**ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΙΑΙΚΤΎΟΥ:**

**ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ**

*Εφαρμογή τουριστικού οδηγού μιας περιοχής*

Μέλη ομάδας

Βλαχοδημητρόπουλος Δημήτριος 1053572

Δασκαλάκης Στυλιανός 1053762

**1.Εισαγωγή**

Η παρούσα έκθεση, αφορά την εργασία με τίτλο «Εφαρμογή τουριστικού οδηγού μιας περιοχής» που εκπονήθηκε στα πλαίσια του μαθήματος «Προγραμματισμός Διαδικτύου». Η εργασία αυτή αποτελεί μια πρακτική εφαρμογή τεχνολογιών *frontend* και *backend* για τον σχεδιασμό σελίδων στο διαδίκτυο. Για την ανάπτυξή της, χρησιμοποιήθηκαν τεχνολογίες όπως η *HTML*, *CSS*, *JavaScript*, *node js* καθώς και το framework *bootstrap*.

Η εφαρμογή παρέχει λειτουργίες περιήγησης σε μη εγγεγραμμένους χρήστες (visitors) καθώς και λειτουργίες εξατομικευμένης περιήγησης σε εγγεγραμμένους χρήστες (users) και στους διαχειριστές της σελίδας (admins).

**2.Περίληψη**

Ένας τουριστικός οδηγός νησιού αποτελεί ένα πολύ χρήσιμο εργαλείο για έναν ταξιδιώτη που επιθυμεί να λάβει σημαντικές πληροφορίες σχετικά με το μέρος το οποίο επρόκειτο να επισκεφθεί. Μία τέτοια εφαρμογή τον βοηθά να γνωρίσει καλύτερα τον προορισμό του, καθώς παρέχει πληροφορίες τόσο για τα αξιοθέατα, όσο και για τις υπηρεσίες του τόπου. Κύριο μέλημα, λοιπόν, της εφαρμογής μας είναι να ικανοποιούνται αρχικά αυτές οι προϋποθέσεις και να παρέχονται στον χρήστη όλες οι απαραίτητες γνώσεις. Συγκεκριμένα, έχοντας επιλέξει σαν περιοχή το ελληνικό νησί της Μήλου, φροντίσαμε ο χρήστης να έχει πρόσβαση σε βασικές πληροφορίες όπως: παραλίες, χώροι εστίασης, ξενοδοχεία και αξιοθέατα όπως αρχαιολογικοί χώροι του νησιού.

Προκειμένου όμως να υπάρχει και μια αλληλεπίδραση μεταξύ της σελίδας και του χρηστή, καθώς και μεταξύ των χρηστών, προσθέσαμε στη σελίδα τη δυνατότητα σχολιασμού από τον χρήστη του εκάστοτε αξιοθέατου/υπηρεσίας. Με αυτό τον τρόπο, ο επισκέπτης μπορεί να λάβει μια καλύτερη εικόνα για το κάθε μέρος μέσα από τις εμπειρίες των άλλων χρηστών. Βέβαια, η δυνατότητα αυτή παρέχεται μόνο στους εγγεγραμμένους χρήστες. Ακόμη δημιουργήσαμε βαθμολογία για το κάθε μέρος ώστε ο χρήστης να έχει περισσότερες πληροφορίες και να μην βασίζεται αποκλειστικά και μόνο στα σχόλια. Ο χρήστης μπορεί να δώσει μόνο μία βαθμολογία για κάθε μέρος, την οποία μπορεί να ανανεώσει.

Για την ομαλή λειτουργία της σελίδα και για να αποτραπεί η κατάχρηση των δικαιωμάτων των χρηστών (πχ. η προσθήκη υβριστικών σχολίων) η σελίδα έχει μια ακόμη κατάταξη χρηστών τους *admin*. Οι *admins* έχουν τον ρόλο του διαχειριστή. Τα εργαλεία που έχουν στην διάθεση τους είναι όλες οι βασικές ιδιότητες των κανονικών χρηστών καθώς και η διαγραφή σχολίου , η προσθήκη ενός νέου άρθρου στις ήδη υπάρχουσες κατηγορίες και η διαγραφή άρθρου. Ο μόνος τρόπος να αναβαθμιστεί ένας χρήστης σε *admin* είναι να τον αναβαθμίσει ένας άλλος *admin* γράφοντας το email του λογαριασμού στην admin φόρμα.

