**Τελική Ομαδική Εργασία**

**Προγραμματισμός Διαδικτύου**

Μουτάκης Δημήτρης ΑΜ:1031535

Νάσιος Στέφανος ΑΜ:1020048

6ο Έτος

Θέμα: Εφαρμογή online κρατήσεων ξενοδοχείου

Hotel Nadia

**Προβλήματα και μέθοδοι επίλυσής τους**

Το πρόβλημα που κλήθηκε να αντιμετωπίσει η ομάδα μας στα πλαίσια της εργασίας εξαμήνου είναι όπως αναγράφεται και στον τίτλο η δημιουργία μιας εφαρμογής που πραγματοποιεί κρατήσεις σε ξενοδοχείο. Επιπλέον απαιτήσεις που θα πρέπει να πληροί αυτή η εφαρμογή είναι όπως υπαγορεύτηκε από τους διδάσκοντες:

* Ο ιστότοπος που θα δημιουργηθεί να είναι δυναμικός και να υποστηρίζεται από εφαρμογή εξυπηρετητή (backend).
* Να υπάρχουν λογαριασμοί χρηστών και η δυνατότητα για προσωποποιημένη περιήγηση.
* Να εξασφαλίζεται ένα ελάχιστο επίπεδο ασφάλειας των χρηστών.

Λαμβάνοντας ως δεδομένο αυτή την «εκφώνηση» το αρχικό σχέδιό μας ήταν να φτιάξουμε έναν ιστότοπο ο οποίος θα παρείχε δυνατότητες για παρουσίαση όλων των ειδών των παροχών του ξενοδοχείο και εκτός από κρατήσεις για δωμάτια και κρατήσεις για αίθουσες εκδηλώσεων , κρατήσεις σε τραπέζια του εστιατορίου του ξενοδοχείο , σε γήπεδο τέννις και σε υπηρεσίες σπα . Ωστόσο μετά από το πέρας της πρώτης φάσης και από υποδείξεις των διδασκόντων αποφασίσαμε ότι θα ήταν πιο συνετό να περιοριστούμε στην επίλυση του υποπροβλήματος κρατήσεων δωματίων.  
Έτσι, τα προβλήματα (φάσεις κατασκευής) που συναντήσαμε και οι λύσεις που δώσαμε παρουσιάζονται παρακάτω:  
  
1η Φάση Εργασίας

Στην 1η φάση της εργασίας είχαμε να δημιουργήσουμε το front-end του site μας μέσω html, css και js. Ουσιαστικά έπρεπε να δημιουργήσουμε σχεδόν όλες τις σελίδες που θα έπρεπε να έχει η ιστοσελίδα μας.

Τι μορφή θα έχουν οι ιστοσελίδες ;

Για να καταλήξουμε σε μια συγκεκριμένη μορφή ψάξαμε για ιδέες σε ήδη υπάρχουσες ιστοσελίδες για ξενοδοχεία. Αυτή που μας άρεσε περισσότερο και αποφασίσαμε να υιοθετήσουμε στοιχεία της είναι η ιστοσελίδα του Intercontinental Athens. Συνεπώς καταλήξαμε σε δομή της μορφής :

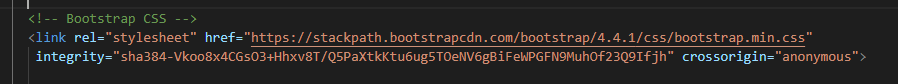
<header>  
<navbar>  
<main-body>  
<footer>  
  
Από σελίδα σε σελίδα αυτή η δομή άλλαζε ( π.χ. στην αρχική σελίδα υπάρχει και το <carousel>) αλλά χωρίς μεγάλες αλλαγές.

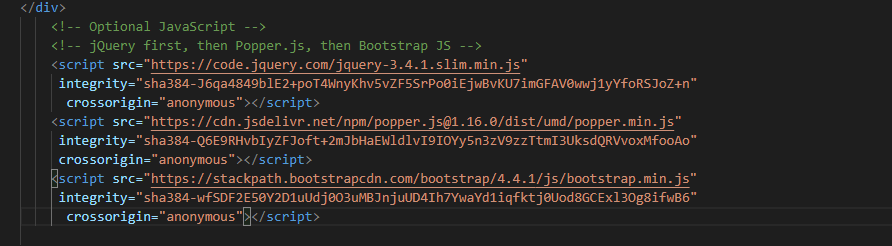
**Logo** και εικόνες χωρίς copyrights:

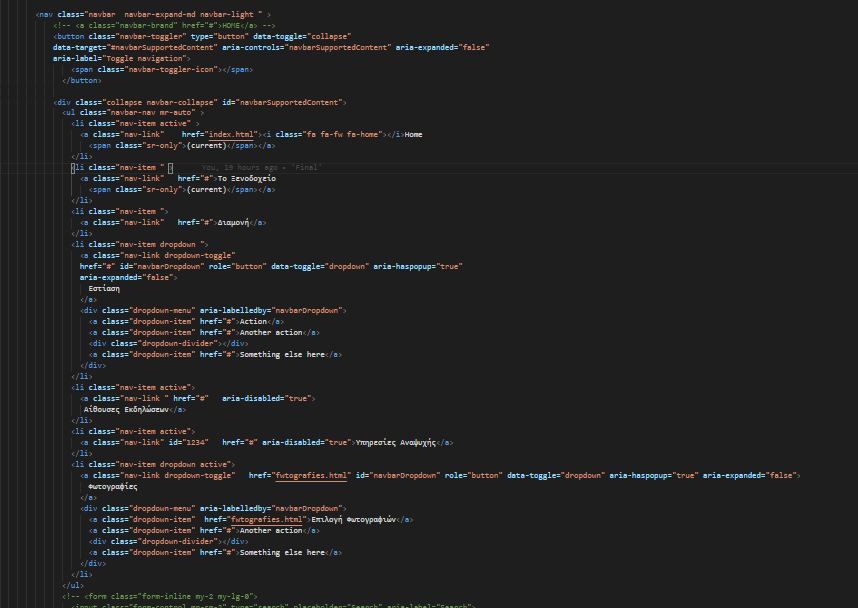
Για την δημιουργία logo χρησιμοποιήσαμε την online εφαρμογή <https://looka.com/onboarding> που σου επιτρέπει να σχεδιάσεις δωρεάν το logo σου ενώ εικόνες για την διακόσμηση του site πήραμε από την σελίδα Pexels.com.

Πώς θα ικανοποιούνται οι απαιτήσεις του responsive design ;

Είναι αρκετά σημαντικό η ιστοσελίδα μας να μπορεί να προσαρμόζεται στο μέγεθος της οθόνης της συσκευής που την προβάλει όπως μας επισημάνθηκε πολλές φορές και από τους διδάσκοντες. Όσο αφορά αυτό το ζήτημα μας έλυσε τα χέρια σε πολλά σημεία η Bootstrap. Κάναμε τις απαραίτητες αναφορές στις βιβλιοθήκες της Bootstrap:





Ακόμη αξιοποιήσαμε το online documentation που υπάρχει στην σελίδα της Bootstrap αλλά και τα παραδείγματα χρήσης του στην σελίδα W3schoolsTutorials απ’ όπου και αντλήσαμε μέχρι και κομμάτια κώδικα που κατασκευάζουν σε μεγάλο βαθμό βασικά δομικά στοιχεία του site όπως navbar , carousel , sticky-top header. Κάποια παραδείγματα είναι τα παρακάτω:   
navbar:  


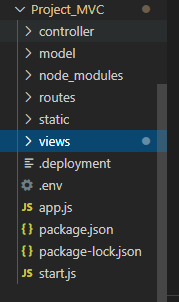
Επίσης οι δυνατότητες για διάταξη της σελίδας σε rows και 12 columns αλλά και η αλλαγή του ποσού των columns με την βοήθεια break-points που εισάγονται ως τιμές στα attributes class των στοιχείων μας βοήθησε πολύ στο να είναι προσαρμοστική.   

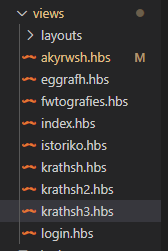

2η φάση της εργασίας

Στην 2η φάση της εργασίας είχαμε να φτιάξουμε και το back-end του ξενοδοχείου μας, το οποίο είναι και το πιο δύσκολο κομμάτι του project.

