες με την σειρά που εμφανίζονται.

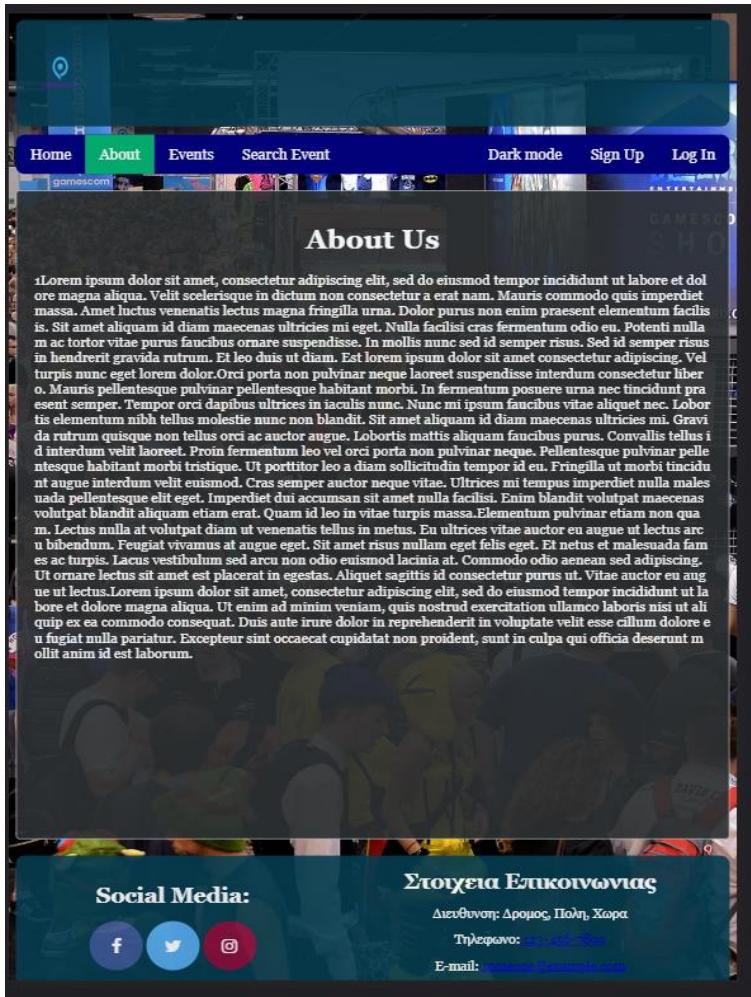
Αρχικά θα πατήσει το "About" στο navbar οπού θα τον ανακατευθύνει στην σελίδα με τις γενικές πληροφορίες της καμπάνιας



Desktop



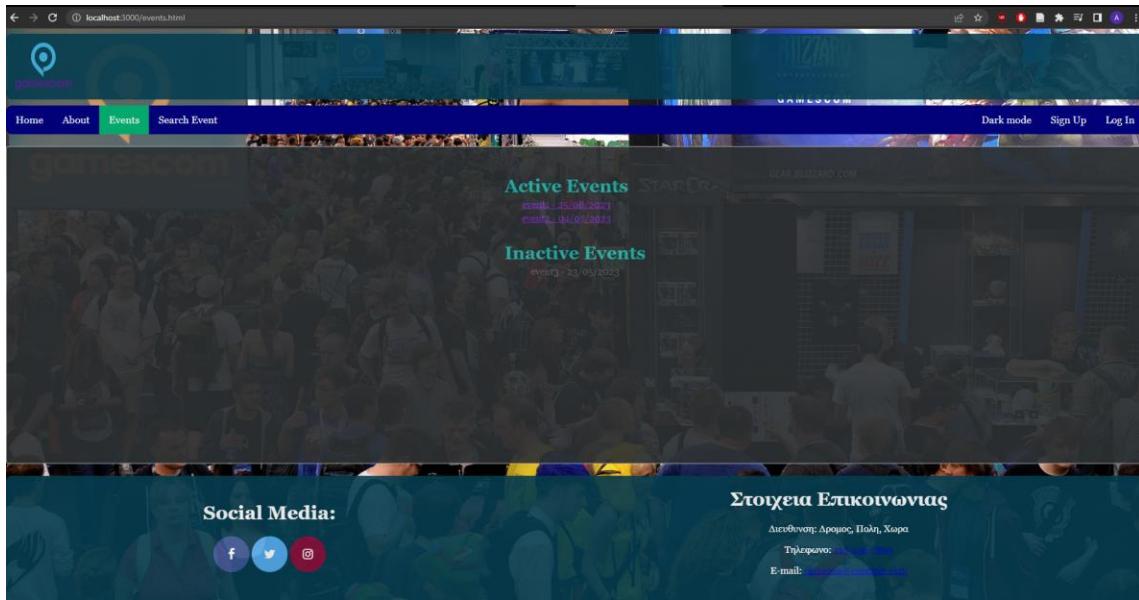
Smartphone



Tablet

(6) Σελίδα Εκδηλώσεων :

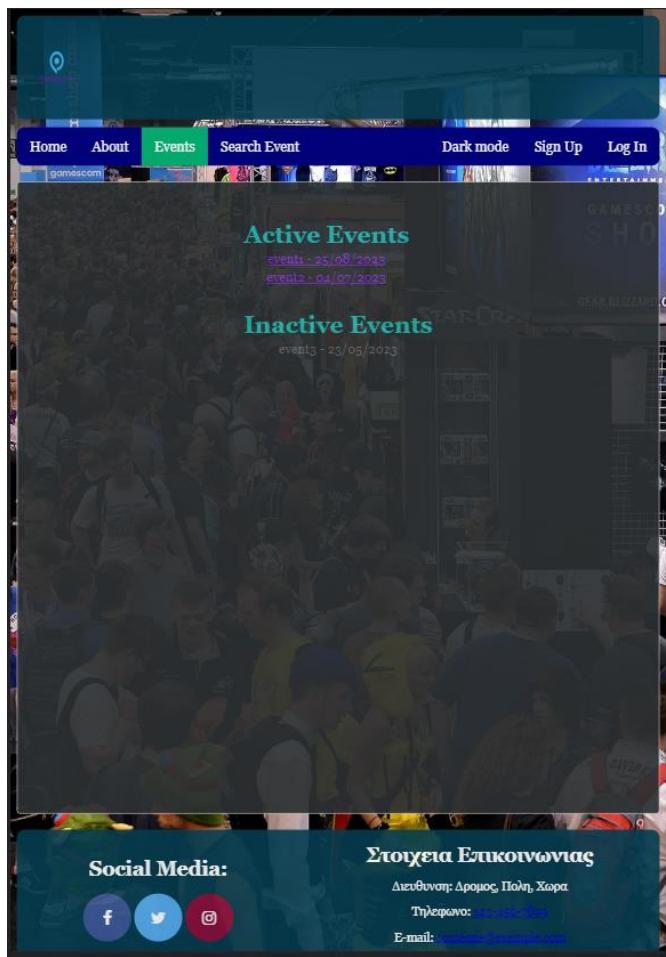
Η επόμενη σελίδα είναι η "Events". Σε εκείνη την σελίδα ο χρήστης μπορεί να δει όλα τα events που έχουν διοργανωθεί από την καμπάνια, μελλοντικά αλλά και events που έχουν περάσει.