**3.Μέθοδος επίλυσης**

**3.1 Προσέγγιση του προβλήματος**

Για τον σχεδιασμό της εφαρμογής οι δύο βασικοί άξονες οι οποίο επηρέασαν της επιλογές μας ήταν ο τρόπος που θα σχεδιαστεί η βάση δεδομένων και ο τρόπος που θα φορτώνονται τα αρχεία *.hbs* ώστε να μπορεί η σελίδα να επεκτείνεται χωρίς να χρειάζονται καινούρια *hbs* αρχεία. Με το σκεπτικό αυτό καταλήξαμε στην απόφαση η βάση δεδομένων μας να έχει αποθηκευμένα όλες τις πληροφορίες των άρθρων. Έτσι κατά την φόρτωση ενός άρθρου γίνεται πάντα render το ίδιο *hbs* με διαφορετικά δεδομένα τα οποία στέλνονται από την βάση.

**3.2 Υλοποίηση της βάσης**

Αποφασίστηκε η βάση να στεγάζεται στο Heroku και να είναι PostgreSQL για να έχουμε την δυνατότητα να στεγάσουμε πολλά δεδομένα δωρεάν. Οι βασικοί πίνακες της βάσης είναι οι εξής:

Α)*Comments*: Αποθηκεύονται τα comments των χρηστών με γνωρίσματα sight\_id (id του άρθρου που έχει γραφτεί), user\_email (email χρήστη που το δημιούργησε), datetime (ημερομηνία και ώρα δημιουργίας comment), comment (το κείμενο του σχολίου), comment\_id (το μοναδικό id του κάθε comment)

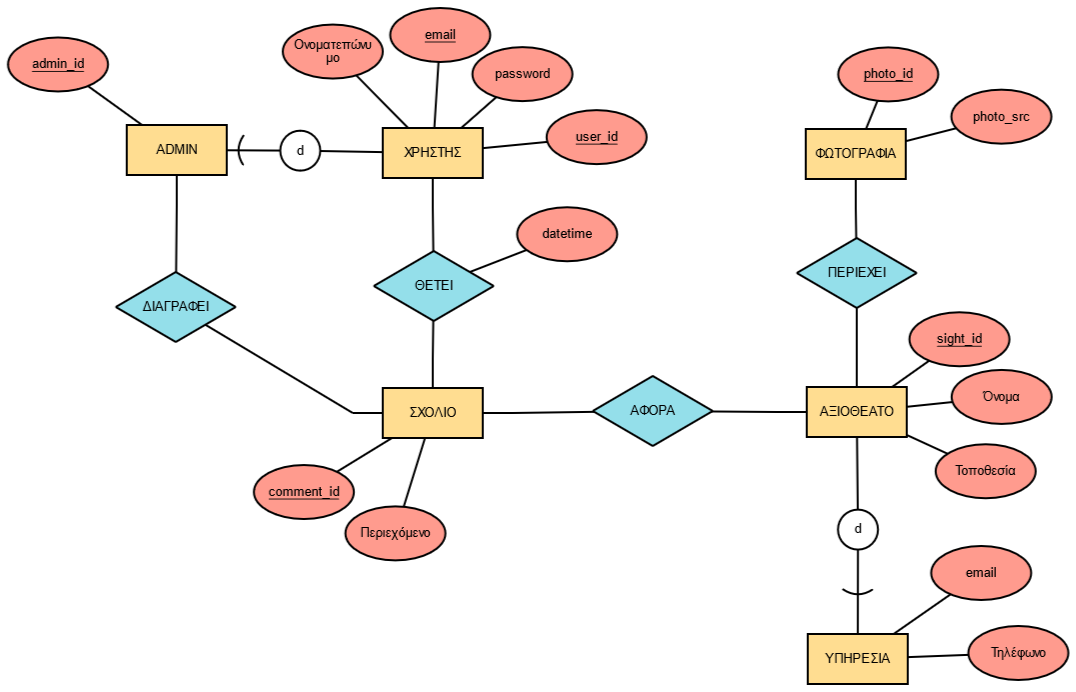
Β)*Users*: Αποθηκεύονται τα στοιχεία των χρηστών της εφαρμογής με γνωρίσματα user\_id (το μοναδικό id του κάθε χρήστη), password (ο κωδικός του χρήστη που αποθηκεύεται σε μορφή hash ώστε να μην φαίνεται πουθενά), email (το email του χρήστη), name (το όνομα του χρήστη), surname (το επώνυμο του χρήστη), role (ο ρόλος του χρήστη δηλαδή αν είναι απλώς BASIC ή διαχειριστής ADMIN)