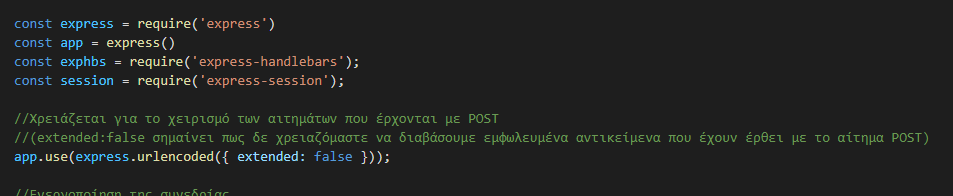
Πώς θα μετατρέψουμε τα μεμονωμένες σελίδες HTML σε δυναμικό ιστότοπο με δυνατότητες εξυπηρετητή ;

Για να το πετύχουμε αυτό αξιοποιήσαμε το μοντέλο MVC model – views – controller στο οποίο μας εισήγαγαν οι διδάσκοντες. Ότι σελίδες html είχαμε δημιουργήσει μέχρι αυτό το στάδιο τροποποιήθηκαν σε σελίδες handlebars:

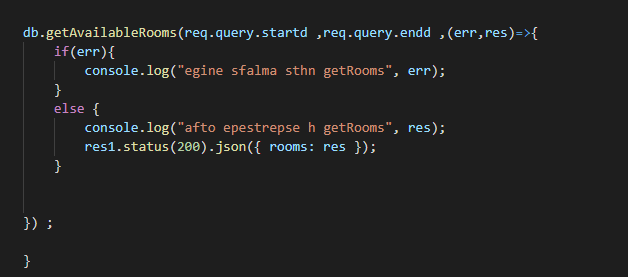
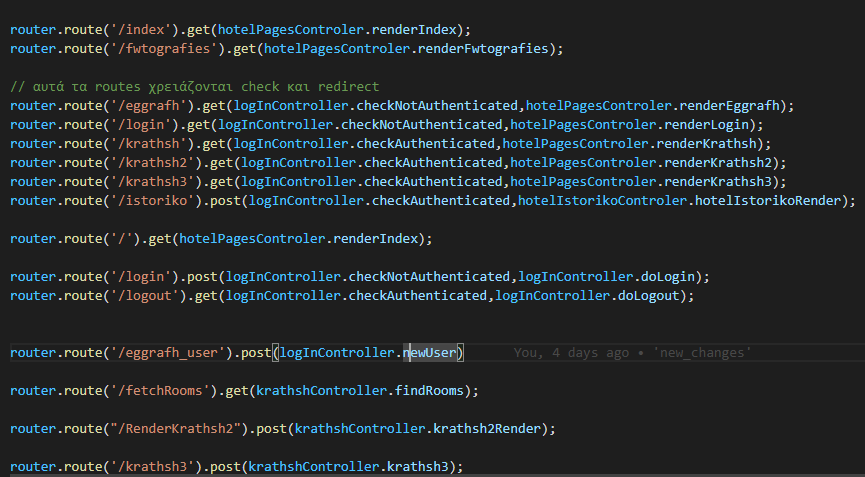




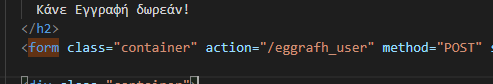
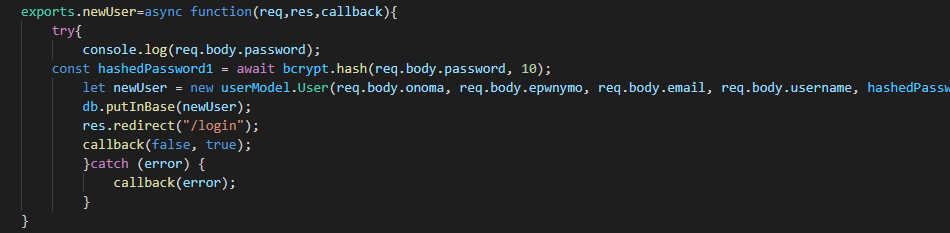
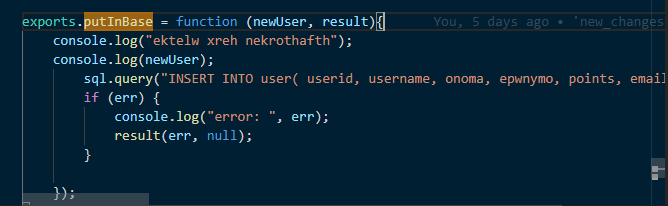
Και στην ουσία αυτά είναι που καταλήγουν στον client.   
Επίσης κάνοντας import το node-module express δημιουργήσαμε το αντικείμενο app στο οποίο ορίζουμε να «περιμένει» http-requests σε ένα συγκεκριμένο port και αναλόγως το request να γίνεται triggered το κατάλληλο route και να καλείται ο αντίστοιχος handler από τον φάκελο του controller και να στέλνει το επιθυμητό response στον client.

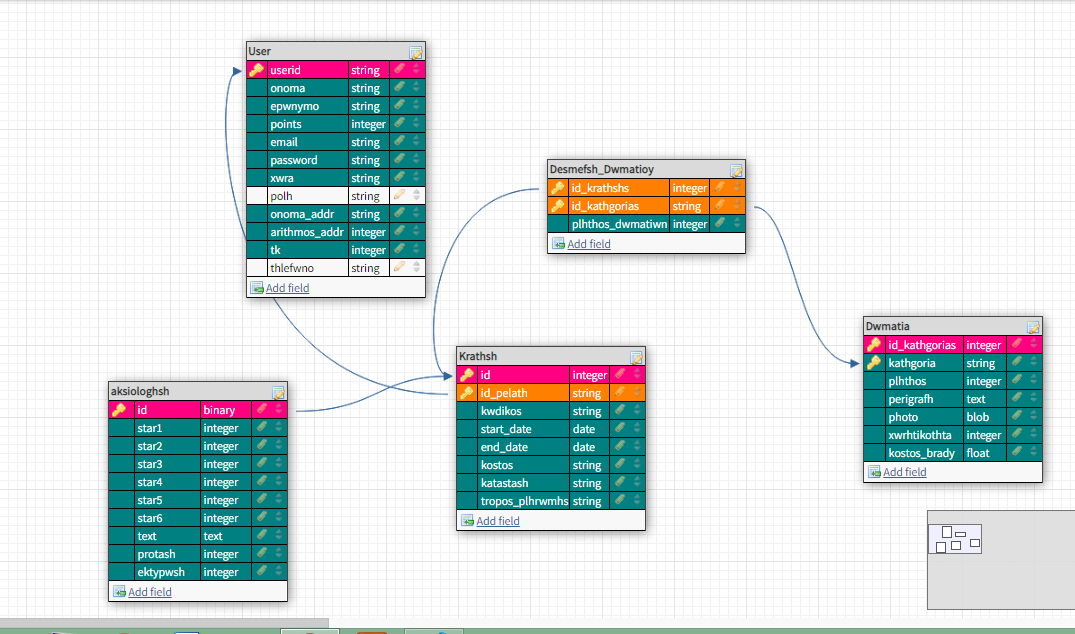




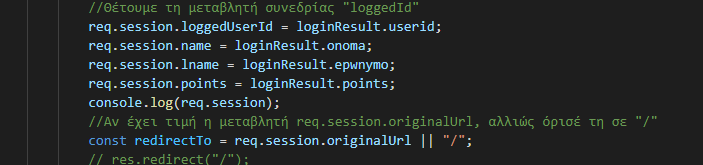
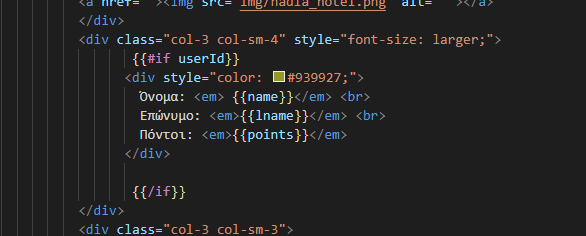
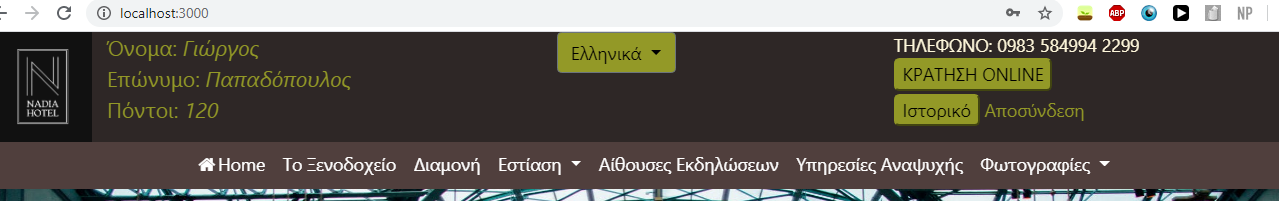