Desktop

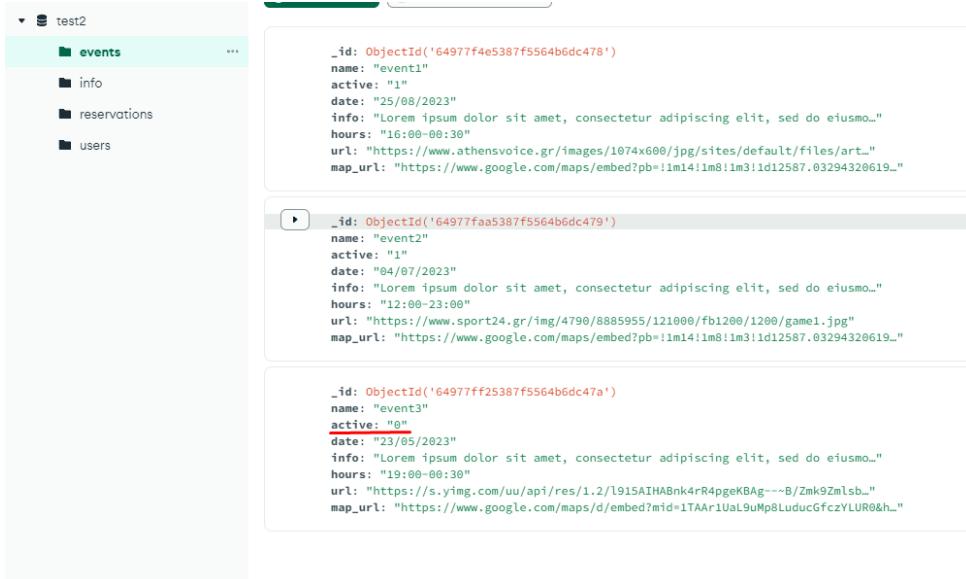


Smartphone



Tablet

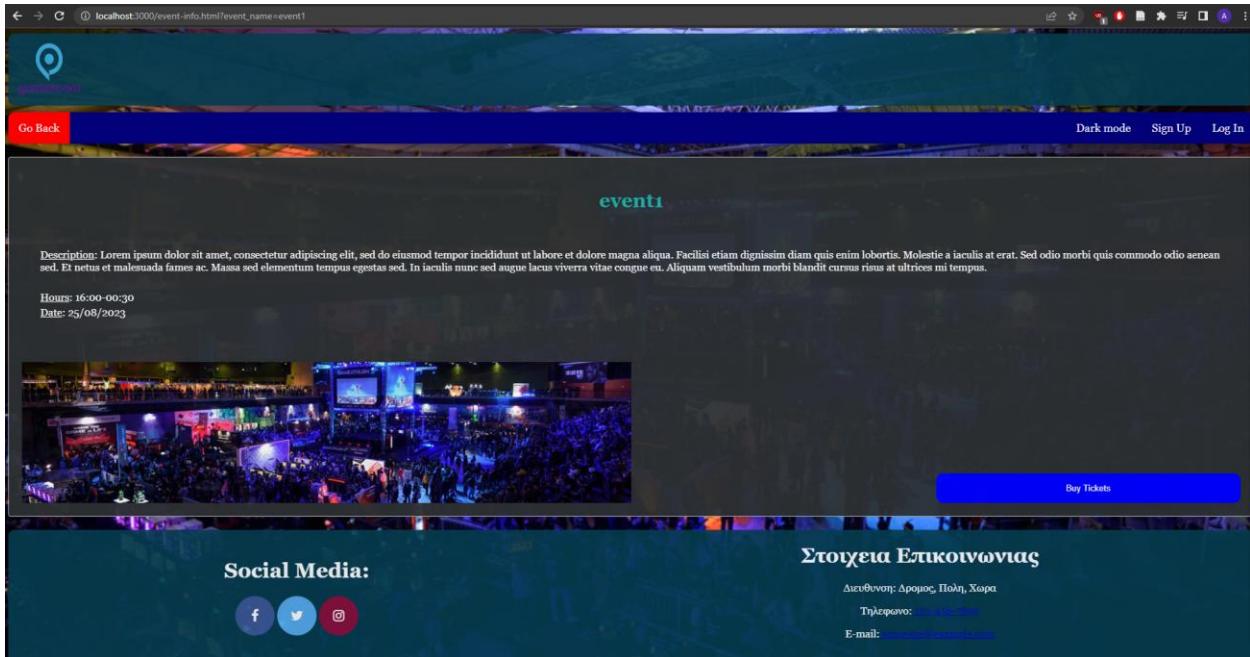
Σαν παραδείγματα έχουμε 3 documents που αφορούν τα events στην βάση δεδομένων μας με το "event3" να μην είναι active ποια. Παρακάτω φαίνονται τα events στο collection της βάσης δεδομένων



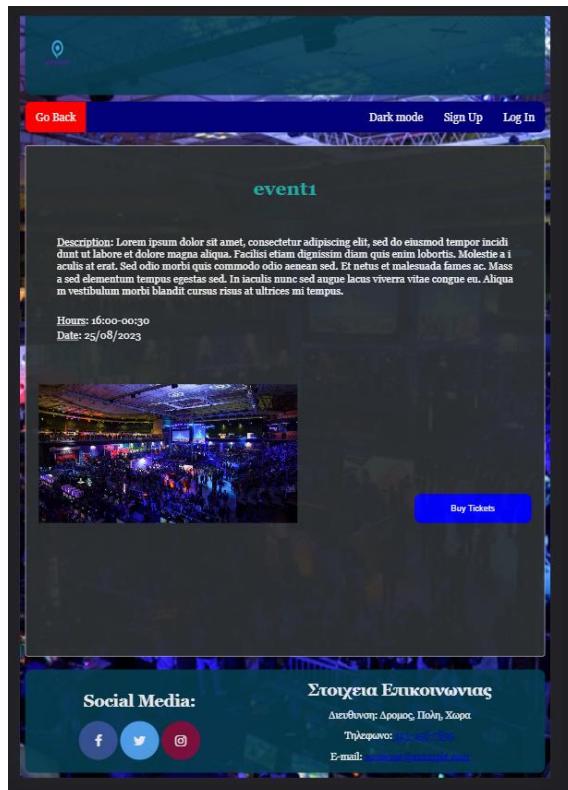
The screenshot shows the MongoDB Compass interface with a database named 'test2'. The 'events' collection is selected. Three documents are listed:

- Document 1:** _id: ObjectId('64977f4e5387f5564b6dc478'), name: "event1", active: "1", date: "25/08/2023", info: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod...", hours: "16:00-00:30", url: "https://www.athensvoice.gr/images/1074x600/jpg/sites/default/files/art...", map_url: "https://www.google.com/maps/embed?pb=!m14!1m8!1m3!1d12587.03294320619..."
- Document 2:** _id: ObjectId('64977faa5387f5564b6dc479'), name: "event2", active: "1", date: "04/07/2023", info: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod...", hours: "12:00-23:00", url: "https://www.sport24.gr/img/4790/8885955/121000/fb1200/1200/gamel.jpg", map_url: "https://www.google.com/maps/embed?pb=!m14!1m8!1m3!1d12587.03294320619..."
- Document 3:** _id: ObjectId('64977ff25387f5564b6dc47a'), name: "event3", active: "0", date: "23/05/2023", info: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod...", hours: "19:00-00:30", url: "https://s.yimg.com/uu/api/res/1.2/l915AIHABnk4rR4pgeKBAg--~B/Zmk9Zmlsb...", map_url: "https://www.google.com/maps/d/embed?mid=1TAAr1UaL9uMp8LuducGfczYLUR0&h..."

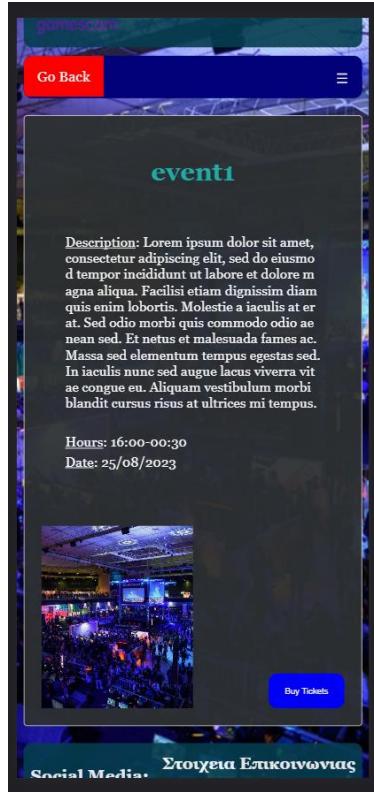
Στην συγκεκριμένη σελίδα ο χρήστης μπορεί να πατήσει πάνω σε ένα event που θα τον ανακατευθύνει στην σελίδα με τις πληροφορίες για το συγκεκριμένο event. Πατώντας δηλαδή πάνω στο "event1" θα πάει στην σελίδα "/event-info.html?event_name=event1" που περνάει το όνομα του event που πάτησε ως παράμετρος για να βρεθούν όλες οι πληροφορίες του από την βάση.