Γ)*Sights*: Αποθηκεύονται τα στοιχεία των άρθρων της σελίδας με γνωρίσματα sight\_id (το μοναδικό id του κάθε άρθρου), sight\_ name (το όνομα του άρθρου), location (η τοποθεσία που βρίσκεται το αξιοθέατο παρμένη από το google maps), service\_email (προαιρετικό για τα αξιοθέατα με email πχ. ξενοδοχεία), service\_phone (προαιρετικό για τα αξιοθέατα με τηλέφωνο πχ. ταβέρνες), sight\_type (το είδος που ανήκει το αξιοθέατο πχ. παραλία, ταβέρνα ή δωμάτια ), photo\_src (το source της κύριας φωτογραφίας του αξιοθέατου), sight\_text (το κείμενο που περιγράφει το αξιοθέατο και εμφανίζεται στο άρθρο)

Δ)*Photos*: Αποθηκεύονται όλα τα στοιχεία των φωτογραφιών που περιέχονται στη σελίδα, με γνωρίσματα photo\_id (το μοναδικό id της κάθε φωτογραφίας), photo\_src (το source της κάθε φωτογραφίας) και το sight\_id που αποτελεί ξένο κλειδί που προσδιορίζει ποιες φωτογραφίες αφορούν το κάθε μέρος.

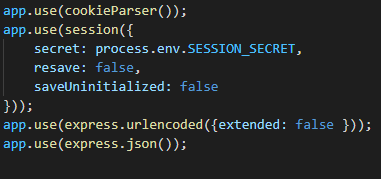
Ε)*Ratings*: Αποθηκεύονται όλε οι βαθμολογίες των χρηστών με γνωρίσματα sight\_id(το id του άρθρου που γίνεται η βαθμολόγηση), User\_email(το email του χρήστη ο οποίος κάνει το rating), rating(α αριθμός της βαθμολόγησης)

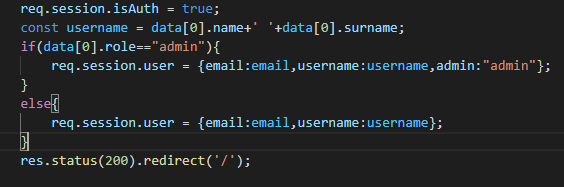
Παρακάτω βλέπουμε και το ERD που υλοποιεί αυτή τη βάση δεδομένων



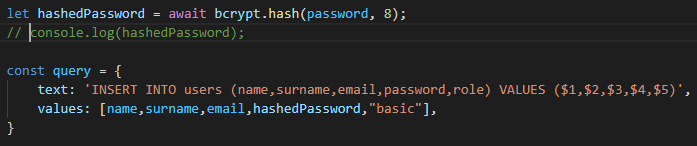
**3.3 Κατασκευή του backend**

Το backend είναι αυτό που φροντίζει την ομαλή λειτουργία της σελίδας διαχειρίζεται όλα τα αιτήματα και οργανώνει την μεταφορά δεδομένων ανάμεσα στον χρήστη και την βάση. Κατασκευάστηκε με την βοήθεια του node js και χρησιμοποιήθηκαν οι βιβλιοθήκες express καθώς και η express sessions για να είναι εφικτή η διαχείριση των δεδομένων. Όταν ο client ζητάει μια σελίδα το hbs της φορτώνεται και στέλνεται μαζί με τα απαραίτητα δεδομένα χρήστη(αν είναι συνδεδεμένος) και τις πληροφορίες από την βάσης δεδομένων που χρειάζονται για να εμφανιστούν τα στίχοι στην οθόνη. Η σύνδεση των εγγεγραμμένων χρηστών και του διαχειριστή διατηρείται με τη χρήση sessions. Όταν γίνεται login από έναν χρήστη ο server κάνει authenticate το session και αποθηκεύει σε αυτό τα δεδομένα του χρήστη(όνομα και ρόλος). Με το logout το session καταστρέφεται αφήνοντας τον client ξανά “άγνωστο” στον server. Το session αυτό διαρκεί για πολύ μεγάλο χρονικό διάστημα. Μετά το πέρας του χρόνου αυτού, θα πρέπει να ξανασυνδεθεί για να συνεχίσει την εξατομικευμένη περιήγηση.





