Τίτλος

Ανάπτυξη Εφαρμογής Ηλεκτρονικών Δημοπρασιών στον Παγκόσμιο Ιστό

Συμμετέχοντες

- Ομάδα Χρηστών 94

- Γιώργος – Μάριος Πατέλης
- A.M : 1115201300140

- Ευθύμιος Πατέλης
- A.M : 1115201300141

Περιεχόμενα

Εισαγωγή

Η παρούσα εργασία έχει ως στόχους :

1. Την ανάπτυξη μιας εφαρμογής για την δημοπράτηση αντικειμένων στον Παγκόσμιο Ιστό

2. Την εξοικείωση και την χρήση συγχρόνων και δημοφιλών εργαλείων της βιομηχανίας Ανάπτυξης Εφαρμογών Διαδικτύου

3. Την εξοικείωση με τις κοινές πρακτικές της βιομηχανίας Ανάπτυξης Εφαρμογών Διαδικτύου

Τα κεφάλαια που ακολουθούν παρουσιάζουν και αναλύουν τις **σημαντικότερες** πτυχές της διαδικασίας ανάπτυξης της εφαρμογής τόσο από την «οπτική γωνιά» του Client αλλά και από αυτή του Server

Παραδοχές

1. Στην παρεχόμενη εκφώνηση δεν αναφέρεται ρητά το αν μια δημοπρασία αποτελείται από ένα ή πολλαπλά αντικείμενα ενώ τα παρεχόμενα δεδομένα (E-Bay Data) υποδεικνύουν ότι κάθε δημοπρασία αποτελείται μόλις ένα αντικείμενο.

   Την χρονική στιγμή που αποσαφηνίστηκε στο πλαίσιο του μαθήματος ότι μια δημοπρασία ενδέχεται να αποτελείται από ένα η περισσότερα αντικείμενα είχαν υλοποιηθεί πολλαπλά τμήματα του Client και του Server ενώ είχε οριστικοποιηθεί η μορφή της υποκείμενης βάσης δεδομένων .

   Η συγκεκριμένη τροποποίηση θα είχε ως αποτέλεσμα τον μερικό ανασχεδιασμό της εφαρμογής το οποίο δεν ήταν δυνατό λόγω χρονικών περιορισμών ενώ δημιουργήθηκαν και κάποια αλλά σημαντικά ερωτήματα όπως :

   - Ένα αντικείμενο διαθέτει προαιρετικά μια ή περισσότερες φωτογραφίες. Σχεδιάζοντας την δημοπρασία ως μια δέσμη αντικειμένων τι θα πρέπει να επιλεχθεί για την παρουσίαση της δημοπρασίας στην μεριά του Client;

   - Ο πλειοδότης δεσμεύεται να κάνει προσφορά σε όλα τα αντικείμενα της δημοπρασίας ή στα όποια ο ίδιος επιθυμεί;

   - Ο δημιουργός της δημοπρασίας θα πρέπει να κάνει μια μικρό-διαχείριση των επιμέρους αντικειμένων;

   Τα παραπάνω λειτουργήσαν συμπληρωματικά ως προς την συγκεκριμένη σχεδιαστική επιλογή / παραδοχή η οποία επιτρέπει στον χρήστη να δημιουργήσει **έμμεσα** μια δημοπρασία με πολλαπλά αντικείμενα κάτι το οποίο φαίνεται να συμβαίνει και σε παρόμοιες πλατφόρμες όπως Amazon, E-Bay, κ.ο.κ.

2. Η εκφώνηση αναφέρει ότι θα αποθηκεύεται **προαιρετικά** μία ή περισσότερες φωτογραφίες ενός αντικειμένου.
   Κατά την περίπτωση που ο χρήστης δεν διατίθεται να προσφέρει φωτογραφικό υλικό στο πλαίσιο της εκάστοτε δημοπρασίας τότε διατίθεται από τον Server μια placeholder φωτογραφία με απώτερο σκοπό την συνεπή παρουσίαση των δημοπρασιών στον Client.