Desktop



Tablet



Smartphone

Επίσης κάνοντας hover πάνω απτή εικόνα του event ο χρήστης μπορεί να δει τον χάρτη της τοποθεσίας που βρίσκεται το event με μια κόκκινη πινέζα να δείχνει ακριβώς που είναι

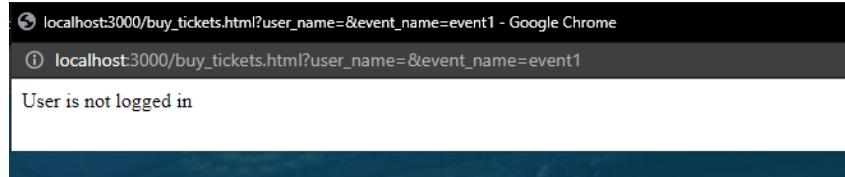
Social Media:

-
-
-

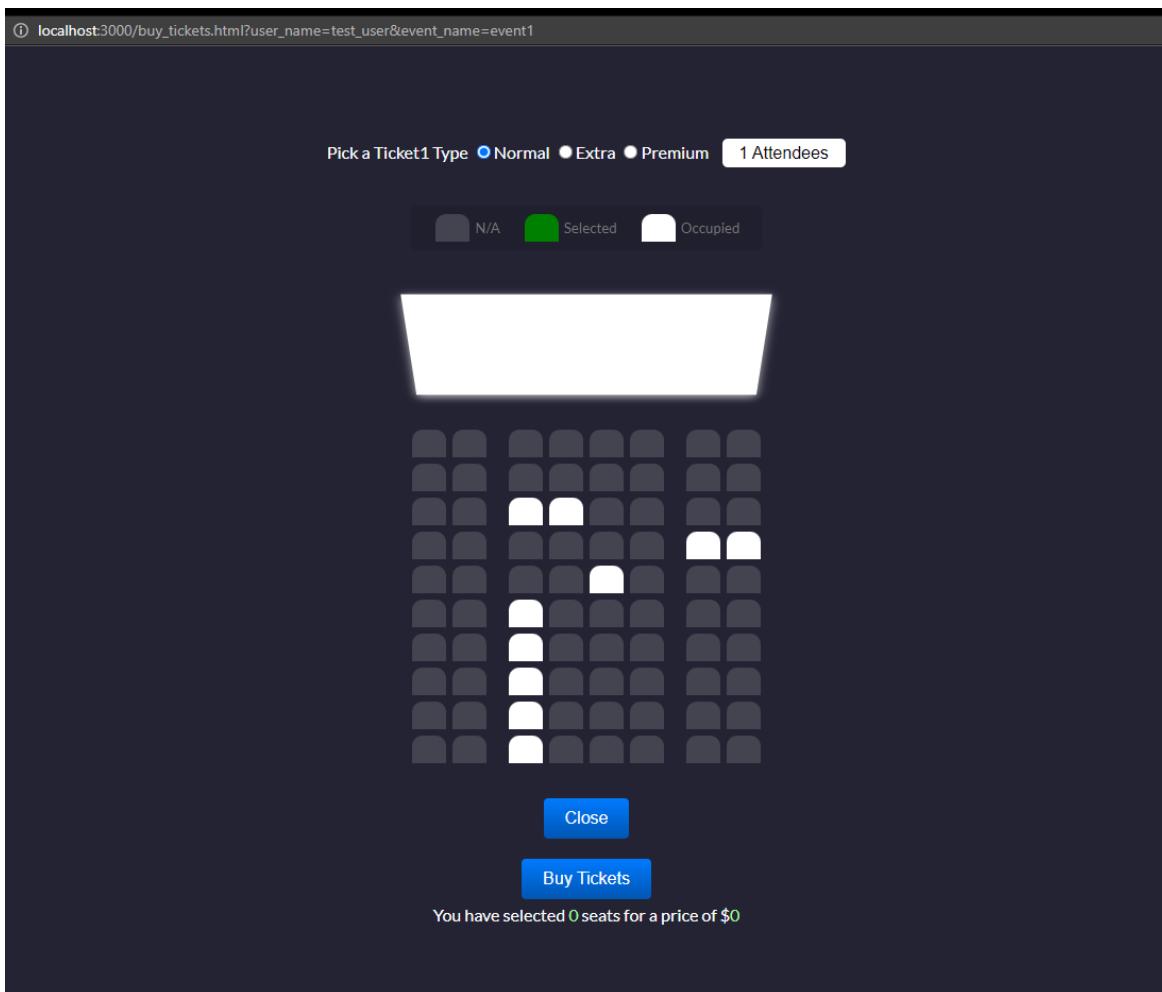
Στοιχεία Επικοινωνίας

Διεύθυνση: Δρόμος, Πόλη, Χώρα
Τηλέφωνο: +30-123-4567
E-mail: info@epikoinonia.com

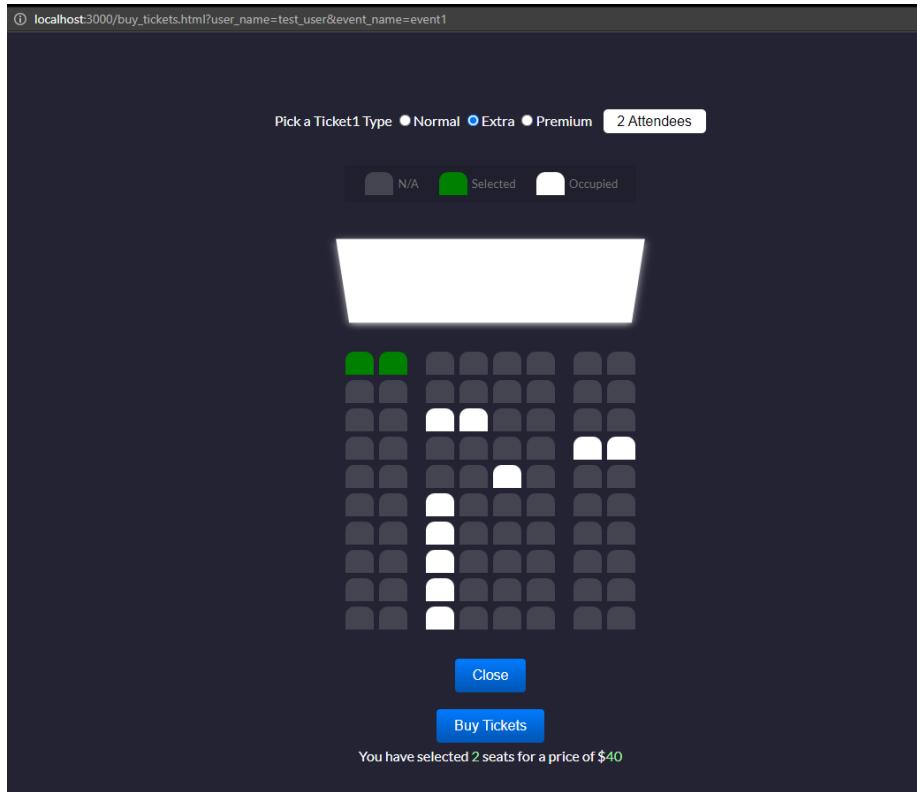
Τέλος υπάρχει και το κουμπί “Buy Tickets” οπού όταν πατηθεί θα ανακατευθύνει τον χρήστη στην σελίδα οπού αγοράζει εισιτήρια για το συγκεκριμένο event, αν είναι ενεργό. Ωστόσο, πρέπει να έχει κάνει πρώτα Log in, επομένως στην περίπτωση που δεν είναι θα του εμφανιστεί το συγκεκριμένο μνήμα:



Αν ο χρήστης έχει κάνει log in και πατήσει στο κουμπί για να αγοράσει εισιτήρια τότε θα ανοίξει ένα καινούργιο window οπού εκεί θα μπορεί να επιλέξει τον τύπο του εισιτήριου, αυτό θα επηρεάζει την τιμή του, τον αριθμό των ατόμων και τέλος πατώντας πάνω στις θέσεις μπορεί να επιλέγει τις θέσεις που θέλει.



Για παράδειγμα, ο test_user επιλέγοντας τον τύπο εισιτήριου "extra" που κοστίζει \$20, επιλέγοντας 2 "Attendees" και επιλέγοντας τις πρώτες 2 θέσεις.