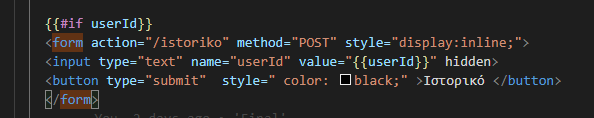
Πώς θα υπάρχει δυνατότητα για δημιουργία λογαριασμού χρηστών και για log in στην εφαρμογή μας ;

Για να επιλυθεί αυτό το ζήτημα φτιάξαμε μια σελίδα που περιέχει μια <form> εγγραφής για τους χρήστες. Αυτή η <form> :  
  
αφού την συμπληρώσει ο χρήστης και κάνει Submit , η εφαρμογή καλεί τον αντίστοιχο handler :  
  
ο οποίος εκτελεί την συνάρτηση:   
  
Σ’αυτό σημείο τονίζουμε ότι το πρόβλημα ασφάλειας του ότι δεν πρέπει να είναι αποθηκευμένο στην βάση το password έτσι όπως το εισήγαγε ο χρήστης αλλά κρύπτογραφημένο το επιλύσαμε χρησιμοποιώντας την hash-function του module bcrypt. Δηλαδή στην βάση αποθηκεύεται το hashed ισοδύναμο του password που εισήγαγε ο χρήστης και αντίστοιχα κατά την ενέργεια του log in γίνεται η αντίστοιχη σύγκριση.   
Στη συνέχεια καλείται συνάρτηση όπου εκτελεί το INSERT στην βάση και γίνεται ανακατεύθυνση του χρήστη στην σελίδα για να κάνει log in.   


Επίσης με την χρήση βάσης δεδομένων επιλύθηκε το ζητούμενο :  
  
 Που θα είναι αποθηκευμένα τα στοιχεία του χρήστη μετά την εγγραφή για log in και τα υπόλοιπα χρήσιμα για την εφαρμογή δεδομένα ;   
  
Λύση σε αυτό το πρόβλημα αποτέλεσε η δημιουργία μια βάσης δεδομένων για την εφαρμογή μας με την τεχνολογία MySQL:  


  
Αυτός ο τρόπος αποθήκευσης δεδομένων αποδείχτηκε πολύ χρήσιμος διότι με κατάλληλα queries έχουμε την δυνατότητα να ανακτούμε δεδομένα από την βάση και να τα χρησιμοποιούμε για να λειτουργούν ομαλά διάφορες δυνατότητες της εφαρμογής μας όπως:  
- εμφάνιση διαθέσιμων δωματίων   
- εμφάνιση παλαιότερων κρατήσεων του χρήστη κτλπ.  
αλλά και εκτός από αυτά μέσω του remote-hosting να μπορεί η βάση μας να είναι 24/7 διαθέσιμη και να έχει σταθερή πρόσβαση σ’αυτήν η εφαρμογή μας όταν γίνει το deployment.

Πως γίνεται η περιήγηση προσωποποιημένη ;   
 Αυτό το καταφέρνουμε χρησιμοποιώντας το αντικείμενο session στα request μας και περνώντας τιμές στις μεταβλητές res.locals στις οποίες έχουν πρόσβαση οι σελίδες handlebars και τροποποιούν το περιεχόμενό τους ανάλογα με τον συνδεδεμένο χρήστη.   
Πιο συγκεκριμένα όταν ο χρήστης κάνει επιτυχημένο log in θέτουμε κατάλληλες μεταβλητές συνεδρίας :  
  
 τις οποίες χρησιμοποιούμε στις σελίδες handlebars   
  


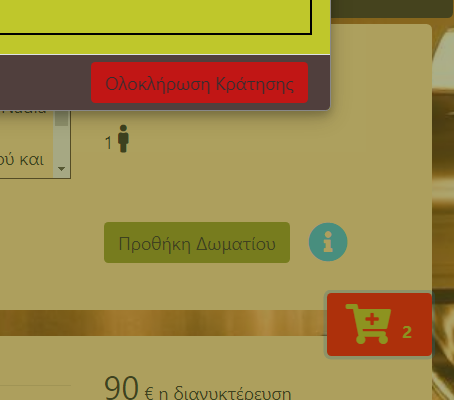
Πώς εξασφαλίζεται το ότι ένας χρήστης έχει πρόσβαση μόνο στα δικά του δεδομένα ;  
  
Πατώντας το κουμπί Ιστορικό o χρήστης μπορεί να δει παλαιότερες κρατήσεις του και να τροποποιήσει τρέχουσες . Ενώ αρχικά αυτή η ανάκτηση δεδομένων γινόταν εισάγοντας ένα query στο URL γρήγορα καταλάβαμε ότι ένας χρήστης θα μπορούσε να εισάγει ένα τυχαίο id και να αποκτήσει πρόσβαση σε δεδομένα άλλου χρήστη. Γι’ αυτό αποφασίσαμε ότι η μετάβαση στην σελίδα ιστορικού θα γίνεται με την βοήθεια ενός hidden text-input και μιας <form method=”POST”> :  


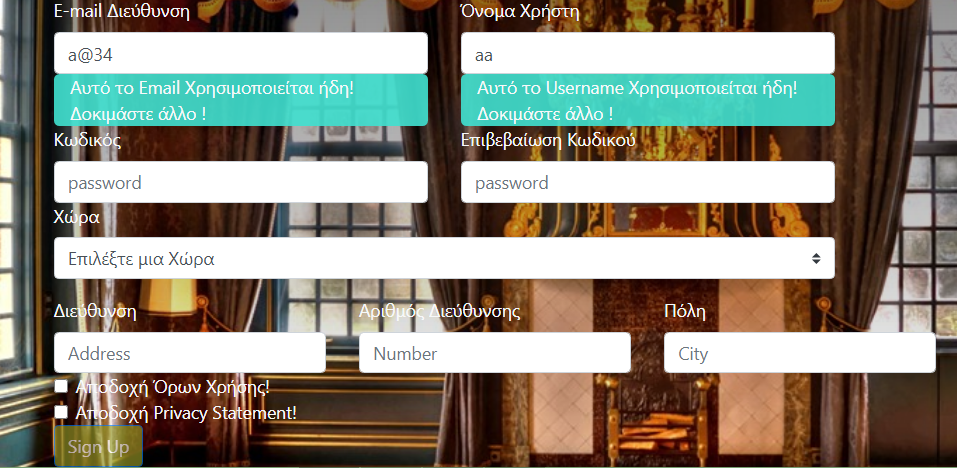
Πως υλοποιείται η λειτουργία πραγματοποίησης κράτησης ;  
  
Η υλοποίηση συνοψίζεται στα εξής βήματα:  
1) Πραγματοποίηση του log in  
2) Μετάβαση στην σελίδα krathsh.hbs και συμπλήρωση αριθμού ατόμων και επιθυμητών ημερομηνιών  
3) Μετάβαση στην σελίδα krathsh2.hbs όπου ο χρήστης βλέπει επιπλέον πληροφορίες για τα δωμάτια και επιλέγει όποια επιθυμεί  
4) μετάβαση στην σελίδα krathsh4.hbs για επιβεβαίωση στοιχείων και ολοκλήρωση κράτησης   
Το πιο αξιοσημείωτο προγραμματιστικά μέρος αυτής της διαδικασίας είναι το πώς θα διατηρήσουμε τα στοιχεία που εισήγαγε ο χρήστης από την σελίδα 2 στην 3 και πως μετά θα περάσουν στον controller και στην βάση για να ολοκληρωθεί η κράτηση.   
Αυτό το θέμα το αντιμετωπίσαμε με την βοήθεια hidden text-inputs όπου και στο πεδίο value αποθηκεύουμε ότι εισάγει ο χρήστης ποιο συγκεκριμένα αυτό στον κώδικα μας φαίνεται στις μεταβλητές epilegmena\_dwmatia , epilegmena\_posa (= πόσα δωμάτια από κάθε κατηγορία επέλεξε) , epilegmena\_kosth . Έτσι στην ουσία η κράτηση του χρήστη αποθηκεύεται σε κάποια stings τα οποία με την βοήθεια μεθόδων .join(‘,’) , .split(‘,’) τα επεξεργαζόμαστε για να τους δώσουμε την μορφή που μας είναι χρήσιμη κάθε στιγμή , είτε array για να μπουν ως όρισμα στην συνάρτηση που γράφει δεδομένα στην βάση , είτε ένα μεγάλο string για να περάσει ως τιμή σε μεταβλητή συνεδρίας και να είναι διαθέσιμο μετά την εναλλαγή σελίδας από κράτηση2 σε κράτηση3 .   
  
