

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Διπλωματική Εργασία

**«Αξιολόγηση ενός Σχεδόν Πραγματικού Χρόνου
Συστήματος Συστάσεων»**

«Ψαριανός Δημήτριος»

«ΠΚΜ202022»

Επιβλέπουσα καθηγήτρια: Ελένη Χριστοπούλου

Κέρκυρα, Ιούνιος 2022



«Αξιολόγηση ενός Σχεδόν Πραγματικού Χρόνου Συστήματος Συστάσεων»

«Ψαριανός Δημήτριος»

«ΠΚΜ202022»

Τριμελής επιτροπή

Δημήτριος Ρίγγας

ΕΔΙΠ

Τμήμα Πληροφορικής

Ιόνιο Πανεπιστήμιο

Κάτια Κερμανίδου

ΕΔΙΠ

Τμήμα
Πληροφορικής
Ιόνιο Πανεπιστήμιο

Ελένη Χριστοπούλου

ΕΔΙΠ

Τμήμα Πληροφορικής

Ιόνιο Πανεπιστήμιο

Κέρκυρα, Ιούνιος 2022

Περιεχόμενα

Περίληψη	4
Abstract.....	5
Εισαγωγή.....	6
1.1 Τι είναι τα συστήματα συστάσεων (RS).....	6
1.2 Λόγοι χρησιμοποίησης RS.....	7
1.3 Τρόπος λειτουργίας RS	8
Θεωρητικό Υπόβαθρο.....	9
2.1 Κατηγορίες Συστημάτων Συστάσεων	9
2.2 Συλλογή δεδομένων εισόδου των συστημάτων συστάσεων	9
2.3 Προβλήματα συστημάτων Συστάσεων.....	10
2.4 Σχεδιασμός εφαρμογών	12
2.4.1 Πλεονεκτήματα και μειονεκτήματα εγγενών εφαρμογών.....	12
2.4.2 Πλεονεκτήματα και μειονεκτήματα web εφαρμογών	13
2.5 Communication Patterns (Publish/Subscribe).....	14
Περιγραφή και Μεθοδολογία Συστήματος και Εφαρμογής	19
3.1 Το σύστημα συστάσεων προς αξιολόγηση.....	19
3.2 Σχεδιασμός της εφαρμογής αξιολόγησης	19
3.3 Εργαλεία που χρησιμοποιήθηκαν	20
3.3.1 Visual Studio Code	20
3.3.2 Docker	21
3.3.3 Apache Kafka.....	22
3.3.4 RapidApi	22
3.3.5 HTML – CSS – JavaScript	23
Υλοποίηση της Εφαρμογής και σύνδεση με το Σύστημα Συστάσεων.....	24
4.1 Αρχιτεκτονική του Συστήματος.....	24
4.2 Ανάλυση Εφαρμογής	26
4.3 Εγκατάσταση συστήματος συστάσεων σε λειτουργικό Linux με την χρήση του Docker.....	29
4.4 Σύνδεση εφαρμογής με το σύστημα συστάσεων.....	31
Παρουσίαση και ερμηνεία αποτελεσμάτων	33
Συμπεράσματα – Προτάσεις – Προβλήματα	34
6.1 Προβλήματα.....	34
6.2 Προτάσεις	34
6.3 Συμπεράσματα / Αξιολόγηση	35
Ο κώδικας της εφαρμογής.....	36
Βιβλιογραφικές Αναφορές.....	43

Πίνακας Εικόνων

Εικόνα 1- Publish/Subscribe Pattern	14
Εικόνα 2 - Data Streaming	17
Εικόνα 3 - Visual Studio Code	20
Εικόνα 4 - Docker Logo	21
Εικόνα 5 - Apache Kafka Logo	22
Εικόνα 6 - RapidApi Logo	22
Εικόνα 7 - RapidApi IMDB Api	23
Εικόνα 8 - HTML\CSS\JS	23
Εικόνα 9 - Διάγραμμα Αρχιτεκτονικής του Συστήματος Συστάσεων	25
Εικόνα 10 - Περιβάλλον Εργασίας VScode	26
Εικόνα 11 - Εφαρμογή αξιολόγησης RS, αρχική σελίδα	27
Εικόνα 12 - Εφαρμογή αξιολόγησης RS, αξιολόγηση ταινιών	27
Εικόνα 13 - Εφαρμογή αξιολόγησης RS, Publish to Topic	28
Εικόνα 14 - Εφαρμογή αξιολόγησης RS, Css code	28
Εικόνα 15 - Docker Web.....	29
Εικόνα 16 - Docker App.....	30
Εικόνα 17 - Docker in Windows Comand Line	30
Εικόνα 18 - VScode Docker Extention.....	31
Εικόνα 19 - Published Objects, UserId:ItemId	32
Εικόνα 20 - RS Json Return.....	32
Εικόνα 21 - Εφαρμογή αξιολόγησης RS, αξιολόγηση συστάσεων	33

Περίληψη

Το αντικείμενο του συγκεκριμένου Capstone Project είναι η αξιολόγηση ενός συστήματος συστάσεων, με την δημιουργία μιας εφαρμογής η οποία θα παράγει συνεχούς ροής δεδομένα.

Το σύστημα συστάσεων δημιουργήθηκε για να δώσει μια λύση στο πρόβλημα το οποίο αντιμετωπίζουν τα συστήματα συστάσεων, σχετικά με τη μείωση της ποιότητας των προτάσεων τους, καθώς αλλάζει το περιεχόμενο των εφαρμογών και ο σκοπός της εφαρμογής είναι να αξιολογήσει το κατά πόσο το συγκεκριμένο σύστημα επιτυγχάνει τον στόχο του.

Αρχικά κατά την εκπόνηση του Capstone project, μελετήθηκαν τα συστήματα συστάσεων γενικότερα σε θεωρητικό υπόβαθρο για τον τρόπο λειτουργίας τους και τον σκοπό τους. Στην συνέχεια μελετήθηκε το σύστημα συστάσεων υπό αξιολόγηση έτσι ώστε να δημιουργηθεί η κατάλληλη εφαρμογή για την αξιοποίηση και την αξιολόγηση του.

Κατά το στάδιο της ανάπτυξης αρχικά έγινε η εγκατάσταση του συστήματος συστάσεων σύμφωνα με τις οδηγίες του δημιουργού του. Στην συνέχεια αναπτύχθηκε η εφαρμογή μέσω της οποίας οι χρήστες της αξιολόγησαν τις προτάσεις του συστήματος.

Τέλος, μελετήθηκαν τα αποτελέσματα της έρευνας και παρουσιάστηκαν για να αξιολογηθεί το κατά πόσο οι συστάσεις που παρείχε το σύστημα ήταν ενημερωμένες και ακριβείς σύμφωνα με τις επιλογές των χρηστών.

Abstract

The object of this Capstone Project is to evaluate a system of recommendations, by creating an application that will produce a continuous flow of data.

The recommendation system was created to provide a solution to the problem that recommendation systems face, regarding the reduction of the quality of their proposals, as the content of the applications changes and the purpose of the application is to assess whether the specific system achieves his goal.

Initially, during the elaboration of the Capstone project, the recommendation systems in general were studied in a theoretical background for their mode of operation and their purpose. Then the system of recommendations under evaluation was studied so as to create the appropriate application for its utilization and evaluation.

During the development phase, the recommendations system was initially installed according to the instructions of its creator. Then the application was developed, through which its users evaluated the proposals of the system.

Finally, the results of the survey were studied and presented to assess whether the recommendations provided by the system were up-to-date and accurate according to the users' choices.

Κεφάλαιο 1

Εισαγωγή

Ο Παγκόσμιος Ιστός έχει αλλάξει δραματικά τον τρόπο ζωής μας. Έχει φέρει πολλές αλλαγές στον τρόπο που επικοινωνούμε, που ψωνίζουμε, που ενημερωνόμαστε και γενικότερα στην καθημερινότητα μας από το τι θα φάμε, πως θα ψυχαγωγηθούμε έως την μόρφωση μας. Πλέον βασιζόμαστε πάρα πολύ στον Παγκόσμιο Ιστό και τις σχετικές τεχνολογίες του, όπως τις μηχανές αναζήτησης για οποιαδήποτε πληροφορία αναζητούμε.

Όμως η τεράστια ανάπτυξη του διαδικτύου έχει επιφέρει και γιγαντιαίο όγκο πληροφορίας με το μεγαλύτερο μέρος αυτής να μην είναι χρήσιμη για την καθημερινότητα του χρήστη και έτσι να αυξάνεται η πολυπλοκότητα της πληροφορίας με αποτέλεσμα να μπερδεύει τον χρήστη και να δυσχεραίνει την λήψη αποφάσεων.

Σε αυτό ακριβώς το σημείο έρχονται τα συστήματα συστάσεων για να δώσουν λύση στο συγκεκριμένο πρόβλημα τα οποία εντοπίζουν υπηρεσίες ή προϊόντα τα οποία είναι χρήσιμα για τον κάθε χρήστη εξατομικευμένα. Τα συστήματα συστάσεων εκπαιδεύονται μέσω αλγορίθμων και επιλογών του χρήστη και προτείνουν σε αυτόν τα προτιμότερα προϊόντα ή υπηρεσίες, μέσα από μια πληθώρα διαθέσιμων επιλογών, τα οποία είναι πιο κοντά στις προτιμήσεις του.

1.1 Τι είναι τα συστήματα συστάσεων (RS)

Τα συστήματα συστάσεων είναι εφαρμογές λογισμικού που παρέχουν εξατομικευμένες προτάσεις στους χρήστες σχετικά με προϊόντα, υπηρεσίες ή πληροφορίες που μπορεί να τους ενδιαφέρουν. Προτείνουν στοιχεία σχετικά με τα ενδιαφέροντα των χρηστών, με βάση τις προτιμήσεις τους οι οποίες εκφράζονται άμεσα (explicitly), είτε έμμεσα (implicitly). Κάθε σύστημα συστάσεων δημιουργείται και ακολουθεί ένα συγκεκριμένο αλγόριθμο βάσει του οποίου συστήνει προϊόντα που είναι πιθανόν να ενδιαφέρουν περισσότερο τον χρήστη. Ένα σύστημα συστάσεων με καλή απόδοση βελτιώνει την εμπειρία του χρήστη και συμβάλει στην ικανοποίηση αυτού ως προς την χρήση του συστήματος. Για παράδειγμα, έστω ότι κάποιος θέλει να αγοράσει μια οικιακή συσκευή, θα κάνει μια έρευνα στο διαδίκτυο, θα ανατρέξει μεγάλο αριθμό ιστοσελίδων, θα αναζητήσει τα χαρακτηριστικά κάθε συσκευής για να δει αν τον καλύπτουν, θα διαβάσει κριτικές για τις συσκευές που τον ενδιαφέρουν και θα αναζητήσει απόψεις από τον περίγυρό του. Το ίδιο ακριβώς ισχύει σχεδόν για το οτιδήποτε κάποιος ενδιαφέρετε να πληροφορηθεί από αυτοκίνητο μέχρι ταινίες, μουσική, φαγητό κλπ. Σε αυτό το σημείο εμπλέκονται τα συστήματα συστάσεων, δεδομένου ότι λαμβάνουν υπόψιν όλα αυτά τα στοιχεία έτσι ώστε να προτείνουν στο χρήστη το επιθυμητό προϊόν ή υπηρεσία χωρίς ο ίδιος να χρειάζεται να πραγματοποιήσει όλη αυτή την έρευνα (Jannach D. et al , 2011).

1.2 Λόγοι χρησιμοποίησης RS

Για να χαρακτηριστεί ένα σύστημα συστάσεων αποδοτικό θα πρέπει να βελτιώνει την εμπειρία και την εμπιστοσύνη του χρήστη ως προς το σύστημα. Αυτό πραγματοποιείται μέσω ενός καθορισμένου αλγόριθμου ο οποίος κάνει συστάσεις προϊόντων / υπηρεσιών στον χρήστη. Οι συστάσεις του συστήματος μπορούν να βασίζονται σε διάφορα στοιχεία όπως τα κορυφαία προϊόντα του καταστήματος, δημογραφικά στοιχεία αγοραστών κλπ. Η σύσταση όμως μπορεί να γίνει και με έναν πιο πολύπλοκο τρόπο λαμβάνοντας υπόψιν πιο σημαντικά στοιχεία για να ικανοποιήσει τις επιθυμίες ενός αγοραστή όπως στοιχεία που προκύπτουν από την ανάλυση της αγοραστικής συμπεριφοράς των πελατών και σύγκριση τους με αντίστοιχες ομάδες με κοινή αγοραστική συμπεριφορά και κοινά χαρακτηριστικά με σκοπό την πρόβλεψη της επιθυμίας του.

Ο τρόπος που γίνονται οι συστάσεις μπορεί να διαφέρει από σύστημα σε σύστημα. Ένα σύστημα μπορεί να κάνει απλές γενικές συστάσεις, μπορεί να προσφέρει εξατομικευμένες συστάσεις για ένα προϊόν ως προς τον εκάστοτε πελάτη, κριτικές πελατών, συστάσεις ανάλογα κοινότητες χρηστών, συστάσεις χρήστες με κοινά χαρακτηριστικά κ.α.

Στόχος ενός συστήματος συστάσεων είναι η αύξηση των πωλήσεων ενός ηλεκτρονικού καταστήματος και η αύξηση της επισκεψιμότητας. Σκοπός λοιπόν είναι να μετατρέψει έναν απλό επισκέπτη του ηλεκτρονικού καταστήματος σε αγοραστή όπου μέσα από έρευνες φαίνεται ότι αυτή η πιθανότητα αυξάνεται κατά πολύ σε ένα ηλεκτρονικό κατάστημα που διαθέτει σύστημα συστάσεων σε αντίθεση με ένα που δεν έχει. Έτσι το σύστημα θα προτείνει στον επισκέπτη προϊόντα ή υπηρεσίες με βάση τα στοιχεία που έχει συλλέξει για αυτόν (προηγούμενες αγορές, προτιμήσεις, κοινωνικά γκρουπ κ.α) και η απόδοση αυτού είναι συνδεδεμένη με το κατά πόσο αυτές οι προτάσεις είχαν επιτυχία.

Πέρα όμως από την αρχική αγορά του καταναλωτή υπάρχουν πολλές ευκαιρίες για τις περισσότερες επιχειρήσεις να αυξήσουν τα κέρδη τους μέσω των πωλήσεων. Μέσω πρακτικών που είναι γνωστές ως cross-selling και upselling μπορούν να βοηθήσουν μια επιχείρηση να τις αξιοποιήσει. Αυτός είναι ένας ακόμα λόγος για την εισαγωγή ενός RS σε ένα κατάστημα. Ο όρος cross-selling χρησιμοποιείται για να υποδείξει τις παράλληλες πωλήσεις οι οποίες έχουν άμεση συσχέτιση με μια αγορά. Παραδείγματος χάριν ένας αγοραστής κάνει μια αγορά ενός κινητού και το σύστημα του προτείνει να αγοράσει επίσης και μια θήκη για το κινητό. Ο όρος upselling χρησιμοποιείται για να ορίσει την πώληση ενός ίδιου προϊόντος αλλά ανώτερης ποιότητας και κατ' επέκτασιν και τιμής. Ένας ακόμα πάρα πολύ σημαντικός λόγος για την χρησιμοποίηση των RS έγκειται στην εμπιστοσύνη των πελατών προς την επιχείρηση. Με τον όλο και αυξανόμενο ρυθμό δημιουργίας νέων ηλεκτρονικών καταστημάτων είναι εξαιρετικά σημαντικό μια επιχείρηση να αποκτήσει σταθερούς και ικανοποιημένους πελάτες και έτσι ως αποτέλεσμα να αυξήσει και την φήμη του. Έτσι όσο περισσότερες φορές επιστρέψει ο πελάτης και όσο περισσότερο έρχεται σε επαφή με το σύστημα που φέρει χαρακτηριστικά αυτό-εκμάθησης, τόσο καλύτερες θα είναι οι υπηρεσίες (συστάσεις που θα λαμβάνει) και ταυτόχρονα τόσο θα αυξάνεται η εμπιστοσύνη του προς αυτό.