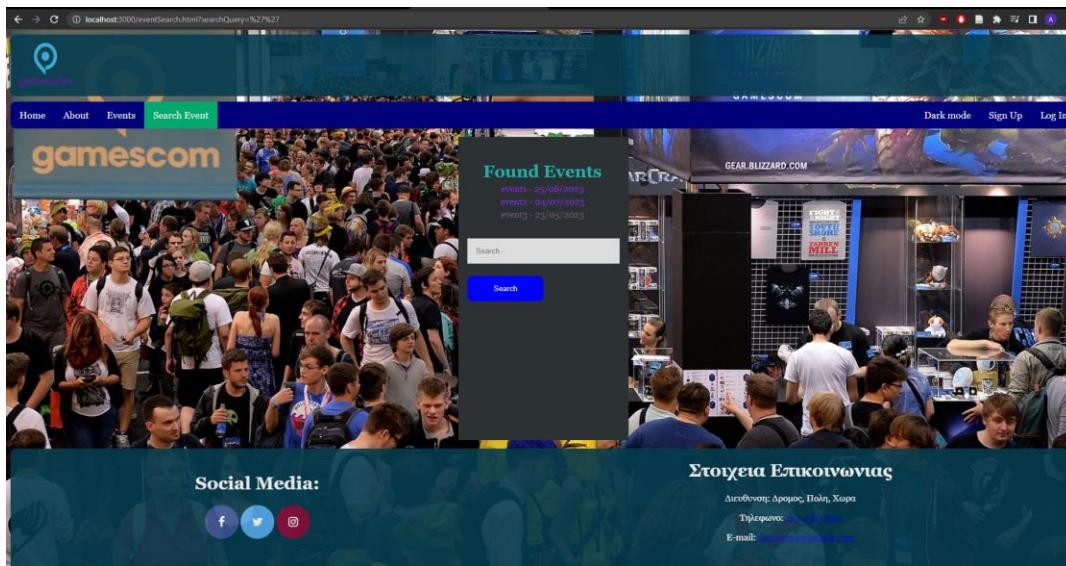


Έτσι θα δημιουργηθεί ένα reservation με τα εξής στοιχεία:

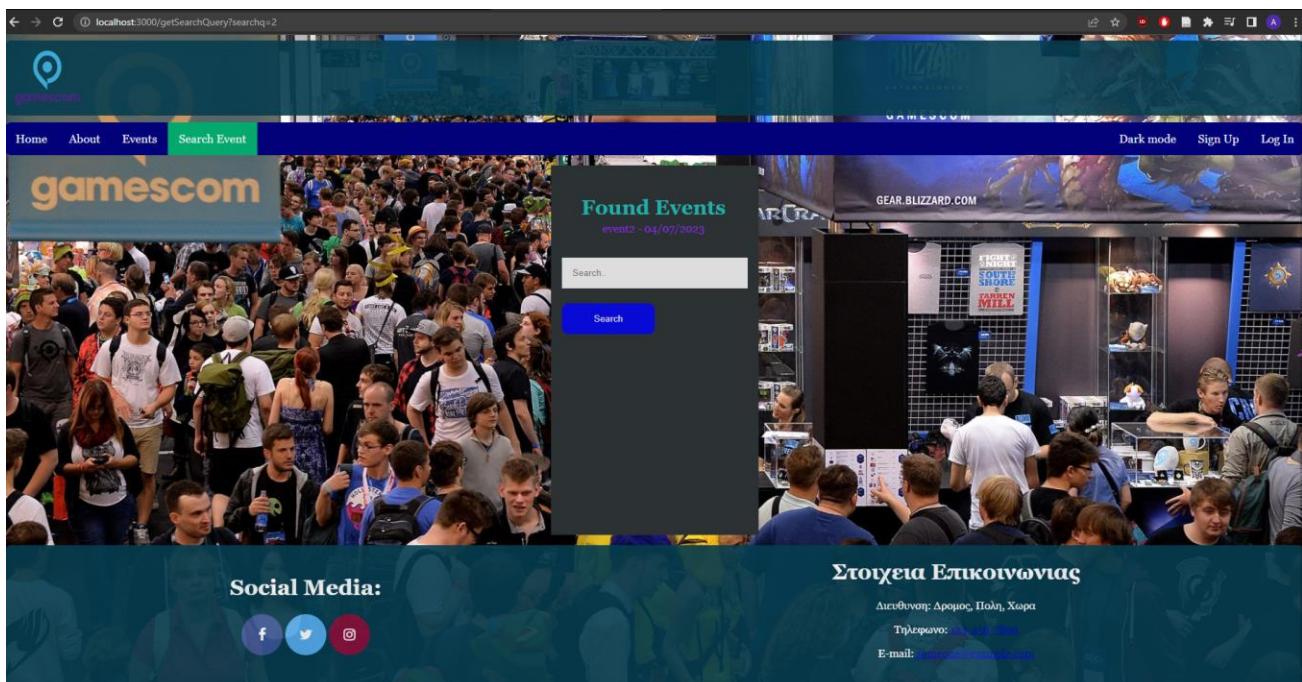
```
_id: ObjectId('6498322d3d0d7dd4e5e7e5e1')
id: "8dedac78-7629-4a47-8cad-2966575418dc"
seats: Array
  0: 0
  1: 1
username: "test_user"
eventname: "event1"
total_price: 40
ticket_amount: 2
ticket_type: "extra"
```

(8) Αναζήτηση Εκδηλώσεων:

Στην συνέχεια ο χρήστης μπορεί να πάει στην σελίδα “Search Events”. Εκεί εμφανίζονται όλα τα events



Εκεί ο χρήστης μπορεί να δει τα events που είναι ενεργά με ένα πιο ανοιχτό χρώμα και πατώντας πάνω σε ένα event θα ανακατευθύνεται στην σελίδα με τις πληροφορίες για το event. Τέλος υπάρχει μια search bar οπού μπορεί να γράψει ένα κομμάτι το ονόματος του event που θέλει να βρει και θα του εμφανιστεί. Για παράδειγμα, γράφοντας τον αριθμό 2 θα του εμφανιστεί το event2:



Αυτό συμβαίνει καθώς παίρνει το input του χρήστη ως παράμετρο και ψάχνει στο collection με τα events τα ονόματα που του ταιριάζουν, παίρνοντας τα σε μια μεταβλητή και κάνοντας ξανά render την σελίδα.

Μέρος Διαχειριστή:

(10) + (11) Αρχική Σελίδα Διαχείρισης & Σύνδεση Διαχειριστή:

Για να συνδεθεί ο διαχειριστής θα πρέπει να κάνει log in με τα στοιχεία που του έχουν δοθεί. Πιο συγκεκριμένα αυτά είναι:

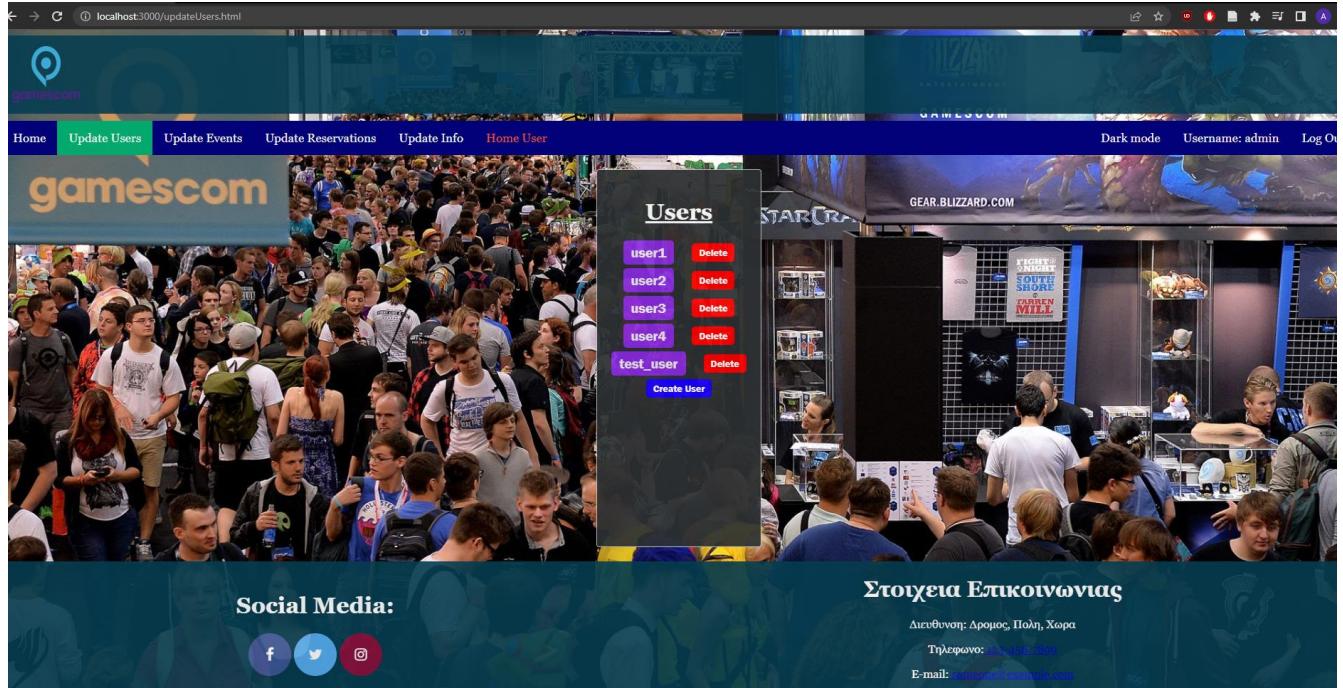
```
_id: ObjectId('6493485b207bab2c8c08a2ec')
username: "admin"
name: "admin"
email: "admin@admin.com"
password: "admin123"
phone: "12345"
```