Επισημαίνεται επιπλέον, ότι οι κωδικοί πρόσβασης των χρηστών που είναι αποθηκευμένοι στην βάση, δεν είναι αποθηκευμένοι σε μορφή απλού κειμένου, αλλά κωδικοποιούνται με τη χρήση του module bcrypt. Έτσι διασφαλίζεται κατά κάποιο τρόπο η ασφάλεια των χρηστών, καθότι η βάση ενδεχομένως να μπορεί να προσπελαστεί από κακόβουλους χρήστες.



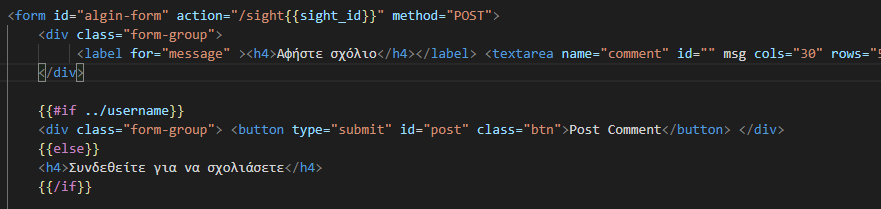
Οι συναρτήσεις που εκτελούνται από τα request των χρηστών ελέγχουν και οι ίδιες αν το request έρχεται από χρήστη με τις κατάλληλες δικαιοδοσίες. Αυτό γίνεται καθώς παρόλο που ανάλογα με το είδος χρήστη που είναι συνδεδεμένο εμφανίζονται μόνο οι δυνατές ενέργειες πρέπει σε περίπτωση που δοθεί request από κάποιον που δεν θα έπρεπε αυτό να μην επηρεάσει το site.

Τέλος οι συναρτήσεις οι οποίες δημιουργούν την επικοινωνία μεταξύ χρήστη και βάσης δεδομένων τα στέλνουν τα δεδομένα στα hbs αρχεία σε μορφή JSON. Για την ανάγνωση των δεδομένων που έρχονται σε μορφή post έχουν χρησιμοποιηθεί οι express (urlencoded) και express (JSON).

**3.4 Δημιουργία Frontend**

Το Frontend δημιουργήθηκε με την πρόθεση να έχει τα λιγότερα δυνατά αρχεία hbs και για την επίτευξη αυτoύ χρησιμοποιήθηκαν οι helpers {{#if}}, {{#each}} του hbs για διαχείριση δεδομένων σε μορφή JSON που έρχονται από τον server.

Το γραφικό περιβάλλον της σελίδας προήλθε κυρίως από βιβλιοθήκες της bootstrap σε συνδυασμό με την χρήση της γλώσσας CSS. Δημιουργήθηκαν επιμέρους αρχεία τα οποία φορτώνονται μαζί με τα αντίστοιχα δεδομένα ανάλογα με το request του χρήστη.



