

ΤΕΧΝΟΛΟΓΙΕΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΙΚΤΥΟΥ

Project – Full stack development με RestFul APIs

ΟΜΑΔΑ 55

Παναγιώτα Ψυχάρη 1115201500184

Κωνσταντίνος Βουτσίνος 1115201200016

BACK END

<https://localhost:8443>

Database

Πριν την εκτέλεση του ολικού προγράμματος (Back και Front) απαιτείται η εκτέλεση κώδικα sql που θα δημιουργήσει το database της εφαρμογής. Το αρχείο βρίσκεται στον φάκελο **Backend/Database**. Με το αρχείο **AuctionWebApp_CD.sql** δημιουργείται η βάση και διαγράφονται οποιεσδήποτε προηγούμενες εγγραφές. Παράλληλα δημιουργείται ένας λογαριαμός χρήστη που χρησιμεύει στο connection μεταξύ βάσης και Spring Boot. Το αρχείο **AuctionWebApp_DD.sql** διαγράφει εντελώς την βάση και τον χρήστη.

Η βάση αποτελείται από τους εξής πίνακες:

- **users**
Table για τους χρήστες του site (απλούς χρήστες και admin).
- **items**
Table για τα αντικείμενα προς δημοπρασία. Περιέχονται βασικές πληροφορίες αντικειμένου, όπως από ποιόν αγοράστηκε (αν αγοράστηκε), τον πωλητή του, αν είναι ενεργή δημοπρασία κτλπ.
- **categories**
Table με ονόματα κατηγοριών. Στην βάση περιέχονται default τιμές.
- **item_category**
Πίνακας συσχέτισης μεταξύ των tables items και categories λόγω της “Many to Many” σχέσης τους.
- **bids**
Table με bids που έχουν τεθεί από κάποιον χρήστη για κάποιο αντικείμενο. Αν ένας χρήστης έχει κάνει bid σε ένα αντικείμενο πάνω από μία φορά, κάθε προσφορά αποτελεί καινούργιο row στον πίνακα.
- **photos**
Table που περιέχει εικόνες των αντικείμενων. Αποθηκεύονται ολόκληρες στην βάση σε μορφή BLOB.
- **locations**
Table με πληροφορίες που αφορούν την τοποθεσία που έχει ορίσει ένας χρήστης για τον εαυτό του (registration) και για αντικείμενα που έχει δημιουργήσει.
- **ratings**
Table που περιέχει το rating που έχει θέσει ένας χρήστης σε έναν άλλον. Γενικά σε κάθε χρήστη στην συγκεκριμένη υλοποίηση δίνεται η δυνατότητα να κάνει rate έναν άλλον χρήστη μόνο μία φορά.

- **messages**

Table με τις πληροφορίες των μηνυμάτων που ανταλλάσσονται μεταξύ των χρηστών (bidder με seller και αντίστροφα). Ένα μήνυμα δεν διαγράφεται από την βάση τελείως. Αν ένας χρήστης επιλέξει διαγραφή από το inbox ή outbox του, διαγράφεται μόνο για αυτόν με κατάλληλη ένδειξη σε πεδίο του πίνακα.

Rest API

Για την υλοποίηση του Back end έχει χρησιμοποιηθεί Spring Framework (Java). Κάθε αντικείμενο οντότητα της άσκησης, ως κλάση, διαθέτει τον δικό της Controller, Service και Repository. Σύμφωνα με τα Restful APIs, ο κώδικας διαρθρώνεται ως εξής:

- **Controllers**

Κλάσεις που περιέχουν την επικοινωνία του Back End με το Front End. Δέχονται αιτήματα http (GET, POST, DELETE, PUT) και απαντούν σε αυτά σύμφωνα με τις πληροφορίες που αποκτούν από την επικοινωνία με τους με το Service layer.

- **Services**

Κλάσεις που αναπαριστούν το Service layer μεταξύ του Controller layer και της βάσης (MySQL).

- **Repositories**

Κλάσεις που δίνουν την άμεση δυνατότητα στο Service layer να επικοινωνήσει με την βάση. Βασίζονται στην λογική των CRUD Repositories. Πραγματοποιούν ερωτήματα στην βάση.