Αφού συνδεθεί τότε θα κατευθυνθεί στην αρχική σελίδα διαχείρισης



(12) Σελίδα επεξεργασίας χρηστών:

Όταν ο admin πατήσει πάνω στο “Update Users” τότε θα κατευθυνθεί στην σελίδα που θα εμφανίζονται όλοι οι users



Πατώντας του κουμπί “delete” διαγράφεται ο συγκεκριμένος user. Αν διαγραφθεί ένας user Που έχει κάνει κάποια κράτηση τότε διαγράφεται και αυτή . Για παράδειγμα αν πάμε να διαγράψουμε τον χρήστη “test_user” τότε:

Δεν υπάρχει ποια ο ‘test_user’ στο collection “users”.

```
_id: ObjectId('6493485b207bab2c8c88a2ec')
username: "admin"
name: "admin"
email: "admin@admin.com"
password: "admin123"
phone: "12345"

_id: ObjectId('6495839f27d095065ed96f53')
username: "user2"
name: "user2"
email: "user2@test"
password: "user2"
phone: "12345"

_id: ObjectId('64977ea95387f5564b0ddc476')
username: "user3"
name: "user3"
email: "user3@test"
password: "user3"
phone: "12345"

_id: ObjectId('64977ea95387f5564b0ddc477')
username: "user4"
name: "user4"
email: "user4@test"
password: "user4"
phone: "12345"

_id: ObjectId('64983d9f3dbd7dd4e5e7e5e2')
username: "made_user"
name: "made_user"
email: "made_user@23.com"
password: "made_user"
phone: "12345"
```

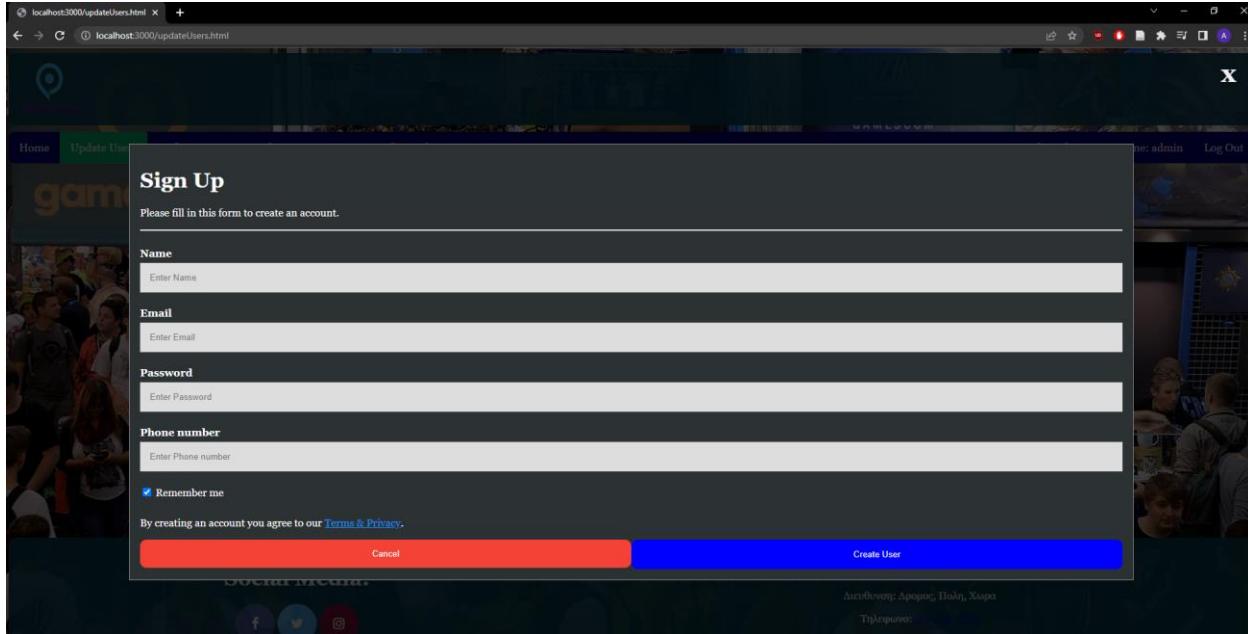


Αλλά δεν υπάρχει ούτε το reservation που είχε δημιουργήσει σε προηγούμενο παράδειγμα:

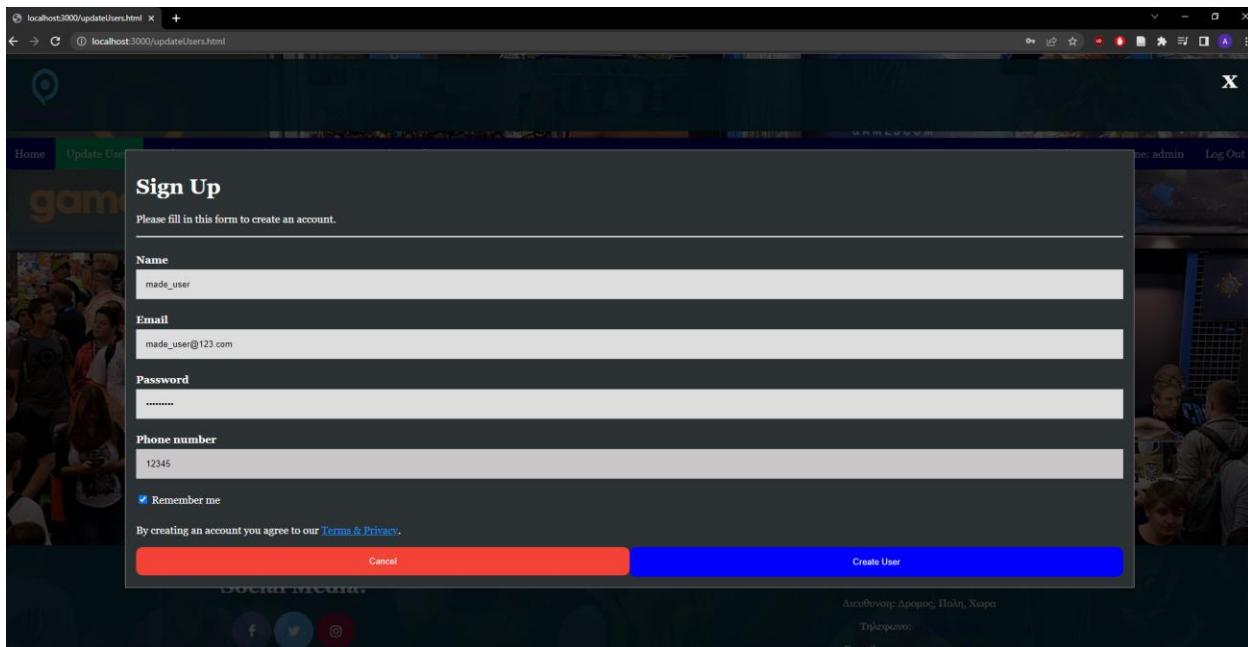
```
_id: ObjectId("6497803d5387f5564b6dc47c")
id: "e4d7eb0b-8d00-4c69-991f-96a64b42713d"
seats: Array
username: "user2"
eventname: "event1"
total_price: 120
ticket_amount: 4
ticket_type: "premium"

_id: ObjectId("6497804e5387f5564b6dc47d")
id: "e88a2e4d-cbb0-4891-be8d-db447c461faf"
seats: Array
username: "user3"
eventname: "event2"
total_price: 30
ticket_amount: 1
```

