

PROJECT ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΔΙΑΔΙΚΤΥΟΥ (2021-2022)

Full stack web-site εφαρμογής αναζήτησης ακινήτου για αγορά/ενοικίαση

Λουζιώτης-Αγγέλης Φίλιππος

AM: 1066504

Ευαγγελάτος Σπυρίδων

AM: 1066532

ΟΜΑΔΑ 7

1 ΕΙΣΑΓΩΓΗ

Στόχος του συγκεκριμένου project ήταν η δημιουργία μιας ιστοσελίδας αναζήτησης ακινήτου για αγορά ή ενοικίαση. Ιδιαίτερη σημασία δόθηκε στην διευκόλυνση της διαδικασίας αναζήτησης και ανάδειξη του ενδιαφέροντος για τα ακίνητα από το χρήστη και ταυτόχρονα, να έχει τη δυνατότητα να δημιουργεί αγγελίες ο ίδιος. Βέβαια, πρέπει να σημειωθεί πως οι αγγελίες αυτές, δεν αναρτώνται κατευθείαν αλλά γίνεται προέγκριση από τον διαχειριστή.

2 ΜΕΘΟΔΟΛΟΓΙΑ

Αρχικά, πραγματοποιήθηκε έρευνα σε διάφορες ιστοσελίδες Real Estate, ώστε να γίνει κατανοητό το αντικείμενο και η δομή αλλά και ο τρόπος υλοποίησης. Έπειτα, σε πρώτη φάση, έγινε η δημιουργία του εννοιολογικού μοντέλου “ERD” και σύντομα ακολούθησε μια γενική κατασκευή της δομής των ιστοσελίδων με χρήση HTML, για χτιστεί μια πρώτη ιδέα.

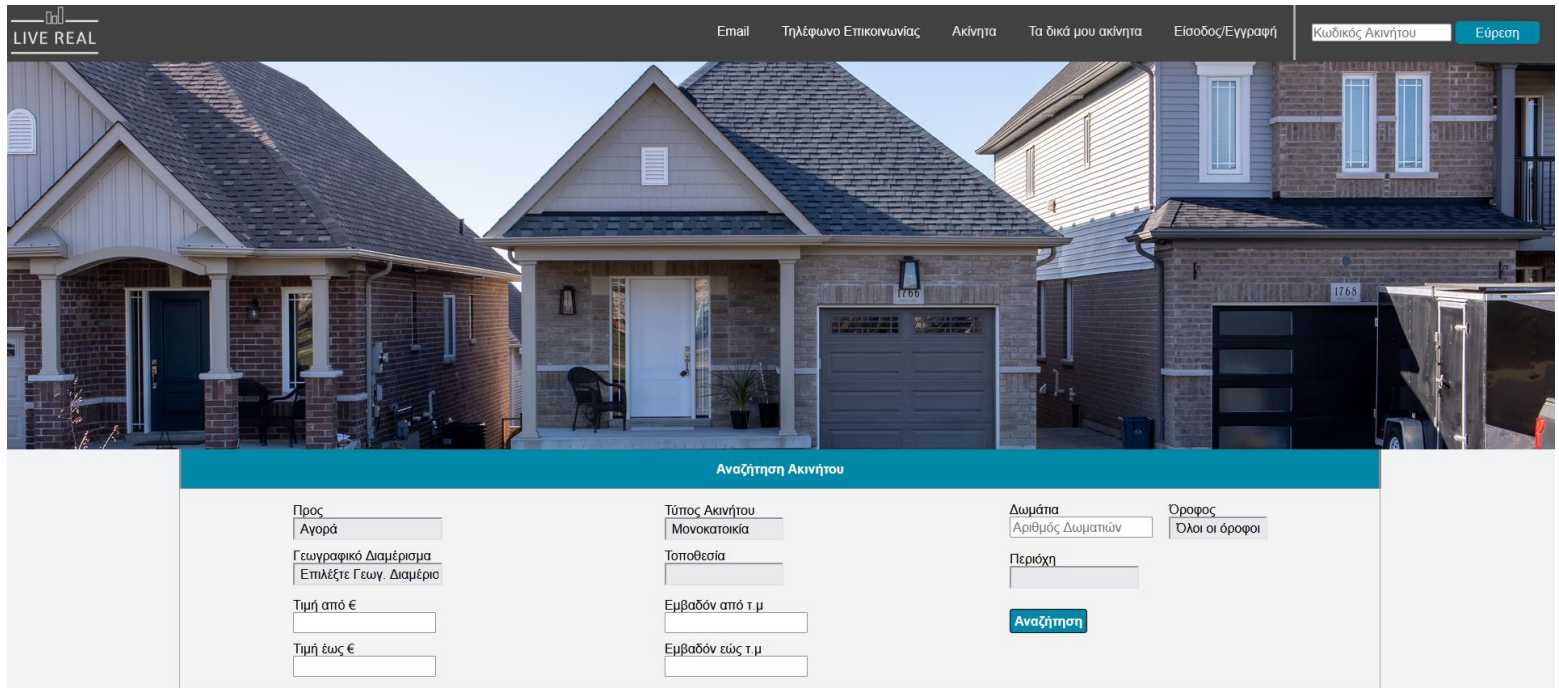
Ο τρόπος με τον οποίο κατασκευάστηκε η ιστοσελίδα ακολούθησε την εξής διαδικασία:

- Πρώτα, έγινε χρήση της HTML για τη δημιουργία των βασικών σελίδων (index.html, listings.html), για να γίνει κατανοητή η βασική δομή, με τη συνοδεία λίγης μορφοποίησης (CSS).
- Στη συνέχεια, ολοκληρώθηκε το μοντέλο συσχετίσεων-οντοτήτων για την ορθή κατασκευή της βάσης δεδομένων.
- Μετά την ενδιάμεση παρουσίαση, που λύθηκαν κάποιες απορίες, ακολούθησε η ολοκλήρωση των παραπάνω σελίδων και η κατασκευή κάποιων καινούργιων που θα συμπλήρωναν τις αναγκαίες λειτουργίες.
- Έπειτα, έγινε η σύνδεση των σελίδων μεταξύ τους και με κώδικα javascript για την πολύ βασική λειτουργικότητα τους.
- Με τη χρήση της postgres, δημιουργήθηκε η βάση δεδομένων, και η πολύ βασική λειτουργικότητα του back end.
- Τέλος, κάνοντας χρήση της Node.js και Express.js, έγινε η σύνδεση της βάσης δεδομένων με το server που τη διαχειρίζεται, ενώ παράλληλα, έγινε και η σύνδεση του front end με το server μέσω των Http Requests της Express, ολοκληρώνοντας και το front end αλλά και το back end.

3 ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΣΙΑΣ

3.1 FRONT-END:

Αρχικά, μπαίνοντας στην ιστοσελίδα, παρουσιάζεται στον χρήστη η βασική σελίδα index.html:



The screenshot shows the front-end of the 'Live Real' website. At the top is a dark navigation bar with the 'LIVE REAL' logo on the left and links for 'Email', 'Τηλέφωνο Επικοινωνίας', 'Ακίνητα', 'Τα δικά μου ακίνητα', and 'Είσοδος/Εγγραφή' in the center. On the right of the navigation bar is a search bar with the placeholder 'Κωδικός Ακινήτου' and a 'Εύρεση' button. Below the navigation bar is a large image of a modern house. Underneath the image is a white search form titled 'Αναζήτηση Ακινήτου'. The form contains several input fields: 'Προς' (with a dropdown menu showing 'Αγορά'), 'Γεωγραφικό Διαμέρισμα' (with a dropdown menu showing 'Επιλέξτε Γεωγ. Διαμέριο'), 'Τιμή από €', 'Τιμή έως €', 'Τύπος Ακινήτου' (with a dropdown menu showing 'Μονοκατοικία'), 'Τοποθεσία', 'Εμβαδόν από τ.μ.', 'Εμβαδόν έως τ.μ.', 'Δωμάτια' (with a dropdown menu showing 'Αριθμός Δωματίων'), 'Όροφος' (with a dropdown menu showing 'Όλοι οι όροφοι'), and 'Περιοχή'. A blue 'Αναζήτηση' button is located at the bottom right of the search form.


Η δομή της αποτελείται από δύο τομείς, το “Navigation Bar” και την εκάστοτε ιστοσελίδα.

Ο πρώτος τομέας, που υπάρχει σε κάθε σελίδα, το “Navigation Bar” έχει τις εξής λειτουργίες:

Αρχικά, πατώντας το Logo “Live Real”, δίνεται η δυνατότητα επιστροφής στην αρχική σελίδα index.html. Στη συνέχεια, το “Email” και το “Τηλέφωνο Επικοινωνίας”, προσφέρουν κάποιες πληροφορίες για τους διαχειριστές μέσω ενός Dropdown Menu όταν κάνει hover ο χρήστης πάνω από αυτά. Τα υπόλοιπα τρία είναι απλά κουμπιά που οδηγούν σε ιστοσελίδες που θα παρουσιαστούν παρακάτω. Τέλος, στη δεξιά πλευρά, μπορεί ο χρήστης να ψάξει αποκλειστικά ένα ακίνητο, γνωρίζοντας τον κωδικό του.

Η κύρια λειτουργία του δεύτερου τομέα, δηλαδή της παραπάνω σελίδας, είναι να προσφέρει στο χρήστη μια εισαγωγή στο περιεχόμενο. Ο ρόλος της, είναι να φιλτράρει τα ακίνητα της βάσης, ανάλογα με τις προτιμήσεις του χρήστη. Πιο συγκεκριμένα, μέσω του πεδίου αναζήτησης ακινήτου, ο χρήστης έχει την δυνατότητα να εισάγει τις ανάγκες του σε περιοχή, τύπο ακινήτου, αγορά/ενοικίαση κλπ., ενώ δεν είναι ανάγκη να συμπληρώσει όλα τα πεδία παρά μόνο τα πολύ βασικά, όπως το «Γεωγραφικό Διαμέρισμα». Τέλος, πατώντας το κουμπί “Αναζήτηση” θα οδηγηθεί στην σελίδα listings.html, που θα επιστρέψει τα ακίνητα που πληρούν τις ανάγκες του.


Η σελίδα listings.html παρουσιάζεται παρακάτω

EmailΤηλέφωνο ΕπικοινωνίαςΑκίνηταΤα δικά μου ακίνηταΕίσοδος/Εγγραφή

Αποτελέσματα αναζήτησης

Ταξινόμηση

Προς Ενοίκιαση




Διαμέρισμα, 50 τ.μ.
Κωδικός: 19

600 €

Πάτρα - Κέντρο (Πάτρα)
Αποτελείται από 1 υπνοδωμάτιο, σαλόνι, κουζίνα, μπ...