Πως θα τηρούνται οι παραδοχές λειτουργίας της εφαρμογής ;

Αρχικά έπρεπε να ορίσουμε κάποιες παραδοχές για να κάνουμε πιο συγκεκριμένο το πρόβλημα «δημιουργία εφαρμογής κρατήσεων ξενοδοχείου». Οι σημαντικότερες από αυτές είναι:

* Οι κρατήσεις γίνονται τουλάχιστον 2 μέρες μετά από την ημέρα που ο χρήστης χρησιμοποιεί την εφαρμογή ώστε να υπάρχει περιθώριο μια μέρα για ακύρωση κράτησης
* Κρατήσεις για μέχρι 10 άτομα
* Οι κρατήσεις πρέπει να περιέχουν αρκετά δωμάτια ώστε να μπορούν να στεγάσουν των αριθμό ατόμων που ορίστηκε αλλά και ένα μέγιστο όριο 10 δωματίων
* Η κράτηση γίνεται οριστική και δεν μπορεί να ακυρωθεί 24 ώρες πριν την ημέρα CHECK IN που έχει οριστεί από τον χρήστη.

Για να τηρηθούν αυτές οι παραδοχές και να ελέγχουμε σε ποια πλαίσια κινείται ο χρήστης όταν χρησιμοποιεί την εφαρμογή εφαρμόσαμε διάφορα τεχνάσματα όπως για παράδειγμα:

Disabled κουμπί καλαθιού μέχρι να ορίσει αρκετά δωμάτια ο χρήστης :  
  
  
  
  
Έλεγχοι μέσω fetch αν το username που εισήγαγε ο χρήστης ήδη χρησιμοποιείται :

  
τα οποία υλοποιούνται με τον κώδικα:  
  
  
  
Εβδομάδα παράτασης Bonus φάση.

Σε αυτή την φάση έχοντας τελειώσει με τις βασικές λειτουργίες του site προσπαθήσαμε να προσθέσουμε και μια επιπλέον λειτουργία.

Πως θα πραγματοποιηθεί η λειτουργία της αξιολόγησης ;

Από άποψη back-end δεν γίνεται κάτι διαφορετικό απ’ ότι έχει γίνει για τις προηγούμενες λειτουργίες. Όπως και στην εγγραφή έτσι και εδώ ο χρήστης συμπληρώνει μια φόρμα και με το κουμπί καταχωρούνται δεδομένα στην βάση. Στην συνέχεια οι αξιολογήσεις εμφανίζονται μέσω link στο footer της αρχικής σελίδας, σε όλους τους χρήστες, παίρνοντας δεδομένα από την βάση.

**Οδηγίες εγκατάστασης**

**# Hotel Nadia**

Πρώτα κατεβάστε τα πακέτα node με `npm install`.

Τρέχει με `npm run debug` αφού εκτελεστούν πρώτα τα βήματα για την βάση σειρά 46.

Το πρόγραμμα ξεκινάει απο το `start.js` δηλώνοντας το port 3000 .

Το `app.js` ακολουθεί την λογική MVC.

Το πρόγραμμα μοιράζεται σε διάφορα τμήματα:

 - `/model` παρέχει την πρόσβαση στα δεδομένα.

 - `/controller` περιέχει συναρτήσεις για να έχουμε πρόσβαση στα δεδομένα που μας δίνει το model.

 Οι φάκελοι `views` και `routes`:

 - Ο `/views` περιέχει template γραμμένα σε handlebars.

 - Ο `/routes` περιέχει τις διαδρομές που αναγνωρίζει η εφαρμογή.

Υπάρχουν αρκετές διαδρομές:

1. Στη βασική **\*\*( διαδρομή "/")\*\***: θα φορτωθεί το view `index.hbs`. Σε κάθε χρήστη εμφανίζεται η αρχική

σελίδα του ξενοδοχείου. Είναι η μοναδική διαδρομή που συνιστάται να διαλέξει ο χρήστης γιατί μέσω αυτής θα έχει πρόσβαση σε όλες τις άλλες διαδρομές.

**#### Το μοντέλο:**

Ο φάκελος `/models` περιέχει το μοντέλο mysql για την αποθήκευση των δεδομένων:

- `hotel-model-mysql.js`, διαβάζει,καταχωρεί,προσθέτει και διαγράφει δεδομένα από μια βάση MySQL.

-  `user-model.js`, διαβάζει τα δεδομένα από μια βάση MySQL.

Τα στοιχεία πρόσβασης για την βάση βρίσκονται στο αρχείο `model/db.mysql.js`.

**#### Ο controller:**

Στο αρχείο `/controller/hotel-istoriko-controller.js` περιέχονται οι συναρτήσεις χειρισμού του μοντέλου. Η βασική διαδρομή που τον χρησιμοποιεί είναι η **\*\*"/istoriko"\*\***. Υπάρχουν και άλλες διαδρομές για περισσότερες λειτουργίες όπως **\*\*"/tropopoihsh"\*\*** , **\*\*"/akyrwsh"\*\*** , **\*\*"/aksiologish"\*\*** .

Το ίδιο και στο αρχείο `/controller/hotel-page-controller.js`. Η βασική διαδρομή που τον χρησιμοποιεί είναι η **\*\*"/"\*\*** και απο το υπόλοιπο site μέσω **\*\*"/index"\*\***.

Το ίδιο και στο αρχείο `/controller/krathsh-controller.js`. Η βασική διαδρομή που τον χρησιμοποιεί είναι η **\*\*"/krathsh"\*\***. Υπάρχουν και άλλες διαδρομές που φορτώνουν άλλες σελίδες όπως **\*\*"/krathsh2"\*\*** , **\*\*"/krathsh3"\*\*** .

Το ίδιο και στο αρχείο `/controller/log-in-controller.js`. Η διαδρομή που τον χρησιμοποιεί είναι η **\*\*"/login"\*\***.

Σε κάθε controller χρησιμοποιήται οποιοδήποτε μοντέλο χρειαστούμε, διαλέγοντας κατάλληλα model στην κορυφή του αρχείου, π.χ.:

```javascript

const model = require('../model/hotel-model-mysql.js');

```

**#### H βάση MySQL:**

Χρησημοποιήθηκε τοπική βάση mysql με την βοήθεια του προγράμματος XAMPP.

Απαιτείται κατέβασμα του XAMPP.

Στην συνέχεια πρέπει να πατηθεί στα actions των module: Apache και MySQL, start.

Αφού τρέξει μετά απο λίγα δευτερόλεπτα θα χρειαστεί να πατηθεί το Admin του module: MySQL και θα ανοιξει μια σελίδα στον browser.