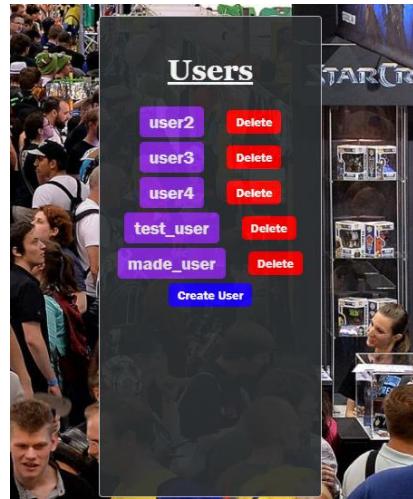
Πατώντας το κουμπί “Create User” εμφανίζεται ένα modal form οπού ο admin καλείται να συμπληρώσει τα στοιχεία του καινούργιου user που θέλει να δημιουργήσει:



Για παράδειγμα, αν ο admin πάει να δημιουργήσει έναν χρήστη με το όνομα “made_user”:



Τον δημιουργεί και μετά ανακατευθύνεται στην σελίδα "Update Users" οπού ανανεώνονται οι users.



Πατώντας πάνω σε έναν user τότε ανακατευθύνεται στην σελίδα επεξεργασίας του.

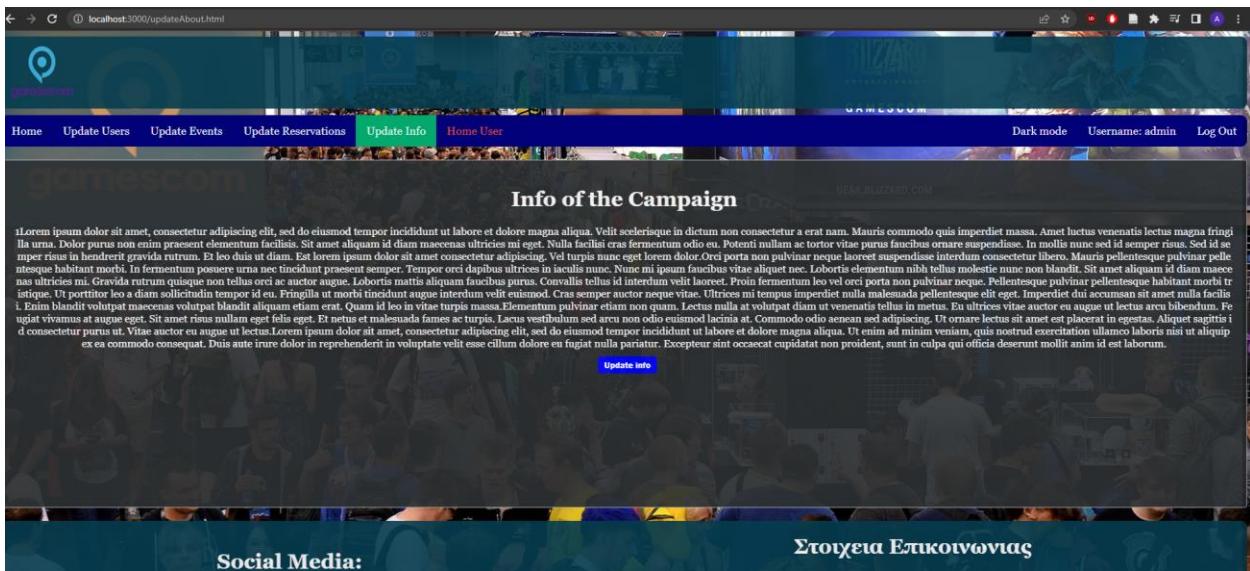
A screenshot of a web browser displaying the "Update User Profile" page for a user named "test_user". The page has a dark theme with a blue header. The main content area is titled "PROFILE" and contains five input fields: "Username" (test_user), "Full Name" (test_user), "Email" (123@123.com), "Password" (*****), and "Phone number" (12345). At the bottom of the form is a blue "Update Profile" button. The background of the browser window shows a blurred image of a gaming convention with people and booths.

Εκεί υπάρχει ένα form οπού δείχνονται τα στοιχεία του user όπως είναι στην βάση δεδομένων. Πατώντας πάνω σε ένα input field μπορεί να αλλάξει πληροφορίες και πατώντας το κουμπί “Update User” αποθηκεύει τις αλλαγές στην βάση. Για παράδειγμα για τον χρήστη “test_user” αλλάζοντας το “phone number” από “12345” σε “123” έχουμε:



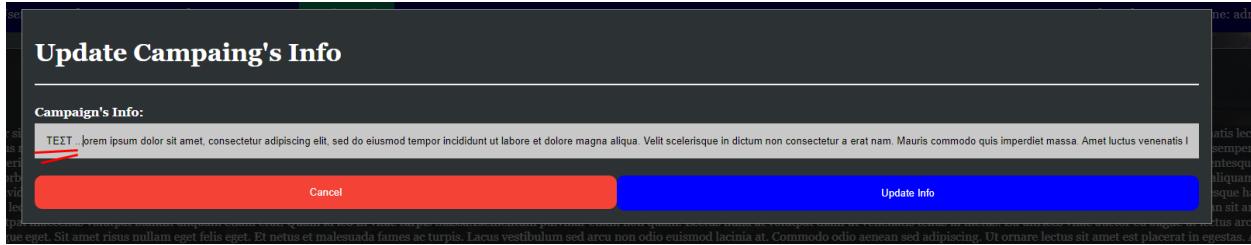
```
_id: ObjectId('649829a53d0d7dd4e5e7e5e0')
username: "test_user"
name: "test_user"
email: "123@123.com"
password: "test_user"
phone: "123"
```

(13) Σελίδα Επεξεργασίας Πληροφοριών:



Πηγαίνοντας στην σελίδα αυτήν μπορεί να δει ο admin τις γενικές πληροφορίες της καμπάνιας που εμφανίζονται και στην "About" σελίδα. Πατώντας πάνω στο κουμπί "Update Info"

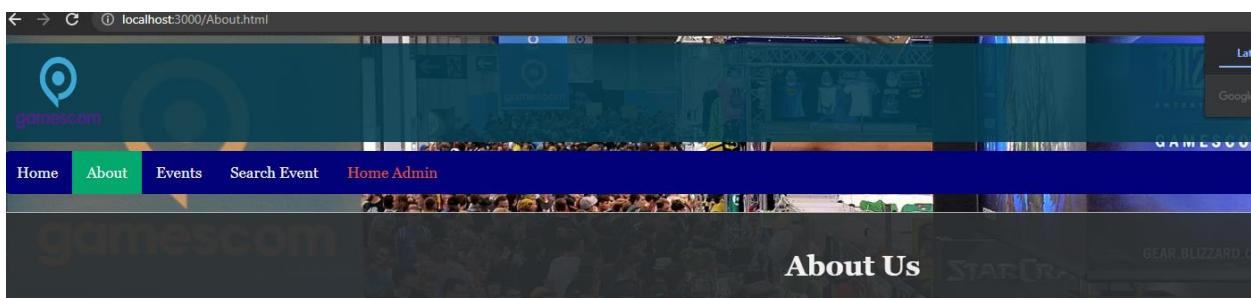
εμφανίζεται μια modal form με το text μέσα στο input field. Εκεί μπορεί να κάνει αλλαγές και στη συνέχεια να το αποθηκεύσει.