3. Για την υλοποίηση του Matrix Factorization έπρεπε να επινοηθεί ένα σύστημα που να αποτυπώνει τις προτιμήσεις των χρηστών.
   Η εκφώνηση αναφέρει ότι βάσει του ιστορικού των συναλλαγών και των δημοπρασιών που έχει επισκεφτεί ο χρήστης, σε κατάλληλο τμήμα της διεπαφής θα προβάλλονται προτεινόμενα αντικείμενα με τη χρήση αλγορίθμου Matrix Factorization.
   Το σύστημα που υλοποιήθηκε αποτυπώνει τις διαδράσεις του χρήστη με την εκάστοτε δημοπρασία με τον παρακάτω τρόπο :

   - Δημιουργία δημοπρασίας - Χ
   - Οι διαδράσεις κάθε χρήστη - Υ με την Χ είναι αρχικά 0
   - Η επίσκεψη της Χ από την χρήστη Υ αυξάνει τις διαδράσεις Χ – Υ κατά 1
   - Η προσφορά για την Χ από τον χρήστη Υ αυξάνει τις διαδράσεις Χ – Υ κατά 1
   - Ο μέγιστος αριθμός διαδράσεων Χ – Υ είναι 10

   Τα σημεία (3) – (4) δεν εφαρμόζονται στον δημιουργό της δημοπρασίας μιας και αυτός είναι ο χρήστης που ενδέχεται να έχει τις περισσότερες διαδράσεις με την εκάστοτε δημοπρασία.

4. Η αναζήτηση των δημοπρασιών λειτουργεί με τον παρακάτω τρόπο :

   1. Αναζήτηση δημοπρασιών βάσει ελευθέρου κειμένου
   2. Αναζήτηση δημοπρασιών βάσει κατηγορίας
   3. Αναζήτηση δημοπρασιών συνδυαστικά βάσει των (1) – (2)
   4. Φιλτράρισμα των αποτελεσμάτων των (1) ή (2) ή (3) βάσει των κριτήριων :

      a. Εύρος τιμής
      b. Τοποθεσίας
      c. Ύπαρξη «Buy - Now» τιμής
      d. Συνδυασμός των (a) – (b) – (c)

5. Η εξαγωγή των δημοπρασιών από τον διαχειριστή λειτουργεί ως εξής :

   a. Επιλογή ημερομηνίας - Α
   b. Επιλογή ημερομηνίας – Β
   c. Εξαγωγή το πολύ 500 δημοπρασιών που αναρτήθηκαν στο διάστημα [Α — Β]

Client

Εκτέλεση

Προαπαιτούμενα :

- npm 8.12.2
- Node.js 16.15.1
- Angular 13.3.4

Εντολή εκτέλεσης Client :

- npm install – [Πριν την 1η εκτέλεση]
- npm i --save-dev @types/leaflet - [Πριν την 1η εκτέλεση]
- ng serve
    - Επιλογή Port : --port XXXX
    - Προσθήκη SSL : --ssl true --ssl-key ./ssl/localhost-key.pem --ssl-cert ./ssl/localhost.pem

Τεχνολογίες

- Angular
- Leaflet
- LocationIQ - Forward/Reverse Geocoding API
- Bootstrap
- CSS - Flexbox

Αρχιτεκτονική

Ακολουθείται το μοντέλο

Model – View – Controller

που παρέχεται από την Angular – αφού αποτελεί ένα υψηλά opinionated framework - ως εξής :

- Model

  Τα **περισσότερα** επιμέρους Services διαδραματίζουν τον παραπάνω ρολό καθώς ευθύνονται για την αναπαράσταση και τις CRUD λειτουργίες επί της υποκείμενης βάσης δεδομένων