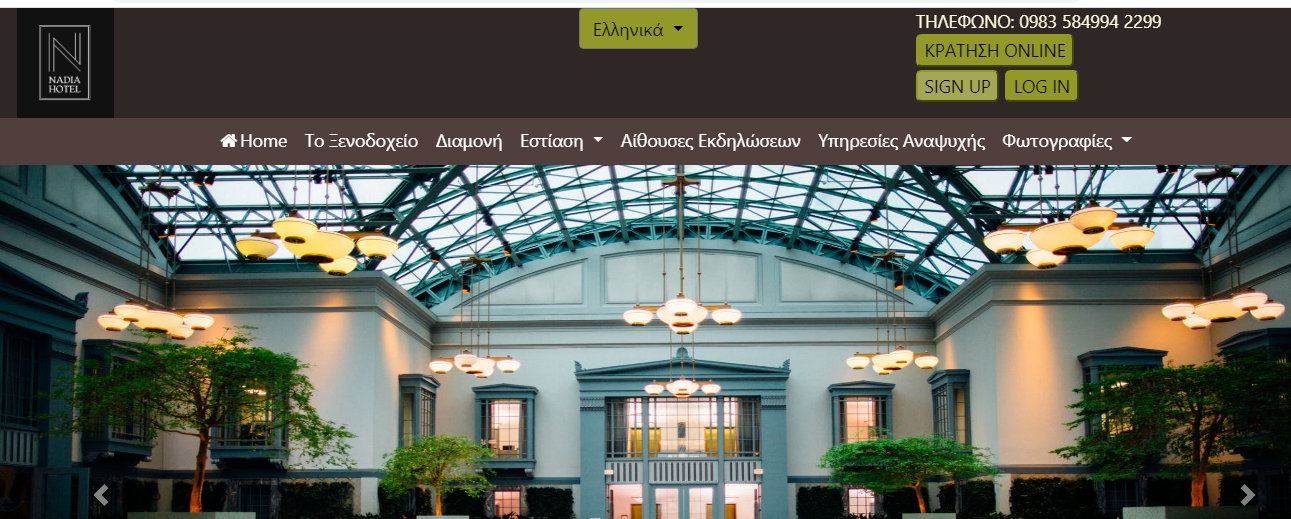
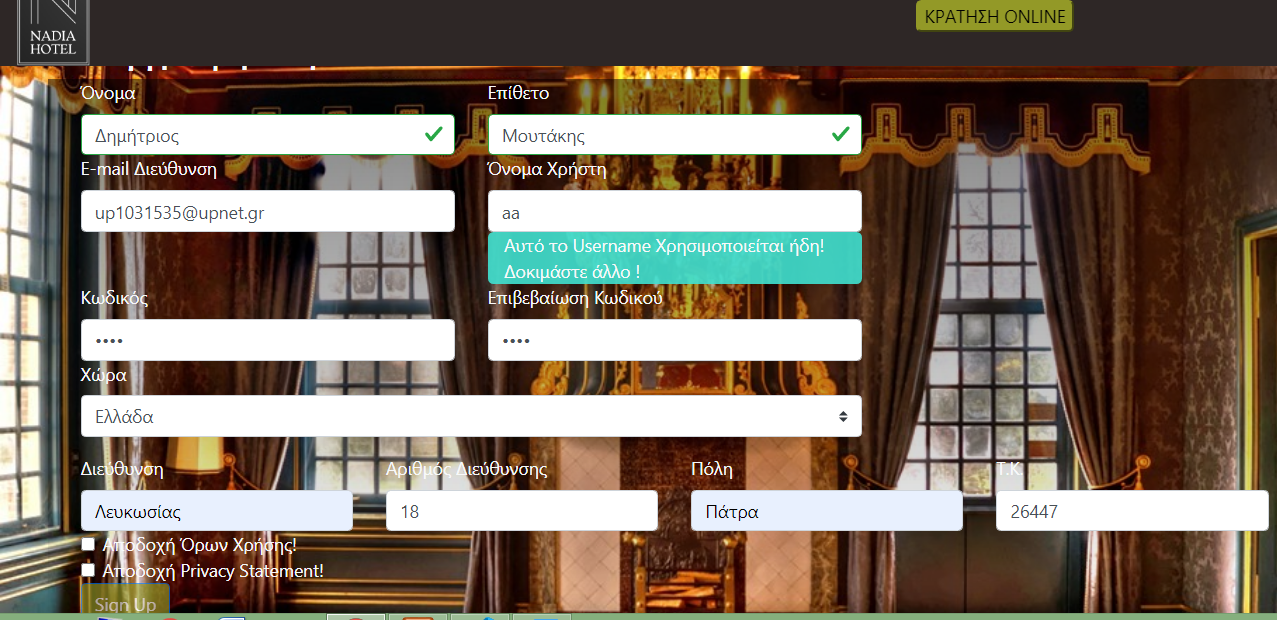
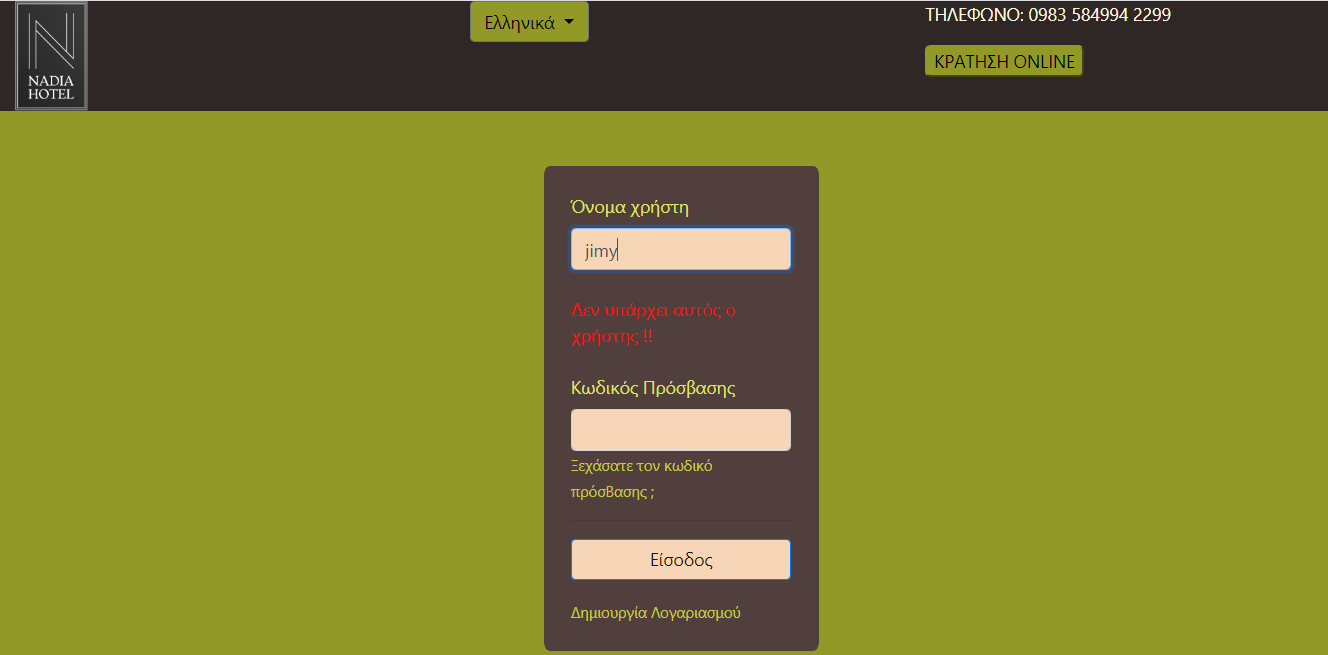
Μέσω της σελίδας που άνοιξε θα πρέπει να δημιουργηθεί ένας φάκελος με ονομα nadia και ένας υποφάκελος με όνομα nadia\_hotel.

Πατόντας στο nadia\_hotel στο κομπί Κώδικας MySQL προστήθεται ο κώδικας του αρχείου `nadia\_hotel2.sql` με copy paste και στην συνέχεια Εκτέλεση.

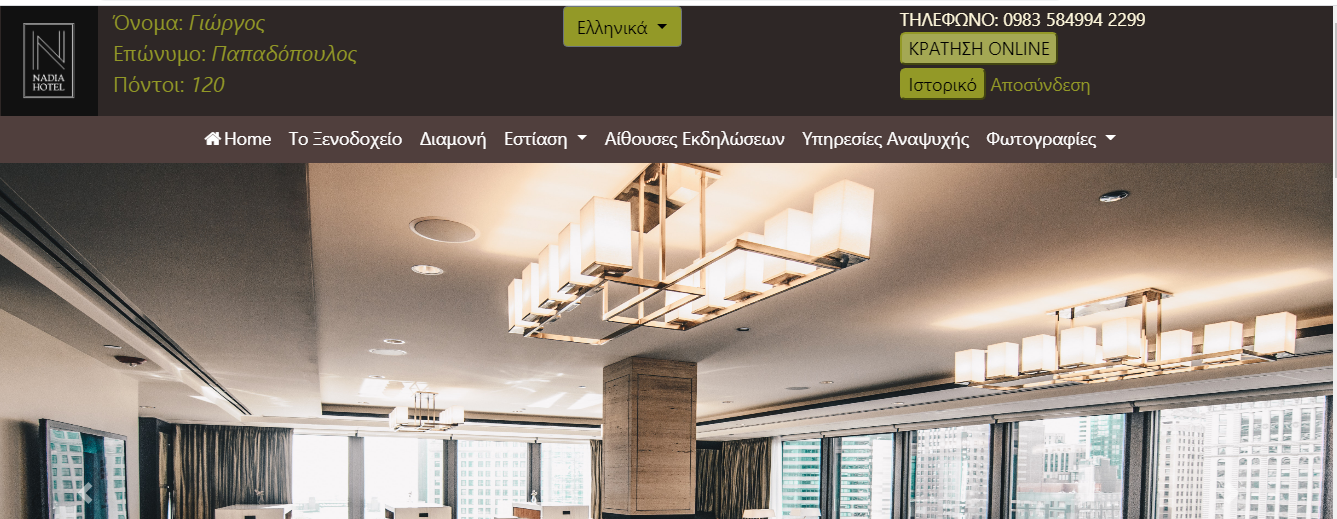
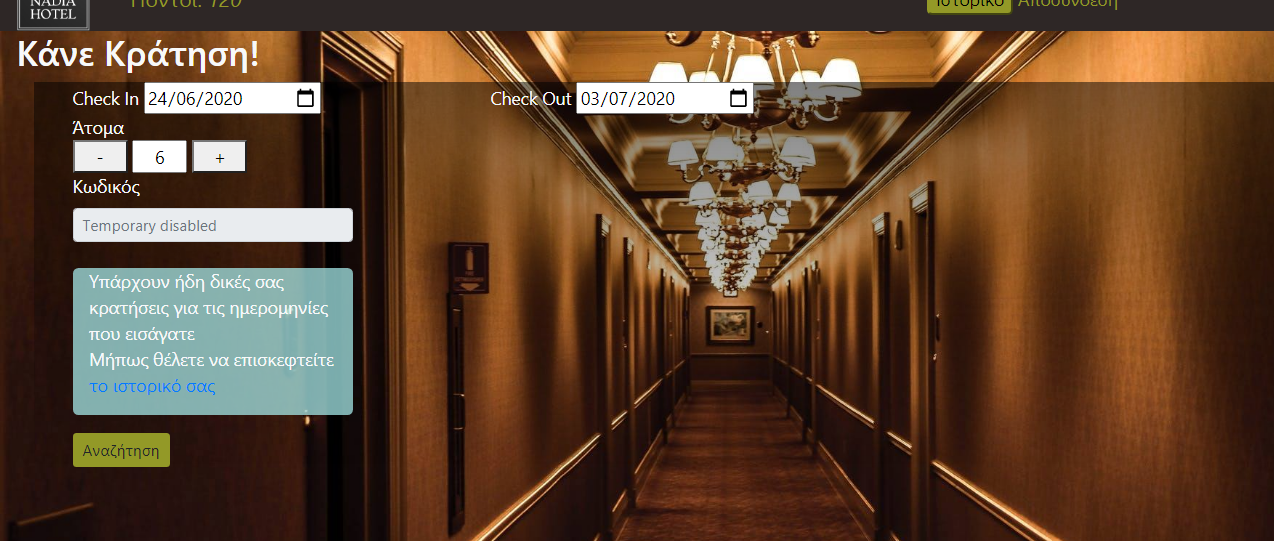
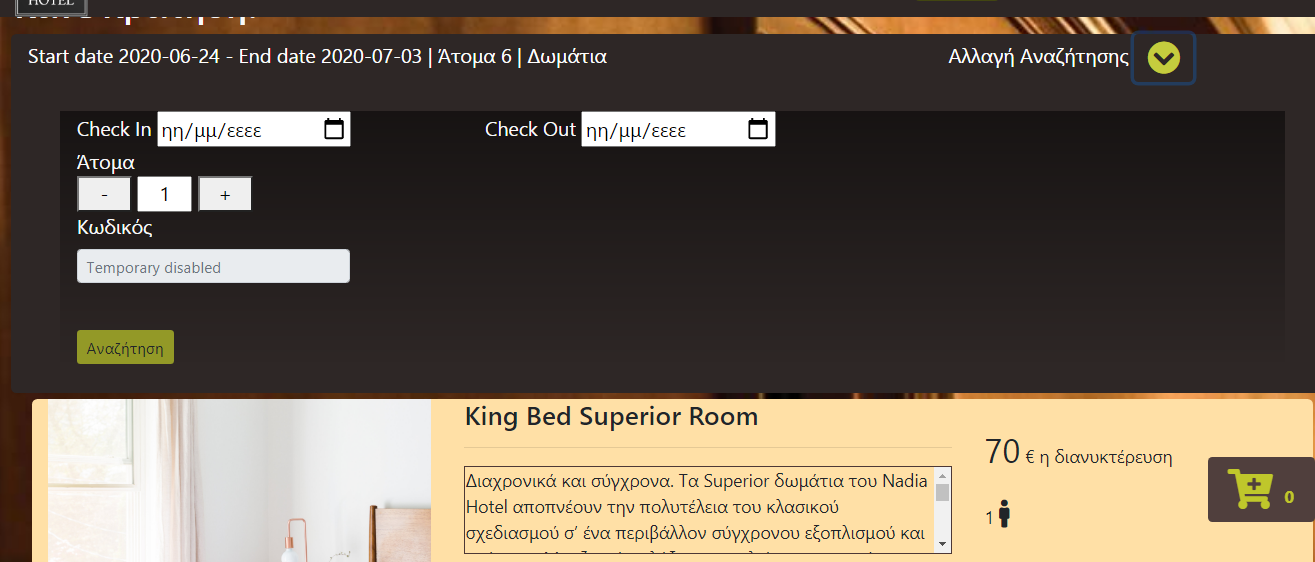
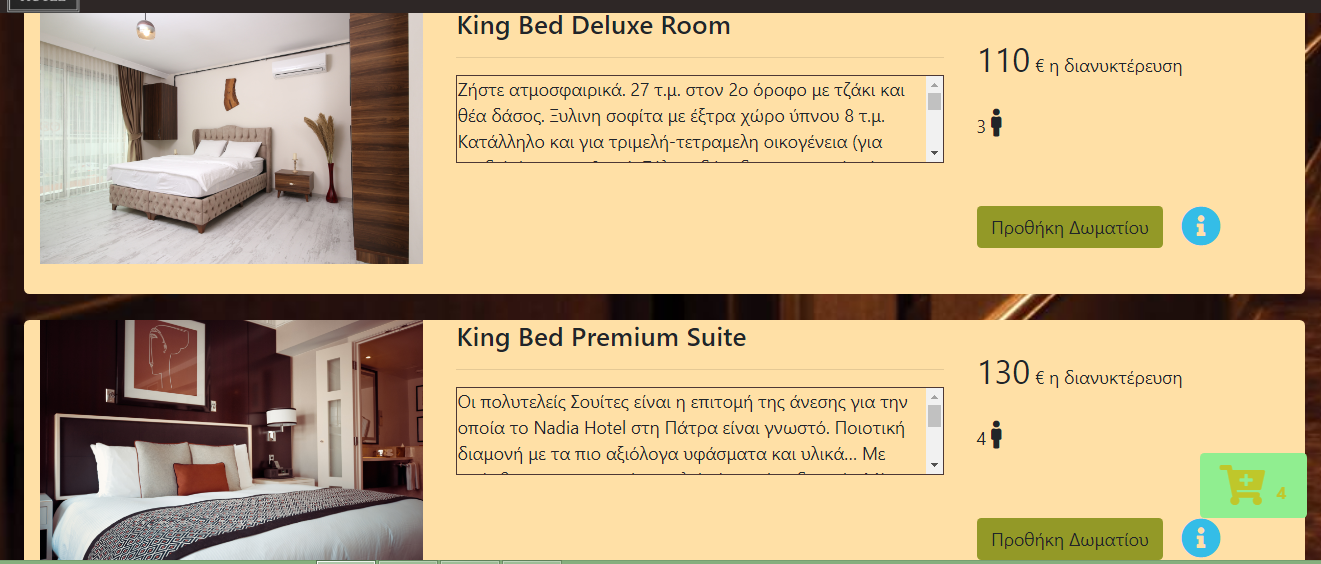
**Παραδείγματα χρήσης προγράμματος**

Ο χρήστης μπορεί να χρησιμοποιήσει την εφαρμογή μας για να κάνει τα εξής:

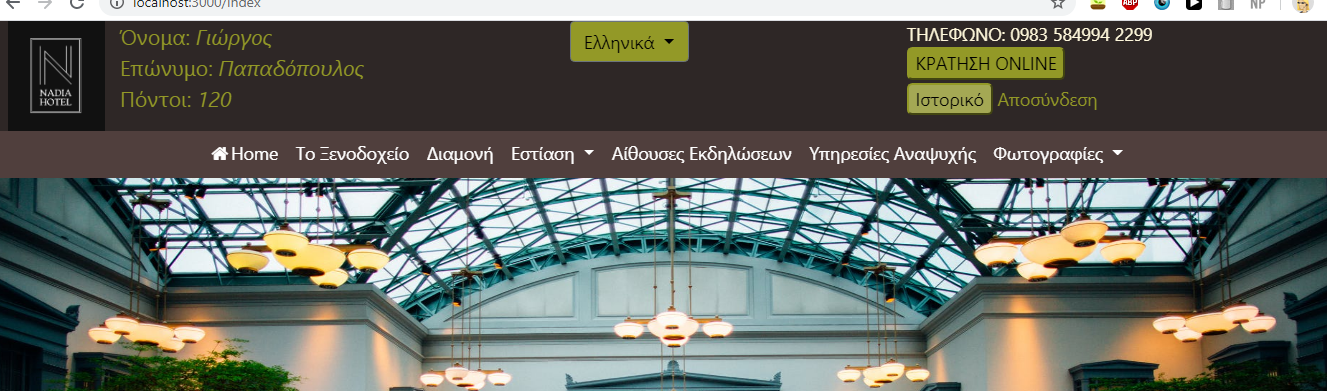
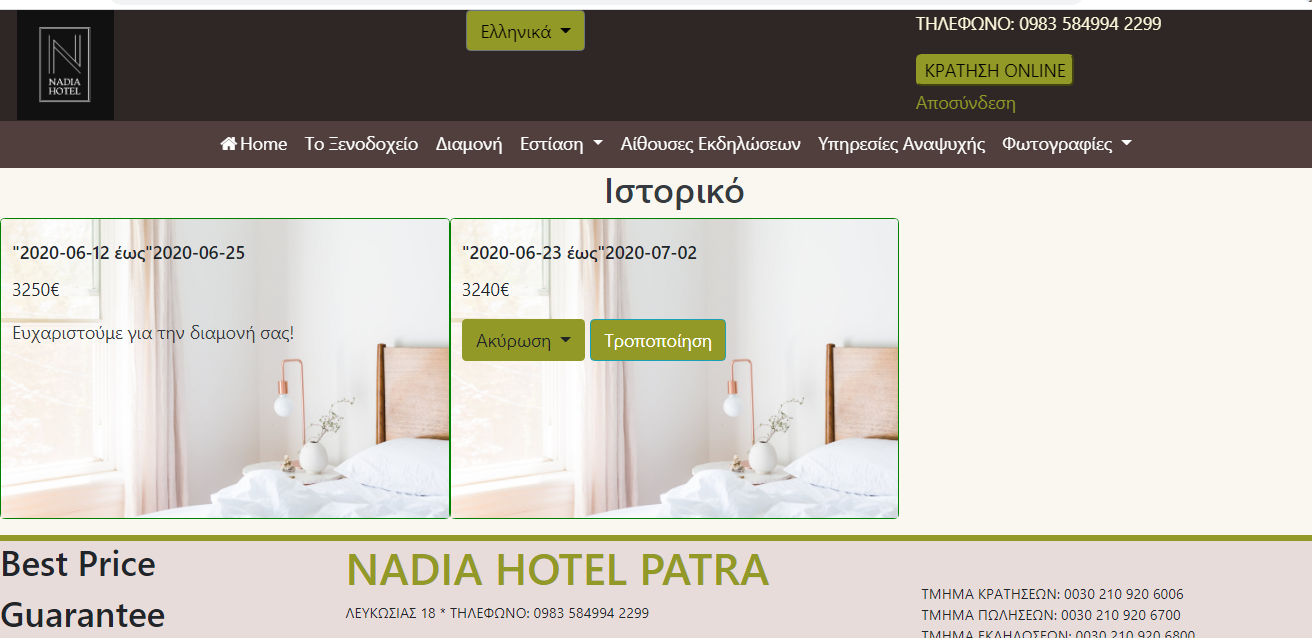
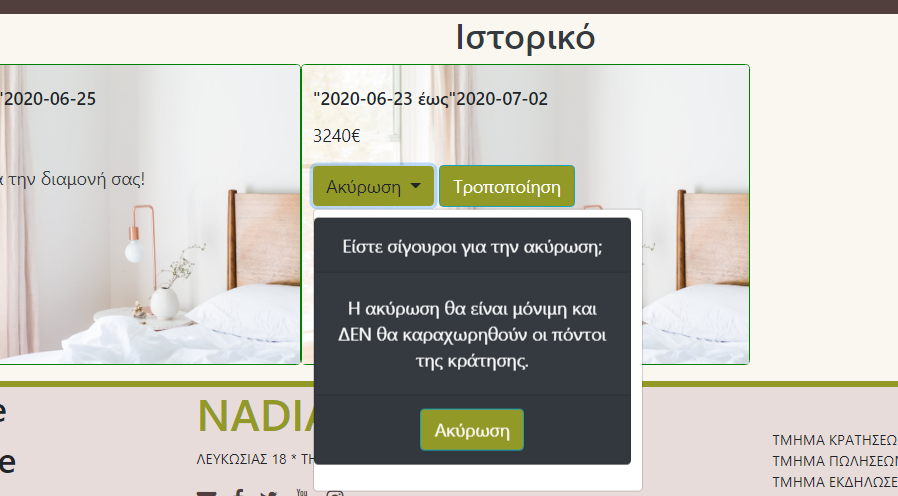
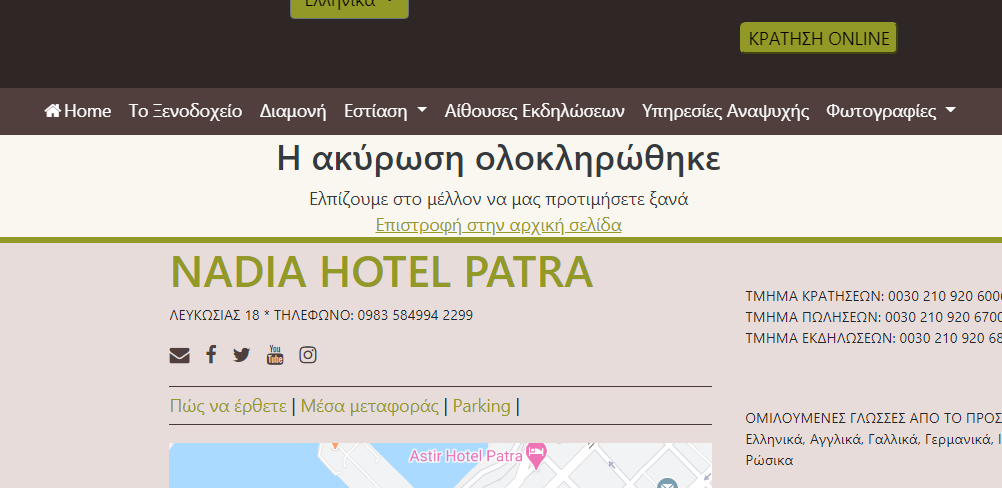
1. Εγγραφή και log in

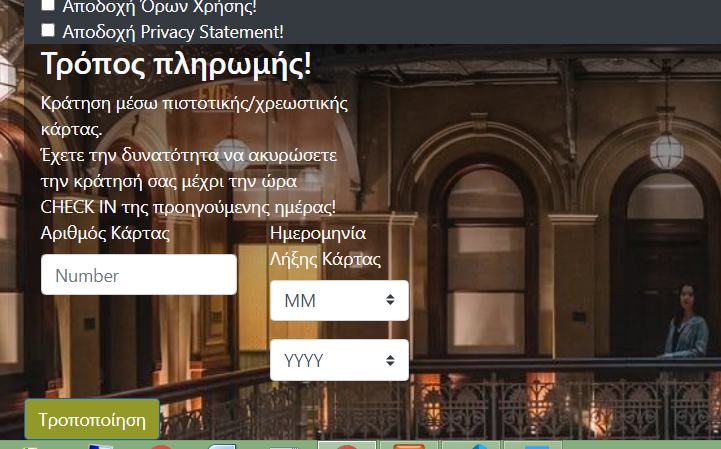
Από την αρχική σελίδα που του εμφανίζεται (index.hbs)  
  
 ο χρήστης πατώντας το κουμπί sign up οδηγείται στην σελίδα eggrafh.hbs που περιέχει φόρμα συμπλήρωσης στοιχείων λογαριασμού  
  
Αφού εισάγει έγκυρα στοιχεία και πατήσει το κουμπί Sign Up μετά το σύστημα τον ανακατευθύνει στην σελίδα log-in.hbs   
  
Μόλις εισάγει σωστά τα στοιχεία του επανέρχεται στην αρχική σελίδα απ’ όπου και μπορεί πλέον να έχει πρόσβαση στις άλλες 3 σημαντικές λειτουργίες της εφαρμογής μας.   
Σημειώνεται πως αν έχει ήδη λογαριασμό πατάει απευθείας στο κουμπί Log-in της αρχικής σελίδας , εισάγει τα στοιχεία του και συνεχίζει κανονικά.

1. Πραγματοποίηση κράτησης

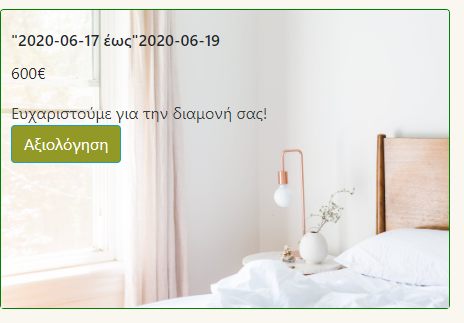
Από την αρχική σελίδα ο χρήστης πατώντας το κουμπί κράτηση online   
  
οδηγείται στην σελίδα krathsh.hbs όπου και χρειάζεται να συμπληρώσει τις ημερομηνίες που τον ενδιαφέρουν και τον αριθμό των ατόμων για τα οποία σκοπεύει να κρατήσει δωμάτια  
  
Έπειτα με το κουμπί αναζήτηση προχωράει στην σελίδα krathsh2.hbs όπου και επιλέγει τα δωμάτια της προτίμησής του με βασικό περιορισμό να μπορούν να στεγάσουν τον αριθμό των ατόμως που έχει ορίσει στην προηγούμενη σελίδα. Επίσης σ’αυτή την σελίδα έχει και δυνατότητα να ορίσει εκ νέου αριθμό ατόμων και ημερομηνίες και να του εμφανιστούν εκ νέου τα διαθέσιμα δωμάτια:  
  
  
  
  
  
  
Με το κουμπί ολοκλήρωση κράτησης ο χρήστης πηγαίνει στην σελίδα κράτηση 3 όπου στα αριστερά υπάρχει μια φόρμα συμπληρωμένη για επιβεβαίωση τον στοιχείων του και εισαγωγή τρόπου πληρωμής. Ακόμη δεξιά υπάρχουν στοιχεία της κράτησης όπως το συνολικό κόστος και δυνατότητα να ζητήση έκπτωση εφόσον έχει το απαραίτητο ποσό πόντων στον λογαριασμό του. Πατώντας το κουμπί κράτηση που βρίσκεται κάτω αριστερά ολοκληρώνεται η κράτηση.

1. Προβολή Ιστορικού παλαιότερων και τρεχόντων κρατήσεων

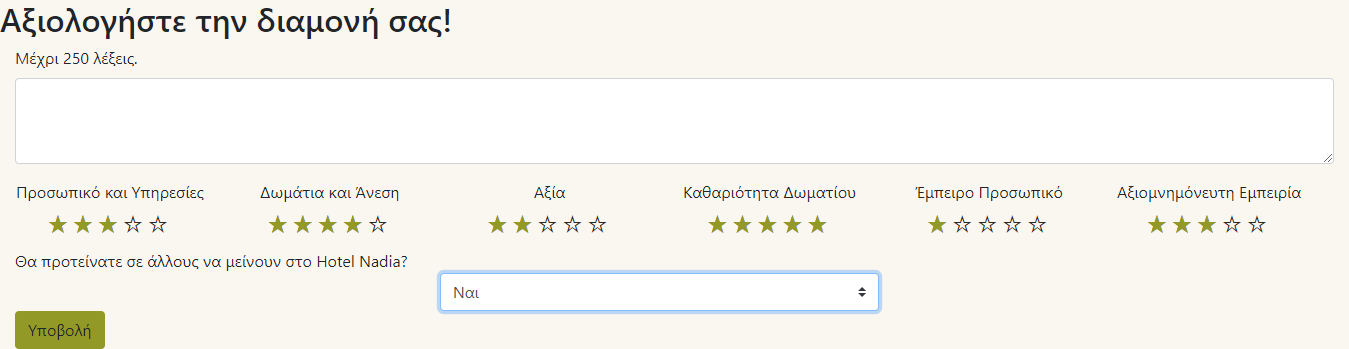
Από την αρχική σελίδα ο χρήστη έχει και την εναλλακτική διαδρομή να πατήσει το κουμπί ιστορικό το οποίο θα τον κατευθύνει στην σελίδα istorik.hbs από την οποία έχει την δυνατότητα να ακυρώσει ή τροποποιήσει τρέχουσες κρατήσεις που δν έχουν οριστικοποιηθεί και να δει/αξιολογήσει παλαιότερες ολοκληρωμένες κρατήσεις.   
  
  
  
  
Επιλέγοντας την ακύρωση εμφανίζεται στον χρήστη κατάλληλο μήνυμα για το μη αναστρέψιμο της ενέργειας που πρόκειται να κάνει και στην συνέχεια οδηγείται σε μια σελίδα με λινκ για επιστροφή στην αρχική σελίδα.   
  
  
  
  
Διαφορετικά ο χρήστης έχει και την δυνατότητα τροποποίησης της κράτησής του . Αφού πατήσει το αντίστοιχο κουμπί μεταπηδά στην 2η σελίδα των κρατήσεων , ακολουθεί την διαδικασία που αναλύθηκε παραπάνω με βασική διαφορά ότι στο τελευταίο βήμα του εμφανίζεται το κουμπί τροποποίηση . Τέλος μπορεί να ξαναδεί την τροποποιημένη κράτησή του στο ιστορικό .



Στην περίπτωση που η κράτηση βρίσκεται σε παρελθοντικό χρόνο ο χρήστης μπορεί να αξιολογήσει την εμπειρία του στο ξενοδοχείο.



Αφού πατήσει το κουμπί μεταφέρεται στην σελίδα και πρέπει να συμπληρώσει μια φόρμα για την αξιολόγηση.



Πατώντας υποβολή επιστρέφει τον χρήστη στην αρχική σελίδα.

Σε αντίθεση με την συμπλήρωση αξιολόγησης, οι εκτυπωμένες αξιολογήσεις εμφανίζονται σε όλους τους χρήστες μέσω του footer.



Εμφανίζονται μόνο οι αξιολογήσεις που έχουν ελεγχθεί από κάποιον που έχει πρόσβαση στην βάση mysql και μπορεί να τροποποιήσει την μεταβλητή ektipwsh, ιδανικά ένας admin.