Info of the Campaign

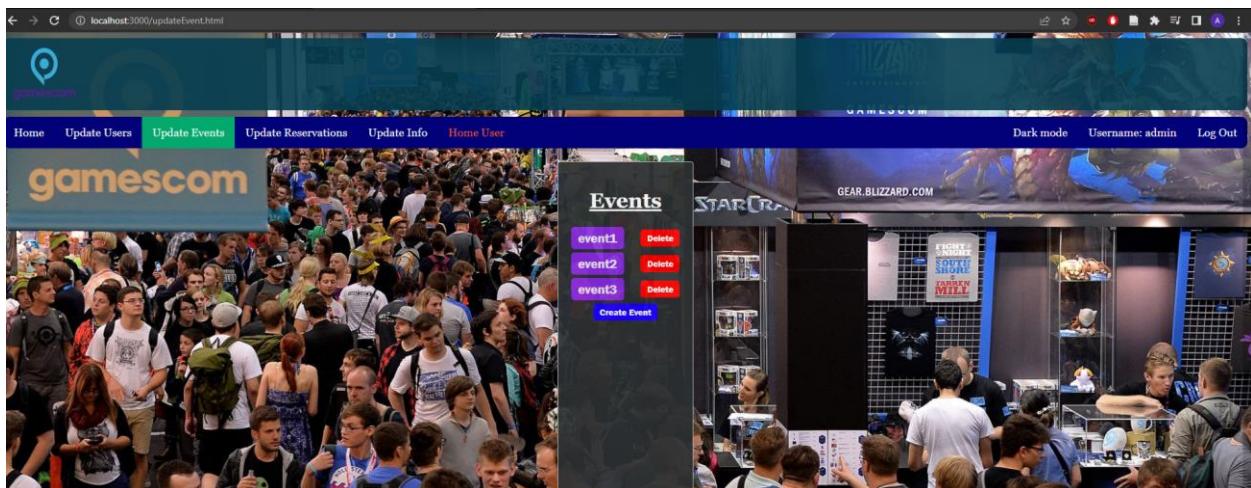
ΤΕΣΤ... lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Και αλλάζει και στην σελίδα "About"



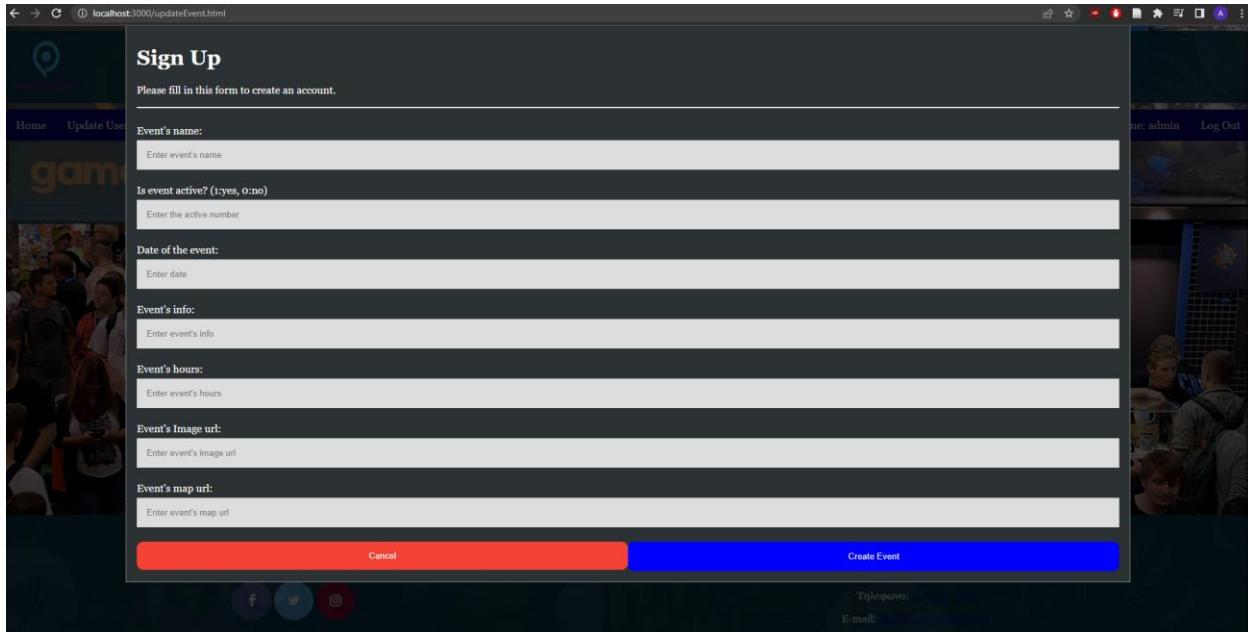
(14) Σελίδα επεξεργασίας Εκδηλώσεων:

Εδώ, με παρόμοιο τρόπο όπως στην "Update Users", εμφανίζονται όλα τα events

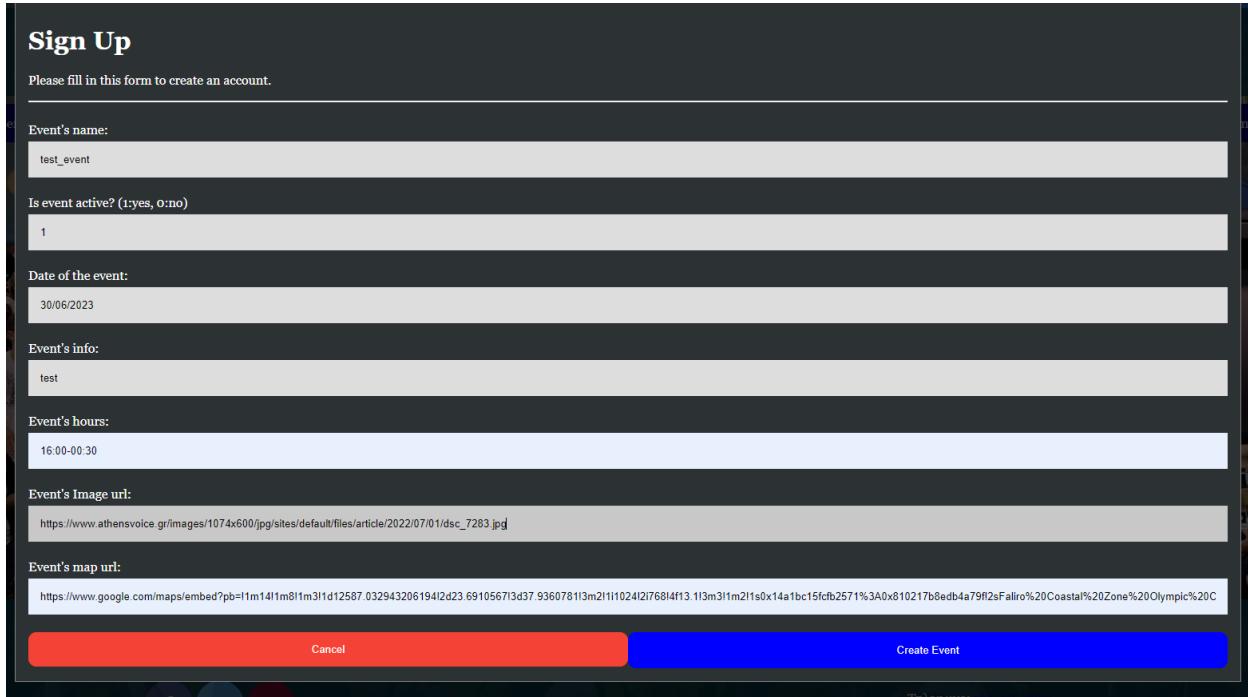


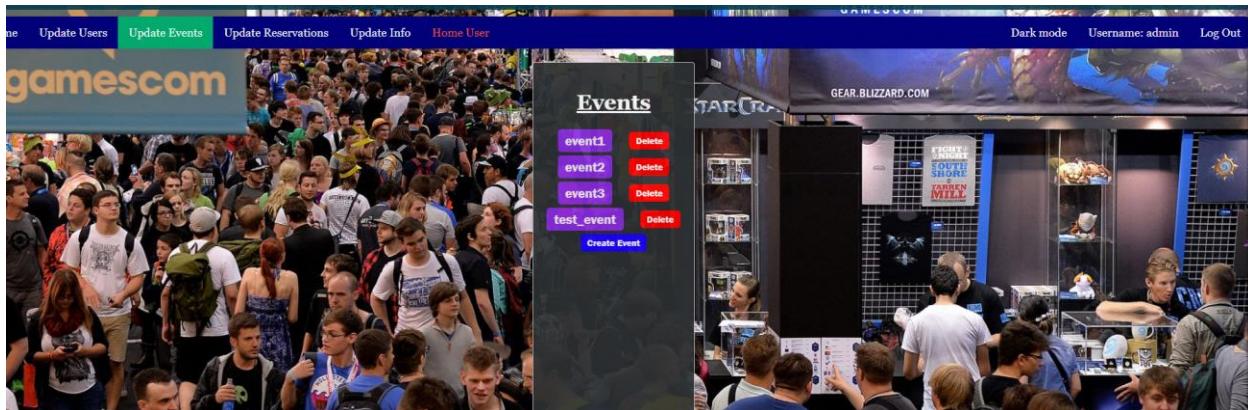
Πατώντας το κουμπί "Delete" διαγράφεται ένα event. Αν διαγράφει κάποιο event για το οποίο έχουν γίνει κρατήσεις τότε διαγράφονται και αυτές

