



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ  
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΙΔΙΚΕΥΣΗ: “ΠΡΟΗΓΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ  
ΣΥΣΤΗΜΑΤΑ”

## Still Series: Recommender System βασισμένο σε Collaborative Filtering Προσέγγιση

---

ΑΠΟΘΗΚΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΕΥΦΥΪΑ

Επιβλέπουσα Καθηγήτρια: Μαρία Χαλκίδη

**Ιωάννα Κανδή, Κωνσταντίνος Μαυρογιώργος**

ΑΜ: ME2136, ME2144

Email: {ioannakandi,kostismvg}@gmail.com

## Περιεχόμενα

Περιεχόμενα.....	1
Κατάλογος Πινάκων.....	2
Κατάλογος Εικόνων .....	3
Περίληψη.....	4
1.Θεματολογία Συστήματος .....	5
1.1 Ορισμός του προβλήματος .....	5
1.2 Στόχοι της εργασίας.....	5
1.3 Δομή της εργασίας .....	5
2. Θεωρητικό Υπόβαθρο .....	6
Ερώτημα Α .....	6
Ερώτημα Β .....	6
Ερώτημα Γ.....	8
3. Υλοποίηση Συστήματος.....	9
3.1. Υλοποιημένες Υπηρεσίες Ιστού.....	9
3.2. Οδηγίες Εκτέλεσης .....	11
Απόκριση Εφαρμογής (responsiveness) .....	16
4. Συμπεράσματα.....	18
Βιβλιογραφία.....	19

## Κατάλογος Πινάκων

Πίνακας 1: Πρότυπο πίνακα για υλοποιημένες υπηρεσίες.....	9
Πίνακας 2: Υπηρεσία δημιουργίας και ανάκτησης προτάσεων .....	9

## Κατάλογος Εικόνων

Εικόνα 1: Η διεπαφή χρήστη του Still Series .....	13
Εικόνα 2: Παροχή προτάσεων βάσει ομοιότητας χρηστών .....	14
Εικόνα 3: Παροχή προτάσεων βάσει ομοιότητας αντικειμένων (σειρών) .....	15
Εικόνα 4: Η διεπαφή του συστήματος, όπως φαίνεται από ένα iPhone X .....	16

## Περίληψη

Η παρούσα εργασία αποτελεί την απαλλακτική εργασία για το μάθημα «Αποθήκες Δεδομένων και Επιχειρηματική Ευφυΐα» του 2<sup>ου</sup> εξαμήνου της μεταπτυχιακής ειδίκευσης «Προηγμένα Πληροφοριακά Συστήματα» του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς. Ειδικότερα, σκοπός της εργασίας αυτής είναι ο σχεδιασμός και η υλοποίηση ενός recommender system που βασίζεται στη collaborative filtering προσέγγιση. Πιο συγκεκριμένα, η εργασία χωρίζεται σε δύο (2) μέρη. Το πρώτο μέρος περιλαμβάνει τις λύσεις σχετικά με τα τρία (3) πρώτα θεωρητικά ερωτήματα της εργασίας, ενώ το δεύτερο μέρος αφορά την υλοποίηση του συστήματος. Στο μέρος αυτό, αναλύονται επαρκώς οι υπηρεσίες που είναι διαθέσιμες στο προαναφερθέν σύστημα ενώ παρέχεται και ένα αναλυτικό εγχειρίδιο χρήσης αυτού.

## 1.Θεματολογία Συστήματος

### 1.1 Ορισμός του προβλήματος

Την σημερινή εποχή παρουσιάζονται διάφορες προκλήσεις οι οποίες επηρεάζουν την καθημερινότητα και οδηγούν στη διαμόρφωση μιας νέας πραγματικότητας. Στα πλαίσια αυτής, η ανάγκη για ανάπτυξη εφαρμογών και συστημάτων που παρέχουν ολοένα και πιο εξατομικευμένες υπηρεσίες στους χρήστες καθίσταται εντονότερη. Ένα χαρακτηριστικό παράδειγμα εφαρμογών τέτοιου τύπου αποτελούν τα συστήματα προτάσεων (recommender systems). Τα συγκεκριμένα συστήματα παρέχουν προτάσεις στους χρήστες βάσει των προτιμήσεών τους και της ομοιότητας της οποίας ενδέχεται να έχουν με άλλους χρήστες [1]. Η χρήση συστημάτων του συγκεκριμένου τύπου πραγματοποιείται από διαφορετικές εταιρείες και πλατφόρμες όπως το Netflix, η Amazon και το Yahoo, προκειμένου να παρέχουν στους χρήστες όσο το δυνατόν πιο εξατομικευμένες προτάσεις. Η συγκεκριμένη εργασία στοχεύει στην δημιουργία ενός τέτοιου συστήματος το οποίο θα παρέχει στους χρήστες του, μέσα από ένα πολύ φιλικό περιβάλλον, προτάσεις σχετικά με τηλεοπτικές εκπομπές που θα μπορούσαν να παρακολουθήσουν, βάσει των προτιμήσεών τους.

### 1.2 Στόχοι της εργασίας

Στη παρούσα εργασία, αρχικά, δίνονται απαντήσεις σε μερικά θεωρητικά ερωτήματα που αφορούν διαφορετικές προσεγγίσεις σχετικά με τα recommender systems. Στη συνέχεια, υλοποιείται ένα σύστημα που ονομάζεται “Still Series” και έχει ως στόχο την παροχή προτάσεων σε χρήστες σχετικά με τηλεοπτικές εκπομπές που θα μπορούσαν να παρακολουθήσουν, με γνώμονα τα διαφορετικά ενδιαφέροντα των χρηστών του.