1.3 Τρόπος λειτουργίας RS

Τα συστήματα συστάσεων λειτουργούν σαν τα υπόλοιπα συστήματα. Δέχεται κάποια δεδομένα σαν είσοδο τα οποία δεδομένα μέσα από διαδικασίες και επεξεργασία παράγουν κάποια δεδομένα ως αποτέλεσμα και έτσι ορίζεται η έξοδος του συστήματος. Πιο συγκεκριμένα ένα σύστημα συστάσεων δέχεται ως είσοδο τις προτιμήσεις και τα χαρακτηριστικά των πελατών καθώς και τις συσχετίσεις μεταξύ αυτών και μέσω διαδικασιών επεξεργασίας (αλγόριθμοι) παράγει συστάσεις (έξοδος) προϊόντων ή υπηρεσιών.

Ένα σύστημα συστάσεων λειτουργεί όπως ακριβώς ορίζει η έννοια του συστήματος. Πιο συγκεκριμένα, ένα σύστημα καθορίζεται από τις εισόδους, τις εξόδους και τις διαδικασίες επεξεργασίας των δεδομένων εισόδου βάσει των οποίων προκύπτουν οι εξόδοι. Ακριβώς με αυτόν τον τρόπο λειτουργεί και ένα σύστημα συστάσεων. Δέχεται μία συλλογή από δεδομένα εισόδου που παίρνουν τη μορφή προτιμήσεων των πελατών και χαρακτηριστικών των προϊόντων μαζί με μία σειρά συσχετίσεων και στη συνέχεια το σύστημα χρησιμοποιεί αυτά τα δεδομένα για να παράγει τις προτάσεις προϊόντων. Τα δεδομένα αυτά μπορούμε να τα χωρίσουμε σε δύο μεγάλες κατηγορίες ανάλογα με την προέλευση τους. Η πρώτη είναι τα δεδομένα που είναι σχετικά με τον πελάτη ο οποίος δέχεται τις συστάσεις, ενώ στη δεύτερη κατηγορία εντάσσονται τα δεδομένα εκείνα που είναι σχετικά με την κοινότητα άλλων πελατών. Σημαντικό ρόλο όμως κατέχει και ο τρόπος με τον οποίο ένα σύστημα λαμβάνει τα δεδομένα εισόδου και ο τρόπος που τα συλλέγει από το χρήστη. Τρόποι με τους οποίους μπορεί ένα σύστημα να λάβει πληροφορίες για έναν πελάτη είναι:

- I. Με καταγραφή των ενεργειών ενός χρήστη (πλοήγηση του χρήστη στα προϊόντα, κλικ σε συγκεκριμένα προϊόντα, χρόνος που έμεινε στην σελίδα ενός προϊόντος κ.α.)
- II. Δεδομένα που προήλθαν από τον ίδιο τον χρήστη (βαθμολόγηση προϊόντων, σχόλια προϊόντων, συμπλήρωση ερωτηματολογίου κ.α)
- III. Λέξεις κλειδιά (πληκτρολόγηση σε μια μηχανή αναζήτησης προτιμήσεων η χαρακτηριστικών κάποιου προϊόντος)
- IV. Ιστορικό αγορών του χρήστη
- V. Δεδομένα που προκύπτουν από μια συγκεκριμένη ομάδα χρηστών (κοινότητα)

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Κατηγορίες Συστημάτων Συστάσεων

Ανάλογα με τον τρόπο τον οποίο γίνονται οι συστάσεις τα συστήματα συστάσεων χωρίζονται σε κατηγορίες:

- Τα συστήματα τα οποία είναι βασισμένα στην συνεργασία (Collaborative Systems). Το συνεργατικό φιλτράρισμα είναι μια τεχνική που μπορεί να φιλτράρει στοιχεία που μπορεί να αρέσουν σε έναν χρήστη με βάση τις αντιδράσεις παρόμοιων χρηστών. Λειτουργεί αναζητώντας μια μεγάλη ομάδα χρηστών και βρίσκοντας ένα μικρότερο σύνολο αυτών με προτιμήσεις παρόμοιες με έναν συγκεκριμένο χρήστη. Εξετάζει τα στοιχεία που τους αρέσουν και τα συνδυάζει για να δημιουργήσει μια ταξινομημένη λίστα προτάσεων.
- Τα συστήματα που είναι βασισμένα στο περιεχόμενο (Content Based). Αυτό το σύστημα λειτουργεί με δεδομένα που παρέχει ο χρήστης, είτε ρητά (explicitly) είτε έμμεσα (π.χ. κάνοντας κλικ σε έναν σύνδεσμο - implicitly). Με βάση αυτά τα δεδομένα, δημιουργείται ένα προφίλ χρήστη, το οποίο στη συνέχεια χρησιμοποιείται για να κάνει προτάσεις στον χρήστη. Καθώς ο χρήστης παρέχει περισσότερες πληροφορίες ή πραγματοποιεί περαιτέρω ενέργειες για τις συστάσεις, το σύστημα γίνεται όλο και πιο ακριβές.
- Τα συστήματα βασισμένα στην γνώση (Knowledge- Based Systems). Τα συστήματα συστάσεων που βασίζονται στη γνώση, χρησιμοποιούν γνώσεις σχετικά με τους χρήστες και τα προϊόντα για να κάνουν προτάσεις. Οι προτάσεις που βασίζονται στη γνώση δεν εξαρτώνται από την αξιολόγηση, ούτε πρέπει να συλλέγουν πληροφορίες για έναν συγκεκριμένο χρήστη για να δώσουν συστάσεις.
- Τα υβριδικά συστήματα συνεργασίας (Hybrid Systems). Ένα υβριδικό σύστημα συστάσεων είναι ένας ειδικός τύπος συστήματος συστάσεων που μπορεί να θεωρηθεί ως ο συνδυασμός του περιεχομένου και της μεθόδου συνεργατικού φιλτραρίσματος.

2.2 Συλλογή δεδομένων εισόδου των συστημάτων συστάσεων

Για να γίνουν οποιεσδήποτε συστάσεις, το σύστημα πρέπει να συλλέγει δεδομένα. Ο στόχος της συλλογής των δεδομένων είναι η γνώση για τις προτιμήσεις των χρηστών, οι οποίες μπορούν αργότερα να χρησιμοποιηθούν για να γίνουν προβλέψεις σχετικά με τις μελλοντικές προτιμήσεις αυτών. Υπάρχουν 2 τρόποι για την συλλογή αυτών των δεδομένων:

- 1) **Explicit data (έμμεση ανατροφοδότηση):** Αυτή η μέθοδος ζητά από τον χρήστη σαφής αξιολογήσεις, συνήθως σε συγκεκριμένη κλίμακα αξιολόγησης (πχ αξιολόγηση ενός βιβλίου από 1 έως το 5 αστέρια). Η εργασία με συλλογή ρητών δεδομένων είναι αρκετά εύκολη. Υποτίθεται ότι οι αξιολογήσεις που παρέχει ένας χρήστης μπορούν να ερμηνευθούν άμεσα ως προτιμήσεις του χρήστη, καθιστώντας ευκολότερο να γίνουν εκτιμήσεις από δεδομένα για την πρόβλεψη μελλοντικών αξιολογήσεων. Ωστόσο, το μειονέκτημα με τα ρητά δεδομένα είναι ότι φέρει την ευθύνη της συλλογής δεδομένων στον χρήστη, ο οποίος μπορεί να μην θέλει να πάρει χρόνο για να καταχωρίσει αξιολογήσεις.

- 2) **Implicit data (ρητή ανατροφοδότηση):** Αυτή η μέθοδος συλλέγει δεδομένα έμμεσα καθώς ο χρήστης χρησιμοποιεί το σύστημα (καταγράφει τις ενέργειες του χρήστη όπως τα κλικ του κέρσορα, το ιστορικό περιήγησης κ.α.). Τα έμμεσα δεδομένα είναι εύκολο να συλλεχθούν σε μεγάλες ποσότητες χωρίς καμία επιπλέον προσπάθεια εκ μέρους του χρήστη αλλά είναι δυσκολότερο διαχειρίσιμα. Ο στόχος είναι η μετατροπή της συμπεριφορά του χρήστη σε προτιμήσεις χρήστη, με το αρνητικό στοιχείο να είναι η μη γνώση αν η ενέργεια του χρήστη σημαίνει αυτόματα και προτίμηση του στο αντίστοιχο αντικείμενο.

2.3 Προβλήματα συστημάτων Συστάσεων

Στο προηγούμενο κεφάλαιο αναφέραμε τους λόγους χρησιμοποίησης των RS και τα θετικά τους αποτελέσματα. Όμως τα συστήματα συστάσεων διαθέτουν και αρκετά προβλήματα όπου για αρκετές επιχειρήσεις τα καθιστούν απαγορευτικά ως προς την χρησιμοποίησή τους. Παρακάτω αναφέρονται κάποια από τα προβλήματα που εμφανίζονται.

1. Απαιτούνται σημαντικές επενδύσεις

Η επένδυση σε ένα σύστημα συστάσεων εκτός από το οικονομικό κομμάτι απαιτεί και επένδυση σε χρόνο καθώς επίσης και καλή τεχνογνωσία για να κατασκευαστεί και να είναι αποδοτικό. Πέρα από τους κατάλληλους επιστήμονες για την δημιουργία του πρέπει να υπολογιστεί και το κόστος της σωστής ανάλυσης των δεδομένων και την συλλογή τους σε περίπτωση που αυτά δεν υπάρχουν, το κόστος διάφορων μελετών, όπως μελέτη σκοπιμότητας, για να αποδειχθεί ότι το σύστημα θα είναι αποδοτικό για την επιχείρηση πριν αυτό εφαρμοστεί, κ.α. Επίσης οι φάσεις μέχρι το τελικό σύστημα να μπορεί να χρησιμοποιηθεί ποικίλουν, από την φάση δημιουργίας πρωτοτύπου, την φάση του ελάχιστου βιώσιμου προϊόντος έως την τελική έκδοση. Ακόμα και μετά την υλοποίηση του συστήματος συστάσεων τα κόστη είναι συνεχή αφού χρειάζεται συνεχή παρακολούθηση και έλεγχο των συστάσεων για να σιγουρευτεί ότι αυτά είναι αποδοτικά.

2. Πλήθος επιλογών

Με την πληθώρα επιλογών συστημάτων συστάσεων και των αριθμό των μεγάλο αριθμό τρίτων (3rd party) που μπορούν να παρέχουν τέτοια συστήματα πως μια επιχείρηση μπορεί να επιλέξει το σωστότερο και αποδοτικότερο σύστημα για την ίδια; Η αξιολόγηση διαφορετικών λύσεων μπορεί να είναι εξαιρετικά χρονοβόρα, καθώς πρέπει να αξιολογήσετε τις περιπτώσιολογικές τους μελέτες, την τεχνολογία, τον τρόπο ενσωμάτωσης της λύσης στην τρέχουσα οργάνωση της εταιρείας σας και ούτω καθεξής.

3. Περίπλοκη διαδικασία ενσωμάτωσης

Η ενσωμάτωση ενός RS σε μια ηλεκτρονική επιχείρηση μπορεί να είναι πολύ περίπλοκη και πολλές φορές να μην αξίζει την επένδυση και τον κόπο. Η δυσκολία μπορεί να έγκειται σε διάφορους παράγοντες όπως μη πλήρης κατανόηση των μοντέλων συστάσεων, κακή σχεδίαση UX (user experience) του ιστοτόπου, ανεπαρκής γνώση του τομέα της επιχείρησής κ.α.

4. Έλλειψη ικανότητας ανάλυσης δεδομένων

Καθώς τα συστήματα συστάσεων βασίζονται σε δεδομένα αυτά είναι απαραίτητα για να είναι ένα σύστημα αξιόπιστο. Ο μεγάλος όγκος όμως δεδομένων δημιουργεί πολλά προβλήματα καθώς δεν είναι όλα τα δεδομένα απαραίτητα για μια σύσταση έτσι αναγκάζει το σύστημα να χρησιμοποιεί αχρείαστα πολλούς πόρους με αμφίβολα αποτελέσματα. Έτσι απαιτείται περαιτέρω προεργασία των

δεδομένων έτσι ώστε να θεωρηθούν υψηλής ποιότητας. Για να θεωρηθούν τα δεδομένα υψηλής ποιότητας πρέπει να ικανοποιούν κάποιες προϋποθέσεις όπως πόσο πρόσφατα είναι, πόσο θόρυβο περιέχουν (δηλαδή πόση άχρηστη πληροφορία περιέχουν για την σύσταση ενός συγκεκριμένου προϊόντος), τι ποικιλία πληροφορίας περιέχουν κ.α. Οι μηχανές συστάσεων που βασίζονται σε βαθιά μάθηση μπορεί να απαιτούν υψηλή υπολογιστική πολυπλοκότητα. Εάν τα δεδομένα που τροφοδοτούνται στο μοντέλο είναι λιγότερο ακριβή ή πολύτιμα, το αποτέλεσμα θα είναι λιγότερο χρήσιμο.

5. Το πρόβλημα της “κρύας εκκίνησης”

Το συγκεκριμένο πρόβλημα εντοπίζεται όταν ένας νέος χρήστης εισάγεται στο σύστημα. Αφού τα RS χρησιμοποιούν δεδομένα χρηστών όπως συσχετίσεις χρήστη – προϊόντος, για την δημιουργία σύστασης είναι απόλυτα κατανοητό ότι για έναν νέο χρήστη που δεν διατίθενται δεδομένα δημιουργείται πρόβλημα στην παροχή σύστασης σε αυτόν.

6. Αδυναμία καταγραφής αλλαγών στη συμπεριφορά των χρηστών (trends are always changing)

Από την στιγμή που οι χρήστες είναι άνθρωποι και όχι μηχανές οι προτιμήσεις τους και οι ανάγκες τους αλλάζουν και εξελίσσονται. Έτσι ένα καλό σύστημα συστάσεων θα πρέπει να εντοπίζει αλλαγές στις προτιμήσεις και την συμπεριφορά των πελατών και θα πρέπει να ενημερώνεται αυτόματα σε πραγματικό χρόνο έτσι ώστε να προσφέρει επιθυμητές συστάσεις.

7. Απόρρητο

Η ακρίβεια ενός συστήματος συστάσεων εξαρτάται κυρίως από τον αριθμό των πληροφοριών που διαθέτει αυτό για έναν χρήστη. Έτσι όσα περισσότερα γνωρίζει το σύστημα για αυτόν τον χρήστη τόσο καλύτερη θα είναι και η σύσταση του. Το να δώσει όμως ένας χρήστης πολλές πληροφορίες για τον εαυτό του δεν είναι εύκολο καθώς πολλοί χρήστες είναι καχύποπτοι για τον λόγο που θα χρησιμοποιηθούν τα δεδομένα του και για την ασφάλεια αυτών και όχι άδικα καθώς υπάρχουν πολλά περιστατικά διαρροής προσωπικών δεδομένων. Έτσι το να χτιστεί μια σχέση εμπιστοσύνης μεταξύ πελάτη και εταιρίας είναι πολύ βασικό και μεγάλη πρόκληση για την ίδια την εταιρία.

2.4 Σχεδιασμός εφαρμογών

Η ανάπτυξη μιας εφαρμογής περιλαμβάνει σχεδιασμό και αρκετές διαδικασίες για να διαμορφώσουν ένα αρμονικό σύνολο. Για την υλοποίηση αυτής υπάρχουν τρεις δυνατότητες, είτε να αναπτυχθεί μια εγγενής εφαρμογή, μια εφαρμογή που έχει αναπτυχθεί ουσιαστικά για μια συγκεκριμένη κινητή συσκευή και εγκαθίσταται απευθείας στη συσκευή, είτε μια εφαρμογή Web, είτε μια Υβριδική εφαρμογή (Hybrid application).