Τέλος πατώντας πάνω στο κουμπί “Create Event” εμφανίζεται ένα modal form με τα απαραίτητα fields για να δημιουργήσει ένα καινούργιο event



Για παράδειγμα: (χρησιμοποιούνται URLs που υπάρχουν ήδη στην βάση)

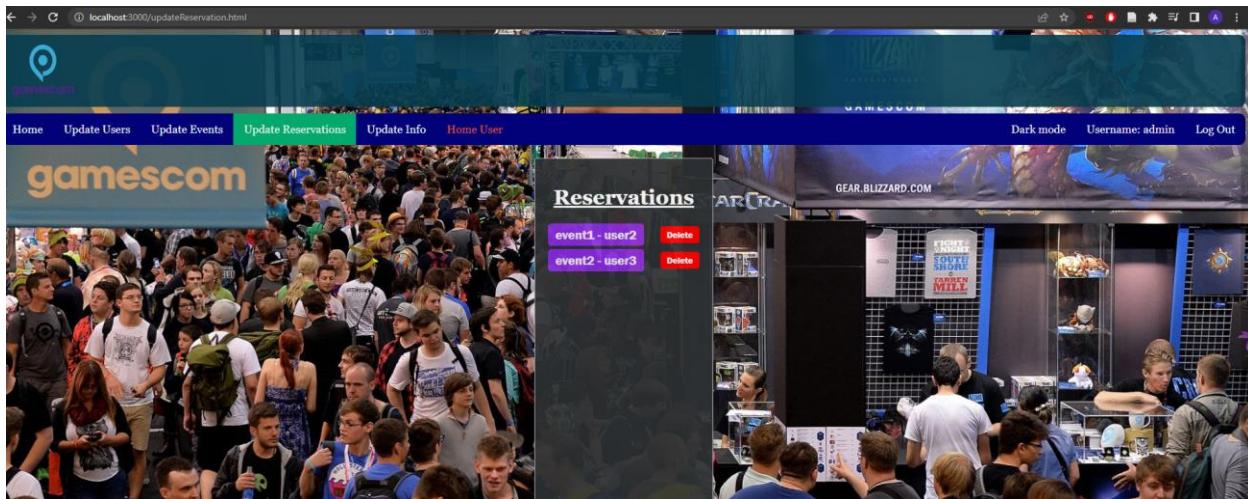




```

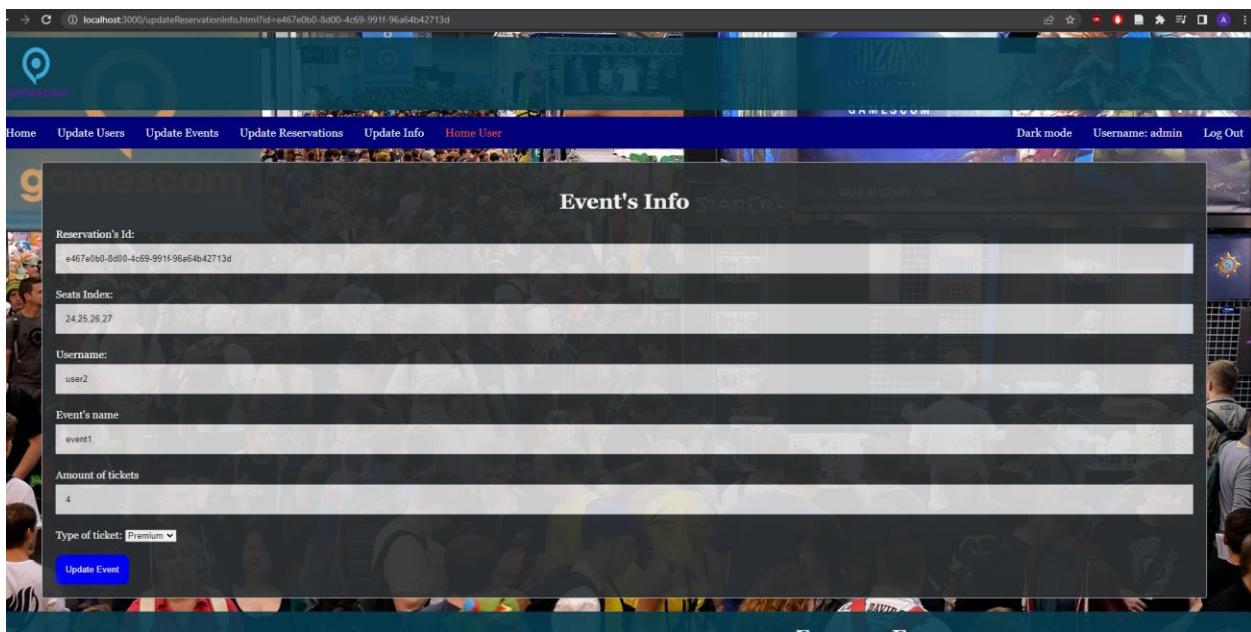
_id: ObjectId('649841433d0d7dd4e5e7e5e3')
name: "test_event"
active: "1"
date: "30/06/2023"
info: "test"
hours: "16:00-00:30"
url: "https://www.athensvoice.gr/images/1074x600/jpg/sites/default/files/art..."
map_url: "https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!1d12587.03294320619..."
```

(14) Σελίδα Επεξεργασίας Κρατήσεων:



Στην συγκεκριμένη σελίδα ο admin μπορεί να δει όλες τις κρατήσεις που υπάρχουν, ποιος χρήστης την έχει κάνει και για ποιο event

(Σημείωση: ο admin δεν μπορεί να δημιουργήσει κρατήσεις, μόνο να τις διαγράψει και αλλάξει τον τύπο των εισιτήριων)



Πατώντας πάνω σε μια κράτηση μπορεί να δει τα στοιχεία της κράτησης οπού μπορεί να αλλάξει το είδος του εισιτήριου με αποτέλεσμα να αλλάζει και η τελική τιμή για τον χρήστη.

Για παράδειγμα, για την 1^η κράτηση "event1 – user2", αλλάζοντας τον τύπο του εισιτήριου από "premium" σε "normal" θα πρέπει να αλλάξει η τιμή από \$120 σε \$40, δεδομένου ότι έχει γίνει κράτηση για 4 θέσεις, συγκεκριμένα τις 24,24,26 και 27

Οπότε έχουμε:

```
_id: ObjectId('6497803d5387f5564b6dc47c')
id: "e467e0b0-8d00-4c69-991f-96a64b42713d"
seats: Array
username: "user2"
eventname: "event1"
total_price: 120
ticket_amount: 4
ticket_type: "premium"
```

Σε:

```
_id: ObjectId('6497803d5387f5564b6dc47c')
id: "e467e0b0-8d00-4c69-991f-96a64b42713d"
seats: Array
username: "user2"
eventname: "event1"
total_price: 40
ticket_amount: 4
ticket_type: "normal"
```

(15) Εναλλαγή σελίδων διαχειριστή σε σελίδες χρήστη:

Πατώντας το "Home User" μπορεί ο admin να αλλάξει τις σελίδες που θα παρουσιάζονται στο topnav bar ώστε να είναι εκείνες των χρηστών. Έτσι μπορεί να έχει πρόσβαση σε αυτές για να βλέπει τις αλλαγές που τυχόν κάνει.



Επίσης όταν εμφανιστούν οι σελίδες χρηστών εμφανίζεται και η επιλογή "Home Admin" οπού θα εμφανίσει ξανά τις σελίδες του διαχειριστή.