- View

  Τα επιμέρους Components και **συγκεκριμένα** τα HTML – CSS πρότυπα/αρχεία διαδραματίζουν τον παραπάνω ρολό ενώ ευθύνονται για την παροχή δεδομένων από τον χρήστη καθώς και για την αναπαράσταση των τελευταίων για αυτόν

- Controller

  Τα επιμέρους Components και **συγκεκριμένα** τα TypeScript αρχεία διαδραματίζουν τον παραπάνω ρολό καθώς λειτουργούν ως μια διεπαφή αναμεσά στο View και στο Model και με αυτό τον τρόπο υλοποιούν την απαιτούμενη λειτουργικότητα της εφαρμογής

Προκλήσεις

- Ζητήματα Αισθητικής

  Η διεπαφή χρήστη ήταν θεμιτό να πληροί τα παρακάτω κριτήρια :

  1. Συστηματική εμφάνιση στην πλειοψηφία των περιπτώσεων
  2. Μοντέρνα αισθητική που να συμβαδίζει με αυτή των σύγχρονων εφαρμογών διαδικτύου

  Τα σημεία (1) - (2) αντιμετωπίστηκαν **κυρίως** με την χρήση του συστήματος πλέγματος της Bootstrap καθώς και την χρήση του συστήματος Flexbox

- Ζητήματα Feedback

  1. Κατά την διάδραση με την εφαρμογή παρατηρήθηκε ότι ήταν ιδιαίτερα σημαντική η παροχή feedback στον χρήστη με απώτερο σκοπό να αντιληφθεί ο τελευταίος τον αντίκτυπο της εκάστοτε πράξης του

  Αναφέρονται ενδεικτικά κάποια από τα μέσα που χρησιμοποιήθηκαν για την αντιμετώπιση του σημείου (1) :

  1. Dialogs
  2. Animations
  3. Color States

- Ζητήματα DRY

1. Συγκεκριμένες σελίδες της εφαρμογής είχαν την ανάγκη των **παρόμοιων** λειτουργιών με μια **ενδεχομένη** διαφοροποίηση ως προς την εμφάνιση αυτών στην διεπαφή χρήστη

2. Για τις περισσότερες διαδράσεις του χρήστη με την εφαρμογή έπρεπε να ελεγχθεί αν ο τελευταίος διαθέτει τα απαραίτητα δικαιώματα

Τα σημείο (1) αντιμετωπίστηκε με τον σχεδιασμό και την υλοποίηση των κατάλληλων Components π.χ. όπως αυτό του χάρτη της εφαρμογής

Τα σημείο (2) αντιμετωπίστηκε με την χρήση των Angular Route Guards

Server

Εκτέλεση

Προαπαιτούμενα :

- JAVA - 18.0.1.1
- Apache Maven - 3.8.6

Εντολή εκτέλεσης Server : mvn spring-boot:run

Στο φάκελο database του Server βρίσκεται το αρχείο Database – Dump.sql με το οποίο είναι δυνατό να δημιουργηθεί η υποκείμενη βάση δεδομένων η οποία περιλαμβάνει έναν χρήστη με το ρόλο του Administrator με τα παρακάτω στοιχεία :

- Username : admin
- Password : 1234

Είναι **υψίστης σημασίας** η υποκείμενη βάση δεδομένων να πληροί τις προδιάγραφες του αρχείου application.properties που βρίσκεται στο φάκελο resources του Server

Τεχνολογίες

- Spring Boot
- Spring Security
- Apache Maven
- MySQL
- Hibernate

Αρχιτεκτονική

Ακολουθείται το μοντέλο

Controller → Service → Repository

με σκοπό των διαχωρισμό των επιμέρους λειτουργιών ως εξής :

- Repository

  Το εκάστοτε Repository συνδέεται αυστηρά με μια οντότητα στην υποκείμενη βάση δεδομένων ενώ ευθύνεται για την ανάκτηση των δεδομένων από αυτή και την αποθήκευση αυτών σε αυτή.