2.4.1 Πλεονεκτήματα και μειονεκτήματα εγγενών εφαρμογών

Πλεονεκτήματα:

- Εφόσον οι εγγενείς εφαρμογές λειτουργούν με τις ενσωματωμένες λειτουργίες της συσκευής - όπως η φωτογραφική μηχανή, το μικρόφωνο, οι υπηρεσίες τοποθεσίας κλπ. είναι πιο εύκολο να λειτουργούν και να εκτελούνται πιο γρήγορα στη συσκευή.
- Οι εγγενείς εφαρμογές λαμβάνουν πλήρη υποστήριξη από τα καταστήματα εφαρμογών και τις αγορές. Οι χρήστες μπορούν εύκολα να βρουν και να κατεβάσουν εφαρμογές της επιλογής τους από αυτά τα καταστήματα.
- Επειδή οι εγγενείς εφαρμογές πρέπει να λάβουν την έγκριση του καταστήματος εφαρμογών για τις οποίες προορίζονται, ο χρήστης μπορεί να διαβεβαιώσει ότι η εφαρμογή είναι ασφαλής και πλήρως συμβατή με τη συσκευή.
- Οι εγγενείς εφαρμογές λειτουργούν καλύτερα για τους προγραμματιστές, στους οποίους παρέχεται το SDK και όλα τα άλλα εργαλεία για να δημιουργήσετε την εφαρμογή με μεγαλύτερη ευκολία.

Μειονεκτήματα:

- Οι εγγενείς εφαρμογές τείνουν να είναι μια ακριβότερη πρόταση για τον προγραμματιστή. Αυτό ισχύει ιδιαίτερα για τους προγραμματιστές που θα ήθελαν η εφαρμογή τους να είναι συμβατή με πολλές κινητές συσκευές και πλατφόρμες.
- Το κόστος συντήρησης της εφαρμογής και ενημέρωσης εφαρμογών είναι υψηλότερο για τις εγγενείς εφαρμογές, ειδικά εάν η εφαρμογή υποστηρίζει περισσότερες από μία πλατφόρμες για κινητά.
- Η διαδικασία λήψης μιας εγγενούς εφαρμογής που έχει εγκριθεί σε ένα κατάστημα εφαρμογών μπορεί να αποδειχθεί μακρά και κουραστική για τον κύριο του έργου και δεν έχει πάντα ως αποτέλεσμα επιτυχία.
- Οι χρήστες διαφόρων φορητών συσκευών ενδέχεται να χρησιμοποιούν διαφορετικές εκδόσεις της εφαρμογής, γεγονός που δυσχεραίνει όχι μόνο τον προγραμματιστή να διατηρεί και να προσφέρει υποστήριξη, αλλά και τους χρήστες να επικοινωνούν αποτελεσματικά, να παίζουν παιχνίδια κλπ. με άλλους χρήστες της εφαρμογής.

2.4.2 Πλεονεκτήματα και μειονεκτήματα web εφαρμογών

Πλεονεκτήματα:

- ✓ Εύκολη Πρόσβαση, αρκεί μια σύνδεση στο internet και ένας web browser για αποκτήσει κανείς πρόσβαση σε αυτή.
- ✓ Πρόσβαση μέσω πλήθους Συσκευών, οι Web εφαρμογές μπορούν να χρησιμοποιηθούν μέσω υπολογιστή, tablet, smart phone, PDA κ.ά. ανεξαρτήτως λειτουργικού.
- ✓ Ευκολότερη Διασύνδεση, οι εφαρμογές υλοποιημένες σε web αρχιτεκτονική διασυνδέονται πολύ πιο εύκολα με άλλες web εφαρμογές. Η διασύνδεση των επί μέρους εφαρμογών διευκολύνει στη βελτίωση της ροής εργασιών και διαδικασιών ενός οργανισμού.
- ✓ Ευκολότερη Εγκατάσταση και Συντήρηση, η εγκατάσταση, βελτίωση, αποκατάσταση προβλημάτων και συντήρηση της εφαρμογής είναι ευκολότερη, καθώς δεν απαιτείται εγκατάσταση των ενημερώσεων σε κάθε τερματικό που έχει πρόσβαση σε αυτή, παρά μόνο στο web server.
- ✓ Εύκολη Αναβάθμιση Υλικού, καθώς οι απαιτήσεις σε υλικό (hardware) αυξάνονται προσθέτοντας νέες λειτουργικότητες ή περισσότερους χρήστες με πρόσβαση στο σύστημα, η μόνη αναβάθμιση που απαιτείται είναι αυτή του server και όχι των υπολογιστών των χρηστών. Η δυναμικότητα μιας web εφαρμογής μπορεί άλλωστε να ενισχυθεί μέσω clustering, ή και εκτέλεση της εφαρμογής σε πολλούς servers ταυτόχρονα.
- ✓ Νέο, βελτιωμένο περιβάλλον: Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι πλέον με την εμφάνιση της HTML5 είναι δυνατό ο δημιουργός της εφαρμογής να την εμπλουτίσει έτσι ώστε να είναι πιο φιλική, εύχρηστη και ευχάριστη προς τον χρήστη με εύκολο τρόπο.

Μειονεκτήματα:

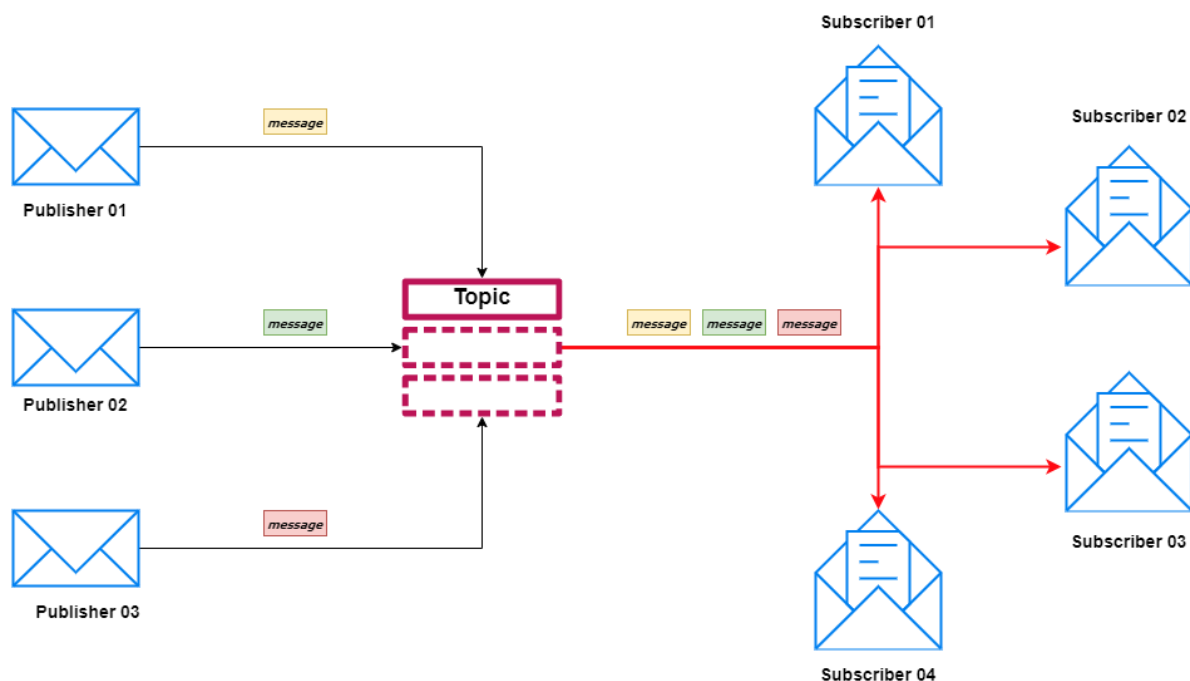
- Ασφάλεια, οι διαδικτυακές εφαρμογές είναι προσβάσιμες από όλους τους υπολογιστές του διαδικτύου. Έτσι είναι ευπρόσβλητες από κακόβουλες ενέργειες. Επιπλέον η αποστολή δεδομένων στον εξυπηρετητή για την επεξεργασία τους, καθιστά ευκολότερη την υποκλοπή τους.
- Κόστος, Οι διαδικτυακές εφαρμογές απαιτούν ένα κόστος για την φιλοξενία τους στο διαδίκτυο. Επειδή όμως εξυπηρετούν πολλούς χρήστες, η υπηρεσία έχει μικρότερο κόστος ανά χρήστη. Στην πράξη πολλές τέτοιες εφαρμογές χρηματοδοτούνται μόνο από διαφημίσεις ή παρέχονται εντελώς δωρεάν στον τελικό χρήστη.
- Μη πλήρης συμβατότητα των περιηγητών: Ένα ακόμα μειονέκτημα που αφορά την τελευταία έκδοση της HTML είναι η μη πλήρης συμβατότητα των περιηγητών με την έκδοση αυτή. Αν και τα πλεονεκτήματα και οι δυνατότητες της HTML5 είναι πολλά αρκετοί από τους περιηγητές δεν είναι ακόμα πλήρως συμβατοί με αυτά. Επίσης, σε περίπτωση που δεν έχει προβλεφθεί η μη λειτουργία κάποιου χαρακτηριστικού της εφαρμογής σε κάποιον περιηγητή, αυτό μπορεί να δημιουργήσει προβλήματα στην εφαρμογή με αποτέλεσμα να μην λειτουργεί σωστά ή να μην λειτουργεί καθόλου.

2.5 Communication Patterns (Publish/Subscribe)

Τα πρότυπα επικοινωνίας (communication patterns) είναι αρχιτεκτονικές έννοιες που περιγράφουν πως κάποιες διεργασίες δικτύου ολοκληρώνονται μέσω της μεταφοράς μηνυμάτων. Για να μπορούν να σχεδιαστούν και να εφαρμοστούν σωστά οι εφαρμογές προς υλοποίηση και για να είναι λειτουργικές και αποδοτικές η καλή γνώση αυτών των προτύπων είναι πολύ σημαντική. Τα βασικά πρότυπα επικοινωνίας είναι το πρότυπο request/response, το asynchronous messaging και το πρότυπο Publish/Subscribe.

Το πρότυπο ανταλλαγής μηνυμάτων Publish/Subscribe

Μέσω του προτύπου Publish/Subscribe η πληροφορία αποστέλλεται στα ενδιαφερόμενα μέρη μαζικά. Η συμφόρηση του δικτύου μειώνεται δραστικά αφού μέσω αυτού του προτύπου επιτρέπεται στο εκδότη (Publisher) να στείλει τις πληροφορίες μόνο μια φορά σε έναν διακομιστή δημοσίευσης/εγγραφής. Στην συνέχεια ο διακομιστής αναμεταδίδει τις πληροφορίες στους συνδρομητές (subscribers) (Tarkoma S. , 2012).



Εικόνα 1- Publish/Subscribe Pattern- πηγή: <https://www.bmc.com/>

Πλεονεκτήματα του προτύπου Publish/Subscribe:

- **Διαχωρισμός της εφαρμογής από τον τρόπο επικοινωνίας.** Έτσι δίνεται η δυνατότητα δημιουργίας ξεχωριστών στοιχείων (components). Αυτό έχει ως αποτέλεσμα πιο καλά δομημένα και ασφαλή στοιχεία (software components) και πιο ποιοτικό και εύκολα συντηρήσιμο κώδικα.
- **Μεγαλύτερη διακριτότητα όλου του συστήματος.** Η απλότητα του συστήματος μας δίνει την δυνατότητα να γνωρίζουμε ακριβώς την προέλευση των πληροφοριών και τον προορισμό τους χωρίς να προσδιορίζουμε ρητά τον αποστολέα ή τον παραλήπτη στον πηγαίο κώδικα.

- **Επικοινωνία σε πραγματικό χρόνο.** Τα μηνύματα παραδίδονται στους συνδρομητές ακαριαία μέσω push, καθιστώντας το την ιδανική επιλογή για απαιτήσεις επικοινωνίας σχεδόν σε πραγματικό χρόνο.
- **Ευκολία ανάπτυξης.** Δεδομένου ότι το pub/sub δεν εξαρτάται από τη γλώσσα προγραμματισμού, το πρωτόκολλο ή μια συγκεκριμένη τεχνολογία, οποιοσδήποτε υποστηριζόμενος μεσίτης μηνυμάτων μπορεί εύκολα να ενσωματωθεί σε αυτό χρησιμοποιώντας οποιαδήποτε γλώσσα προγραμματισμού. Επιπλέον, το pub/sub μπορεί να χρησιμοποιηθεί ως γέφυρα για να επιτρέψει την επικοινωνία μεταξύ στοιχείων που έχουν δημιουργηθεί με χρήση διαφορετικών γλωσσών μέσω της διαχείρισης των επικοινωνιών μεταξύ συστατικών. Αυτό οδηγεί σε εύκολες ενσωματώσεις με εξωτερικά συστήματα χωρίς να χρειάζεται να δημιουργηθεί λειτουργικότητα για τη διευκόλυνση των επικοινωνιών ή την ανησυχία για επιπτώσεις στην ασφάλεια. Μπορούμε απλώς να δημοσιεύσουμε ένα μήνυμα σε ένα θέμα και να αφήσουμε την εξωτερική εφαρμογή να εγγραφεί στο θέμα, εξαλείφοντας την ανάγκη για άμεση αλληλεπίδραση με την υποκείμενη εφαρμογή.
- **Αυξημένη επεκτασιμότητα και αξιοπιστία.** Δεν είναι απαραίτητο να ξέρουμε εκ των προτέρων τον αριθμό των εκδοτών και των συνδρομητών. Μπορεί να γίνει επέκταση αυτών σε ένα θέμα ανάλογα με την χρήση. Ακόμα τα προβλήματα αντιμετωπίζονται ευκολότερα αφού υπάρχει διαχωρισμός εφαρμογής και επικοινωνίας και έτσι οι προγραμματιστές μπορούν να διορθώσουν ένα συγκεκριμένο στοιχείο χωρίς τον φόβο επηρεασμού της υπόλοιπης εφαρμογής.

2.6 Συνεχούς ροής δεδομένα (Data Streaming)

Τα δεδομένα ροής (streaming data) είναι η συνεχής ροή δεδομένων που παράγονται από διάφορες πηγές. Συνήθως αυτές οι ροές (streams) τροφοδοτούνται σε κάποιο λογισμικό επεξεργασίας για την εξαγωγή πληροφοριών. Τα δεδομένα αντιπροσωπεύουν ένα event και η ροή δεδομένων αποτελείται από πολλά events ταξινομημένα χρονικά. Αυτά τα event αποτελούν αλλαγές σε μια επιχείρηση (όπως νέοι χρήστες, αλλαγή στις προτιμήσεις ενός χρήστη κλπ.) και είναι αναγκαίο για την ίδια να ενημερώνεται και να τις αναλύει σε πραγματικό χρόνο (Andrew G. Psaltis, 2017).

Παραδείγματα ροών δεδομένων είναι πολλά όπως αρχεία καταγραφής δραστηριότητας, αρχεία καταγραφής οικονομικών συναλλαγών, δεδομένα αισθητήρων κ.α. Με την ανάπτυξη του Internet of Things (IoT) και τις προσδοκίες των πελατών, η σημασία της ροής δεδομένων και της επεξεργασίας ροής έχει αυξηθεί. Σημαντικοί τομείς όπως της υγείας χρησιμοποιούν συνεχούς ροής δεδομένων. Για παράδειγμα ένα health monitor που χρησιμοποιεί data streaming για να παρακολουθεί τον καρδιακό ρυθμό, το οξυγόνο και άλλες τιμές του ασθενή. Η έγκαιρη ανάλυση αυτών των δεδομένων είναι απαραίτητη, καθώς η ασφάλεια του ασθενή μπορεί να εξαρτάται από αυτήν.

Γενικά χαρακτηριστικά των Data Streams είναι:

➤ Ευαισθησία στον χρόνο

Κάθε event μιας ροής δεδομένων έχει και μια χρονική σήμανση. Η σημασία μετά από ένα ορισμένο χρονικό διάστημα ενός event ελαττώνεται καθώς η ροές δεδομένων είναι εξαρτώνται σημαντικά από τον χρόνο. Για παράδειγμα αν ένας αισθητήρας θερμοκρασίας λάβει μια υψηλή τιμή μια συγκεκριμένη χρονική στιγμή η ανάλυση της και η δράση του συστήματος πρέπει να γίνει σε σύντομο χρονικό διάστημα καθώς αργότερα έχει χαθεί η αξία αυτού του δεδομένου.

➤ Συνεχής

Οι ροές δεδομένων δεν έχουν αρχή και τέλος, συμβαίνουν σε πραγματικό χρόνο και στο σύστημα εμφανίζονται ανάλογα με τις απαιτήσεις του ίδιου.

➤ Ετερογενής

Επειδή τα δεδομένα ροής μπορούν να προέρχονται από πάρα πολλές πηγές των οποίων τα χαρακτηριστικά μπορούν να διαφέρουν έτσι και η μορφή τους μπορεί να ποικίλει.

➤ Ατελής

Καθώς όπως είπαμε οι πηγές των ροών δεδομένων ποικίλουν και οι μηχανισμοί τροφοδότησης των δεδομένων από αυτές να είναι διαφορετικοί, τα δεδομένα μπορούν να καταστραφούν, να λείπουν ή ακόμα και να ληφθούν εκτός χρονικής σειράς.

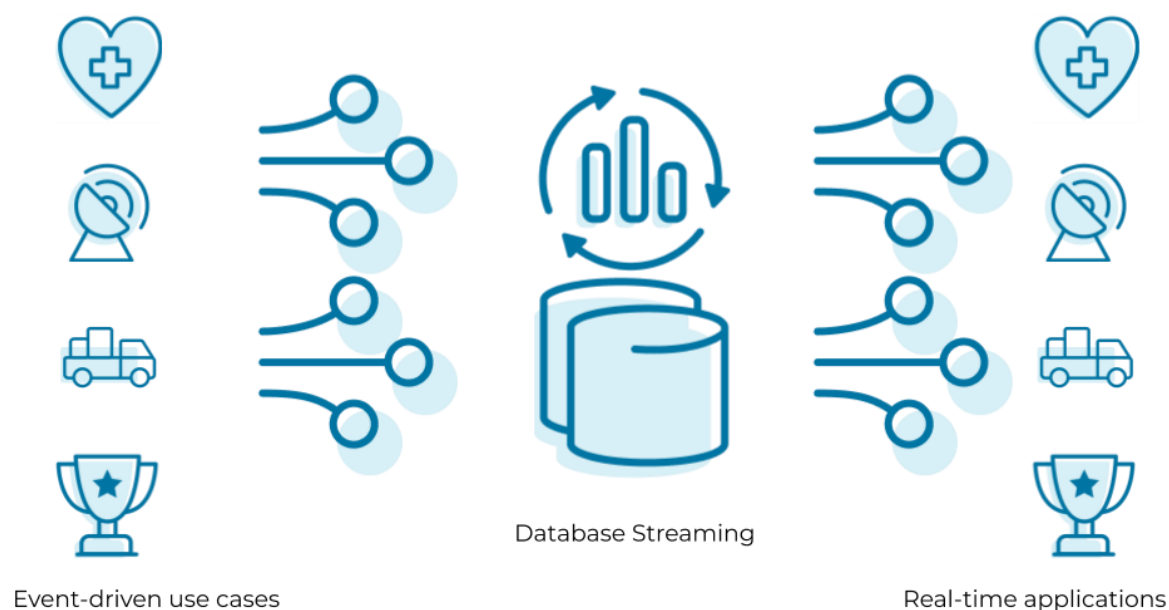
➤ Ευμετάβλητο και μη επαναλαμβανόμενο

Όπως προαναφέραμε η ροή των δεδομένων γίνεται σε πραγματικό χρόνο και ενώ υπάρχουν μηχανισμοί αναμετάδοσης τα δεδομένα ενδέχεται να έχουν μεταβληθεί. Σε σύγχρονα συστήματα υπάρχει αρχείο καταγραφής δεδομένων και έτσι μπορεί να γίνει ανάλυση αυτών οποιαδήποτε στιγμή αυτό απαιτηθεί.

2.7 Η σημασία των συνεχούς ροών δεδομένων για τις επιχειρήσεις

Με την ανάπτυξη του IoT ο όγκος των δεδομένων που παράγονται από τους χρήστες μέσω των συσκευών κάθε δευτερόλεπτο είναι τεράστιος και η ροή τους συνεχόμενη. Για τους οργανισμούς η ανάλυση αυτών των δεδομένων αποτελεί ευκαιρία και μια τεράστια πρόκληση.

Παραδοσιακά η συλλογή και η αποθήκευση των δεδομένων για έναν οργανισμό γίνεται μετά από την πάροδο του χρόνου σε ειδικές αποθήκες με σκοπό την επεξεργασία τους όταν αυτό απαιτηθεί εξοικονομώντας έτσι τους πόρους του συστήματος. Πλέον όμως τα χαρακτηριστικά και η δομή των δεδομένων έχουν αλλάξει καθώς η εισαγωγή αισθητήρων, οι διαδικτυακές συναλλαγές ακόμα και η χρήση καρτών δημιουργούν συνεχώς δεδομένα σε πραγματικό χρόνο όπου όπως αναφέραμε στο προηγούμενο κεφάλαιο η ανάλυση τους έχει αξία να γίνει άμεσα. Επίσης ο όγκος των δεδομένων μπορεί να φτάσει σε γιγάντια επίπεδα αλλά η περισσότερη πληροφορία που διατίθεται από αυτά δεν είναι χρήσιμη. Τα συστήματα μπορούν να αναλύσουν και να μεταδώσουν δεδομένα που είναι όντως χρήσιμα για έναν οργανισμό ώστε να αποθηκεύσει μόνο αυτά με αποτέλεσμα να μειώνεται η απώλεια δεδομένων και να ελαχιστοποιείται το κόστος των υποδομών.



Εικόνα 2 - Data Streaming – πηγή: <https://www.confluent.io/>

Περιπτώσεις χρήσης Data Stream:

- Internet of Things
- Monitor χρηματιστήριου σε πραγματικό χρόνο
- Μάρκετινγκ, πωλήσεις και ανάλυση επιχειρήσεων
- Αρχεία καταγραφής δραστηριοτήτων και συναλλαγών
- Παρακολούθηση και αναφορά σε εσωτερικά συστήματα πληροφορικής
- Παρακολούθηση αρχείων καταγραφής: Αντιμετώπιση προβλημάτων συστημάτων, διακομιστών, συσκευών κλπ.
- Ανίχνευση απάτης
- Απόθεμα λιανικής/αποθήκης: διαχείριση αποθέματος σε όλα τα κανάλια και τοποθεσίες και παροχή απρόσκοπτης εμπειρίας χρήστη σε όλες τις συσκευές
- Μηχανική μάθηση και Α.Ι.

2.8 Επεξεργασία Ροής (Data Streaming)

Για να πραγματοποιηθεί η επεξεργασία της ροής των δεδομένων απαιτείται μια διαδικασία την οποία αναλαμβάνει ένας επεξεργαστής ροής (Data processor πχ apache kafka). Ένας επεξεργαστής ροής συλλέγει, αναλύει και οπτικοποιεί μια συνεχή ροή δεδομένων. Οι απαιτήσεις ενός επεξεργαστή ροών είναι οι ακόλουθες:

✓ Χαμηλή καθυστέρηση

Καθώς τα δεδομένα έρχονται σε συνεχή ροή ο επεξεργαστής πρέπει να διαθέτει ταχύτητα γιατί σε περίπτωση που χάσει δεδομένα δεν μπορεί να επιστρέψει πίσω. Επίσης τα δεδομένα όπως έχουμε αναφέρει χάνουν την αξία με τον χρόνο οπότε αν ο επεξεργαστής είναι αργός θα παράξει άχρηστα αποτελέσματα.

✓ Επεκτασιμότητα

Συνήθως τα δεδομένα δεν έχουν σταθερό ρυθμό στον όγκο τους. Για παράδειγμα στους αισθητήρες τις περισσότερες φορές τα δεδομένα ροής έχουν μικρό όγκο όμως για κάποιο χρονικό διάστημα αυτός ο όγκος μπορεί να κλιμακωθεί. Επειδή ο όγκος δεν μπορεί να προβλεφθεί ο επεξεργαστής απαιτείται να μπορεί να διαχειριστεί μεγάλους όγκους δεδομένων αν αυτό απαιτηθεί.

✓ Διαθεσιμότητα

Επειδή όπως έχουμε αναφέρει τα δεδομένα είναι συνεχή σε πραγματικό χρόνο ένας επεξεργαστής δεν μπορεί να σταματήσει να δουλεύει για μεγάλο χρονικό διάστημα και έτσι οποιαδήποτε σφάλματα παρουσιάσει (εκτός των critical) δεν θα πρέπει να διακόπτουν την λειτουργία του.

Περιγραφή και Μεθοδολογία Συστήματος και Εφαρμογής

3.1 Το σύστημα συστάσεων προς αξιολόγηση

Αναφορικά με την τεχνική, την οποία θα χρησιμοποιεί το σύστημα ώστε να παράγει συστάσεις, το συνεργατικό φιλτράρισμα είναι μία επιλογή, που μπορεί να υποστηρίξει το σκοπό του συστήματος. Άλλωστε, η χρήση μιας τεχνικής που βασίζεται στο περιεχόμενο της εφαρμογής, θα ήταν εμπόδιο για την εκπλήρωση της λειτουργικής απαίτησης, η οποία θέλει το σύστημα να λειτουργεί σε οποιαδήποτε εφαρμογή. Παράλληλα, το είδος των δεδομένων εισόδου, με το οποίο θα τροφοδοτείται το σύστημα, είναι δεδομένα ρητής ανατροφοδότησης, μιας και έχουν μεγαλύτερη διαθεσιμότητα σε σχέση με τα δεδομένα έμμεσης ανατροφοδότησης, ενώ δίνεται μεγαλύτερη ευελιξία στις εφαρμογές, καθώς μπορούν να τα ερμηνεύουν όπως εκείνες επιθυμούν. Αυτό σημαίνει πως η κάθε εφαρμογή μπορεί να ορίζει την αλληλεπίδραση ενός χρήστη με ένα αντικείμενο με διαφορετικό τρόπο (π.χ. χρόνος ανάγνωσης, συχνότητα αγοράς, κ.ο.κ.).

3.2 Σχεδιασμός της εφαρμογής αξιολόγησης

Στόχος της εφαρμογής είναι να μπορεί ο χρήστης να επιλέξει εύκολα τις προτιμήσεις του σε διάφορα πεδία τα οποία αυτός επιθυμεί και αυτή να του προτείνει αντίστοιχα προϊόντα ή υπηρεσίες ανάλογα με τα αποτελέσματα του συστήματος αξιολόγησης που χρησιμοποιείται για να τα παράγει. Στην συνέχεια θα δίνετε στον χρήστη η δυνατότητα να αξιολογήσει τα προϊόντα ή τις υπηρεσίες που του προτάθηκαν.

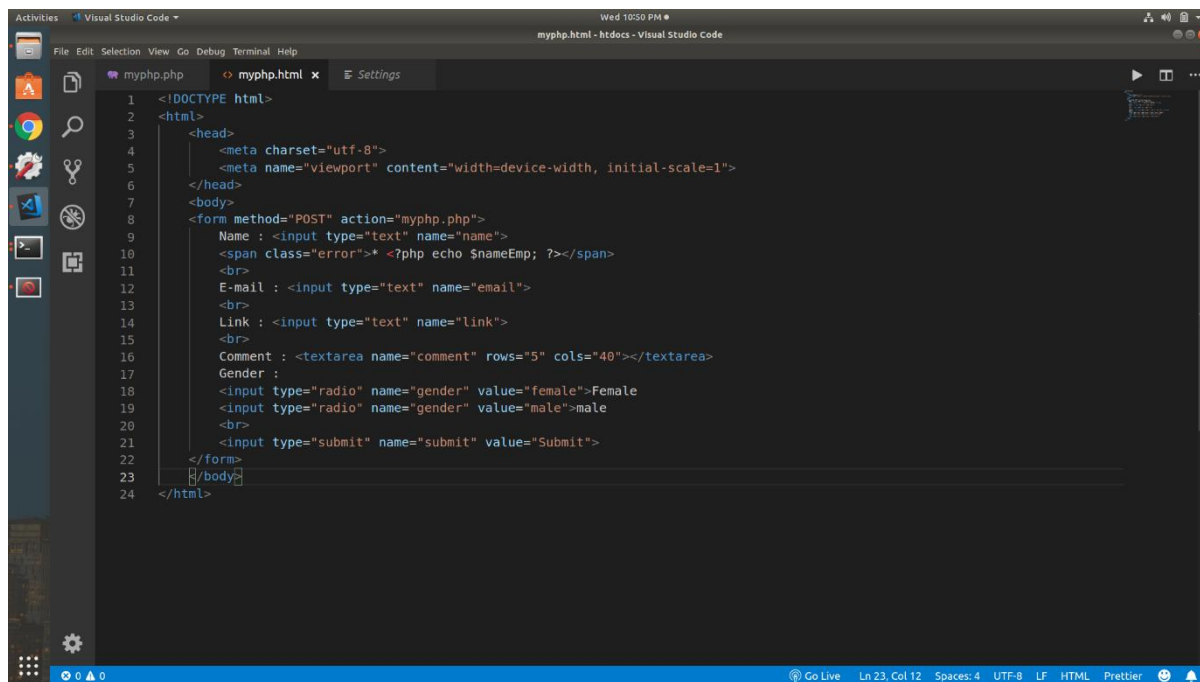
Η εφαρμογή θα πρέπει να έχει την δυνατότητα να ενσωματώσει εύκολα το αντίστοιχο σύστημα αξιολόγησης που επιθυμείτε κάθε φορά να αξιολογηθεί, καθώς επίσης και διαφορετικά πεδία στα οποία θα απαιτηθεί να γίνει η αξιολόγησή του. Επίσης το σύστημα θα πρέπει να μπορούν να το χρησιμοποιήσουν εύκολα και γρήγορα πολλοί χρήστες από οπουδήποτε.

Για την εφαρμογή που επιλέξαμε να δημιουργήσουμε ως αντικείμενο αξιολόγησης του RS επιλέχθηκαν ταινίες και σειρές τις οποίες μπορεί να αξιολογήσει ο χρήστης ως προς την προτίμηση του. Μέσω της εφαρμογής ο χρήστης μπορεί να βαθμολογήσει οποιαδήποτε σειρά ή ταινία βρίσκεται στην μεγαλύτερη βάση δεδομένων ταινιών IMDB.

3.3 Εργαλεία που χρησιμοποιήθηκαν

3.3.1 Visual Studio Code

Για την συγγραφή του κώδικα της εφαρμογής χρησιμοποιήθηκε το εργαλείο visual studio code ή αλλιώς γνωστό ως VS Code (Ockert J. du Preez ,2019).



Εικόνα 3 - Visual Studio Code

Το Visual Studio Code είναι ένα δωρεάν πρόγραμμα επεξεργασίας κώδικα (code editor).

Μπορεί να χρησιμοποιηθεί σε οποιαδήποτε γλώσσα προγραμματισμού χωρίς την ανάγκη ξεχωριστών code editors για την κάθε γλώσσα. Υποστηρίζει πολλές γλώσσες, όπως Python, Java, C++, JavaScript, php, HTML και άλλες. Στην περίπτωση μας χρησιμοποιήθηκε για την συγγραφή κώδικα σε JavaScript, HTML και CSS.

Γιατί χρησιμοποιήσαμε το εν λόγω εργαλείο:

- Το Visual Studio Code επισημαίνει λέξεις-κλειδιά στον κώδικά με διαφορετικά χρώματα για να βοηθήσει να προσδιοριστούν εύκολα τα μοτίβα κωδικοποίησης.
- Καθώς γίνεται η συγγραφή του κώδικα, ο κώδικας του Visual Studio δίνει προτάσεις για να ολοκληρωθούν οι γραμμές του και προτείνει επιδιορθώσεις για συνηθισμένα λάθη.
- Διαθέτει επίσης πρόγραμμα εντοπισμού σφαλμάτων όπου ελέγχει κάθε γραμμή κώδικα για την κατανόηση τυχόν προβλημάτων.
- Διαθέτει πάρα πολλά extensions που βοηθούν πάρα πολύ στην επίτευξη του επιθυμητού αποτελέσματος.(Για παράδειγμα στην εργασία μας χρησιμοποιήσαμε το extension Docker για την σύνδεση της εφαρμογής μας με το σύστημα συστάσεων).

3.3.2 Docker

Το Docker είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα με την οποία μπορούμε να κάνουμε virtualization σε επίπεδο λειτουργικού συστήματος. Δίνεται δηλαδή η δυνατότητα εγκατάστασης μόνο της εφαρμογής / υπηρεσίας (χωρίς έξτρα λειτουργικό σύστημα), σε ένα απομονωμένες περιοχές που ονομάζονται Software Containers. Το λογισμικό χρησιμοποιεί τεχνολογίες του πυρήνα του Linux και έτσι επιτρέπει σε ξεχωριστά containers να εκτελούνται στον ίδιο λειτουργικό σύστημα (Waud E., 2018).



Εικόνα 4 - Docker Logo

Τα βασικά χαρακτηριστικά του είναι:

- το **dockerfile**: script που παρέχει πληροφορίες για το container που θα δημιουργηθεί
- το **image**: αρχείο που χρησιμοποιείται για να εκτελεστεί ο κώδικας του container
- το **container**: η "φούσκα" μέσα στην οποία τρέχουν οι εφαρμογές
- το **network**: συνδέει τα containers μεταξύ τους

Στην περίπτωση μας χρησιμοποιήθηκε για την εγκατάσταση του λειτουργικού συστήματος (Ubuntu) στο οποίο εκτελείται το σύστημα συστάσεων. Επίσης δημιουργήθηκαν διαφορετικά containers στα οποία εκτελείται ο Apache Kafka και ο Apache Zookeeper.

3.3.3 Apache Kafka

Ο Apache Kafka είναι μια πλατφόρμα event-streaming. Δηλαδή είναι μια πλατφόρμα ανταλλαγής μηνυμάτων η οποία ακολουθεί το μοτίβο Publish / Subscribe (PubSub). Τα δεδομένα ροής (streaming data) είναι δεδομένα τα οποία παράγονται συνεχώς από πάρα πολλές πηγές δεδομένων, οι οποίες συνήθως στέλνουν τις εγγραφές δεδομένων ταυτόχρονα. Μια πλατφόρμα ροής όπως ο Apache Kafka χειρίζεται αυτή τη συνεχή εισροή δεδομένων και επεξεργάζεται τα δεδομένα διαδοχικά και σταδιακά. Οι βασικές λειτουργίες που παρέχει ο Apache Kafka είναι η δημοσίευση (Publish) και εγγραφή (Subscribe) σε records ροών δεδομένων, αποθήκευση των ροών των records με την σειρά δημιουργίας τους και η επεξεργασία των ροών των records σε πραγματικό χρόνο (Narkhede N. et al ,2017).



Εικόνα 5 - Apache Kafka Logo

Ο Apache Kafka χρησιμοποιείται για την δημιουργία data pipelines και εφαρμογών real-time streaming. Ένα data pipeline επεξεργάζεται και μετακινεί δεδομένα από το ένα σύστημα στο άλλο. Μια εφαρμογή real-time streaming είναι μια εφαρμογή που επεξεργάζεται ροές δεδομένων.

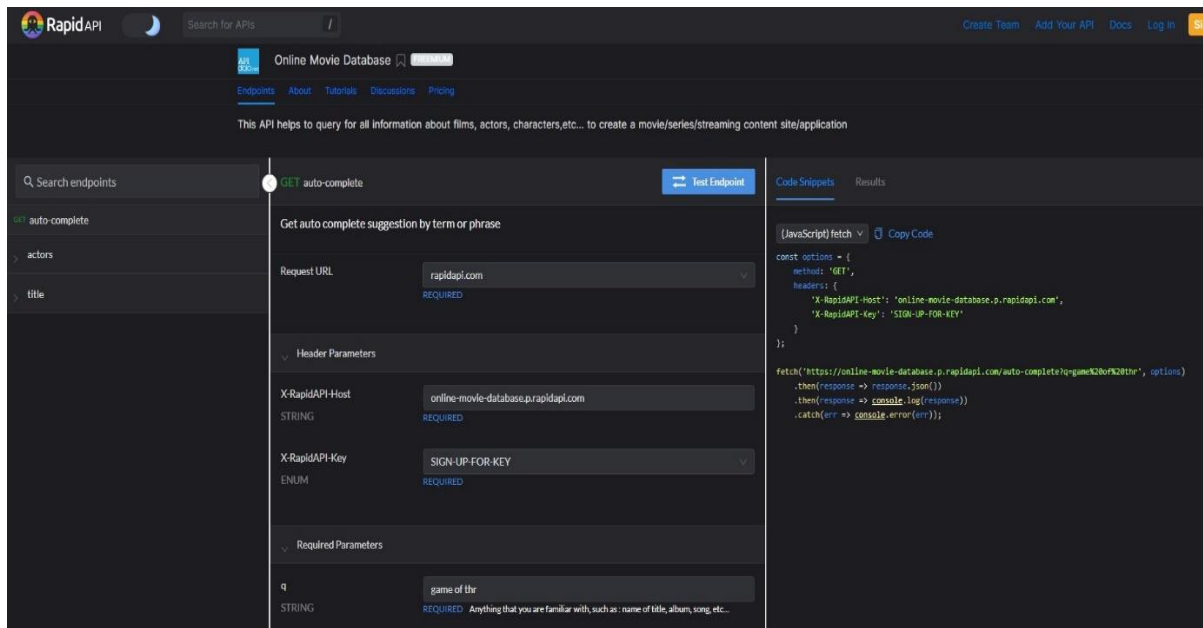
3.3.4 RapidAPI

Το RapidAPI είναι ένα Hub για API που χρησιμοποιείται από προγραμματιστές για την εύρεση, δοκιμή και σύνδεση πολλών API απλά με την χρήση ενός κλειδιού και του dashboard.

Μέσω του RapidAPI υπάρχει η δυνατότητα να αναζητηθούν έτοιμα API απαραίτητα για οποιοδήποτε εφαρμογή και παρακολουθήση αυτών μέσω ενός ενιαίου Dashboard. Επίσης υπάρχει η δυνατότητα διαμοιρασμού προσωπικών API σε όλους τους χρήστες.



Εικόνα 6 - RapidAPI Logo



Εικόνα 7 - RapidAPI IMDB Api

3.3.5 HTML – CSS – JavaScript

Η HTML, η CSS και η JavaScript είναι οι τρεις κύριες γλώσσες που χρησιμοποιούνται για την κατασκευή ιστοτόπων από τους web developers.

Η JavaScript είναι η γλώσσα προγραμματισμού, η HTML χρησιμοποιείται για την δομή του ιστοτόπου και η CSS χρησιμοποιείται για τον σχεδιασμό και την διάταξη της ιστοσελίδας (Meloni J.C. & Kyrnin J. , 2018).



Εικόνα 8 - HTML\CSS\JS

Κεφάλαιο 4

Σε αυτό το κεφάλαιο, παρουσιάζεται η αρχιτεκτονική της εφαρμογής για την αξιολόγηση του σχεδόν πραγματικού χρόνου συστήματος συστάσεων. Έπειτα, περιγράφεται βήμα προς βήμα, ο κώδικας τόσο των βασικών προγραμμάτων του συστήματος, όσο και των βοηθητικών προγραμμάτων, ενώ παράλληλα επιδεικνύονται οι λειτουργίες τους. Τέλος, περιγράφονται οι τρόποι ενσωμάτωσης και χρήσης του συστήματος από μία εφαρμογή.

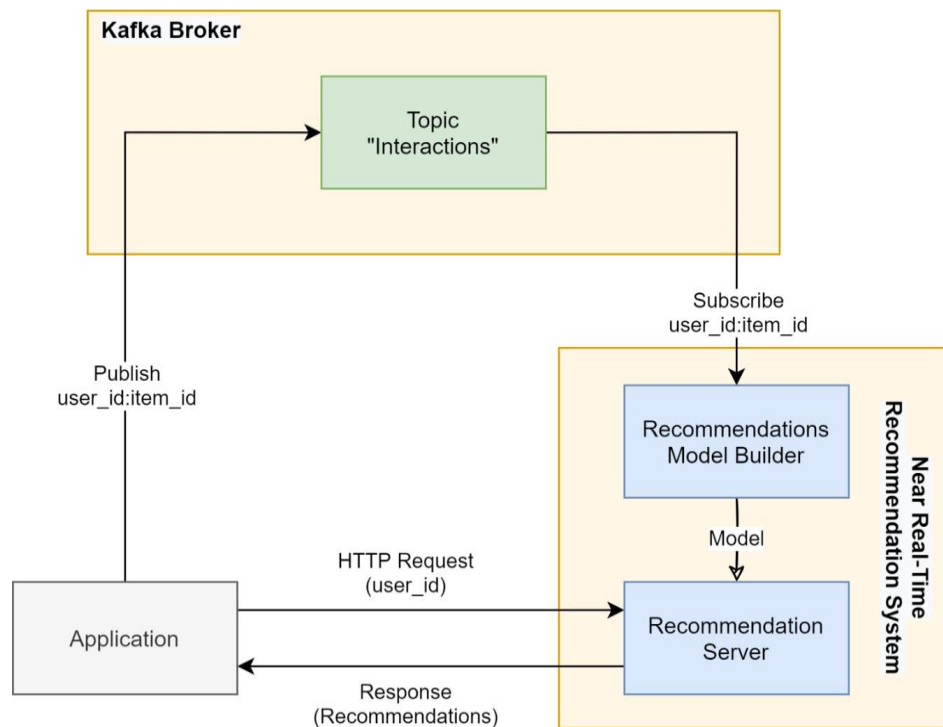
Υλοποίηση της Εφαρμογής και σύνδεση με το Σύστημα Συστάσεων

4.1 Αρχιτεκτονική του Συστήματος

Η αρχιτεκτονική όλου του συστήματος περιλαμβάνει και την εφαρμογή για την συλλογή των δεδομένων, το σύστημα συστάσεων σχεδόν πραγματικού χρόνου και ο εξυπηρετητής Kafka.

Η εφαρμογή είναι υπεύθυνη να συλλέξει τις προτιμήσεις του χρήστη και να την αναμεταδώσει στο RS μέσω του εξυπηρετητή Kafka και να συλλέξει και να παρουσιάσει τα δεδομένα εξόδου του RS. Το RS απαρτίζεται από το πρόγραμμα που συλλέγει τα δεδομένα για την εκπαίδευση του μοντέλου συστάσεων και το πρόγραμμα το οποίο δέχεται αιτήματα για παραγωγή συστάσεων, παράγει και επιστρέφει συστάσεις χρησιμοποιώντας το μοντέλο. Τέλος ο εξυπηρετητής Kafka, που είναι υπεύθυνος για την συνεχή ροή δεδομένων ανατροφοδότησης (Grivas, P. ,2021).

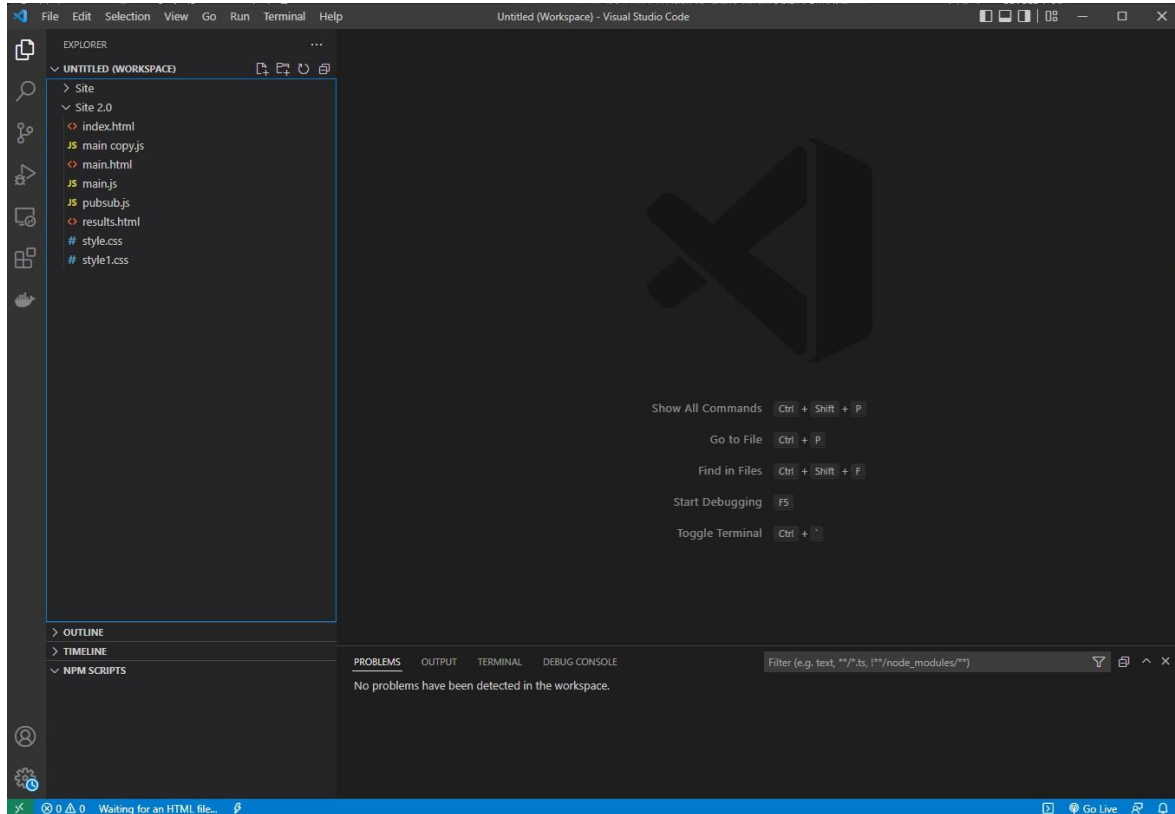
Στην Εικόνα 8 βλέπουμε το διάγραμμα αρχιτεκτονικής από την πτυχιακή εργασία “Αξιοποιώντας Συνεχούς Ροής Δεδομένα για την Δημιουργία ενός Σχεδόν Πραγματικού Χρόνου Συστήματος Συστάσεων” του Πασχάλη Γρίβα



Εικόνα 9 - Διάγραμμα Αρχιτεκτονικής του Συστήματος Συστάσεων

4.2 Ανάλυση Εφαρμογής

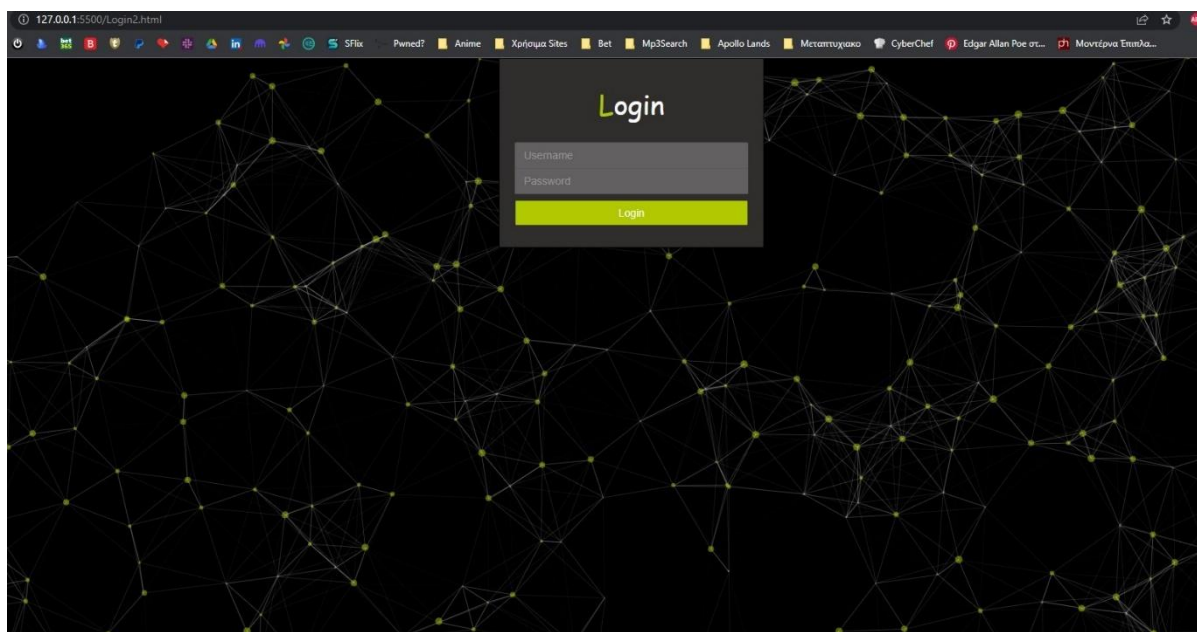
Όπως αναφέραμε στο προηγούμενο κεφάλαιο για την δημιουργία του συστήματος χρησιμοποιήθηκε το προγραμματιστικό εργαλείο VScode.



Εικόνα 10 - Περιβάλλον Εργασίας VScode

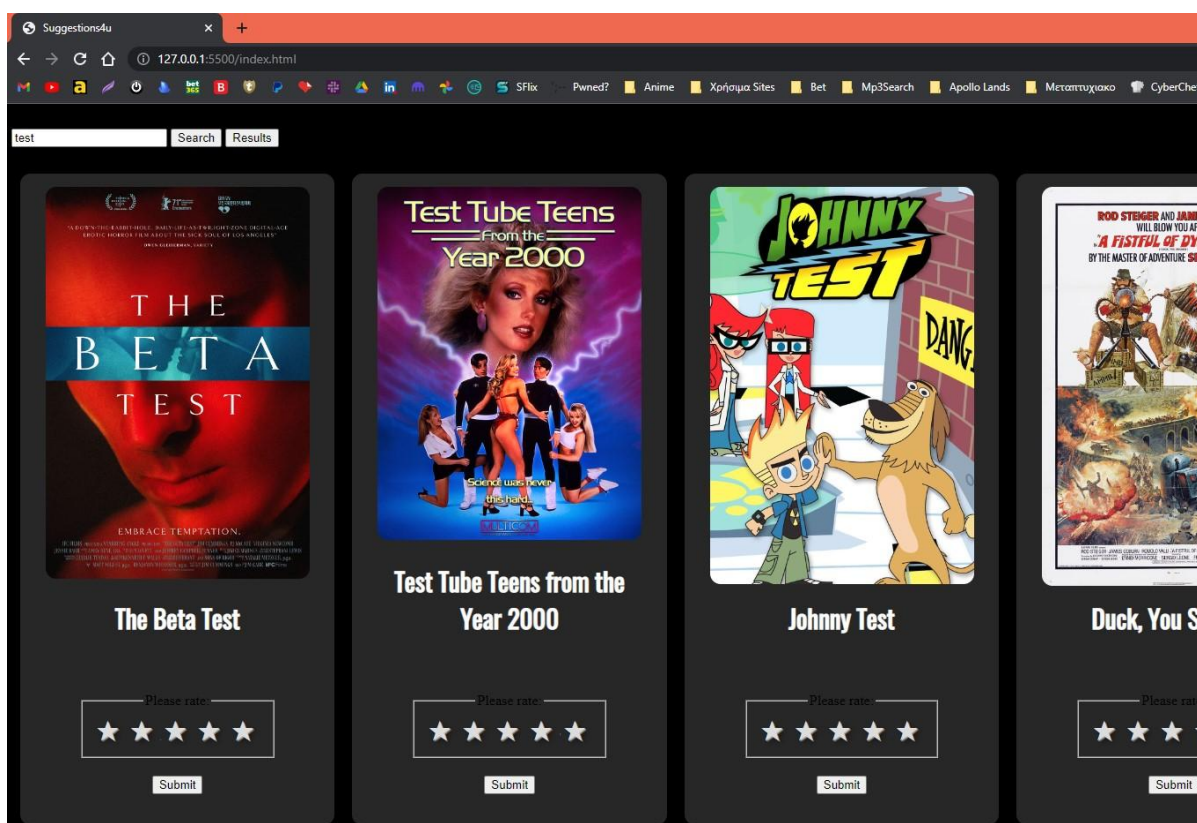
Το πληροφοριακό σύστημα που δημιουργήθηκε είναι στην ουσία μια ιστοσελίδα και για την δημιουργία της χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java η οποία υποστηρίζεται από το VScode.

Στην κεντρική σελίδα το σύστημα ζητά από τον χρήστη τα στοιχεία του έτσι ώστε να του δώσει έναν μοναδικό αριθμό (ID).



Εικόνα 11 - Εφαρμογή αξιολόγησης RS, αρχική σελίδα

Στην συνέχεια ζητείτε από τον χρήστη να εισάγει τις ταινίες που επιθυμεί και να τις αξιολογήσει με αστέρια από το 1 έως το 5.

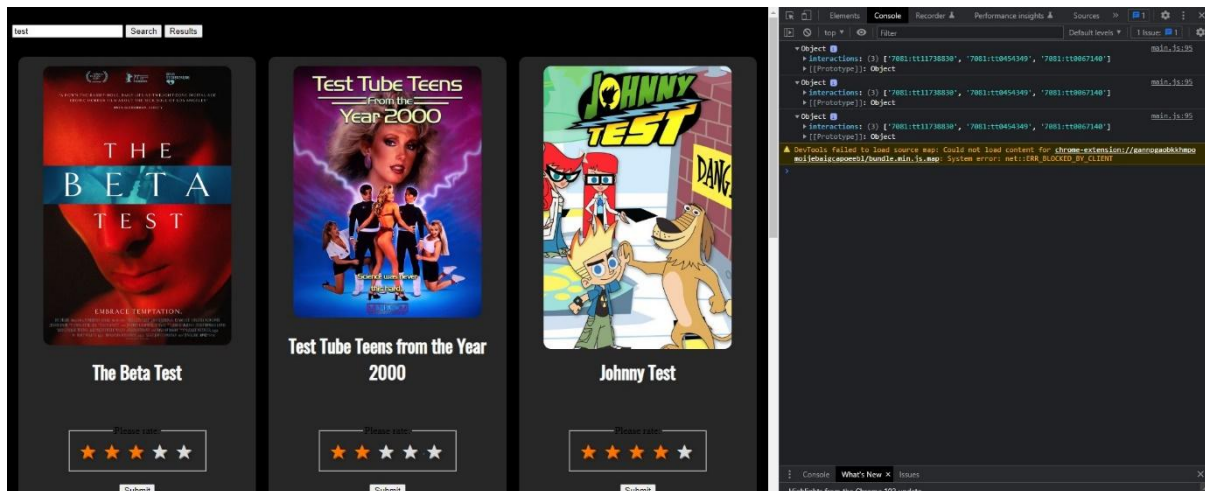


Εικόνα 12 - Εφαρμογή αξιολόγησης RS, αξιολόγηση ταινιών

Η αναζήτηση και προβολή των ταινιών προς αξιολόγηση γίνεται με την χρήση του RapidApi σε μορφή Javascript για την δυναμική ενημέρωση του περιεχομένου. Στην συνέχεια ο χρήστης μπορεί να βαθμολογήσει όποια ταινία του εμφανιστεί με τα αντίστοιχα αστέρια που επιθυμεί.

Πατώντας submit το σύστημα ελέγχει αν ο χρήστης έχει επιλέξει 3 ή περισσότερα αστέρια για την αξιολόγηση του και στην συνέχεια κάνει της εξής ενέργειες:

- Αν ο χρήστης επιλέξει 2 αστέρια ή λιγότερα τότε το σύστημα δεν υπολογίζει την αξιολόγηση του
- Αν ο χρήστης επιλέξει 3 ή περισσότερα αστέρια τότε το σύστημα μέσω μιας function η οποία κάνει Post σε ένα topic το ID της ταινίας και το ID του χρήστη έτσι ώστε να μπορεί το σύστημα συστάσεων να διαβάσει τις επιλογές του εκάστοτε χρήστη.

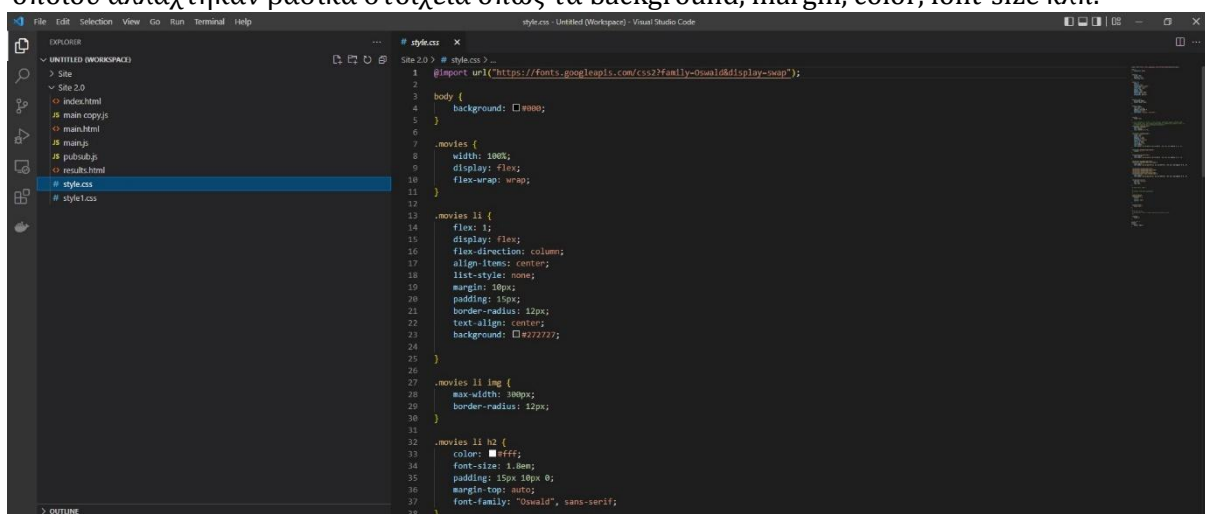


Εικόνα 13 - Εφαρμογή αξιολόγησης RS, Publish to Topic

Αφού ο χρήστης ολοκληρώσει την αξιολόγηση του μπορεί κάνοντας κλικ στην επιλογή suggest me να δει τις προτάσεις του συστήματος συστάσεων και να τις αξιολογήσει κατά πόσο αυτές τον ικανοποιούν.

Css

Για να γίνει πιο προσιτό προς τον χρήστη το σύστημα χρησιμοποιήθηκε κώδικας css μέσω του οποίου αλλάχτηκαν βασικά στοιχεία όπως τα background, margin, color, font-size κλπ.



Εικόνα 14 - Εφαρμογή αξιολόγησης RS, Css code

4.3 Εγκατάσταση συστήματος συστάσεων σε λειτουργικό Linux με την χρήση του Docker

Το σύστημα συστάσεων προς αξιολόγηση είναι μια εφαρμογή που εκτελείται σε περιβάλλον Linux και για την εγκατάστασή χρησιμοποιήθηκε η πλατφόρμα Docker όπως προαναφέρθηκε στο προηγούμενο κεφάλαιο.

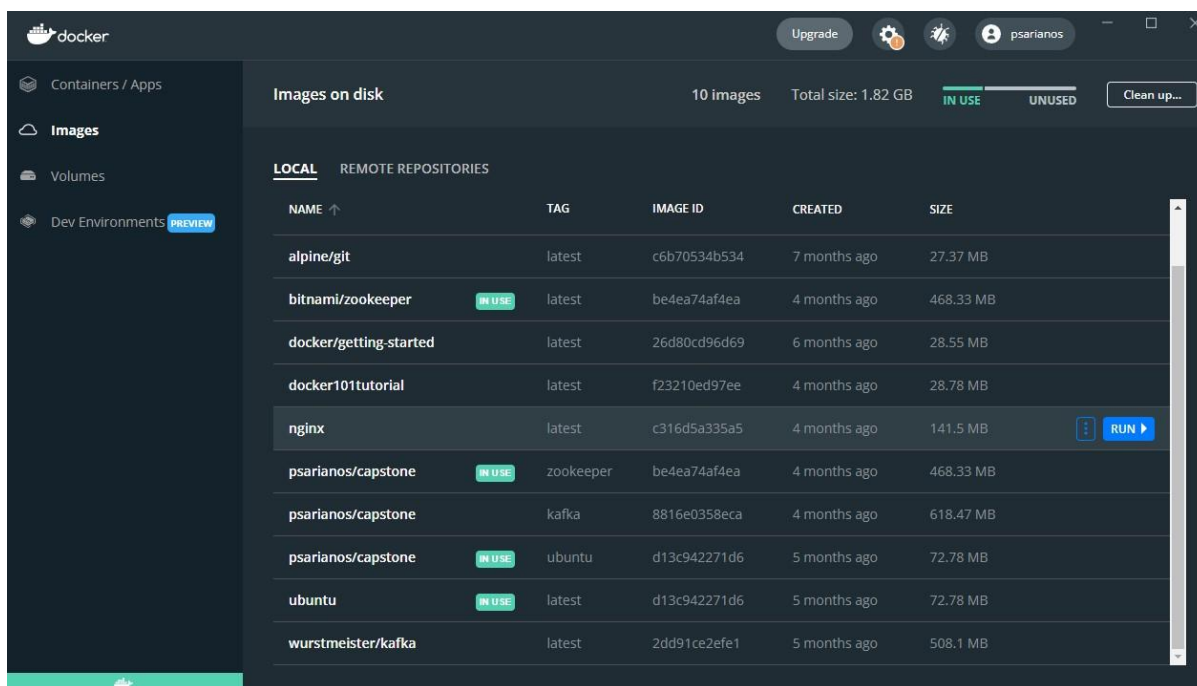
Αρχικά δημιουργήθηκε ένας λογαριασμός στην πλατφόρμα και ένα repository με την ονομασία capstone. Μέσα στο συγκεκριμένο repository δημιουργήθηκαν 3 images. Τα 2 για την συνδεσιμότητα μέσω apache kafka των δύο συστημάτων και 1 με το λειτουργικό (linux) στο οποίο έγινε η εγκατάσταση και εκτέλεση του συστήματος συστάσεων.

The screenshot displays the Docker Hub interface for the repository **psarianos / capstone**. It includes a header with the repository name, a description placeholder, and the last push time (2 months ago). A 'Docker commands' section provides a command to push a new tag. Below this, the 'Tags and Scans' section lists three tags: **ubuntu**, **zookeeper**, and **kafka**, each with its OS, pull/push status, and time. A 'VULNERABILITY SCANNING - DISABLED' warning is also present. The 'Automated Builds' section explains how to connect to GitHub or Bitbucket for automatic builds and includes an 'Upgrade to Pro' button.