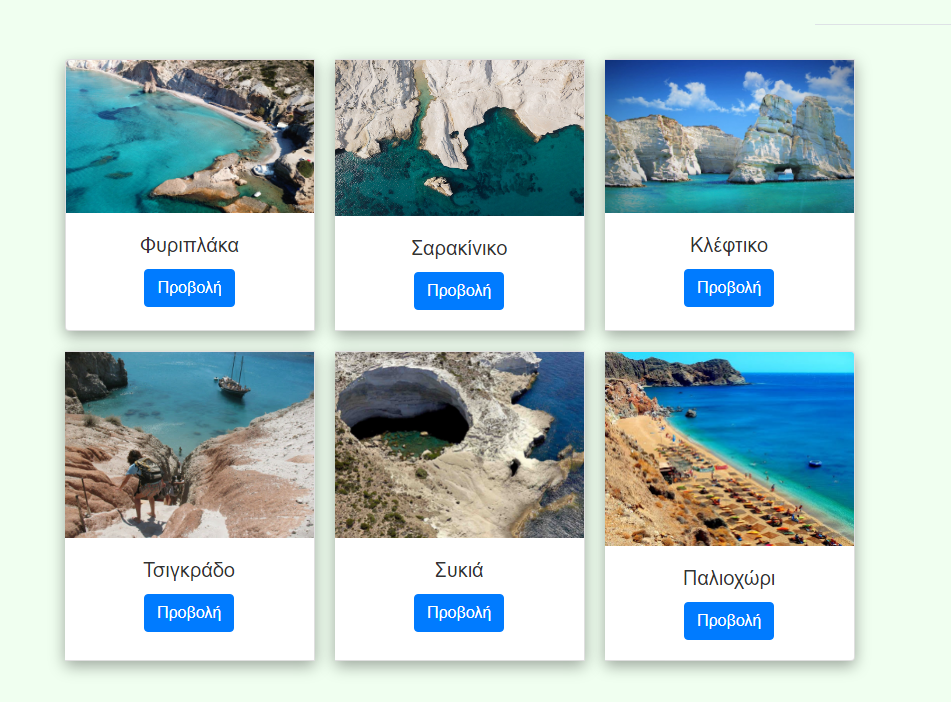
**4. Σενάρια χρήσης**

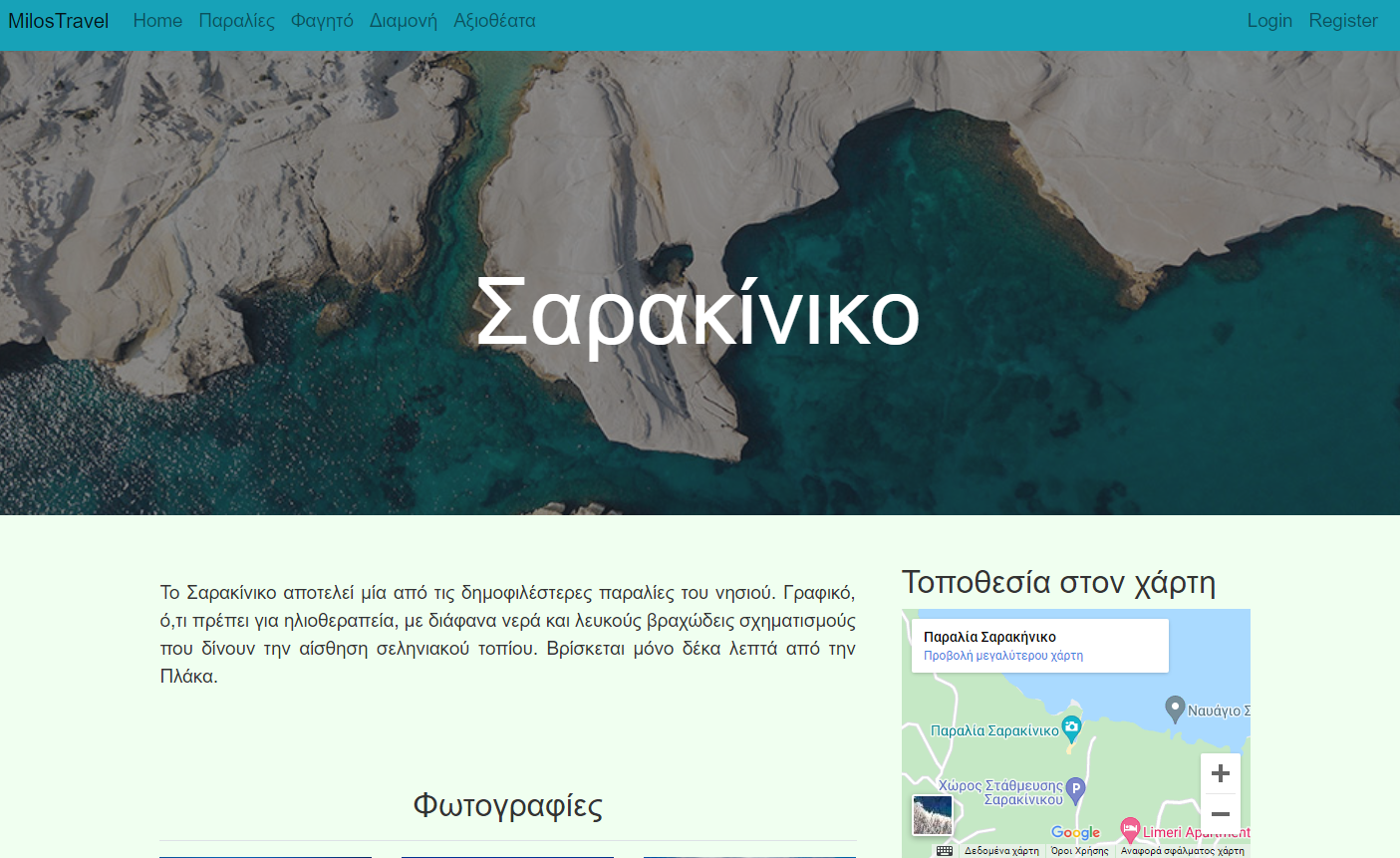
**4.1 Όχι εγγεγραμμένος χρήστης**

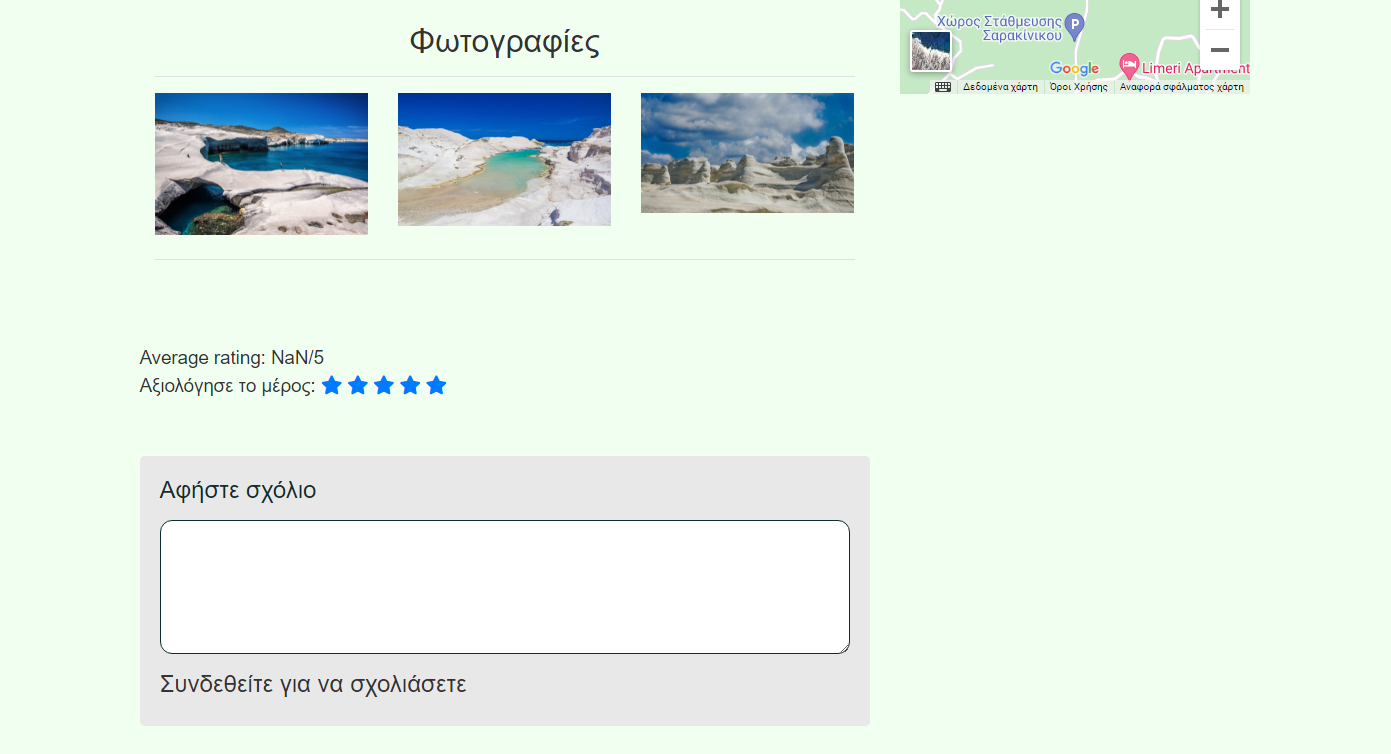
Όταν ένας χρήστης συνδέεται στο σαιτ μπορεί να περιηγηθεί ελεύθερα σε όλες τι σελίδες. Μπορεί να δει όποιο άρθρο θέλει καθώς και την βαθμολογία και τα σχόλια τους . Ωστόσο οι δυνατότητες του τελειώνουν εκεί καθώς δεν μπορεί να κάνει κάποιο σχόλιο η να αφήσει βαθμολογία.





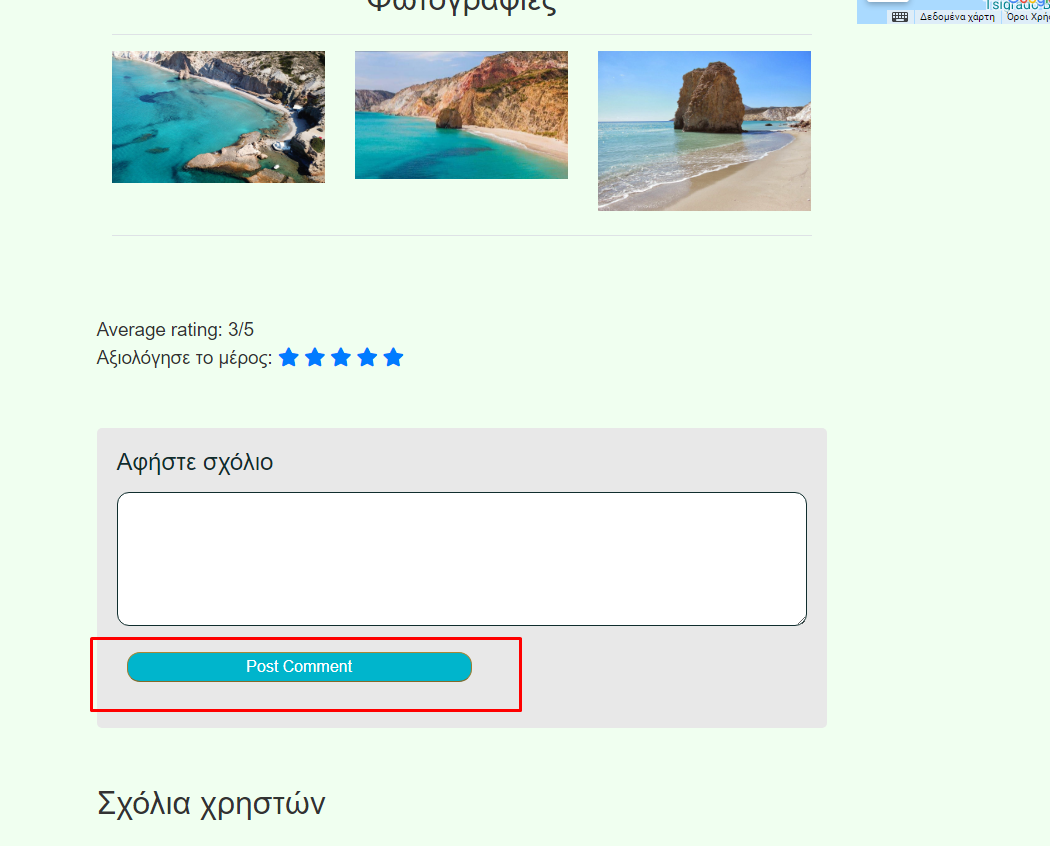






**4.2 Εγγεγραμμένος χρήστης**

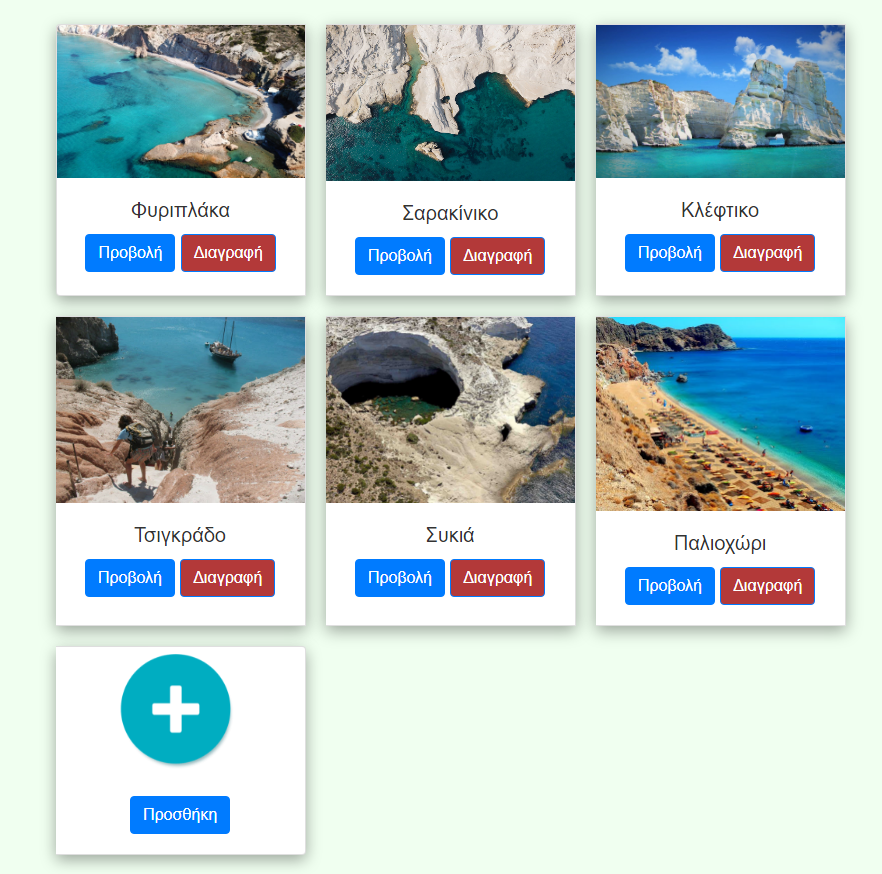
Οι εγγεγραμμένοι χρήστες έχουν δυνατότητα να περιηγηθούν ακριβώς όπως οι μη εγγεγραμμένοι απλώς μπορούν επίσης να προσθέσουν σχόλια και βαθμολογία στην κάθε σελίδα. Μπορούν να κάνουν όσα σχόλια θέλουν όμως μπορούν να δώσουν μια βαθμολογία στο κάθε άρθρο. Σε περίπτωση που πατηθεί ξανά η βαθμολόγηση τότε αλλάζει η προηγούμενη και μπαίνει η τιμή της νέα. Τέλος οι απλοί χρήστες μπορούν να αλλάξουν τα στοιχεία του λογαριασμού τους ανά πάσα χρονική στιγμή.



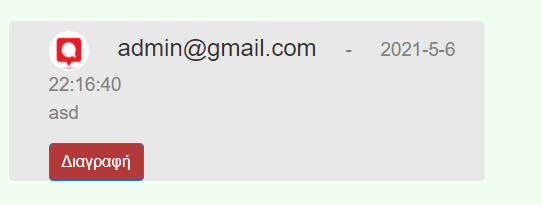
**4.3 Admin**

Οι Admins έχουν όλες τις δυνατότητες των κανονικών χρηστών και κάποιες επιπλέον. Μπορούν να προσθέσουν καινούρια άρθρα.

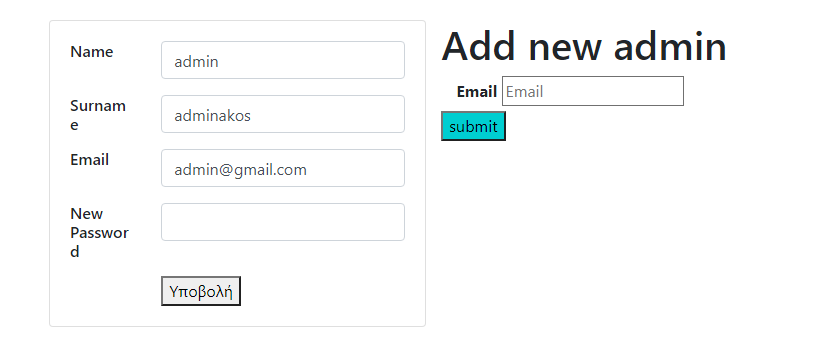
Να διαγράψουν ήδη υπάρχων άρθρα



Να διαγράψουν σχόλια χρηστών



Να προάγουν έναν κανονικό χρήστη σε admin



**5 Οδηγίες Χρήσης**

Οδηγίες χρήσης:

1.Κατεβάζουμε όλα τα αρχεία

2.Φροντίζουμε να είναι εγκατεστημένα στον υπολογιστή μας τα: Node.js και npm

3.Κατεβάζουμε τα αντίστοιχα extensions στο VS Code (node.js & npm)

4.Στο VS Code πατάμε “Open Folder” και ανοίγουμε τον φάκελο που περιέχει το project

5.Ανοίγουμε το terminal (Ctrl + J) και

a.Τρέχουμε το command npm install

b.Τρέχουμε το command npm run debug

5.Στον browser επισκεπτόμαστε την διεύθυνση http://localhost:3001/

6.Περιηγούμαστε ελεύθερα στην ιστοσελίδα!

Ήδη Διαθέσιμοι Λογαριασμοί:

Τυπικοί χρήστες:

username: normaluser@gmail.com

password:456

Διαχειριστής:

username: [admin@gmail.com](mailto:admin@gmail.com)

password: 123