- Service

  Το εκάστοτε Service με την χρήση των κατάλληλων Repositories προσφέρει την υλοποίηση των απαιτούμενων λειτουργιών ανεξαρτήτως της πολυπλοκότητας των τελευταίων όπως :

  - Εγγραφή χρήστη
  - Δημιουργία δημοπρασίας
  - Αποστολή μηνύματος

- Controller

  Το εκάστοτε Controller με την χρήση των κατάλληλων Services συνδέει το εκάστοτε API End - Point με την κατάλληλη λειτουργικότητα.

Προκλήσεις

- Ζητήματα Συγχρονισμού

1. Ένα αριθμός χρηστών X κάνει μια προσφορά σε μια δημοπρασία ταυτόχρονα.

2. Ένας πλειοδότης - X υποβάλλει μια προσφορά για την δημοπρασία - Y ενώ ο δημιουργός – Z της τελευταίας αποφασίζει να την τροποποιήσει/διαγράψει την ιδιά στιγμή.

Το (1) σημείο καθώς και ένα τμήμα του (2) αντιμετωπίστηκε με PESSIMISTIC – WRITE το οποίο επιτρέπει την παροχή ενός αποκλειστικού lock επί του εκάστοτε record στην υποκείμενη βάση δεδομένων.

Ο έλεγχος X προσφορών **«σειριακά»** για την εκάστοτε δημοπρασία ελέγχθηκε με την χρήση του Apache JMeter και την δημιουργία 500 **παράλληλων νημάτων** τα οποία υπέβαλλαν **συνεχώς** προσφορές του **ιδίου ποσού** για την **ιδιά δημοπρασία** στο πλαίσιο 10 δευτερολέπτων και σε κάθε περίπτωση μόνο 1 από τις X προσφορές έγινε αποδεκτή.

Το (2) σημείο αντιμετωπίστηκε με την χρήση του Versioning οπού κάθε χρήστης υποβάλλει - χωρίς να το γνωρίζει - μια προσφορά για το εκάστοτε Version της εκάστοτε δημοπρασίας που του προβάλλεται και σε περίπτωση οπού αυτό αλλάξει η προσφορά δεν γίνεται δεκτή διότι μπορεί η οποία αλλαγή επί της δημοπρασίας να καθιστά την τελευταία μη ελκυστική για τον χρήστη.

- Ζητήματα Απόδοσης

1. Μια οντότητα ενδέχεται να χαρακτηρίζεται από πολλαπλές σχέσεις $1-N$ ή/και $M-N$ το οποίο δημιουργεί ζητήματα απόδοσης λόγω του αυξημένου μεγέθους των Join Products που προκύπτουν κατά την ανάκτηση των οντοτήτων αυτών και των σχέσεων τους από την υποκείμενη βάση δεδομένων

2. Η χρήση του Pagination στο πλαίσιο της εφαρμογής συνδυαστικά με το 1^0 σημείο παρουσίασε **σημαντικά ζητήματα απόδοσης** μιας και η Hibernate εκτελούσε το Pagination στην μνήμη με το warning HHH000104.

Το (1) σημείο αντιμετωπίστηκε με την κάθε σχέση/**collection** της εκάστοτε οντότητας να ανακτάται ξεχωριστά και συνεπώς για **μια** οντότητα με $|X|$ $1-N$ ή/και $M-N$ σχέσεις απαιτούνται $|X|$ Queries ενώ σε ένα από τα $|X|$ Queries γίνεται ανάκτηση όλων των $1-1$ και $N-1$ σχέσεων αν αυτές υπάρχουν και απαιτούνται.