TAG	OS	PULLED	PUSHED
ubuntu	linux	2 months ago	2 months ago
zookeeper	linux	4 months ago	4 months ago
kafka	linux	4 months ago	4 months ago

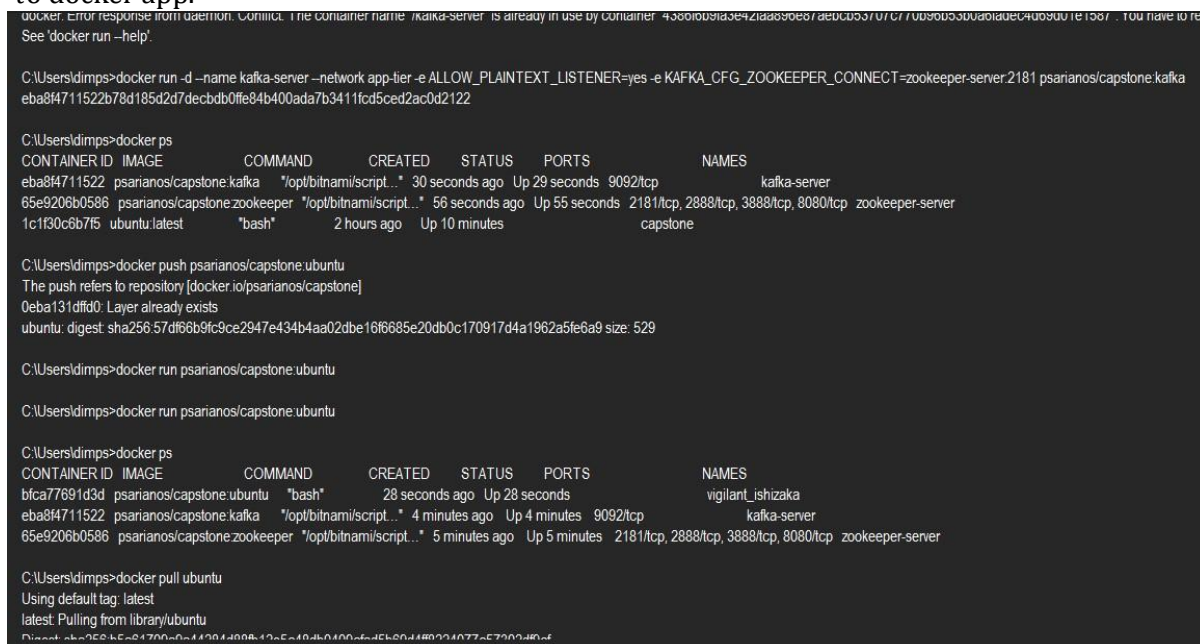
Εικόνα 15 - Docker Web

Για την χρήση της πλατφόρμας Docker και την δημιουργία των images εγκαταστάθηκε σε λειτουργικό Windows το Docker App μέσω του οποίου μπορεί να γίνει δημιουργία και διαχείριση των repositories και των images.



Εικόνα 16 - Docker App

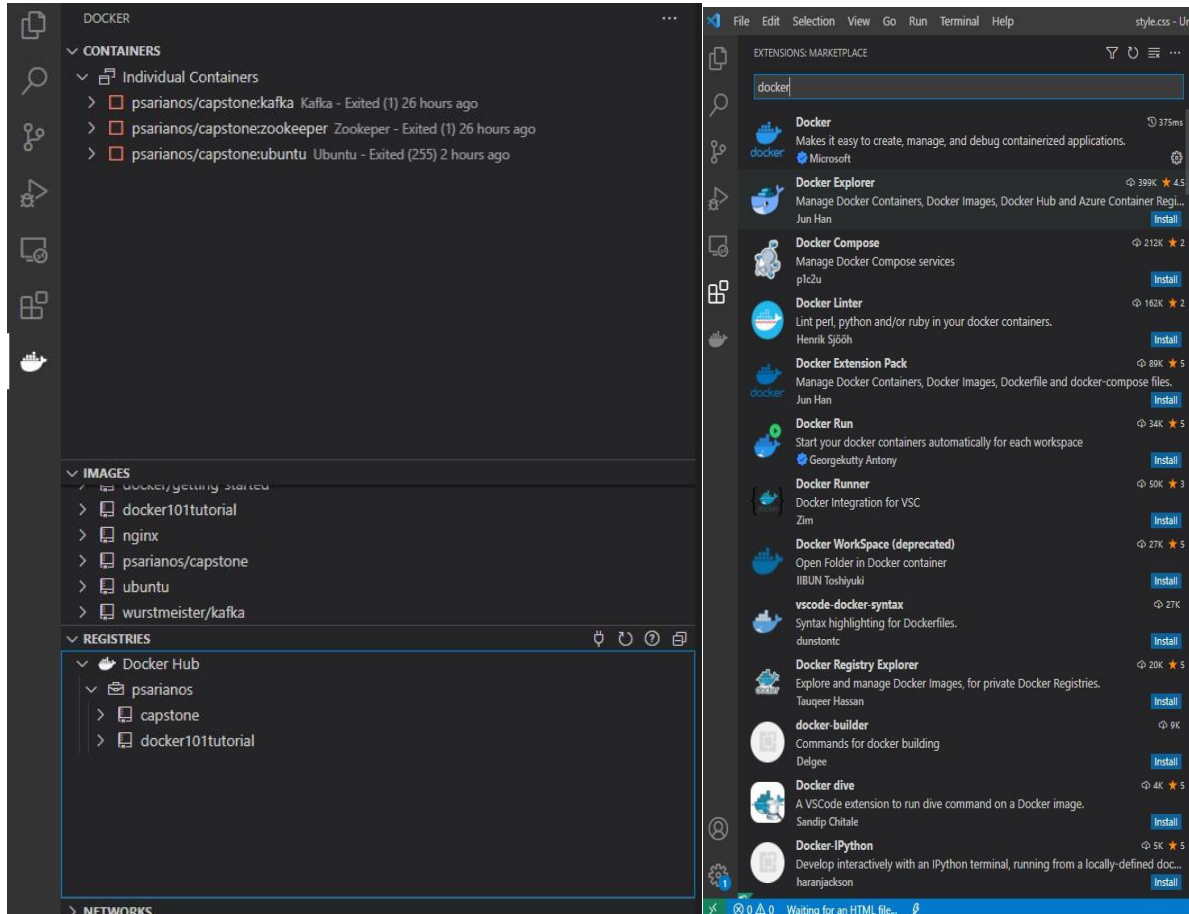
Για την εκτέλεση εντολών στο περιβάλλον των Linux που έχουν δημιουργηθεί στα images του repository χρησιμοποιείται το command line και οι αντίστοιχες επιλογές που παρέχονται από το docker app.



Εικόνα 17 - Docker in Windows Comand Line

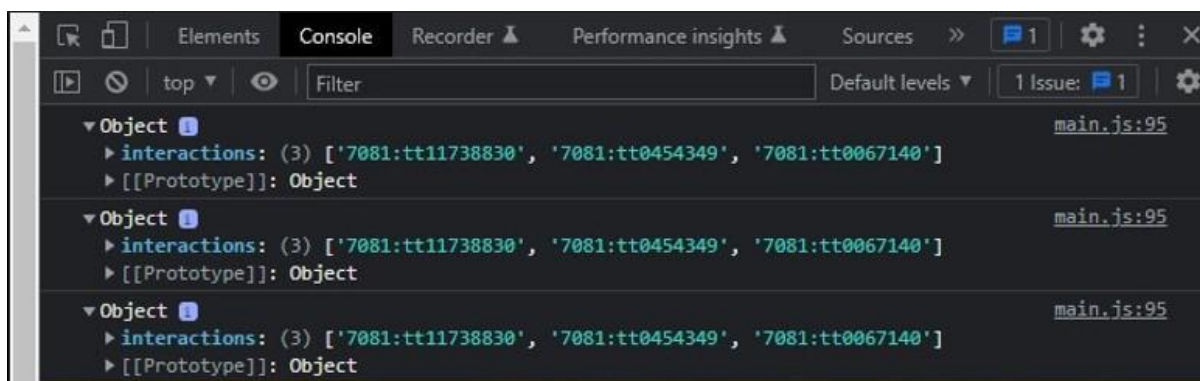
4.4 Σύνδεση εφαρμογής με το σύστημα συστάσεων

Στο VScode δίνεται επίσης και η δυνατότητα να συνδεθεί μέσω του docker extension το docker hub με το ίδιο το VSCode.



Εικόνα 18 - VScode Docker Extension

Όπως προαναφέραμε σε προηγούμενο κεφάλαιο η εφαρμογή για την αξιολόγηση του συστήματος δεν είναι άμεσα συνδεδεμένη με το RS αλλά η σύνδεση των δεδομένων πραγματοποιείται μέσω ενός broker. Έτσι η εφαρμογή, όταν επιλέξει ο χρήστης να κάνει μια αξιολόγηση δημιουργεί μια εγγραφή τύπου Post σε ένα Topic το οποίο έχουμε ονομάσει interactions με παραμέτρους το ID του χρήστη και το ID της ταινίας (UserId:MovieID).



Εικόνα 19 - Published Objects, UserId:ItemId

Στην συνέχεια το σύστημα συστάσεων “διαβάζει” σε αυτό το Topic και τραβάει τα δεδομένα προς επεξεργασία.

Όταν ο χρήστης επιθυμήσει να δεχτεί συστάσεις από το σύστημα πατώντας το κουμπί suggest me τότε το σύστημα, στέλνει ένα αίτημα τύπου GET (HTTP Request) προς το πρόγραμμα παραγωγής συστάσεων (Recommendation Server), με παραμέτρους το ID του χρήστη και το πλήθος των συστάσεων που πρέπει να επιστραφούν. Στην συνέχεια η εφαρμογή λαμβάνει από το RS τα αποτελέσματα σε μορφή JSON και της εμφανίζει στον χρήστη.

```
{
  "recommended_items": [
    tt442356,
    tt881222,
    tt881167,
    tt114555
  ],
  "user_id": 23
}
```

Εικόνα 20 - RS Json Return

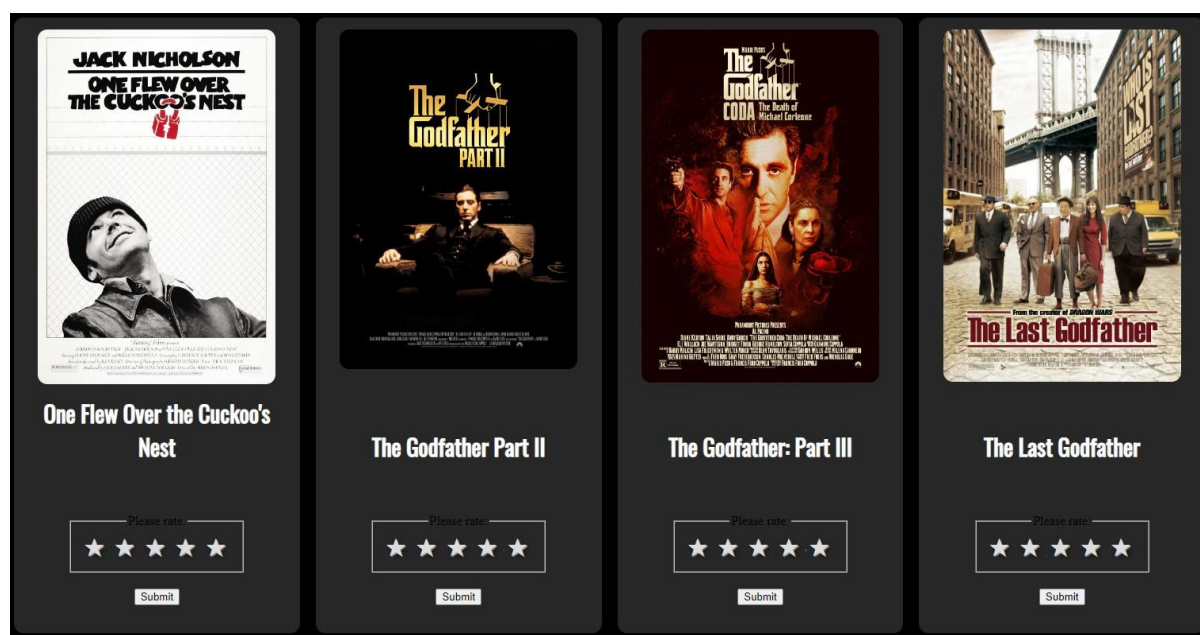
Κεφάλαιο 5

Παρουσίαση και ερμηνεία αποτελεσμάτων

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο η εγκατάσταση του συστήματος συστάσεων έγινε με χρήση του docker και για να τρέχει το σύστημα ήταν απαραίτητο να δουλεύει και ο τοπικός υπολογιστής. Κατά την χρήση της εφαρμογής εκτελούνταν παράλληλα και το σύστημα συστάσεων με τις αξιολογήσεις των χρηστών και τις συστάσεις να γίνονται σε πραγματικό χρόνο και μόνο κατά την χρήση του Η/Υ που έτρεχαν τα δύο συστήματα. Αυτό είχε σαν αποτέλεσμα όταν απενεργοποιούνταν ο Η/Υ να χάνονταν τα δεδομένα και στην επόμενη εκκίνηση του συστήματος να χρειαζόντουσαν νέα.

Κατά την διάρκεια υλοποίησης του capstone project έγινε η αγορά ενός Domain και έγινε εγκατάσταση του wordpress. Στην συνέχεια μια αρχική έκδοση της εφαρμογής ανέβηκε στην σελίδα αλλά καθώς δεν ήταν δυνατό να βρεθεί ένας linux server το οποίο απαιτεί το σύστημα συστάσεων για να εκτελεστεί δεν ήταν δυνατή η αξιοποίηση του και εγκαταλείφθηκε.

Η αξιολόγηση έγινε από μόνο 15 χρήστες τοπικά αφού δεν υπήρξε η δυνατότητα και οι πόροι να ανέβει το σύστημα στο διαδίκτυο έτσι ώστε να είναι εύκολα προσβάσιμο από μεγάλο αριθμό χρηστών. Οι αξιολογήσεις των χρηστών ως προς τις συστάσεις που έδινε το RS αποθηκεύονται συνολικά σε μια μεταβλητή στον κώδικα της εφαρμογής και στην συνέχεια υπολογίζεται ο μέσος όρος των βαθμολογιών από 1-5 αστέρια.



Εικόνα 21 - Εφαρμογή αξιολόγησης RS, αξιολόγηση συστάσεων

Όπως προαναφέραμε η αξιολόγηση έγινε από 15 χρήστες και ο μέσος όρος βαθμολογίας των συστάσεων ήταν 3,7. Εδώ πρέπει να αναφερθεί ότι υπήρξαν και άλλα προβλήματα τα οποία παρουσιάζονται στο επόμενο κεφάλαιο.

Συμπεράσματα – Προτάσεις – Προβλήματα

6.1 Προβλήματα

Κατά την διάρκεια της υλοποίησης του project υπήρξαν διάφορα προβλήματα όπου κάποια αντιμετωπίστηκαν ενώ κάποια άλλα όχι. Τα προβλήματα που παρουσιάστηκαν ήταν τα εξής:

- ❖ Καθώς το σύστημα συστάσεων δεν ήταν εγκατεστημένο κάπου και η χρησιμοποίηση του δεν ήταν εκ των προτέρων εφικτή η εύρεση ενός λειτουργικού συστήματος όπου θα ήταν δυνατή η εγκατάσταση του και η αδιάκοπη λειτουργία του δεν ήταν εφικτή. Το λειτουργικό απαιτεί λειτουργικό σύστημα linux και η επεξεργασία μεγάλων δεδομένων πολλούς πόρους του συστήματος που θα γινόταν η εγκατάσταση και λειτουργία του (linux server όπου θα χρησιμοποιούταν μόνο για το σκοπό της εκτέλεσης του RS).
- ❖ Το συγκεκριμένο API από το RapidAPI που χρησιμοποιήσαμε όπου τραβάει στοιχεία για ταινίες από το IMDB διατίθεται σε δωρεάν χρήση μόνο για 500 αναζητήσεις ταινιών το μήνα και από εκεί και πέρα υπάρχει χρέωση.
- ❖ Η χρήση του Docker απαιτεί μεγάλη προσοχή καθώς τα Images στα repositories μπορούν να αντικατασταθούν και να χαθούν όλα τα δεδομένα (στην περίπτωση μας χρειάστηκε να εγκατασταθεί στο image των linux το λειτουργικό σύστημα, το σύστημα συστάσεων καθώς και τα απαραίτητα προγράμματα, όπως η γλώσσα προγραμματισμού, τρεις φορές).
- ❖ Καθώς το σύστημα συστάσεων προτείνει ταινίες στον χρήστη σε λαμβάνοντας υπόψιν κριτικές άλλων χρηστών, για να αξιολογήσει ο χρήστης τις ταινίες που του προτείνονται χρειάζεται κάποιο χρονικό διάστημα καθώς υπάρχει μεγάλη πιθανότητα και είναι και επιθυμητό ο χρήστης να μην έχει παρακολουθήσει την προτεινόμενη ταινία. Στην εφαρμογή μας όμως δεν έχει προβλεφθεί αποθήκευση των προτάσεων για μεταγενέστερη αξιολόγηση.

6.2 Προτάσεις

Για την καλύτερη αξιολόγηση του συστήματος συστάσεων που ασχοληθήκαμε ή ακόμα και την αξιολόγηση και άλλων RS θα μπορούσαν να γίνουν οι παρακάτω βελτιώσεις – υλοποιήσεις.

- ❖ Εύρεση ενός linux server και εγκατάσταση του RS έτσι ώστε να επιτευχθεί η αδιάκοπη λειτουργία του και στην συνέχεια ανέβασμα της εφαρμογής αξιολόγησης σε έναν domain και με ελάχιστες τροποποιήσεις να μπορεί να επιτευχθεί η αξιολόγηση για οποιοδήποτε RS
- ❖ Αγορά του API μέσω RapidAPI για αναζήτηση πέρα του δωρεάν ορίου.
- ❖ Με το ανέβασμα της εφαρμογής σε κάποιο domain ο χρήστης θα μπορούν πολύ απλά να εισέρχονται στο σύστημα με τα στοιχεία τους και θα υπάρχει η δυνατότητα να αποθηκεύονται πολύ εύκολα οι συστάσεις του RS για μεταγενέστερη αξιολόγηση.
- ❖ Υπάρχει η δυνατότητα με μικρές αλλαγές (πχ αλλαγή μόνο του API και μικρών σημείων του κώδικα) να ενταχθούν στην εφαρμογή και άλλα στοιχεία όπως ταινίες, ειδήσεις κλπ όπου θα μπορούσαν να επιφέρουν μια διαφορετική αξιολόγηση του RS.

6.3 Συμπεράσματα / Αξιολόγηση

Τα συστήματα συστάσεων είναι μια ισχυρή νέα τεχνολογία η οποία συμβάλει στο να παρέχει πρόσθετη αξία για μια επιχείρηση με την αξιοποίηση των βάσεων δεδομένων των χρηστών της. Τα συστήματα αυτά βοηθούν τους χρήστες να βρουν αντικείμενα ή υπηρεσίες σύμφωνα με τις προτιμήσεις τους μια επιχείρηση. Έτσι, βοηθούν μια επιχείρηση δημιουργώντας περισσότερες πωλήσεις. Τα συστήματα συστάσεων έγιναν ένα κρίσιμο εργαλείο στο ηλεκτρονικό εμπόριο με ταχύτατους ρυθμούς καθώς ο όγκος των δεδομένων χρηστών σε εταιρικές βάσεις είναι πάρα πολύ μεγάλος και ο ρυθμός ανάπτυξης αυτών είναι τεράστιος αφού όλο και περισσότεροι χρήστες χρησιμοποιούν το διαδίκτυο. Από την άλλη μεριά αυτό δημιουργεί και ένα μεγάλο πρόβλημα στα συστήματα αυτά μιας και δεν έχουν σχεδιαστεί να επεξεργάζονται τόσο μεγάλο όγκο δεδομένων γι' αυτό απαιτούνται νέες τεχνολογίες που μπορούν να βελτιώσουν δραματικά την επεκτασιμότητα των συστημάτων συστάσεων.

Μέχρι σήμερα υπάρχουν διάφορες τεχνικές για την δημιουργία συστάσεων όπως συστήματα βασισμένα στην γνώση, στην συνεργασία και στο περιεχόμενο μέσω διαφορετικών αλγορίθμων. Ένα βασικό πρόβλημα όμως που εντοπίζεται σε όλες τις τεχνικές είναι η δυναμική ενημέρωσή τους και έτσι η απόδοσή τους, δηλαδή το πόσο ακριβής είναι στις συστάσεις τους, φθίνει. Η λύση σε αυτό το πρόβλημα και αυτή που προσπαθήσαμε να δώσουμε στο συγκεκριμένο project, είναι η τροφοδότηση του συστήματος με συνεχούς ροής δεδομένα. Έτσι με αυτά τα δεδομένα μπορεί να κατασκευαστεί ένα συγκεκριμένο μοντέλο συστάσεων αλληλεπίδρασης χρήστη – αντικειμένου με την επανεκπαίδευση όποτε αυτό λαμβάνει νέα δεδομένα. Όπως είναι εύκολα αντιληπτό εδώ δημιουργείται ένα μεγάλο πρόβλημα αφού ο όγκος των δεδομένων κάθε φορά θα μεγαλώνει μέχρι να γιγαντωθεί αρκετά γρήγορα και άρα οι πόροι και ο χρόνος που θα χρειάζεται το σύστημα θα γίνονται απαγορευτικοί. Η λύση που δόθηκε στο συγκεκριμένο σύστημα που αξιολογήσαμε ήταν να δίνεται στο σύστημα ένας συγκεκριμένος αριθμός εκπαίδευσης αντικειμένων χρήστη – προϊόν όπου κάθε φορά αυτά θα είναι και τα πιο πρόσφατα του συστήματος αξιολόγησης. Έτσι το σύστημα συστάσεων ενημερώνεται από το σύστημα αξιολόγησης σε σχεδόν πραγματικό χρόνο από την άλλη όμως υπάρχει μια μείωση στην ακρίβεια του συστήματος μιας και δεν υπολογίζει παλαιότερες προτιμήσεις των χρηστών.

Όπως αναφέρεται και παραπάνω η αξιολόγηση του συστήματος δεν έγινε με το καλύτερο δυνατό τρόπο λόγω των προαναφερθέντων προβλημάτων. Έτσι εν τέλει το project επικεντρώθηκε περισσότερο στην δημιουργία μια εφαρμογής που θα μπορεί να χρησιμοποιηθεί για την αξιολόγηση (με μικρές τροποποιήσεις του κώδικα της) οποιουδήποτε συστήματος συστάσεων.

Ο κώδικας της εφαρμογής

Κώδικας σε HTML για την αρχική σελίδα για το Login του χρήστη

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Animated Background Login Screen</title>
  <link rel="stylesheet" href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css'><
  link rel="stylesheet" href="./style3.css">
</head>
<body>
<!-- partial:index.partial.html -->
<div class="container">
  <div id="login-box">
    <div class="logo">
      
      <h1 class="logo-caption"><span class="tweak">L</span>ogin</h1>
    </div><!-- /.logo -->
    <div class="controls">
      <input type="text" name="username" placeholder="Username" class="form-control" />
      <input type="text" name="password" placeholder="Password" class="form-control" />
      <button type="button" class="btn btn-default btn-block btn-custom" onclick="WriteToFile(); window.location.href='index.html';" >Login</button>
      <!-- <button onclick="WriteToFile(); window.location.href='index.html';">Insert</button> -->
    </div>
  </div>
</div>
<div id="particles-js"></div>
<!-- partial -->
<script src='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css'></script>
<script src='https://code.jquery.com/jquery-1.11.1.min.js'></script><script src="./script.js"></script>
<script src="main.js"></script>
</body>
</html>
```

Ο κώδικας σε HTML για την σελίδα που ανακατευθύνεται ο χρήστης μετά το Login

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Suggestions4u</title>
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scrip
t>

</head>

<body>
<div class="topnav clearfix">
  <a class="active"></a>
  <input id="searchId" type="text" placeholder="Search..">
  <button onclick="returnText()">Search</button>
  <button onclick="window.location.href='results.html'">Suggest Me</button>

</div>
<div class="movies"></div>

<script src="main.js"></script>
</body>

</html>
```

Ο κώδικας σε javascript για την αναζήτηση και προβολή των ταινιών. Επίσης σε αυτό το αρχείο εκτελείται και μια Function για το Publish userId:ItemId σε ένα Topic.

```
var userId;
const ps = pubSub();

function WriteToFile() {
  alert(userId = Math.floor(Math.random() * 10000) + 1);
  localStorage.setItem("user", userId);
}

function returnText() {
```

```

let SearchString = document.getElementById("searchId").value;

fetch(`https://imdb8.p.rapidapi.com/auto-complete?q=${SearchString}`, {
  "method": "GET",
  "headers": {
    "x-rapidapi-host": "imdb8.p.rapidapi.com",
    "x-rapidapi-key": "1138d29f44msha71675d708c5410p15d291jsn680473fc88a9"
  }
})
.then(response => response.json())
.then(data => {
  const list = data.d;

  list.map((item) => {
    const id = item.id;
    const name = item.l;
    const photo = item.i.imageUrl;
    const movie = `-  <h2>${name}</h2>

    <div id="status"></div>
    <form id="ratingForm">
      <fieldset class="rating">
        <legend>Please rate:</legend>
        <input type="radio" id="${id}_star5" name="${id}_rating"
value="${id}_star_5" /><label for="${id}_star5" title="Rocks!">5 stars</label>
        <input type="radio" id="${id}_star4" name="${id}_rating"
value="${id}_star_4" /><label for="${id}_star4" title="Pretty good">4
stars</label>
        <input type="radio" id="${id}_star3" name="${id}_rating"
value="${id}_star_3" /><label for="${id}_star3" title="Meh">3 stars</label>
        <input type="radio" id="${id}_star2" name="${id}_rating"
value="${id}_star_2" /><label for="${id}_star2" title="Kinda bad">2
stars</label>
        <input type="radio" id="${id}_star1" name="${id}_rating"
value="${id}_star_1" /><label for="${id}_star1" title="Sucks big time">1
star</label>
      </fieldset>
      <div class="clearfix"></div>
      <button id="button_${id}" type="button" value="send" class="btn btn-
primary" onclick="Something(this.id)">Submit</button>

    </form>

    </li>`
    document.querySelector('.movies').innerHTML += movie;
  })
})

```

```

    })
    .catch(err => {
        console.error(err);
    });
}

function Something(id) {
    let form_id = id.substring(7);
    let text1 = form_id;
    let text2 = "_rating";
    let name = text1.concat(text2);

    radio_id = document.querySelector('input[name="' + name +
    '"]:checked').value;

    var rating_stars = radio_id.substr(radio_id.length - 1);
    var movie_id = radio_id.substr(0, radio_id.length - 7);

    if (rating_stars > 2) {
        var user_id = localStorage.getItem("user");
        alert(user_id + ' ' + movie_id + ' ' + rating_stars);

        ps.subscribe('interactions', user_id + ':' + movie_id);
    }
}

```

Ο κώδικας σε CSS για την μορφοποίηση της προβολής των αντικειμένων.

```

@import url("https://fonts.googleapis.com/css2?family=Oswald&display=swap");

body {
    background: #000;
}

.movies {
    width: 100%;
    display: flex;
    flex-wrap: wrap;
}

.movies li {
    flex: 1;
    display: flex;
    flex-direction: column;
    align-items: center;

```

```

    list-style: none;
    margin: 10px;
    padding: 15px;
    border-radius: 12px;
    text-align: center;
    background: #272727;
}

.movies li img {
    max-width: 300px;
    border-radius: 12px;
}

.movies li h2 {
    color: #fff;
    font-size: 1.8em;
    padding: 15px 10px 0;
    margin-top: auto;
    font-family: "Oswald", sans-serif;
}

.rating {
    float: left;
}

/* :not(:checked) is a filter, so that browsers that don't support :checked
don't
    follow these rules. Every browser that supports :checked also supports
:not(), so
    it doesn't make the test unnecessarily selective */
.rating:not(:checked)>input {
    position: absolute;
    top: -9999px;
    clip: rect(0, 0, 0, 0);
}

.rating:not(:checked)>label {
    float: right;
    width: 1em;
    padding: 0 .1em;
    overflow: hidden;
    white-space: nowrap;
    cursor: pointer;
    font-size: 200%;
    line-height: 1.2;
    color: #ddd;
    text-shadow: 1px 1px #bbb, 2px 2px #666, .1em .1em .2em rgba(0, 0, 0, .5);
}

```

```

}

.rating:not(:checked)>label:before {
  content: '★ ';
}

.rating>input:checked~label {
  color: #f70;
  text-shadow: 1px 1px #c60, 2px 2px #940, .1em .1em .2em rgba(0, 0, 0, .5);
}

.rating:not(:checked)>label:hover,
.rating:not(:checked)>label:hover~label {
  color: gold;
  text-shadow: 1px 1px goldenrod, 2px 2px #B57340, .1em .1em .2em rgba(0, 0, 0, .5);
}

.rating>input:checked+label:hover,
.rating>input:checked+label:hover~label,
.rating>input:checked~label:hover,
.rating>input:checked~label:hover~label,
.rating>label:hover~input:checked~label {
  color: #ea0;
  text-shadow: 1px 1px goldenrod, 2px 2px #B57340, .1em .1em .2em rgba(0, 0, 0, .5);
}

.rating>label:active {
  position: relative;
  top: 2px;
  left: 2px;
}

/* end of Lea's code */

/*
 * Clearfix from html5 boilerplate
 */

.clearfix:before,
.clearfix:after {
  content: " ";
  /* 1 */
  display: table;
  /* 2 */
}

```

```

.clearfix:after {
    clear: both;
}

/*
 * For IE 6/7 only
 * Include this rule to trigger hasLayout and contain floats.
 */

.clearfix {
    *zoom: 1;
}

/* my stuff */
#status,
button {
    margin: 20px 0;
}

```

Ο κώδικας για το HTTPS αίτημα GET για να ληφθούν τα αποτελέσματα του RS

```

new Ajax.Request( 'http://127.0.0.1/', {
    method: 'get',
    parameters: { 'userId': '4' },
    onSuccess: function(response){
        alert(response.responseText);
    },
    onFailure: function(){
        alert('ERROR');
    }
});

```

Κεφάλαιο 7

Βιβλιογραφικές Αναφορές

Balabanovic & Shoham, <https://www.ischool.utexas.edu/>, (1997). ΓΡΑΦΟΘΕΩΡΗΤΙΚΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΩΝ ΣΥΣΤΑΣΗΣ [13]

Grivas, P. (2021). A Near Real-Time Python Recommendation System. <https://github.com/p17griv>

Jannach D., Zanker M., Felfenrig A. & Friedrich J. (2011). Recommender Systems: An Introduction

Falk K. (2019). Practical Recommender Systems

Tarkoma S. (2012). Publish / Subscribe Systems: Design and Principles

Narkhede N., Shapira G. & Palino T. (2017) Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale. O' Reilly Media.

Waud E. (2018). Docker Quick Start Guide

Debolt V. (2005). Integrated HTML and CSS

Meloni J.C. & Kyrnin J. (2018). HTML, CSS, and JavaScript All in One,

Hu, Y., Koren, Y., & Volinsky, C.. (2008). Collaborative Filtering for Implicit Feedback Datasets. Pisa, Italy: Eighth IEEE International Conference on Data Mining. pp. 263-272. doi: 10.1109/ICDM.2008.22

Oard, D.W., & Kim, J. (1998). Implicit Feedback for Recommender Systems. College Park, Maryland: Proc. 5th DELOS Workshop on Filtering and Collaborative Filtering.

Ockert J. du Preez (2019). Visual Studio 2019 in depth.

Griffiths I., Flanders J. & Chris Sells Ch. (2003). Mastering Visual Studio .NET

Ekstrand M. D., Riedl T. J., Konstan A. J. (2011). Collaborative Filtering Recommender Systems

Gabbaro S. A. (2007). Web Application Design and Implementation

Vora P. (2009). Web Application Patterns

Pureval S. (2014). Learning Web App Development: Build Quickly with Proven JavaScript Techniques

Messenlehner B., Coleman J. (2014). Building Web Apps with WordPress: WordPress as an Application Framework

Andrew G. Psaltis (2017). Streaming Data: Understanding the real-time pipeline