Για τους σκοπούς της εργασίας έχουν δημιουργηθεί συγκεκριμένες κλάσεις αντικειμένων που αντιπροσωπεύουν τις οντότητες της βάσης.

- **Entities**

Αντικείμενα που αντιπροσωπεύουν πίνακες της βάσης. Είναι ουσιαστικά η αντικειμενοστραφής αναπαράσταση των αποθηκευμένων οντοτήτων. Χρησιμοποιούνται για την λήψη πληροφοριών από την βάση και την αποθήκευσή τους σε αυτή.

- **DTOs**

Αντικείμενα που χρησιμοποιούνται για την μεταφορά πληροφορίας από layer σε layer.

ΠΑΡΑΤΗΡΗΣΕΙΣ:

- Τα υπόλοιπα αντικείμενα χρησιμοποιούνται για ειδικού σκοπούς (πχ επιστροφή στον χρήστη συγκεκριμένων πληροφοριών από κάποιο αίτημά του για κάποιο item ή άλλο χρήστη).
- Όταν ο admin επιλέξει την εξαγωγή των items σε XML ή JSON, τα αρχεία θα αποθηκευτούν στα paths **BackEnd/XMLS** και **BackEnd/JSONS** αντίστοιχα.

FRONT END

<https://localhost:4200>

Για το Front End έχει χρησιμοποιηθεί Javascript σε Angular 7. Για την δομή και εμφάνιση των pages χρησιμοποιείται Bootstrap.

Μεταξύ των αρχείων του παραδοτέου διατίθεται ο φάκελος src που περιέχει τον κώδικα του Front End του site. Για την εκτέλεσή του, στον φάκελο της επιλογής σας δημιουργείτε καινούργιο angular project και στην θέση του default src φακέλου προσθέστε τον φάκελο src που σας δίνεται.

ΠΡΟΣΟΧΗ:

Για να εκτελεστούν ορισμένα features των σελίδων χρειάζεται η εγκατάσταση ορισμένων modules στο working directory που θα επιλέξετε (τα modules θα αποθηκευτούν στον default φάκελο *node-modules*):

- `npm install --save @angular/material @angular/cdk`
- `npm install --save ngx-tags-input`
- `npm install --save ngx-chips`
- `npm install --save @ng-bootstrap/ng-bootstrap`

Το περιεχόμενο του site διαρθρώνεται σε :

- **Components**
Αρχεία κλάσεων που αντιπροσωπεύουν ξεχωριστά pages. Τα pages δομούνται με βάση Html και CSS κώδικα και η διαχείριση των data γίνεται μέσω Typescript.
- **Services**
Οι κλάσεις αυτές αναπαριστούν το layer που στέλνει αιτήματα στο Back End (τα οποία αναλαμβάνουν οι Controllers).
- **Routings**
Περιέχουν πληροφορίες σχετικά με την αντιστοίχιση urls με components.
- **Models**
Αντιπροσωπεύουν αντικείμενα που χρησιμοποιούνται στην διάρθρωση του site και στην παροχή και χρήση των πληροφοριών που στέλνονται και αποκτούνται αντιστοίχως από το Back End (User, Item, Message κτλπ).

ΠΑΡΑΤΗΡΗΣΕΙΣ:

- Έχει χρησιμοποιηθεί στον front τακτική η οποία να αποτρέπει έναν χρήστη να έχει πρόσβαση σε σελίδες στις οποίες δεν έχει δικαίωμα να μεταβεί. Η λειτουργία αυτή βρίσκεται στο `auth_gaurd.ts` και παρατηρείται και στα αρχεία των routings (user-routing και admin-routing)
- Όταν ένας χρήστης συνδεθεί με επιτυχία στον λογαριασμό του έχει στην διάθεση του ένα JWT token που αντιπροσωπεύει το session του και του δίνει την δυνατότητα να πληγηθεί στον λογαριασμό του.
- Αν τυχόν υπάρξει κάποιο πρόβλημα με τον server (πχ έχει διακοπεί η λειτουργία του Spring Boot) ο χρήστης αποσυνδέεται αυτόματα.