### 1.3 Δομή της εργασίας

Στο Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο παρατίθενται οι λύσεις που αφορούν τα πρώτα τρία (3) ερωτήματα της εργασίας. Στο Κεφάλαιο 3 – Υλοποίηση Συστήματος περιγράφονται λεπτομερώς οι υπηρεσίες που παρέχονται από την εφαρμογή, όπως παρουσιάζεται, επίσης, συνοπτικά η αρχιτεκτονική της, σε συνδυασμό με την ροή που ακολουθείται κατά την εκτέλεσή της. Επιπρόσθετα, αναλύονται όλες οι τεχνολογίες που αξιοποιήθηκαν για την υλοποίηση της προαναφερθείσας εφαρμογής. Τέλος, στο Κεφάλαιο 4 – Συμπεράσματα πραγματοποιείται μία ανασκόπηση των πεπραγμένων της εργασίας και καταγράφονται μελλοντικές βελτιώσεις που θα μπορούσαν να γίνουν στην εφαρμογή, ούτως ώστε να ανταποκρίνεται επαρκώς σε ακόμα πιο σύνθετα σενάρια χρήσης.

## 2. Θεωρητικό Υπόβαθρο

Το παρόν κεφάλαιο αφορά τις λύσεις των πρώτων τριών (3) ασκήσεων που περιγράφονται στην εκφώνηση της εργασίας, οι οποίες παρατίθενται παρακάτω. Για την επίλυση των ασκήσεων αξιοποιήθηκε το λογισμικό MATLAB.

### Ερώτημα Α

Ο συγκεκριμένος πίνακας  $T$  είναι συμμετρικός. Τα στοιχεία της κύριας διαγωνίου αναφέρονται στο πλήθος των αντικειμένων που έχει βαθμολογήσει ο εκάστοτε χρήστης, ενώ τα υπόλοιπα στοιχεία αναφέρονται στα κοινά αντικείμενα που έχουν βαθμολογήσει οι εκάστοτε χρήστες που αντιστοιχούν στο αντίστοιχο κελί του πίνακα. Για παράδειγμα, στο ακόλουθο πίνακα  $T$  το τονισμένο στοιχείο υποδεικνύει ότι ο χρήστης 1 έχει αξιολογήσει 2 αντικείμενα.

$T =$

2	1	2	1
1	1	1	0
2	1	3	1
1	0	1	1

Επίσης, στον ίδιο πίνακα, το δεύτερο στοιχείο στην Τρίτη γραμμή υποδεικνύει πως ο χρήστης 3 και ο χρήστης 2 έχουν αξιολογήσει ένα κοινό αντικείμενο.

$T =$

2	1	2	1
1	1	1	0
2	1	3	1
1	0	1	1

### Ερώτημα Β

Έστω ο πίνακας ομοιότητας  $R$ . Υπολογίζοντας την ομοιότητα μεταξύ των στηλών (δηλαδή των αντικειμένων), προκύπτει ο ακόλουθος πίνακας  $S1$ .

$R =$

1	0	1
1	0	0
1	1	1
0	0	1

```
>> similarities = cosineSimilarity([1 1 1 0],[0 0 1 0])
```

```
similarities =
```

```
0.5774
```

```
>> similarities = cosineSimilarity([1 1 1 0],[1 0 1 1])
```

```
similarities =
```

```
0.6667
```

```
S1 =
```

```
1.0000    0.5774    0.6667
0.5774    1.0000    0.5774
0.6667    0.5774    1.0000
```

Η προς απόδειξη σχέση είναι η ακόλουθη :

```
>> S1 = (sqrtQ*tR)*(R*sqrtQ)
```

Όπου:

```
>> sqrtQ = [1/sqrt(3) 0 0; 0 1 0; 0 0 1/sqrt(3)]
```

```
sqrtQ =
```

```
0.5774    0    0
0    1.0000    0
0    0    0.5774
```

Επιλύοντας την προαναφερθείσα σχέση, προκύπτει ότι:



```
>> S1 = (sqrtQ*tR)*(R*sqrtQ)
```

```
S1 =
```

```
1.0000    0.5774    0.6667
0.5774    1.0000    0.5774
0.6667    0.5774    1.0000
```

### Ερώτημα Γ

#### Item-Item collaborative filtering

Βάσει των δεδομένων της άσκησης και της δοθείσας σχέσης ( $\Gamma = RQ^{-1/2}R^TRQ^{-1/2}$ ) του ερωτήματος Γ, ο πίνακας προτάσεων που προκύπτει είναι ο ακόλουθος:

```
CItem =
```

```
1.6667    1.1547    1.6667
1.0000    0.5774    0.6667
2.2440    2.1547    2.2440
0.6667    0.5774    1.0000
```

Ειδικότερα, για τον πρώτο χρήστη θα προταθεί είτε το πρώτο είτε το τρίτο αντικείμενο. Για τον δεύτερο χρήστη, θα προταθεί το πρώτο αντικείμενο και για τον τρίτο, τα αντικείμενα 1 και 3. Τέλος, στον τέταρτο χρήστη θα προταθεί το τρίτο αντικείμενο.

#### User-User collaborative filtering

Στη συγκεκριμένη περίπτωση θα πρέπει να ληφθεί υπόψη και η μη-κανονικοποιημένη ομοιότητα ανάμεσα σε χρήστες ( $T=R^TR$ ), η οποία έχει υπολογιστεί στο ερώτημα Α. Συνεπώς, η σχέση που προκύπτει είναι η ακόλουθη:  $\Gamma = TRQ^{-1/2}R^TRQ^{-1/2}$ . Ο αντίστοιχος πίνακας προτάσεων που προκύπτει είναι ο παρακάτω:

```
CUser =
```

```
9.4880    7.7735    9.4880
4.9107    3.8868    4.5774
11.7321    9.9282    11.7321
4.5774    3.8868    4.9107
```

Παρατηρούμε ότι, οι προτάσεις που προκύπτουν βάσει του παραπάνω πίνακα ταυτίζονται με αυτές της item-item προσέγγισης.

### 3. Υλοποίηση Συστήματος

#### 3.1. Υλοποιημένες Υπηρεσίες Ιστού

Στη συγκεκριμένη ενότητα περιγράφεται η υλοποιημένη υπηρεσία σε μορφή πινάκων. Ο πίνακας έχει την ακόλουθη μορφή:

Όνομα Υπηρεσίας	
Περιγραφή	Περιγραφή της λειτουργίας της υπηρεσίας
Endpoint URL	http://localhost:5000/...
HTTP μέθοδος	Δυνατές τιμές: GET, POST, PUT, DELETE
Παράμετροι	Οι παράμετροι που χρειάζεται η υπηρεσία (αν υπάρχουν)
Επιστροφή	Τι επιστρέφει η υπηρεσία στον χρήστη

Πίνακας 1: Πρότυπο πίνακα για υλοποιημένες υπηρεσίες.

getShows	
Περιγραφή	Χρησιμοποιείται για την παραγωγή προτάσεων και την ανάκτησή τους από τον εκάστοτε χρήστη
Endpoint URL	http://localhost:5000/getShows/{search_parameter}/{search_method}
HTTP μέθοδος	GET
Παράμετροι	search_parameter: η παράμετρος βάσει της οποίας θα παραχθούν οι προτάσεις. Μπορεί να είναι είτε ο κωδικός ενός χρήστη, είτε το όνομα μίας σειράς search_method: ο τρόπος με τον οποίο θα παραχθεί η πρόταση. Αν πρόκειται για user-based, τότε λαμβάνει τη τιμή «user». Διαφορετικά, αν πρόκειται για item-based, τότε λαμβάνει τη τιμή «item»
Επιστροφή	Επιστρέφει δέκα (10) προτάσεις

Πίνακας 2: Υπηρεσία δημιουργίας και ανάκτησης προτάσεων

Το σύστημα/εφαρμογή της εργασίας είναι web-based και αποτελείται από το front-end, το οποίο αποτελεί την διεπαφή του χρήστη, και το back-end που είναι όλες οι υπηρεσίες που έχει πρόσβαση ο χρήστης από την διεπαφή. Για το front-end χρησιμοποιήθηκαν οι ακόλουθες τεχνολογίες:

- HTML5 για την οργάνωση των επιμέρους τμημάτων από τα οποία αποτελείται η διεπαφή και την προσθήκη κανόνων στις φόρμες υποβολής του χρήστη για διασφάλιση της εγκυρότητας αυτών πριν σταλούν στο back-end.
- Bootstrap 4: για την μορφοποίηση των παραπάνω τμημάτων ούτως ώστε να είναι ευδιάκριτα και ελκυστικά προς τον χρήστη. Το Bootstrap είναι ένα framework το οποίο αποτελείται από HTML, CSS και JavaScript. Ουσιαστικά, παρέχει στους προγραμματιστές έτοιμες κλάσεις τις οποίες μπορούν να χρησιμοποιήσουν έτσι ώστε να υλοποιήσουν γρήγορα μία πολύ ελκυστική διεπαφή χρήστη.
- CSS για την μορφοποίηση των παραπάνω τμημάτων ούτως ώστε να είναι ευδιάκριτα και ελκυστικά προς τον χρήστη.

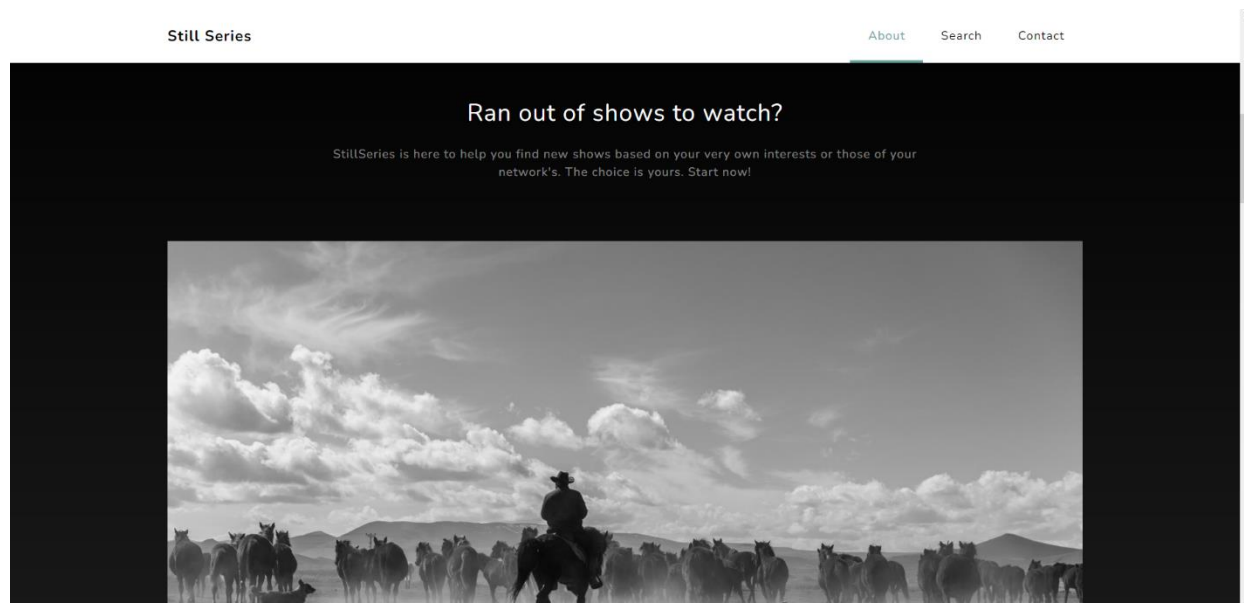
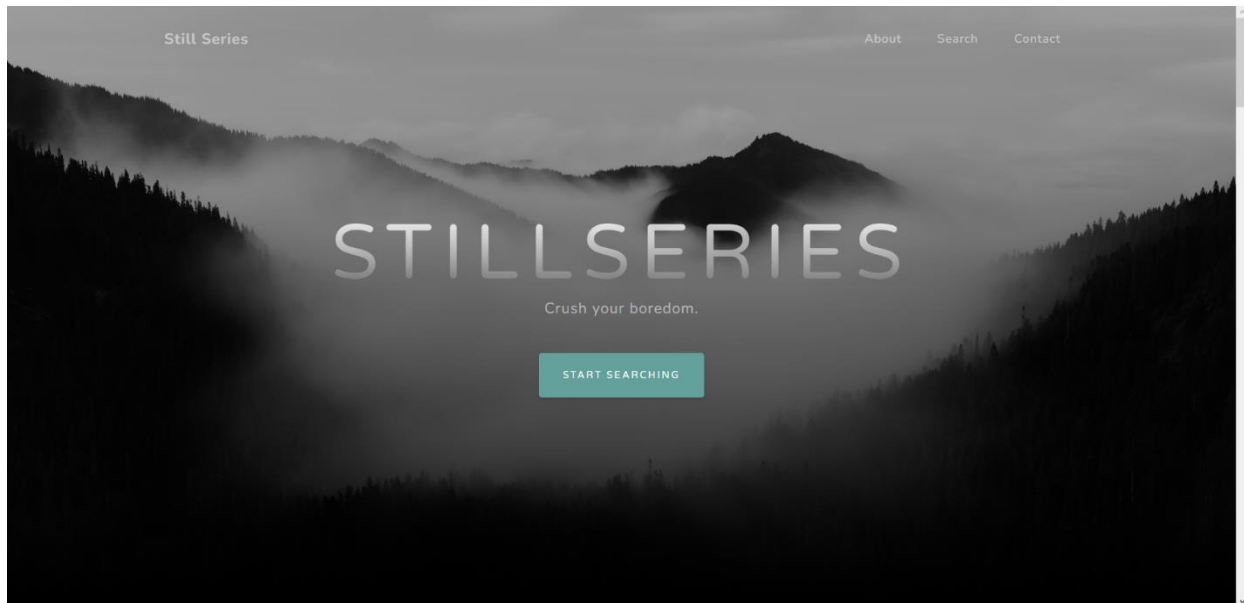
- JavaScript για την προσθήκη του στοιχείου της δυναμικότητας στο front-end, διατήρηση cookies και γενικά διαχείριση διάφορων τιμών στο front-end.
- jQuery, η οποία αποτελεί μία βιβλιοθήκη JavaScript και ουσιαστικά παίρνει πολλές κοινές εργασίες που απαιτούν πολλές γραμμές κώδικα JavaScript για να ολοκληρωθούν και τις «τυλίγει» σε μεθόδους που μπορούν να χρησιμοποιηθούν με μία γραμμή κώδικα. Μέσω της jQuery γίνονται και οι κλήσεις Ajax στα διάφορα API με τα οποία επικοινωνεί η εφαρμογή, όπως επίσης και η εφαρμογή τεχνικών form validation, αξιοποιώντας την αντίστοιχη βιβλιοθήκη (jQuery Validation Plugin).

Για το back-end χρησιμοποιήθηκαν οι ακόλουθες τεχνολογίες:

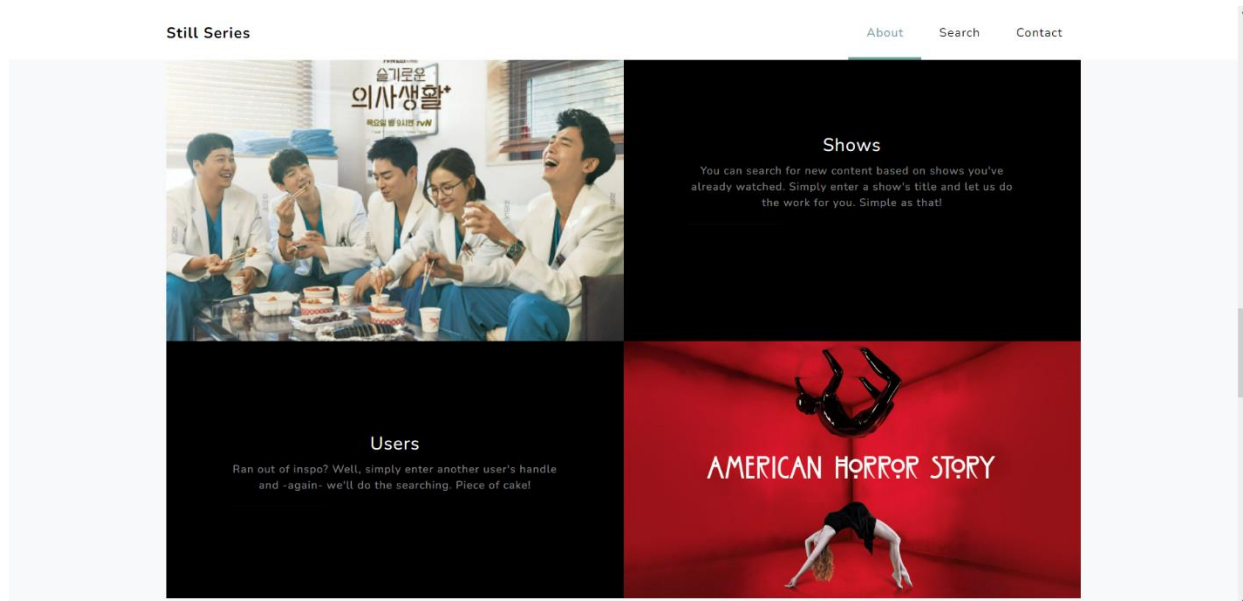
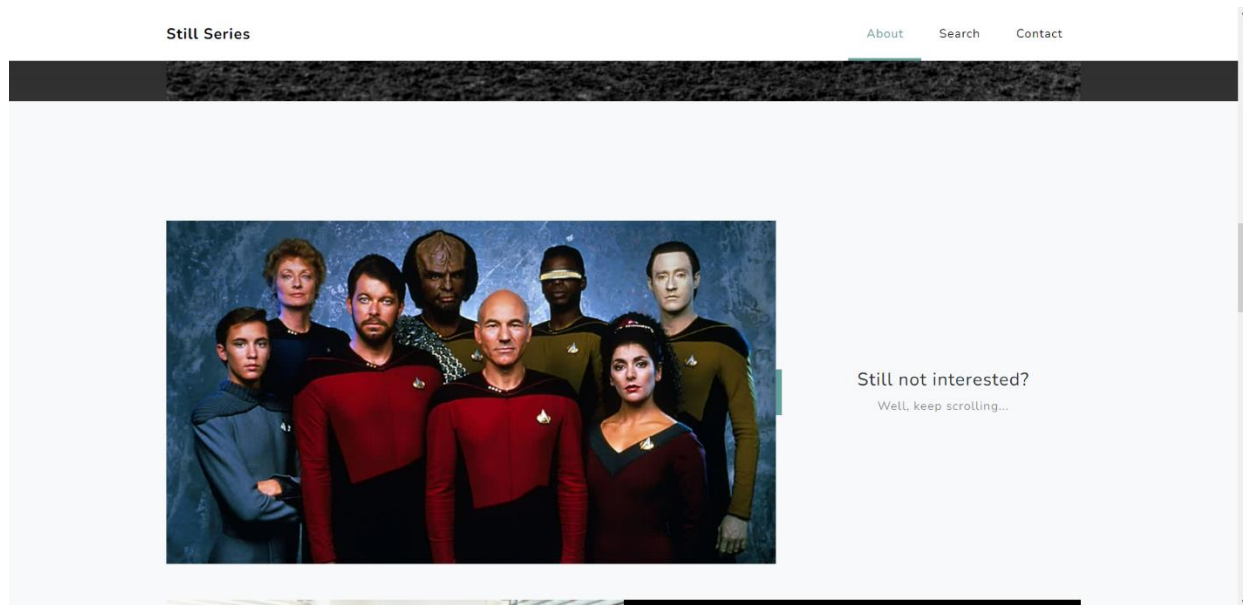
- Python για την υλοποίηση των παρεχόμενων υπηρεσιών.

### 3.2. Οδηγίες Εκτέλεσης

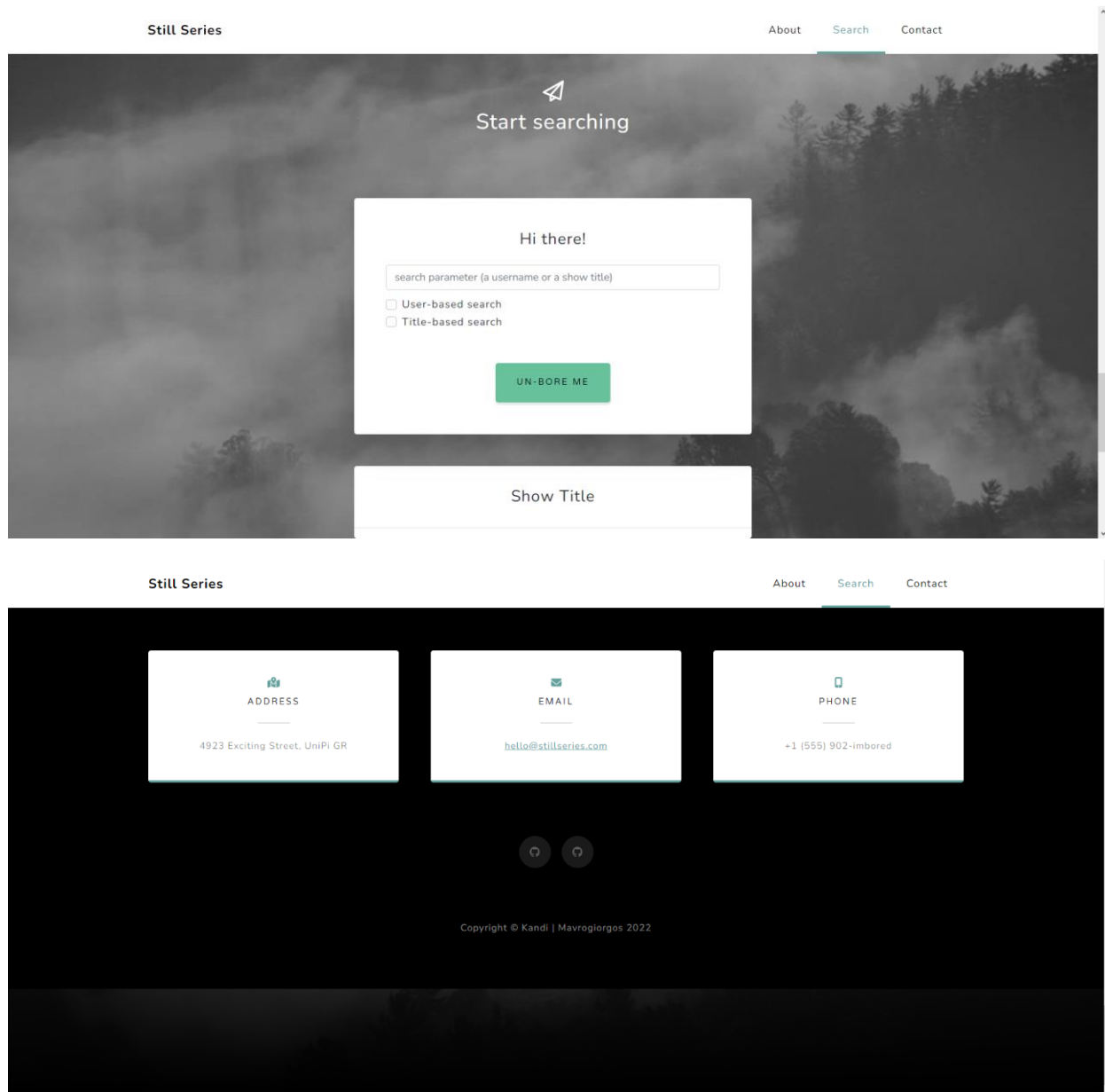
Η διεπαφή του συστήματος αποτελείται από μία ενιαία σελίδα (Εικόνα 1) και είναι διαθέσιμη στη διεύθυνση <http://localhost:5000>, δεδομένου ότι η εφαρμογή έχει εκκινήσει. Στη διεπαφή του χρήστη είναι διαθέσιμες όλες οι πληροφορίες που αφορούν τις παρεχόμενες υπηρεσίες, ενώ προς το τέλος της σελίδας παρέχεται μία φόρμα την οποία ο χρήστης μπορεί να συμπληρώσει, έτσι ώστε να ανακτήσει προτάσεις σχετικά με τηλεοπτικές σειρές που θα μπορούσε να παρακολουθήσει.



## Ιωάννα Κανδή & Κωνσταντίνος Μαυρογιώργος – Still Series

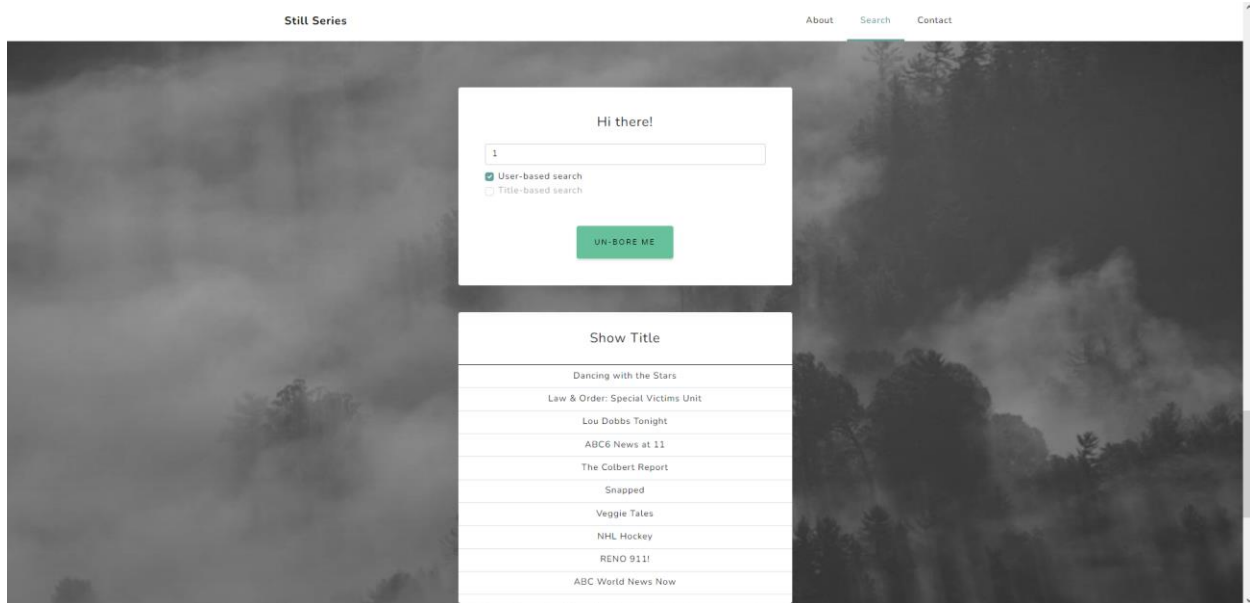


## Ιωάννα Κανδή & Κωνσταντίνος Μαυρογιώργος – Still Series



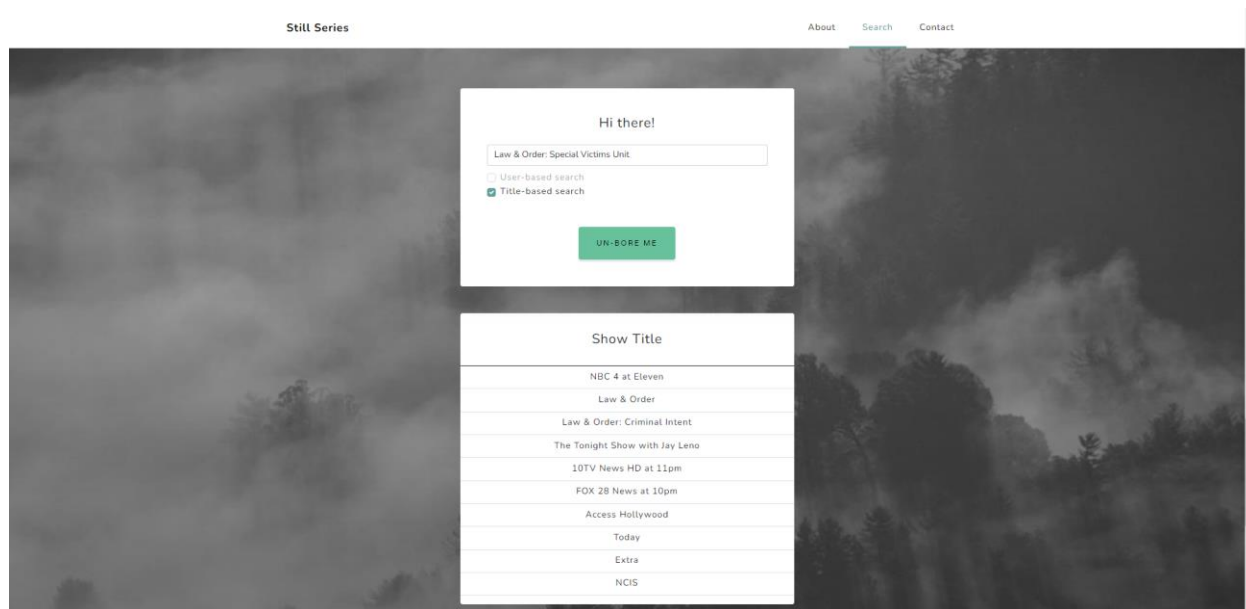
Εικόνα 1: Η διεπαφή χρήστη του Still Series

Σχετικά με την παροχή προτάσεων, αυτή μπορεί να πραγματοποιηθεί με δύο (2) τρόπους. Ο πρώτος τρόπος αφορά την παροχή προτάσεων βάσει ομοιότητας μεταξύ χρηστών. Σε αυτή τη περίπτωση, ο χρήστης επιλέγει τον κωδικό κάποιου χρήστη και του επιστρέφονται δέκα (10) προτάσεις με βάση τις σειρές που έχουν παρακολουθήσει άλλοι «παρόμοιοι» χρήστες. Ενδεικτικό παράδειγμα, όπου επιλέγεται ο χρήστης με κωδικό «1», φαίνεται στην Εικόνα 2. Για την παροχή προτάσεων βάσει ομοιότητας μεταξύ χρηστών, αξιοποιείται ο αλγόριθμος KNN και η ακρίβειά του είναι περίπου 80%.



Εικόνα 2: Παροχή προτάσεων βάσει ομοιότητας χρηστών

Ο δεύτερος τρόπος αφορά την παροχή προτάσεων βάσει ομοιότητας αντικειμένων. Σε αυτή τη περίπτωση, ο χρήστης επιλέγει μία σειρά και του επιστρέφονται οι δέκα (10) πιο «παρόμοιες» σειρές σε σχέση με αυτή που επέλεξε. Ενδεικτικό παράδειγμα, όπου επιλέγεται η σειρά «Law & Order: Special Victims Unit», παρουσιάζεται στην Εικόνα 3. Για την παροχή προτάσεων βάσει ομοιότητας μεταξύ αντικειμένων, αξιοποιείται ο αλγόριθμος KNN και η ακρίβειά του είναι περίπου 90%.



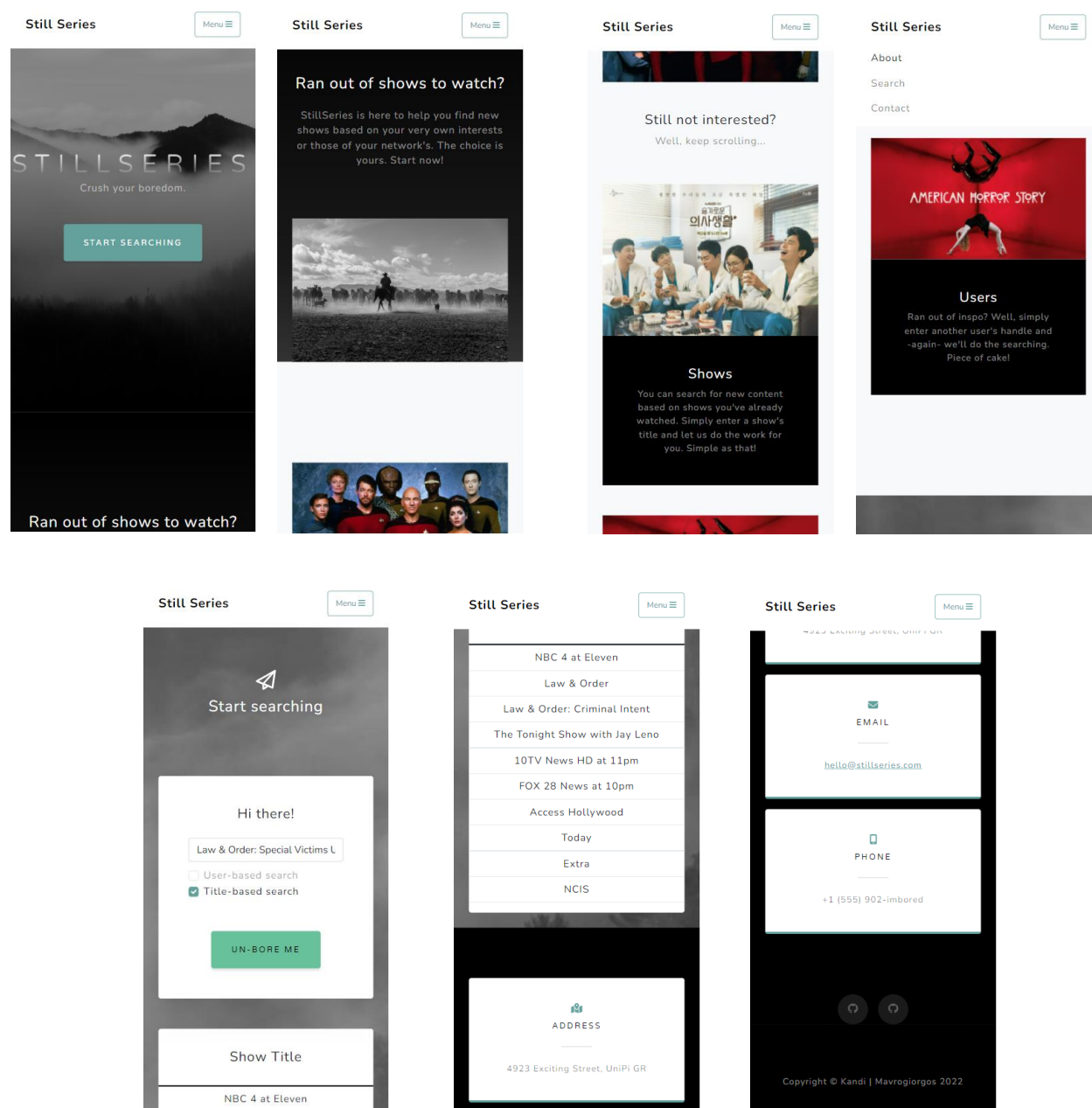
Εικόνα 3: Παροχή προτάσεων βάσει ομοιότητας αντικειμένων (σειρών)

Παρατηρείται ότι η ακρίβεια των προτάσεων στην περίπτωση της ομοιότητας μεταξύ αντικειμένων είναι μεγαλύτερη από αυτή της ομοιότητας μεταξύ χρηστών. Αυτό οφείλεται στο γεγονός ότι τα αντικείμενα είναι πιο «απλά», ενώ οι χρήστες είναι πιο «σύνθετοι» [3].



## Απόκριση Εφαρμογής (responsiveness)

Αξίζει να σημειωθεί ότι το σύστημα / εφαρμογή της εργασίας έχει σχεδιαστεί με τέτοιο τρόπο, έτσι ώστε ο χρήστης να μπορεί να έχει πρόσβαση και από διαφορετικές συσκευές, εκτός του υπολογιστή. Πιο συγκεκριμένα, η μορφή της εφαρμογής προσαρμόζεται ανάλογα με την συσκευή που χρησιμοποιεί ο χρήστης. Παρακάτω παρατίθενται στιγμιότυπα οθόνης που αφορούν την χρήση της εφαρμογής σε κινητές συσκευές (Εικόνα 4).



Εικόνα 4: Η διεπαφή του συστήματος, όπως φαίνεται από ένα iPhone X.

Επίσης, αξίζει να σημειωθεί ότι ο κώδικας της εφαρμογής είναι ανεβασμένος και σε ιδιωτικό repository στο GitHub, στη διεύθυνση <https://github.com/ioannakandi/recSys>. Για να έχει κανείς πρόσβαση θα πρέπει να επικοινωνήσει σε ένα από τα emails που αναγράφονται στο εξώφυλλο της εργασίας, έτσι ώστε να προστεθεί ως contributor.

## 4. Συμπεράσματα

Συνοψίζοντας, η παρούσα εργασία και κατ' επέκταση η ανάπτυξη της εφαρμογής 'Still Series' ήταν απαραίτητη για να επιτευχθεί μια πρώτη επαφή με τη λειτουργία των συστημάτων προτιμήσεων (recommender systems) . Η προαναφερθείσα εφαρμογή εκτελεί συγκεκριμένες διεργασίες και διαθέτει γραφική διεπαφή φιλική προς τον χρήστη έτσι ώστε να είναι πιο αρμονική η πλοήγηση, η εισαγωγή και η εμφάνιση των εκάστοτε επιθυμητών στοιχείων.

Ωστόσο, υπάρχουν περιθώρια βελτίωσης. Ειδικότερα, η εφαρμογή αυτή θα μπορούσε να επεξεργάζεται και να εκτελεί λειτουργίες αυξημένης πολυπλοκότητας · καλό θα ήταν να γίνεται επεξεργασία μεγαλύτερου όγκου δεδομένων, το οποίο θα μπορούσε να επιτευχθεί μέσω της χρήσης κάποιας βάσης δεδομένων, η οποία δεν αξιοποιείται στα πλαίσια της εργασίας. Εφόσον γίνεται λόγος για μεγαλύτερο όγκο δεδομένων θα ήταν αξιοσημείωτη η συσταδοποίηση (clustering) των δεδομένων σε διάφορες κατηγορίες, έτσι ώστε η πρόβλεψη του συστήματος να είναι πιο ακριβής [2]. Τέλος, σημαντική θα ήταν η χρήση πληθώρας analytics για τη σύσταση μιας πιο ολοκληρωμένης εικόνας των χρηστών, αλλά και της ίδιας της εφαρμογής, κάτι το οποίο δεν είναι υλοποιήσιμο την περίοδο υλοποίησης της εφαρμογής.

## Βιβλιογραφία

- [1] Lu J, Wu D, Mao M, Wang W, Zhang G. Recommender system application developments: a survey. Decision Support Systems. 2015 Jun 1;74:12-32.
- [2] Truong K, Ishikawa F, Honiden S. Improving accuracy of recommender system by item clustering. IEICE TRANSACTIONS on Information and Systems. 2007 Sep 1;90(9):1363-73.
- [3] Μ. Χαλκίδη, Διαφάνειες Μαθήματος «Αποθήκες Δεδομένων και Επιχειρηματική Ευφυΐα», 2022