Προς Πώληση



Μονοκατοικία, 90 τ.μ.
Κωδικός: 20

230.000 €

Ψαροφάι (Πάτρα)
Αποτελείται από 2 υπνοδωμάτια, σαλόνι, κουζίνα, μπ...

Αναζήτηση

Περιοχή

Τύπος Ακινήτου

Προς

Τιμή από €

Τιμή έως €

Εμβαδόν από τ.μ

Εμβαδόν έως τ.μ

Δωμάτια

Όροφος


<< PREVIOUS PAGE

NEXT PAGE >>

Στην δεξιά πλευρά, ο χρήστης έχει την δυνατότητα να ταξινομήσει τα ακίνητα ανάλογα με τις τιμές τους (αύξουσα, φθίνουσα), αλλά και μπορεί να γίνει ένα επιπλέον φιλτράρισμα ανάλογα με τις παραμέτρους του χρήστη, στο πεδίο «Αναζήτηση».

Στην αριστερή πλευρά, βρίσκονται τα σπίτια που επιστρέφονται από τη βάση δεδομένων που πληρούν τις προτιμήσεις του χρήστη αν έχει πραγματοποιήσει κάποια αναζήτηση από τις λειτουργίες που προαναφέρθηκαν, ενώ αν απλά πατήσει το κουμπί «Ακίνητα» του “Navigation Bar”, θα παρουσιαστούν αφιτράριστα, όλα τα ακίνητα της βάσης. Παράλληλα, ώστε να μην φαίνονται όλα τα ακίνητα σε μία σελίδα, πραγματοποιήθηκε η διαδικασία του “Pagination”, που είναι η απλή φόρτωση ορισμένου αριθμού στοιχείων ανά page. Τέλος, πατώντας πάνω στην εικόνα ή το τίτλο του κάθε ακινήτου, ο χρήστης οδηγείται σε σελίδα που παρουσιάζονται πιο εκτεταμένα οι πληροφορίες του ακινήτου. Η σελίδα αυτή είναι η property.html.


Η property.html παρουσιάζεται παρακάτω:



[Email](#) [Τηλέφωνο Επικοινωνίας](#) [Ακίνητα](#) [Τα δικά μου ακίνητα](#) [Έξοδος](#)

[Εύρεση](#)

Προς Πώληση



Μονοκατοικία, 280 τ.μ.
Κωδικός: 21

350.000 €

Κρήνη (Μεσσαπίδα)

Αποτελείται από 6 υπνοδωμάτια, 2 σαλόνια, 2 κουζίνες, 2 μπάνια και WC. Είναι κατασκευασμένη το 2002 και διαθέτει θέρμανση αυτόνομη - πετρελαίου, απεριόριστη θέα, κουφώματα αλουμινίου, πατώματα από πλακάκι, εντοιχισμένες ντουλάπες, πρόσβαση ΑΜΕΑ, πάρκινγκ, αποθήκη 5 τ.μ., κήπο, τζακι, A/C, συναγερμό, τέντες, σίτες, διπλά τζάμια, ηλιακό θερμοσίφωνα, Boiler, σκαλιά εσοδού, νυχτερινό ρεύμα, ανοιχτωσιά, εσωτερική σκάλα, μπαλκόνια 60 τ.μ. απόσταση από θάλασσα 4000 μέτρα

Με ενδιαφέρει

[Αποστολή](#)

Η σελίδα είναι απλή σε δομή, φανερώνονται όλα τα δεδομένα του συγκεκριμένου ακινήτου. Συνεχίζοντας, αν ο χρήστης πατήσει πάνω στην “Είσοδο/Εγγραφή”, θα παρατηρήσει την παρακάτω εικόνα του login.html:

Username:

Password:

[Login](#)

[Register](#)

Συγκεκριμένα, ο χρήστης μπορεί να αποκτήσει δικό του λογαριασμό στην ιστοσελίδα, εφόσον με αυτό το τρόπο, δίνονται παραπάνω δικαιώματα, όπως η πρόσθεση αγγελιών ή και η ένδειξη ενδιαφέροντος για ήδη υπάρχουσες.

Αν δεν έχει λογαριασμό, πατώντας “Register” του δίνεται η δυνατότητα να κάνει εγγραφή και να αποθηκευτούν τα στοιχεία του στη βάση, μέσω της φόρμας:

Email:


Username:

Password:

Repeat Password:

[Register](#)

Επιπροσθέτως, έχοντας κάνει είσοδο στην ιστοσελίδα, το κουμπί “Είσοδο/Εγγραφή” αλλάζει σε “Εξοδος”, που πατώντας το οποίο, εκτελείται η λειτουργία της αποσύνδεσης του λογαριασμού του χρήστη. Επιπλέον, ο χρήστης μπορεί να πατήσει στο κουμπί “Τα δικά μου ακίνητα” και να οδηγηθεί στην παρακάτω σελίδα:




[Email](#) [Τηλέφωνο Επικοινωνίας](#) [Ακίνητα](#) [Τα δικά μου ακίνητα](#) [Εξοδος](#)

[Εύρεση](#)

[Προσθήκη Νέου Ακινήτου](#)

Προς Ενοίκηση




Διαμέρισμα, 50 τ.μ.
Κωδικός: 19

Πάτρα - Κέντρο (Πάτρα)
Αποτελείται από 1 υπνοδωμάτιο, σαλόνι, κουζίνα, μπάνιο . Είναι κατασκευασμένο το 1985 με ενεργειακή ...

600 €

[Διαγραφή](#)

Προς Πώληση



Μονοκατοικία, 90 τ.μ.
Κωδικός: 20


Ψαροφάι (Πάτρα)
Αποτελείται από 2 υπνοδωμάτια, σαλόνι, κουζίνα, μπάνιο . Είναι κατασκευασμένο το 2010 με ενεργειακή ...

230.000 €

[Διαγραφή](#)

Εδώ, παρατίθενται όλα τα ακίνητα που έχει υποβάλει ο ίδιος ανάλογα με το λογαριασμό του, χωρίς να έχει σημασία αν έχουν εγκριθεί ή όχι.

Προς Ενοίκηση



Μονοκατοικία, 120 τ.μ.
Κωδικός: 18

Αγία Σοφία (Πάτρα)
Αποτελείται από 3 υπνοδωμάτια, σαλόνι, κουζίνα, μπάνιο . Είναι κατασκευασμένο το 1980 και διαθέτει θ...

500 €

[Διαγραφή](#) [Εκκρεμεί έγκριση](#)

Πρέπει να σημειωθεί, πως αν πατήσει το κουμπί “Τα δικά μου ακίνητα” χωρίς να έχει κάνει είσοδο στην ιστοσελίδα, θα οδηγηθεί πάλι στο login.html.

5

Μέσω του κουμπιού “Προσθήκη Νέου Ακινήτου”, μπορεί να δημιουργήσει νέα αγγελία, συμπληρώνοντας την παρακάτω φόρμα:

Πατέρα - Κέντρο (Πάτρα)

Προσθήκη Νέου Ακινήτου

Ονοματεπώνυμο
Τηλέφωνο

Γεωγραφικό Διαμέρισμα
Επιλέξτε Γεωγ. Διαμέρισμα

Τοποθεσία
Περίοχη

Διεύθυνση
π.χ. 'Διεύθυνση Νούμερο'

Προς
Αγορά

Εμβαδόν τ.μ

Τιμή

Τύπος Ακινήτου
Μονοκατοικία

Δωμάτια
Αριθμός Δωματιών

Οροφος
0

Περιγραφή

Πιστοποιητικό Ιδιοκτησίας
Επιλογή αρχείου Δεν επιλέχθη... κανένα αρχείο.

Φωτογραφία
Επιλογή αρχείου Δεν επιλέχθη... κανένα αρχείο.

Προσθήκη

ενεργειακή ...

230.000 €

ενεργειακή ...

350.000 €

ένη το 2002...

Από την άλλη πλευρά, πέρα από τον κανονικό χρήστη, υπάρχει και το διαχειριστής της ιστοσελίδας admin, που έχει δικαιώματα διαγραφής αγγελιών όπως και αποδοχής αυτών που δεν έχουν εγκριθεί ακόμα. Ειδικότερα, από τη στιγμή που θα γίνει σύνδεση με λογαριασμό admin, το κουμπί “Τα δικά μου ακίνητα” του γίνεται “Εκκρεμή Αιτήματα”, πατώντας το οποίο, οδηγείται στο παρακάτω:

Εκκρεμή Αιτήματα

Μονοκατοικία, 120 τ.μ.
Κωδικός: 18
Αγία Σοφία (Πάτρα)
Αποτελείται από 3 υπνοδωμάτια, σαλόνι, κουζίνα, μπάνιο. Είναι κατασκευασμένο το 1980 και διαθέτει θ...
[18.pdf](#)

Διαγραφή **Αποδοχή Αιτήματος**

500 €

Στην παραπάνω σελίδα, αναδεικνύονται οι αγγελίες που δεν έχουν εγκριθεί, καθώς και ένα link για το pdf του πιστοποιητικού ιδιοκτησίας του ακινήτου. Με την “Αποδοχή Αιτήματος”, αμέσως προστίθεται στη λίστα με τα ακίνητα που φαίνονται δημόσια, ενώ προφανώς με τη “Διαγραφή”, διαγράφεται ολόκληρη η αγγελία από τη βάση.

Πρέπει να σημειωθεί, πως ο admin έχει τη δυνατότητα να διαγράφει ακόμα και στα ακίνητα που έχουν εγκριθεί.

Εν τέλει, είναι αξιοσημείωτο πως όλες οι σελίδες είναι responsive και ενδεικτικά θα παρουσιαστούν κάποιες.

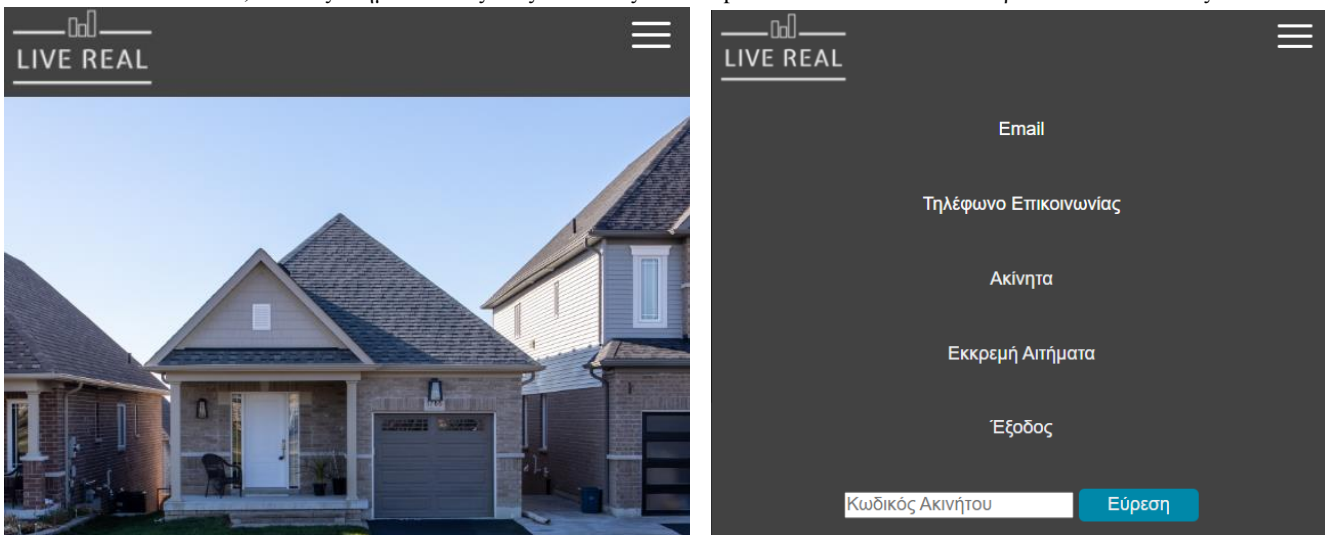


Figure 1: index.html με minimized Navigation Bar

Figure 2: index.html minimized

The screenshot shows the 'index.html' page of the 'LIVE REAL' website. It features a search form titled 'Αναζήτηση Ακινήτου' (Property Search). The form includes several input fields and a search button.

Search Form Fields:

- Προς:** Αγορά
- Γεωγραφικό Διαμέρισμα:** Επιλέξτε Γεωγ. Διαμέρισμα
- Τιμή από €:**
- Τιμή έως €:**
- Τύπος Ακινήτου:** Μονοκατοικία
- Τοποθεσία:**
- Εμβαδόν από τ.μ:**
- Εμβαδόν έως τ.μ:**
- Διαμέρισμα:** Αριθμός Διαμετρίων
- Όροφος:** Όλοι οι όροφοι
- Περιοχή:**

Search Button: Αναζήτηση

Figure 3: listings.html minimized

The screenshot shows the 'listings.html' page of the 'LIVE REAL' website. It displays search results for properties. The page includes a header with the 'LIVE REAL' logo and a navigation menu. Below the header, there are filters for 'Αποτελέσματα αναζήτησης' (Search Results) and 'Ταξινόμηση' (Sorting). The search results are displayed in a list format, showing property details and a 'Delete' button for each listing.

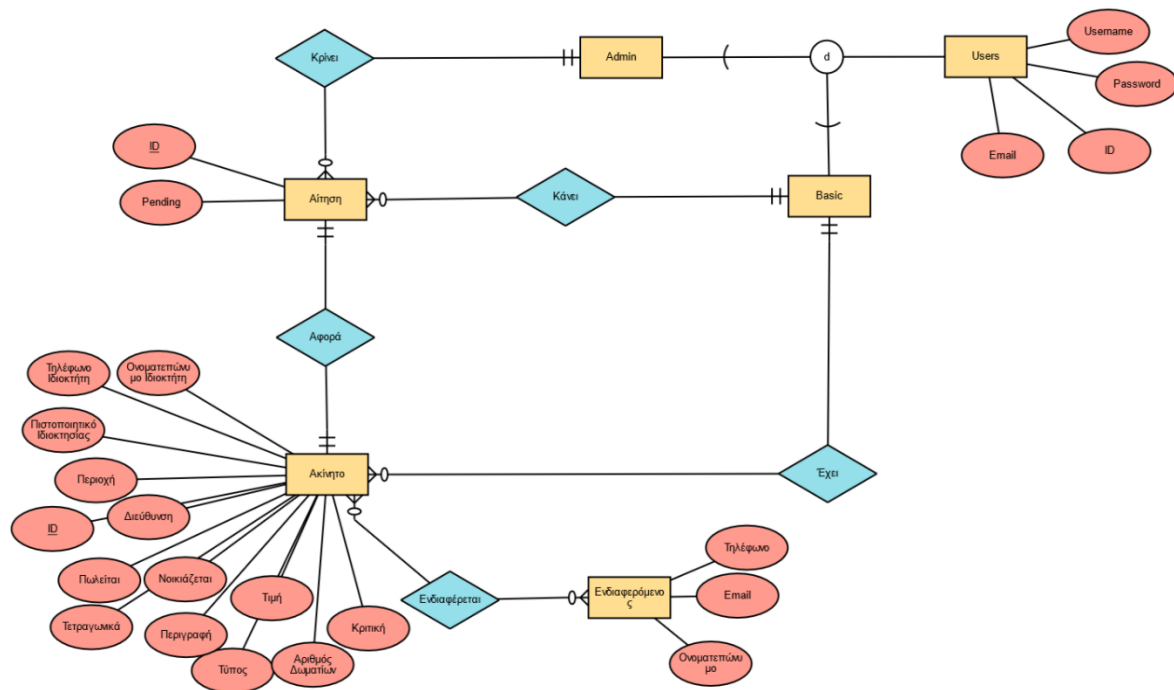
Search Results:

- Προς Ενοίκιαση:** Διαμέρισμα, 50 τ.μ. Κωδικός: 19. Πάτρα - Κέντρο (Πάτρα). Αποτελείται από 1 υπνοδωμάτιο, σαλόνι, κουζίνα, μπ... **Delete**
- Προς Πώληση:** Μονοκατοικία, 90 τ.μ. Κωδικός: 20. Ψαροφάι (Πάτρα). Αποτελείται από 2 υπνοδωμάτια, σαλόνι, κουζίνα, μπ... **Delete**

Navigation: << PREVIOUS PAGE, Αναζήτηση, NEXT PAGE >>

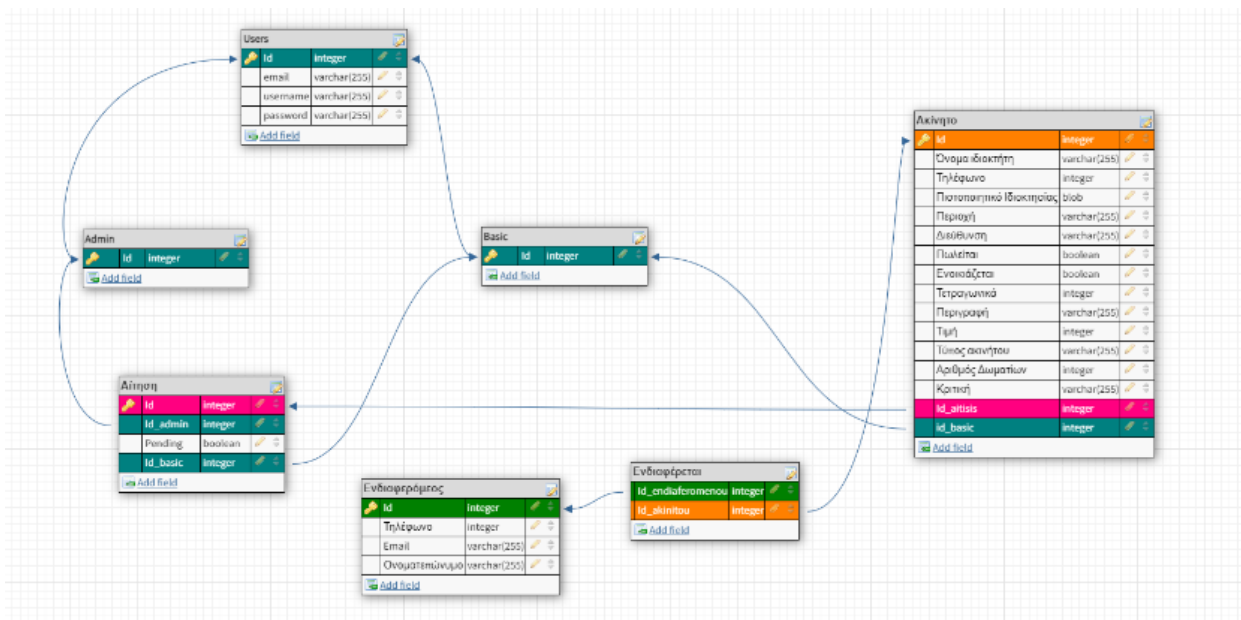
3.2 BACK-END:

Διάγραμμα οντοτήτων-συσχετίσεων:



Η λογική του διαγράμματος είναι πως οι χρήστες χωρίζονται σε 2 είδη, τον admin και τον basic, όπως περιγραφίκαν παραπάνω. Ο Basic μπορεί να κάνει μια αίτηση για αγγελία ακινήτου, η οποία καταλήγει στο table των αιτήσεων (Pending) και πρέπει να την εγκρίνει ο admin για να εμφανιστεί το ακίνητο δημοσία. Παράλληλα του basic μπορεί να του ανήκουν ήδη υπάρχουσες αγγελίες ακινήτων, στις οποίες μπορεί να δηλώσει οποιοσδήποτε ενδιαφέρον για το συγκεκριμένο ακίνητο.

Παρακάτω παρουσιάζεται το relational model:



Με τα παραπάνω δεδομένα δημιουργήθηκε η βάση δεδομένων σε postgres.

List of relations			
Schema	Name	Type	Owner
public	Admin	table	postgres
public	Basic	table	postgres
public	Interested	table	postgres
public	Property	table	postgres
public	Request	table	postgres
public	Users	table	postgres
public	isInterested	table	postgres

Στην συνέχεια δημιουργήθηκε ο server του backend, ώστε να λαμβάνει requests στο port 3000 και έγινε η σύνδεση με την βάση δεδομένων όπως φαίνεται παρακάτω:

```
const Pool = require("pg").Pool;

const pool = new Pool({
  user: "postgres",
  password: "postgres",
  host: "localhost",
  port: 5432,
  database: "liveReal",
});

module.exports = pool;
```

```
app.listen(3000, () => {
  console.log("Listening on port 3000");
});
```

Παράλληλα χρησιμοποιήθηκαν τα παρακάτω middleware:

```
// middleware
app.use(cors());
app.use(express.json()); // req.body
app.use(fileupload());
```

Αρχικά, το cors χρησιμοποιείται ώστε header των http requests, έτσι ώστε το Front-end να αποδέχεται τα resources που του στέλνονται από διαφορετικό server. Επίσης, το express.json χρησιμοποιείται για τις απαντήσεις των http requests και με το fileUpload υπάρχει δυνατότητα να ληφθούν και αρχεία στα requests.

Ενδεικτικά παρουσιάζονται κάποια http requests, που δέχεται το back-end:

```
// Get all properties from db
app.get("/properties", async (req, res) => {
  try {
    const listings = await pool.query(
      `SELECT "Property"."Id", "Property"."nameOfOwner", "Property"."phoneOfOwner", "Property"."province",
        "Property"."placeInTown", "Property"."town", "Property"."address", "Property"."toSell", "Property"."squareMeters",
        "Property"."description", "Property"."price", "Property"."typeOfProperty", "Property"."numOfRooms", "Property"."Id_request"
        FROM "Property", "Request"
        WHERE "Request"."Id" = "Property"."Id_request" AND "Request"."Pending" = FALSE;`
    );
    res.status(200).json(listings.rows);
  } catch (err) {
    console.error(err);
  }
});
```

Το παραπάνω δέχεται request με μέθοδο get στην διεύθυνση “/properties”, κάνει ένα query στην βάση δεδομένων, π.χ. που ζητά όλα τα ακίνητα χωρίς φιλτράρισμα και τα επιστρέφει ως response.

```
app.post("/login", async (req, res) => {
  try {
    const { username, password } = req.body;
    const user = await pool.query(
      `SELECT * FROM "Users" WHERE "username" = '${username}';`
    );
    if (user.rows.length > 0) {
      bcrypt.compare(
        password,
        user.rows[0].password,
        async (error, response) => {
          if (response) {
            const { Id } = user.rows[0];
            await bcrypt.hash(
              user.rows[0].username + Id,
              saltRounds,
              (err, hash) => {
                if (err) {
                  console.log(err);
                }
                tokens.push(Id + hash);
                setExpiration(tokens, tokens.indexOf(Id + hash), 1200000);
                res.status(200).json({ success: true, id: Id + hash });
              }
            );
          }
        }
      );
    }
  } else {
    res.send({ success: false, message: "Wrong username or password." });
  }
} catch (err) {
  console.error(err.message);
}
});
```

Όπως φαίνεται από την εικόνα, η μέθοδος του συγκεκριμένου request είναι post στην διεύθυνση “/login”. Ειδικότερα, χρησιμοποιούνται οι μεταβλητές από το body του request και γίνεται ένα query στην βάση για έλεγχο ύπαρξης του χρήστη, αν δεν υπάρχει γίνεται response του error, αλλιώς ελέγχεται ο κωδικός με τον hashed κωδικό του συγκεκριμένου χρήστη που υπάρχει ήδη στο database, μέσω του πακέτου bcrypt. Αν είναι επιτυχής ο έλεγχος το response απαρτίζεται από μήνυμα

επιτυχίας και ένα hash, για έλεγχο των δικαιωμάτων του χρήστη σε μελλοντικά request. Επιπροσθέτως το ίδιο hash αποθηκεύεται σε έναν πίνακα στο back-end, που ελέγχεται κάθε φορά στα request του χρήστη αν είναι σωστό, και διαγράφεται αν κάνει Logout ή αν περάσουν 20 λεπτά, όλα αυτά γίνονται για επιπρόσθετη ασφάλεια.

Παράλληλα, στο επόμενο request, που θα παρουσιαστεί, αναδεικνύεται ο τρόπος με τον οποίο ελέγχεται το hash του χρήστη, συγκρινόμενο με του προαναφερθέντα πίνακα και με ένα καινούργιο hash, που κατασκευάζεται με τον ίδιο τρόπο. Αν επιτευχθούν όλα τα παραπάνω, με ένα απλό query στην βάση, αποφασίζεται το είδος του χρήστη (Admin ή basic).

```
app.post("/verify", async (req, res) => {
  try {
    let { hash } = req.body;
    if (hash === null) {
      res.send({ success: true, role: "Basic" });
    } else {
      if (tokens.indexOf(hash) !== -1) {
        const id = hash.split("$")[0];
        hash = hash.replace(id, "");
        const user = await pool.query(
          `SELECT * FROM "Users" WHERE "Id" = '${id}'`
        );
        bcrypt.compare(
          user.rows[0].username + id,
          hash,
          async (err, response) => {
            if (response) {
              const role = await pool.query(
                `SELECT * FROM "Admin" WHERE "Id" = '${id}'`
              );
              if (role.rows.length > 0) {
                res.send({ success: true, role: "Admin" });
              } else {
                res.send({ success: true, role: "Basic" });
              }
            }
          }
        );
      } else {
        res.status(401).json({ message: "Unauthorized." });
      }
    }
  } catch (err) {
    console.error(err);
  }
});
```

4 ΑΞΙΟΛΟΓΗΣΗ

Ο αρχικός στόχος που είχε τεθεί ήταν να δημιουργηθεί μια ιστοσελίδα, που θα μπορεί να εξυπηρετεί τις ανάγκες των χρηστών ως προς την αναζήτηση ακίνητων και να μοιάζει στην ευκολία χρήσης στις υπόλοιπες σελίδες που υπάρχουν στο διαδίκτυο πάνω στον ίδιο τομέα. Αργότερα, έγινε κατανοητό πως αυτό δεν ήταν αρκετό, οπότε δοθήκαν δυνατότητες εισόδου-εξόδου (χρήστη), ανάρτησης αγγελιών και εκδήλωσης ενδιαφέροντος για ακίνητα. Από την άλλη πλευρά, υλοποιήθηκε η δυνατότητα εύκολου χειρισμού από την μεριά των διαχειριστών της ιστοσελίδας με την ύπαρξη admin account, ο οποίος μπορεί να διαγραφεί και να αποδέχεται αιτήματα. Ως κριτήριο επιτυχίας θεωρήθηκε η λειτουργία όλων των παραπάνω.

5 ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ

Η εισαγωγή δεδομένων πραγματοποιήθηκε στα αρχικά στάδια υλοποίησης με την βοήθεια του site: <https://www.mockaroo.com/> για testing, που προσφέρει αρκετές επιλογές για την παραγωγή διάφορων τύπων δεδομένων με ταχύτατο ρυθμό. Ενώ και το admin account έγινε εισαγωγή στην βάση με την εντολή INSERT της sql. Σε τελική φάση, η εισαγωγή των δεδομένων γίνεται αυτόματα μέσω της ιστοσελίδας με την βοήθεια του register, αλλά και της φόρμας για την εισαγωγή ακινήτων.

```
app.post("/register", async (req, res) => {
  try {
    const { email, username, password } = req.body;
    const checkEmail = await pool.query(
      `SELECT * FROM "Users" WHERE "email" = '${email}'`
    );
    if (checkEmail.rows.length > 0) {
      res.send({ success: false, message: "Email already exists." });
      return;
    }
    const checkUser = await pool.query(
      `SELECT * FROM "Users" WHERE "username" = '${username}'`
    );
    if (checkUser.rows.length > 0) {
      res.send({ success: false, message: "Username already exists." });
      return;
    }
    await bcrypt.hash(password, saltRounds, async (err, hash) => {
      if (err) {
        console.log(err);
      }
      await pool.query(
        `INSERT INTO "Users" ("email", "username", "password") VALUES ('${email}', '${username}', '${hash}');`
      );
      const id = await pool.query(
        `SELECT * FROM "Users" WHERE "username" = '${username}' AND "email" = '${email}'`
      );
      await pool.query(`INSERT INTO "Basic" ("Id") VALUES (${id.rows[0].Id});`);
      await bcrypt.hash(
        username + id.rows[0].Id,
        saltRounds,
        async (err, hash) => {
          if (err) {
            console.error(err);
          }
          tokens.push(id.rows[0].Id + hash);
          setExpiration(tokens, tokens.indexOf(id.rows[0].Id + hash), 1200000);
        }
      );
    });
  }
});
```

Figure 3: Register

```

app.post("/createListings", async (req, res) => {
  try {
    let { hash } = req.body;
    if (tokens.indexOf(hash) !== -1) {
      const id = hash.split("$")[0];
      hash = hash.replace(id, "");
      const user = await pool.query(
        `SELECT * FROM "Users" WHERE "Id" = '${id}'`
      );
      bcrypt.compare(
        user.rows[0].username + id,
        hash,
        async (err, response) => {
          if (response) {
            const {
              nameOfOwner,
              phoneOfOwner,
              province,
              town,
              placeInTown,
              address,
              toSell,
              squareMeters,
              description,
              price,
              typeOfProperty,
              numOfRooms,
              numOfFloors,
            } = req.body;

            await pool.query(
              `INSERT INTO "Request" ("Pending","Id_basic") values (true,${id})`
            );

            const id_request = await pool.query(
              `SELECT * FROM "Request" WHERE "Id_basic" = '${id}' and "Pending"=true ORDER BY "Id" DESC`;
            );
          }
        }
      );
    }
  }
});

```

Figure 4: Εισαγωγή ακινήτων

6 ΚΥΡΙΕΣ ΕΝΕΡΓΕΙΕΣ

Στην αρχή δεν υπήρχε μεγάλη οικειότητα με το συγκεκριμένο θέμα. Οπότε χρειάστηκε η διεξοδική αναζήτηση και έρευνα για τον τρόπο λειτουργίας ενός website Real-Estate. Συγκεκριμένα παρατίθενται κάποια ενδεικτικά link που βοήθησαν στην κατανόηση του θέματος:

<https://www.akinitapatras.gr>

<https://www.spitogatos.gr/>

Υστέρα έγινε έρευνα στον τρόπο δημιουργίας responsive ιστοσελίδων, αλλά και έρευνα στην δημιουργία ασφαλούς και αποτελεσματικού back-end.

Ο τρόπος εργασίας ήταν η συνάντηση από την 4^η εβδομάδα και μετά, μέσω κλήσης, ανά 3-4 μέρες. Η εργασία πραγματοποιούνταν παράλληλα στο περιβάλλον της κλήσης, κατά συνέπεια δεν υπήρξε συγκεκριμένος διαχωρισμός των τομέων του project. Υπήρχε συνεχής αλληλεπίδραση μεταξύ των μελών σε κάθε κομμάτι της εργασίας.

7 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

- 4^η-7^η εβδομάδα → Έρευνα πάνω σε ήδη υπάρχοντες ιστοσελίδες Real-Estate. Πρώτα από όλα, πραγματοποιήθηκε ενασχόληση με html και CSS για την δημιουργία της βασικής δομής των σελίδων. Ενώ από την συνάντηση με τους καθηγητές στην ενδιάμεση παρουσίαση, έγινε αντιληπτό πως το αρχικό πλάνο δεν ήταν αρκετό και χρειαζόταν επιπλέον πρόσθεση δυνατοτήτων και λειτουργιών (είσοδος - έξοδος (χρήστη), ανάρτησης αγγελιών και εκδήλωσης ενδιαφέροντος για ακίνητα, αλλά και admin account).
- 8^η εβδομάδα → Δημιουργία του erd και μεταφορά του σε relation model.
- Πάσχα-11^η εβδομάδα → Χρήση της JavaScript για την επίτευξη λειτουργικότητας, αλλά και χρήση της node.js και express.js για την σύνδεση της βάσης με τον server και το Front-end.
- 12^η εβδομάδα → Ολοκλήρωση και τελειοποίηση των παραπάνω και σύνταξη της έκθεσης (αναφοράς) και δημιουργία του power point.