Το (2) σημείο αντιμετωπίστηκε ως εξής :

- Ανάκτηση των $|X|$ - Paginated IDS των οντοτήτων που πληρούν τα εκάστοτε κριτήρια από την υποκείμενη βάση

- Ανάκτηση του συνολικού αριθμού $|Y|$ των οντοτήτων που πληρούν τα εκάστοτε κριτήρια από την υποκείμενη βάση

- Αντιμετώπιση σημείου (1) με την χρήση των $|X|$ - Paginated IDS

- Ζητήματα DRY

    1. Κατά το εκάστοτε Server response παρουσιάστηκε η ανάγκη της έκθεσης των κατάλληλων δεδομένων οπού έπρεπε να γίνει χρήση των Data Transfer Objects

    2. Κατά την εξαγωγή των δημοπρασιών από τον διαχειριστή έπρεπε ανάλογα τον απαιτούμενο μορφότυπο (JSON − XML) να εκτεθούν τα αντίστοιχα δεδομένα με διαφορετικό τρόπο

Το (1) σημείο αντιμετωπίστηκε με την συνδυαστική χρήση των IntelliJ IDEA και MapStruct για την παραγωγή (IntelliJ IDEA) των Data Transfer Objects καθώς και την μετατροπή (MapStruct) των οντοτήτων στα καταλληλά Data Transfer Objects.

Το (2) σημείο αντιμετωπίστηκε με την χρήση της βιβλιοθήκης Jackson και του MapStruct τόσο για τον JSON αλλά και για τον XML μορφότυπο

- Κανονικοποίηση Βάσης

    1. Η υποκείμενη βάση δεδομένων δεν έχει υποστεί πλήρη κανονικοποίηση υπό την έννοια ότι συγκεκριμένα αλλά **ελάχιστα** πεδία/τιμές ενώ είναι δυνατό να ανακτηθούν από ένα ερώτημα σε αυτήν αλλά έγινε η επιλογή να συμπεριληφθούν ως πεδία στην κατάλληλη οντότητα για την αποφυγή θεμάτων απόδοσης.

        Π.χ. :

        - Ο συνολικός αριθμός των προσφορών επί μιας δημοπρασίας

Matrix Factorization

1. Η υλοποίηση του Matrix Factorization ακολουθήσε πιστά τις σημειώσεις που μας δοθήκαν στο πλαίσιο του μαθήματος.

2. Η αρχική υλοποίηση του Matrix Factorization πραγματοποιήθηκε σε Python για την απεικόνιση του σφάλματος (RMSE) τόσο στο πλαίσιο του Training - Set αλλά και στο πλαίσιο του Validation – Set με σκοπό την αποφυγή του φαινομένου της υπέρ – προσαρμογής

3. Οι υπέρ – παράμετροι (Latent Features, Learning Rate, …) επιλέχθηκαν μετά από τον πειραματισμό τόσο με Dense αλλά και Sparse Matrices αφού δεν είναι γνωστός εκ των προτέρων ο διαθέσιμος όγκος δεδομένων

4. Κατά την Python υλοποίηση επιλέχθηκε η διαδικασία του 4 — Cross Validation οπού ο αρχικός πίνακας χωρίζεται νοητά σε 4 υπό - πίνακες και λαμβάνουν μέρος 4 Training – Sessions οπού στο καθένα ο εκάστοτε υπό - πίνακας $X \in [1,4]$ λειτουργεί ως Validation – Set ενώ οι υπόλοιποι 3 λειτουργούν ως Training – Set

5. Η διαδικασία του Matrix Factorization μετά από επικοινωνία με τον Ιωάννη Χαμόδρακα περιλαμβάνει την λειτουργικότητα του Early Stopping η οποία υλοποιήθηκε χρησιμοποιώντας έναν από τους 4 υπό – πίνακες του αρχικού πίνακα ως Validation – Set για την αξιολόγηση του Training – Session

6. Η διαδικασία του Matrix Factorization είναι χρονοπρογραμματισμένη ενώ κάθε φορά διαλέγονται μόλις οι πρώτες 150 ενεργές δημοπρασίες με απώτερο σκοπό να μην αυξηθεί δραματικά ο χρόνος εκτέλεσης της διαδικασίας.

7. Χρησιμοποιήθηκε συμβουλευτικά η παρακάτω έρευνα :

    a. MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS