



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**Μελέτη αλγορίθμων βελτιστοποίησης με πολλαπλά κριτήρια και εφαρμογή τους σε εξέλιξη ηλεκτρονικού παιχνιδιού**

**Διπλωματική Εργασία**

**Στέφανος Παπαναστασίου**

**Επιβλέπουσα:** Παναγιώτα Τσομπανοπούλου

Φεβρουάριος 2022





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**Μελέτη αλγορίθμων βελτιστοποίησης με πολλαπλά κριτήρια και εφαρμογή τους σε εξέλιξη ηλεκτρονικού παιχνιδιού**

**Διπλωματική Εργασία**

**Στέφανος Παπαναστασίου**

**Επιβλέπουσα:** Παναγιώτα Τσομπανοπούλου

Φεβρουάριος 2022





**UNIVERSITY OF THESSALY**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**Study of optimization algorithms with multiple criteria and their  
application in the development of electronic game**

Diploma Thesis

**Stefanos Papanastasiou**

**Supervisor:** Panagiota Tsompanopoulou

February 2022



Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπουσα

**Τσομπανοπούλου Παναγιώτα**

Αναπληρώτρια Καθηγήτρια, Τμήμα Ηλεκτρολόγων  
Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο  
Θεσσαλίας

Μέλος

**Βασιλακόπουλος Μιχάλης**

Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και  
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

**Σταμούλης Γιώργος**

Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και  
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

## **ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ**

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελούν αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλουν οποιασδήποτε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχουν έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

**Ο Δηλών**

**Στέφανος Παπαναστασίου**

**Ημερομηνία**

**22/02/22**



## **DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS**

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant

Stefanos Papastasiou

## Ευχαριστίες

Με αφορμή την ολοκλήρωση της διπλωματικής μου εργασίας αλλά και γενικότερα φτάνοντας στο τέλος της φοιτητικής μου πορείας, θα ήθελα να ευχαριστήσω μια σειρά από ανθρώπους που στάθηκαν δίπλα μου σε όλη αυτή τη διαδρομή.

Αρχικά θα ήθελα να ευχαριστήσω την καθηγήτρια και υπεύθυνη της διπλωματικής μου εργασίας, κ. Παναγιώτα Τσομπανοπούλου, για την βοήθεια που μου παρείχε αλλά και για την πολύτιμη στήριξή της σε αυτές τις περιέργες για την ανθρωπότητα συνθήκες στις οποίες ζούμε εν μέσω της πανδημίας . Επιπλέον τους καθηγητές κ. Βασιλακόπουλο Μιχάλη και κ. Σταμούλη Γιώργο που δέχτηκαν να συμμετάσχουν ως μέλη Επιτροπής.

Τέλος , ευχαριστώ την οικογένειά μου τόσο για την οικονομική όσο και για την ψυχολογική στήριξη που μου παρείχαν όλα αυτά τα χρόνια. Φυσικά τους φίλους μου αλλά και γενικότερα τους συμφοιτητές μου που συνεργαστήκαμε στα πλαίσια της ακαδημαϊκής μας πορείας.

## Διπλωματική Εργασία

# Μελέτη αλγορίθμων βελτιστοποίησης με πολλαπλά κριτήρια και εφαρμογή τους σε εξέλιξη ηλεκτρονικού παιχνιδιού

Στέφανος Παπαναστασίου

## Περίληψη

Σκοπός του συστήματος είναι να βοηθήσει την εμπειρία των χρηστών του επώνυμου video game FIFA Ultimate Team . Το συγκεκριμένο video game βασίζεται πάνω στον πραγματικό κόσμο του ποδοσφαίρου. Ο κάθε χρήστης έχει την δυνατότητα να δημιουργήσει την ομάδα της αρεσκείας του με απώτερο στόχο να παίξει ποδόσφαιρο με τους αγαπημένους του παίκτες. Συνεπώς το σύστημα δεν βοηθάει την χρήση κατά την διάρκεια του gameplay αλλά στην προετοιμασία .Όπως εξηγούμε στις παρακάτω ενότητες το σύστημα καλείται να βρει την καλύτερη επιλογή βοηθώντας τον χρήστη να επιλέξει την ομάδα με την μεγαλύτερη δυνατή χημεία και το ελάχιστο συνολικό κόστος. Για την επίλυση αυτού του ερωτήματος το οποίο αποτελεί ένα πρόβλημα αντικρουόμενων συμφερόντων με πολλαπλά κριτήρια μελετήθηκαν και εφαρμόστηκαν διάφοροι αλγόριθμοι . Βέβαια προηγήθηκε η κατάλληλη επεξεργασία των δεδομένων και η εφαρμογή διαφόρων αλγορίθμων ,ώστε να βρεθούν τα δεδομένα στην κατάλληλη μορφή για την εφαρμογή των αλγορίθμων λήψης αποφάσεων πολλαπλών κριτηρίων . Η διαδικασία μορφοποίησης των δεδομένων καθώς και η εφαρμογή των αλγορίθμων περιγράφεται αναλυτικά στα κεφάλαια που ακολουθούν.

### Λέξεις-κλειδιά:

αλγόριθμος λήψης απόφασης με πολλαπλά κριτήρια, ηλεκτρονικό παιχνίδι, πρόβλημα αντικρουόμενων συμφερόντων, διαδικτυακή εφαρμογή, Python Django ,εξόρυξη δεδομένων ,weighted sum model ,weighted product model , VIKOR , TOPSIS

# **Study of optimization algorithms with multiple criteria and their application in the development of electronic game**

Stefanos Papanastasiou

## **Abstract**

The purpose of the system is to help the user experience of the branded video game FIFA Ultimate Team. This video game is based on the real world of football. Each user has the ability to create the team of their choice with the ultimate goal of playing football with their favorite players. Therefore the system does not help the user during the gameplay but in the preparation. As we explain in the following sections the system is called to find the best option helping the user to choose the team with the highest possible chemistry and the lowest overall cost. Various algorithms have been studied and applied to solve this question which is a problem of conflicting interests with multiple criteria. Of course, the proper processing of the data and the application of various algorithms preceded, in order to find the data in the appropriate format for the application of the multi-criteria decision-making algorithms. The data formatting process as well as the application of the algorithms are described in detail in the following chapters.

### **Keywords:**

multi-criteria decision making algorithm, video game, conflict of interest problem, web application, Python Django, data mining ,weighted sum model ,weighted product model , VIKOR , TOPSIS

# Πίνακας περιεχομένων

<i>Ευχαριστίες</i>	<i>x</i>
<i>Περίληψη</i>	<i>xi</i>
<i>Abstract</i>	<i>xii</i>
<i>Πίνακας περιεχομένων</i>	<i>xiii</i>
<b>Κεφάλαιο 1 Περιγραφή FIFA ULTIMATE TEAM</b>	<b>1</b>
1.1 Αλγόριθμος υπολογισμού χημείας	2
1.2 Τεχνικά Χαρακτηριστικά	7
<b>Κεφάλαιο 2 Django</b>	<b>8</b>
2.1 Ιδιότητες και χαρακτηριστικά Django	8
2.2 Γιατί χρησιμοποίησα το Django	9
2.3 Μειονεκτήματα του Django	14
2.4 Η αρχιτεκτονική Django	15
2.5 Εφαρμογές και βασικά αρχεία Django	15
2.6 Django administration page	18
<b>Κεφάλαιο 3 Λειτουργικότητα</b>	<b>20</b>
3.1 Αλγόριθμος υπολογισμού αξιολόγησης της ομάδας	21
3.2 Weighted sum model	22
3.3 Weighted product model	22
3.4 VIKOR vs TOPSIS	23
3.5 Αποτελέσματα και συμπεράσματα	30
<b>Κεφάλαιο 4 Τίτλος κεφαλαίου</b>	<b>36</b>
4.1 Squad building	37
4.1 Players	38
4.1 Teams	39

**Κεφάλαιο 5 Επίλογος**

**40**

**Βιβλιογραφία**

**41**

## ΚΕΦΑΛΑΙΟ 1 ΠΕΡΙΓΡΑΦΗ FIFA ULTIMATE TEAM

Στην παρακάτω παράγραφο περιγράφεται η δομή και τα βασικά στοιχεία του videogame FIFA ULTIMATE TEAM [3] , με στόχο να καταλάβει ο αναγνώστης την βάση της μελέτης. Στο συγκεκριμένο video game κάθε επαγγελματίας ποδοσφαιριστής παριστάνεται από μία κάρτα η οποία περιέχει τα εξής βασικά χαρακτηριστικά(όνομα παίκτη, ομάδα,εθνικότητα,ύψος,βάρος,στατιστικά που περιγραφουν τις ικανότητες του παίκτη στο παιχνίδι κλπ). Επίσης το παιχνίδι διαθέτει live transfer market όπου ο χρήστης μπορεί να πουλήσει ή να αγοράσει κάποιον παίκτη (οι κάρτες δεν είναι μοναδικές υπάρχουν χιλιάδες διπλότυπα αντικείμενα). Επομένως ο κάθε παίκτης έχει αξία στο μάρκετ που αυξομειώνεται αναλόγως τις περιστάσεις(κατά κανόνα όσο μεγαλύτερη συνολική αξία έχει ο παίκτης τόσο πιο ακριβός είναι).Ο χρήστης καλείται να δημιουργήσει μια ενδεκάδα. Βασικά χαρακτηριστικά της κάθε ενδεκάδας είναι η χημεία και η συνολική αξιολόγηση της ομάδας (Overall Rating). Η χημεία της ομάδας παίρνει τιμές από 0 έως 100 και εξαρτάται από το πως συνδέονται οι παίκτες μεταξύ τους. Συγκεκριμένα θεωρείται ότι ένας παίκτης σχετίζεται με τον γειτονικό του εαν:

- και οι δύο παίζουν στην ίδια ομάδα ή
- παίζουν στο ίδιο πρωταθλημα αλλά είναι διαφορετικής εθνικότητας ή
- έχουν ίδια εθνικότητα αλλά δεν παίζουν στο ίδιο πρωτάθλημα ή
- έχουν ίδια εθνικότητα και παίζουν στο ίδιο πρωτάθλημα

Κάθε περίπτωση έχει διαφορετική βαρύτητα στην τελική χημεία της ομάδας. Ο όρος γειτονικός περιγράφεται από το σύστημα που έχει επιλέξει ο χρήστης, δηλαδή ο κάθε παίκτης σχετίζεται με παίκτες που ενώνεται , το οποίο αλλάζει αναλόγως με το σύστημα που έχει επιλεχθεί.Το κάθε σύστημα ενώνει τους παίκτες με μοναδικό τρόπο.Η συνολική χημεία της ομάδας είναι το άθροισμα της ατομικής χημείας των ένδεκα παικτών, η οποία εξαρτάται τόσο από τους γειτονικούς παίκτες όσο και από το αν ο παίκτης έχει τοποθετηθεί στην θέση που αγωνίζεται[4]. Η συνολική αξιολόγηση της ομάδας ακολουθεί έναν αλγόριθμο που υπολογίζει την συνολική αξιολόγηση της ομάδας. [5]

## 1.1 Αλγόριθμος υπολογισμού χημείας

### 1.1.1 Χημεία ομάδας

Η χημεία της ομάδας είναι το άθροισμα της ατομικής χημείας των παικτών που την αποτελούν και παίρνει τιμές από 0 έως 100. Υπολογίζεται μέσω του παρακάτω τύπου και παρότι μπορεί να φτάσει μέχρι το 110, η μέγιστη τιμή της είναι το 100.

$$TC = IC1 + IC2 + \dots + IC11$$

όπου:

TC είναι η συνολική χημεία της ομάδας (team chemistry)

IC είναι η ατομική χημεία κάθε παίκτη (individual chemistry)

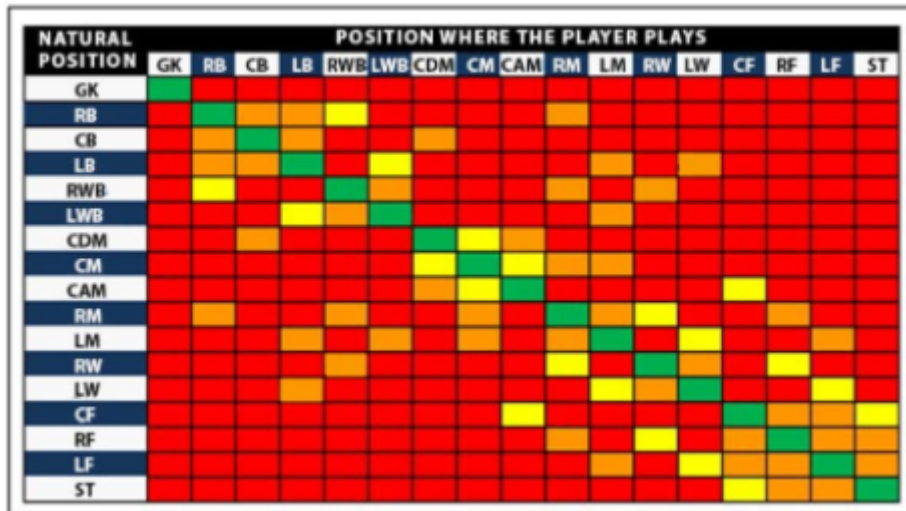
### 1.1.2 Ατομική χημεία

Η ατομική χημεία υπολογίζεται με τον παρακάτω τρόπο και παίρνει τιμές από 0 έως 10. Αρχικά υπολογίζεται το

$$L = \frac{\text{άθροισμα των δεσμών από σχετικούς παίκτες}}{\text{συνολικό άθροισμα των δεσμών}}$$

Η θέση που είναι τοποθετημένος ο παίκτης σε σχέση με την κανονική του θέση καθώς και το **L** συνδυάζονται για να υπολογιστεί η τελική ατομική χημεία με την βοήθεια των παρακάτω πινάκων





Σχήμα 1.1: Υπολογισμός της θέσης του παίκτη [4]

QPL LINKS	POSITION			
$L < 0.3$	0	1	2	3
$0.3 \leq L < 1$	1	3	5	6
$1 \leq L \leq 1.6$	2	5	8	9
$L > 1.6$	2	5	9	10

Σχήμα 1.2: Υπολογισμός ατομικής χημείας [4]

Οι παραπάνω αλγόριθμοι περιγράφουν τον υπολογισμό της ατομικής και ομαδικής χημείας, ωστόσο δεν αναφέρεται πώς το πρόγραμμα είναι σε θέση να γνωρίζει ποιος παίκτης σχετίζεται με ποιον. Όπως ήδη αναφέραμε ο τρόπος με τον οποίο συνδέονται οι παίκτες εξαρτώνται αποκλειστικά από το σύστημα που έχει επιλεγεί. Με την χρήση look-up tables το σύστημα είναι κάθε στιγμή ενήμερο για το ποιος παίκτης σχετίζεται με ποιον στο κάθε σύστημα. Ακολουθούν όλα τα διαθέσιμα συστήματα και οι δεσμοί των παικτών. Κάθε παίκτης αποτελεί έναν κόμβο, ο οποίος έχει διάφορες συνδέσεις. Ο αριθμός των συνδέσεων διαφέρει από παίκτη σε παίκτη και από σύστημα σε σύστημα. Παρακάτω παραθέτουμε όλα τα πιθανά συστήματα και τους επιμέρους συνδέσμους για κάθε παίκτη-κόμβο.



Σχήμα 1.3- 3-1-4-2 Γράφημα



Σχήμα 1.4- 3-4-2-1 Γράφημα



Σχήμα 1.5- 3-4-3 Γράφημα



Σχήμα 1.6- 3-5-2 Γράφημα



Σχήμα 1.7- 4-1-2-1-2 Γράφημα



Σχήμα 1.8- 3-4-1-2 Γράφημα



Σχήμα 1.9- 4-3-1-2 Γράφημα



Σχήμα 1.10- 4-3-3(2) Γράφημα



Σχήμα 1.11- 4-1-3-2 Γράφημα



Σχήμα 1.12- 4-3-3(4) Γράφημα



Σχήμα 1.13- 4-3-2-1 Γράφημα



Σχήμα 1.14- 4-3-3(5) Γράφημα



Σχήμα 1.17- 4-5-1 (2) Γράφημα



Σχήμα 1.18- 4-4-2 (2) Γράφημα



Σχήμα 1.19- 4-5-1 Γράφημα



Σχήμα 1.20- 5-3-2 Γράφημα



Σχήμα 1.19- 4-5-1 Γράφημα



Σχήμα 1.20- 5-3-2 Γράφημα



Σχήμα 1.21- 4-4-2 Γράφημα



Σχήμα 1.22- 5-2-1-2 Γράφημα



Σχήμα 1.23- 5-2-2-1 Γράφημα



Σχήμα 1.24- 5-4-1 Γράφημα



## 1.2 Τεχνικά Χαρακτηριστικά

Όσον αφορά τα τεχνικά χαρακτηριστικά, το σύστημα είναι ανεπτυγμένο με το Python Django Framework . Για την βάση δεδομένων έχει χρησιμοποιηθεί SQLITE 3 ενώ για το front end HTML, CSS και JavaScript. Επίσης χρησιμοποιήθηκαν οι βιβλιοθήκες pandas για την ανάλυση των δεδομένων και NumPy η οποία προσφέρει μια πληθώρα μαθηματικών συναρτήσεων. Το pandas είναι μία βιβλιοθήκη ανοικτού κώδικα που παρέχει χειρισμό δεδομένων υψηλής απόδοσης στην Python. Είναι χτισμένο πάνω από το πακέτο NumPy . Χρησιμοποιείται για ανάλυση δεδομένων στην Python και αναπτύχθηκε από τον Wes McKinney το 2008. Πριν από το Pandas, η Python ήταν ικανή για προετοιμασία δεδομένων , αλλά παρείχε μόνο περιορισμένη υποστήριξη για την ανάλυση δεδομένων . Έτσι , το pandas ενίσχυσε τις δυνατότητες ανάλυσης δεδομένων αφού είναι σε θέση να εκτελέσει πέντε σημαντικά βήματα που απαιτούνται για την επεξεργασία και την ανάλυση δεδομένων ανεξάρτητα από την προέλευση των δεδομένων, δηλαδή , φόρτωση , χειρισμό , προετοιμασία , μοντελοποίηση και ανάλυση . Η βιβλιοθήκη pandas δουλεύει με πίνακες δεδομένων ενώ τα πιο ισχυρά εργαλεία της είναι τα Series και τα DataFrame . Είναι ιδανική για μεγάλο όγκο δεδομένων καθώς έχει καλύτερη απόδοση όταν οι γραμμές που επεξεργάζεται είναι παραπάνω από 500.000 , ωστόσο συγκριτικά με το NumPy καταναλώνει περισσότερη μνήμη . Τα αντικείμενα που προσφέρει είναι πίνακες δύο διαστάσεων που ονομάζονται DataFrame. Το NumPy είναι κυρίως γραμμένο σε γλώσσα C και είναι μία επέκταση της Python. Ορίζεται ως ένα πακέτο Python που χρησιμοποιείται για την εκτέλεση των διαφόρων αριθμητικών υπολογισμών και την επεξεργασία των πολυδιάστατων και μονοδιάστατων στοιχείων πίνακα. Οι υπολογισμοί που χρησιμοποιούν πίνακες NumPy είναι ταχύτεροι από τον κανονικό πίνακα Python. Η βιβλιοθήκη NumPy δουλεύει με αριθμητικά δεδομένα και το πιο ισχυρό εργαλείο της είναι τα Arrays που ουσιαστικά είναι πίνακες πολλών διαστάσεων. Το NumPy προτείνεται όταν οι σειρές που θέλουμε να επεξεργαστούμε δεν υπερβαίνουν τις 500000.

## ΚΕΦΑΛΑΙΟ 2 DJANGO

Το Django είναι ένα δωρεάν και ανοικτού κώδικα web framework γραμμένο σε Python. Ένα web framework είναι ένα σύνολο από στοιχεία που μας βοηθά να αναπτύξουμε ιστοσελίδες πιο γρήγορα και εύκολα [2]. Όταν κατασκευάζουμε μία ιστοσελίδα, χρειαζόμαστε πάντα ένα παρόμοιο σύνολο στοιχείων: έναν τρόπο για να χειριζόμαστε τον έλεγχο ταυτότητας χρήστη (εγγραφή, σύνδεση, αποσύνδεση), ένα πάνελ διαχείρισης για την ιστοσελίδα μας, φόρμες, έναν τρόπο για να ανεβάζουμε αρχεία κλπ.

### 2.1 Ιδιότητες και χαρακτηριστικά Django

#### 2.1.1 Ταχύτητα

Το Django σχεδιάστηκε για να βοηθήσει τους προγραμματιστές να μεταφέρουν εφαρμογές από την ιδέα στην ολοκλήρωση το συντομότερο δυνατό.

#### 2.1.2 Πρόσθετα πακέτα

Το Django περιλαμβάνει δεκάδες πρόσθετα που μπορεί να χρησιμοποιήσουμε για να χειριστούμε κοινές εργασίες ανάπτυξης ιστού. Το Django φροντίζει για τον έλεγχο ταυτότητας χρήστη, τη διαχείριση περιεχομένου, τους χάρτες ιστότοπων, τις ροές RSS και πολλές άλλες εργασίες.

#### 2.1.3 Ασφάλεια

Το Django λαμβάνει σοβαρά υπόψη την ασφάλεια και βοηθά τους προγραμματιστές να αποφύγουν πολλά συνηθισμένα λάθη ασφαλείας, όπως η έγχυση SQL, η δέσμη ενεργειών μεταξύ ιστότοπων, η πλαστογραφία αιτήματος μεταξύ ιστότοπων και το clickjacking. Το σύστημα ελέγχου ταυτότητας χρήστη παρέχει έναν ασφαλή τρόπο διαχείρισης λογαριασμών χρηστών και κωδικών πρόσβασης.

#### 2.1.4 Κλιμάκωση

Ορισμένες από τις πιο πολυσύχναστες εφαρμογές στον πλανήτη χρησιμοποιούν την ικανότητα του Django για γρήγορη και ευέλικτη κλιμάκωση για να καλύψουν τις μεγαλύτερες απαιτήσεις κυκλοφορίας.

#### 2.1.5 Προσαρμοστικότητα

Εταιρείες, οργανισμοί και κυβερνήσεις έχουν χρησιμοποιήσει το Django για να δημιουργήσουν κάθε είδους λογισμικό, από συστήματα διαχείρισης περιεχομένου έως κοινωνικά δίκτυα έως πλατφόρμες επιστημονικής υπολογιστικής.

## 2.2 Γιατί χρησιμοποιήσα το Django

### 2.2.1 Χτισμένο με Python

Όλοι γνωρίζουμε ότι η Python είναι απλή, εύκολη στην ανάγνωση και εύκολη στην εκμάθηση. Πολλοί αρχάριοι επιλέγουν αυτήν τη γλώσσα ως την πρώτη γλώσσα προγραμματισμού λόγω της απλότητας και της εύκολης καμπύλης εκμάθησης. Κολέγια και πανεπιστήμια χρησιμοποιούν αυτήν τη γλώσσα για να διδάξουν προγραμματισμό στους φοιτητές. Όχι μόνο οι αρχάριοι αλλά και οι ειδικοί της τεχνολογίας χρησιμοποιούν αυτή τη γλώσσα για την επιστήμη των δεδομένων, την εκμάθηση μηχανών και σε διάφορους άλλους τομείς. Η Python είναι μια αρκετά σταθερή γλώσσα και το Django κληρονομεί πολλά βασικά οφέλη της Python. Αν εξετάσουμε τα βασικά στοιχεία του Django, όλα τα αρχεία και οι κωδικοί εξαιρέσεων είναι γραμμένα σε Python.

### 2.2.2 Cross-Platform

Το Django είναι ένα φορητό πλαίσιο και μπορούμε να εκτελέσουμε τον κώδικα του σε οποιαδήποτε πλατφόρμα, συμπεριλαμβανομένου του υπολογιστή, του Mac, των Windows, του Linux, κ.λπ. Το Django έχει ένα επίπεδο που ονομάζεται ORM (αντικείμενο-σχεσιακός χαρτογράφος) μεταξύ του προγραμματιστή και της βάσης δεδομένων. Με τη βοήθεια αυτού του επιπέδου, μπορούμε να μεταφέρουμε ολόκληρο το έργο σε άλλες μεγάλες βάσεις δεδομένων με λίγες γραμμές αλλαγής στον κώδικα.

### 2.2.3 Ανοικτού κώδικα και τεράστια υποστήριξη

Το Django είναι ένα δωρεάν και ανοιχτού κώδικα πλαίσιο διαθέσιμο στο Github. Υπάρχουν σχεδόν 2000+ συνεισφέροντες και πολλοί άλλοι συμμετέχουν κάθε μέρα. Υποστηρίζεται από την τεράστια κοινότητα προγραμματιστών και ο κώδικας ενημερώνεται πάντα από τους προγραμματιστές που τον χρησιμοποιούν. Πολλές νέες βιβλιοθήκες εισάγονται επίσης από την κοινότητα για την επίλυση σχετικών με την κωδικοποίηση ζητημάτων που αντιμετωπίζουν συχνά οι προγραμματιστές κατά την κατασκευή ενός έργου. [13]

Αν ψάχνουμε για κάτι που μπορεί να πει πώς μπορεί να λειτουργήσει ένα συγκεκριμένο πράγμα στο Django, οι πιθανότητες είναι μεγαλύτερες ότι το πρόβλημά μας έχει ήδη λυθεί από κάποιον άλλο προγραμματιστή στην κοινότητα και θα λάβουμε τη λύση εύκολα μόλις προκύψει το πρόβλημά μας. Υπάρχουν πολλές λίστες αλληλογραφίας, ιστολόγια, τεκμηρίωση, Slack κανάλια, συναντήσεις, εργαστήρια και άλλοι διαδικτυακοί πόροι για αυτό το πλαίσιο.

### 2.2.4 Περιλαμβάνονται μπαταρίες

Το Django είναι δημοφιλές για το "περιλαμβάνονται μπαταρίες", που σημαίνει ότι μπορείτε να βρείτε σχεδόν τα πάντα για να αναπτύξετε μια πλήρη εφαρμογή. Αυτές οι μπαταρίες περιλαμβάνουν ORM, Έλεγχος ταυτότητας, πρότυπο HTML, υποστήριξη διαχείρισης συνεδρίων, δρομολόγηση URL, Middlewares, βιβλιοθήκες HTTP, υποστήριξη πολλαπλών τοποθεσιών, i18n, μηχανή προτύπων, φόρμες, επίπεδα προβολής, επίπεδα μοντέλων, συμβατότητα rython κλπ. Τα πλαίσια προσφέρουν πολλές βιβλιοθήκες και εργαλεία για περιπτώσεις κοινής χρήσης. Αυτού του είδους η τεράστια υποστήριξη επιτρέπει στους προγραμματιστές να επικεντρωθούν σε αυτό που έχει μεγαλύτερη σημασία, αντί να επανεφεύρουν τον τροχό για τα πάντα.

### 2.2.5 Ασφάλεια

Ένα από τα καλύτερα πράγματα για το Django είναι ότι μπορούμε να δημιουργήσουμε την εφαρμογή με μεγαλύτερη ταχύτητα και να την παραδώσουμε χωρίς να διακυβεύεται η ασφάλεια της εφαρμογής. Οι λειτουργίες ασφαλείας είναι ενεργοποιημένες από προεπιλογή σε αυτό το πλαίσιο. Διαθέτει ενσωματωμένη προστασία για ορισμένα κοινά ζητήματα ασφαλείας, όπως σενάρια σε διάφορα



σημεία, πλαστογράφηση αιτημάτων, εξαγωγή κλικ και έγχυση SQL. Το Django κυκλοφορεί αρκετά συχνά νέες ενημερώσεις κώδικα ασφαλείας και ανταποκρίνεται άμεσα στις ευπάθειες ασφαλείας και ειδοποιεί άλλα πλαίσια.

#### 2.2.6 Ενσωματωμένο UI διαχειριστή

Στα περισσότερα από τα πλαίσια πρέπει να δημιουργήσουμε μόνοι μας τον πίνακα διαχείρισης και αυτό απαιτεί πολύ χρόνο. Το Django προσφέρει μια πλήρως εξοπλισμένη διεπαφή ιστού που δημιουργείται αυτόματα για κάθε εφαρμογή που δημιουργείται. Ο πίνακας διαχείρισης είναι καλά δομημένος και επιτρέπει στους προγραμματιστές να δημιουργούν/ενημερώνουν/διαγράφουν χρήστες και οποιαδήποτε άλλα αντικείμενα βάσης δεδομένων ειδικά για την εφαρμογή. Λαμβάνουμε επίσης ενότητες άδειας και ελέγχου ταυτότητας από το κουτί. Δεν χρειάζεται να περάσουν εβδομάδες ή ημέρες για να το φτιάξουμε από την αρχή. Στις περισσότερες εφαρμογές για τη δημιουργία προφίλ χρηστών, χρειάζονται διάφορες λεπτομέρειες, όπως όνομα χρήστη, email, διεύθυνση, αριθμό τηλεφώνου κλπ. Το Django παρέχει τις περισσότερες από τις απαραίτητες ανάγκες για το προφίλ χρήστη. Αυτό το είδος τεράστιας υποστήριξης δεν είναι διαθέσιμο σε πολλά πλαίσια ή βιβλιοθήκες.

#### 2.2.7 ORM (Αντιστοίχιση σχέσης χαρτογράφησης)

Το Django προσφέρει ένα πλήρως λειτουργικό ORM, συμβατό με πολλές βάσεις δεδομένων. Βασικά το ORM είναι μια βιβλιοθήκη που βοηθά τους προγραμματιστές να αλληλεπιδρούν με βάσεις δεδομένων και τους επιτρέπει να μεταφέρουν τα δεδομένα αυτόματα από τις βάσεις δεδομένων (όπως MySQL και PostgreSQL) στα αντικείμενα. Μπορούμε να εργαστούμε ταυτόχρονα με πολλές βάσεις δεδομένων.

Ορισμένες τεχνολογίες όπως το Eloquent, το greenDAO, το Yii AR, κ.λπ. χειρίζονται πολύ καλά τα βασικά ερωτήματα, αλλά κάποια στιγμή, εάν το ORM δεν αντιμετωπίσει την περίπτωση χρήσης, θα πρέπει να γράψουμε τα ακατέργαστα ερωτήματα. Αυτό δεν συμβαίνει με το Django. Είναι καλά κατασκευασμένο και χειρίζεται πολύ καλά τα ερωτήματα της βάσης δεδομένων.

Η ικανότητα του Django ORM να αντλεί δεδομένα επιταχύνει τη διαδικασία ανάπτυξης της εφαρμογής. Δεν χρειάζεται να γνωρίζουμε τη γλώσσα που

χρησιμοποιείται για την επικοινωνία της βάσης δεδομένων για τον χειρισμό δεδομένων. Επίσης, μπορούμε να κάνουμε εναλλαγή μεταξύ πολλαπλών σχεσιακών βάσεων δεδομένων με λιγότερες γραμμές πρόσθετου κώδικα. Έτσι, για ένα έργο, εάν χρησιμοποιούμε το SQLite για τοπική ανάπτυξη, μπορούμε να μεταβούμε στην παραγωγή MySQL.

#### 2.2.8 DRY (Don't Repeat Yourself)

Στον προγραμματισμό η επαναχρησιμοποίηση κώδικα ή η τήρηση της αρχής DRY είναι πραγματικά σημαντική, ειδικά όταν ενημερώνουμε τον κώδικα τακτικά. Όταν ακολουθούμε την αρχή DRY απλά δεν χρησιμοποιούμε τον υπάρχοντα κώδικα, αποφεύγουμε επίσης τις περιττές γραμμές κώδικα, σφάλματα ή σφάλματα στην εφαρμογή. Με την αρχή DRY, μπορούμε να αξιοποιήσουμε στο έπακρο τον πολύ λίγο κώδικα και αυτό εξοικονομεί πολύ χρόνο στους προγραμματιστές όταν πρόκειται να λειτουργήσει ο κώδικας μας ή να τροποποιήσουμε τον κώδικα για οποιονδήποτε λόγο.

Στο μεγαλύτερο μέρος του πλαισίου, πρέπει να καταβάλλουμε προσπάθειες για να κάνουμε τον κωδικό μας DRY συμβατό και κάθε φορά δεν είναι δυνατό να συνεχίσουμε να ελέγχουμε τον κωδικό μας, ειδικά όταν εργαζόμαστε σε ομάδα. Το Django ακολουθεί την αρχή DRY και έχει σχεδιαστεί με τέτοιο τρόπο ώστε να πρέπει να βγούμε από το δρόμο μας για να παραβιάσουμε την αρχή DRY. Αυτή η δυνατότητα μάς επιτρέπει να χρησιμοποιήσουμε ξανά τον υπάρχοντα κώδικα και να εστιάσουμε στον μοναδικό.

#### 2.2.9 Επεκτάσιμο και αξιόπιστο

Σήμερα κάθε νεοσύστατη εταιρεία ή εταιρεία ανησυχεί για την κλιμάκωση της εφαρμογής. Τι γίνεται αν ο ιστότοπος φτάσει σε κλίμακα και χρειάζεται να χειριστεί τη μεγάλη επισκεψιμότητα και τον μεγάλο όγκο πληροφοριών; Σίγουρα πρέπει να χρησιμοποιήσετε το πλαίσιο που μπορεί να χειριστεί τον τεράστιο όγκο δεδομένων. Λοιπόν, το Django είναι σε θέση να αντιμετωπίσει το έργο οποιουδήποτε μεγέθους είτε πρόκειται για εφαρμογή ιστού μικρής κλίμακας είτε για εφαρμογή Ιστού με υψηλή φόρτωση. Το Django συνοδεύεται από μια σειρά ενσύρματων εξαρτημάτων, έτοιμα να ξεκινήσουν από προεπιλογή. Αυτά τα εξαρτήματα αποσυνδέονται, έτσι σύμφωνα με τις απαιτήσεις ή συγκεκριμένες

λύσεις σε μια εφαρμογή, η ανάπτυξη μπορεί να κλιμακωθεί ή να μειωθεί με την αντικατάσταση ή την αποσύνδεση των εξαρτημάτων.

Το Django είναι ένα πολύ δημοφιλές και ευρέως χρησιμοποιούμενο πλαίσιο εφαρμογών Ιστού σε όλες τις βιομηχανίες. Αυτός είναι ο λόγος που πολλοί πάροχοι υπηρεσιών cloud λαμβάνουν όλα τα μέτρα για να αναπτύξουν την εφαρμογή γρήγορα και εύκολα στις πλατφόρμες τους. Ας πάρουμε για παράδειγμα το Heroku. Μόλις ρυθμιστεί, μπορούμε να ενεργοποιήσουμε την ανάπτυξη και να διαχειριστούμε την εφαρμογή με μία μόνο εντολή από οποιονδήποτε εξουσιοδοτημένο προγραμματιστή. Οι ειδικοί του Django που εργάζονται σε αυτούς τους τομείς αναπτύσσουν μια πιο λειτουργική, αξιόπιστη και αποτελεσματική εφαρμογή.

Γνωστές εφαρμογές που χρησιμοποιούν το Django:

- Disqus
- Instagram
- Spotify
- Youtube
- The Washington Post
- BitBucket
- DropBox
- Eventbrite
- Mozilla
- Pinterest
- Prezi
- Nasa Official Website
- Quora
- The Onion

#### 2.2.10 Καλή τεκμηρίωση

Για γρήγορη αναφορά, η καλή τεκμηρίωση για οποιοδήποτε πλαίσιο ή γλώσσα έχει μεγάλη σημασία, ειδικά όταν κολλήσουμε κάπου κατά τη φάση ανάπτυξης οποιουδήποτε έργου. Το Django παρέχει καλά οργανωμένη τεκμηρίωση με

παράδειγμα κώδικα, ο οποίος είναι πολύ χρήσιμος στη δημιουργία διαφόρων ειδών εφαρμογών πραγματικού κόσμου.

## 2.3 Μειονεκτήματα του Django

### 2.3.1 Δεν είναι κατάλληλο για έργα μικρής κλίμακας

Το Django μερικές φορές μπορεί να είναι υπερβολικό, αλλά η Python μας επιτρέπει να χρησιμοποιούμε άλλα πλαίσια για να αναπτύξουμε απλές λύσεις. Για παράδειγμα, εάν πρέπει να σχεδιάσουμε μια απλή συνομιλία, το Django μπορεί να είναι πολύ μεγάλο πλαίσιο και μπορούμε να χρησιμοποιήσουμε το Flask, ένα πλαίσιο μικροϋπηρεσίας.

### 2.3.2 Δεν υπάρχει προεπιλεγμένη υποστήριξη για WebSockets

Τα WebSockets μας επιτρέπουν να ενημερώνουμε πληροφορίες ή συμβάντα σε πραγματικό χρόνο. Το Django δεν υποστηρίζει ακόμη εφαρμογές ιστού σε πραγματικό χρόνο. Επομένως, πρέπει να χρησιμοποιήσουμε άλλα πλαίσια όπως το aiohttp.

### 2.3.3 Μονολιθικό

Ορισμένες εσωτερικές ενότητες Django, όπως το ORM και οι φόρμες, είναι δύσκολο να αντικατασταθούν. Θα χρειαστεί πολλή προσπάθεια από τους προγραμματιστές για να αλλάξουν την εσωτερική δομή.

### 2.3.4 Η συμπεριφορά του Django είναι μερικές φορές δύσκολο να συντονιστεί

Ορισμένες εσωτερικές ενότητες Django, όπως ο πίνακας διαχείρισης, είναι δύσκολο να συντονιστούν λόγω της φιλοσοφίας του Django. Για παράδειγμα, εάν θέλουμε να προσθέσουμε έναν σύνδεσμο, δυναμικά στατιστικά ή κάτι μοναδικό που δεν περιλαμβάνεται στο οικοσύστημα Django, αυτό μπορεί κυριολεκτικά να διαρκέσει ώρες. Τα πλεονεκτήματα του Django υπερτερούν των μειονεκτημάτων, αλλά δεν πρέπει ποτέ να περιοριζόμαστε σε ένα μόνο πλαίσιο και να μην εξετάζουμε εναλλακτικές λύσεις.

## 2. 4 Η αρχιτεκτονική Django

Το django ακολουθεί **Model-View-Controller(MVC)** αρχιτεκτονική , χωρισμένη σε τρία διαφορετικά μέρη:

- Το **Model** είναι η δομή δεδομένων κάτω από όλη την εφαρμογή , η οποία παριστάνεται από μία βάση δεδομένων.(MySQL,Postgres κλπ)
- Το **View** είναι ένα user interface , το οποίο ευθύνεται για το τι βλέπει ο χρήστης όταν επισκέπτεται την ιστοσελίδα. Αποτελείται από HTML/CSS/Javascript αρχεία.
- Ο **Controller** βρίσκεται ανάμεσα στο Model και View και ουσιαστικά τα συνδέει , περνώντας δεδομένα από το model στο view.

Με το MVC, η εφαρμογή μας θα περιστρέφεται γύρω από το μοντέλο - είτε το εμφανίζει είτε το χειρίζεται.Ας πούμε λοιπόν ότι ένας χρήστης θα εισαγάγει μια διεύθυνση URL στο πρόγραμμα περιήγησής του, το αίτημα αυτό θα περάσει από τα πρωτόκολλα διαδικτύου, στον διακομιστή μας, ο οποίος θα καλέσει το Django. Στη συνέχεια, το Django θα επεξεργαστεί τη συγκεκριμένη διαδρομή URL και εάν ταιριάζει με μια διαδρομή URL που έχουμε δηλώσει ρητά, θα καλέσει τον ελεγκτή, ο οποίος στη συνέχεια θα εκτελέσει μια συγκεκριμένη ενέργεια, όπως να λάβει μια καταχώρηση από το μοντέλο μας (βάση δεδομένων) και στη συνέχεια να αποδώσει μια Προβολή (δηλαδή: κείμενο JSON, ιστοσελίδα HTML/CSS/JavaScript).[1]

## 2.5 Εφαρμογές και βασικά αρχεία Django

Κάθε Django project θα έχει τουλάχιστον μία Django app.Ένα project Django περιλαμβάνει την εφαρμογή και όλα τα συστατικά της, ενώ μια Django app είναι ένα υπο-δοχείο της εφαρμογής με τη δική της λειτουργικότητα που, θεωρητικά, μπορεί να επαναχρησιμοποιηθεί σε άλλη εφαρμογή χωρίς πολλές τροποποιήσεις. Μια τυπική δομή έργου Django :

```

MySite/
  manage.py
  MySite/
    __init__.py
    settings.py
    urls.py
    wsgi.py
  myapp/
    views.py
    models.py
    __init__.py
    admin.py
    tests.py
    urls.py
    static/
      myapp/
        style.css
        index.js
        background.jpg
    templates/
      index.html
      aboutus.html
      post.html
    migrations/
      __init__.py
      0001_initial.py

```

Σχήμα 1.25: Τυπική δομή έργου Django

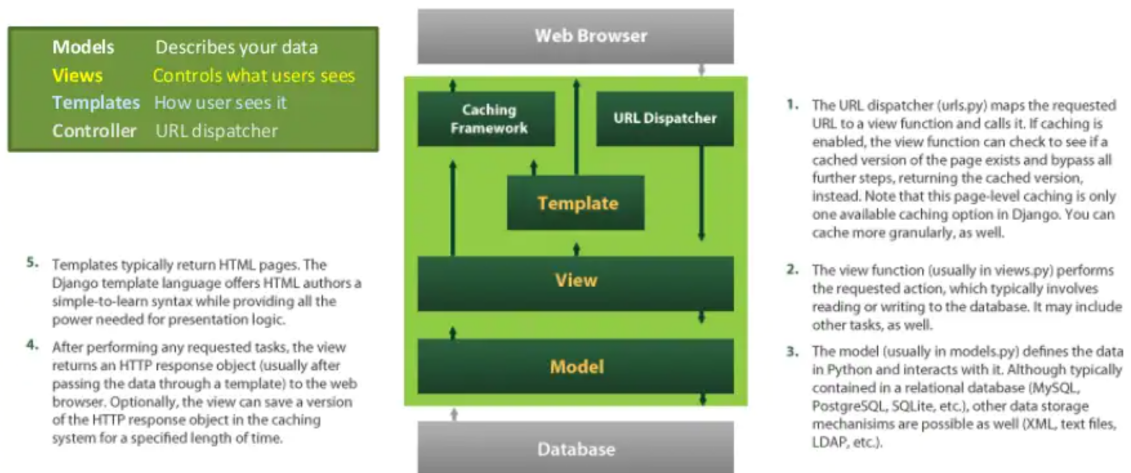
Όλα τα αρχεία στο `myapp/` περιγράφουν την εφαρμογή Django που ονομάζεται `myapp`.

- Το `myapp/models.py` είναι το Μοντέλο ή όπου ορίζουμε τη βάση δεδομένων μας. Ας πάρουμε για παράδειγμα έναν ιστότοπο blogging. Στο ελάχιστο, θα έχουμε έναν πίνακα δημοσιεύσεων ιστολογίου και έναν πίνακα σχολίων. Κάθε δημοσίευση ιστολογίου έχει έναν τίτλο, συγγραφέα, ημερομηνία και το περιεχόμενό της, ενώ τα σχόλια έχουν το ίδιο πράγμα.
- Το `myapp/views.py` είναι ο ελεγκτής. Μέσα στο `views.py` θα ορίσουμε διαφορετικές λειτουργίες/κλάσεις (το Django προσφέρει υποστήριξη και για τα δύο, αλλά οι περισσότεροι εξακολουθούν να χρησιμοποιούν λειτουργίες) Μια συνάρτηση καθορίζει τι συμβαίνει όταν υπάρχει πρόσβαση σε ένα συγκεκριμένο URL και υποβάλλεται αίτημα HTTP στο διακομιστή. Συνήθεις ενέργειες θα ήταν η αναζήτηση της βάσης δεδομένων για εγγραφές και η απόδοση ενός συγκεκριμένου αρχείου προτύπου HTML.
- Όλα κάτω από το `myapp/templates/myapp/` είναι πρότυπα ή αρχεία HTML που καθορίζουν την προβολή μας. Όταν μια συνάρτηση στο `views.py` αποδίδει ένα αρχείο HTML, μπορεί να περάσει αντικείμενα όπως μια λίστα σχολίων στα οποία μπορούμε να χρησιμοποιήσουμε ειδική σύνταξη για να εμφανίσουμε αυτά τα σχόλια. Σε κάθε πρότυπο, μπορούμε να λάβουμε

στατικά αρχεία όπως CSS, αρχεία Javascript ή εικόνες για να δώσουμε ζωή στην ιστοσελίδα.

Το `urls.py` είναι το αρχείο διαμόρφωσης URL. Αυτό είναι το αρχείο που μας επιτρέπει να αντιστοιχίσουμε μια συγκεκριμένη διεύθυνση URL σε μια συγκεκριμένη συνάρτηση στο `views.py`. Ας υποθέσουμε ότι έχουμε διακομιστή που λαμβάνει αιτήματα από τον τομέα `mydomain.com`. Εάν μεταβούμε στο `mydomain.com/about-us`, μια συγκεκριμένη συνάρτηση όπως το `aboutus ()` θα δημιουργήσει μια απάντηση. Το Django θα μας επιτρέψει επίσης να συμπεριλάβουμε σημαντικές πληροφορίες στη διεύθυνση URL. Ένα παράδειγμα θα ήταν αν θέλουμε να λάβουμε μια ανάρτηση ιστολογίου με αναγνωριστικό 4. Αναγκάζουμε το Django να στείλει τον αριθμό "4" σε μια συνάρτηση που ονομάζεται `blogspot (blog_id)` όταν πηγαίνουμε `tomydomain.com/blogs/4/`

## Django Architecture



Σχήμα 1.26: Αρχιτεκτονική Django [1]

## 2.6 Django administration page

Το django μέσω του administration page προσφέρει την δυνατότητα να διαχειριστούμε τα μοντέλα και την βάση δεδομένων.

### Django administration

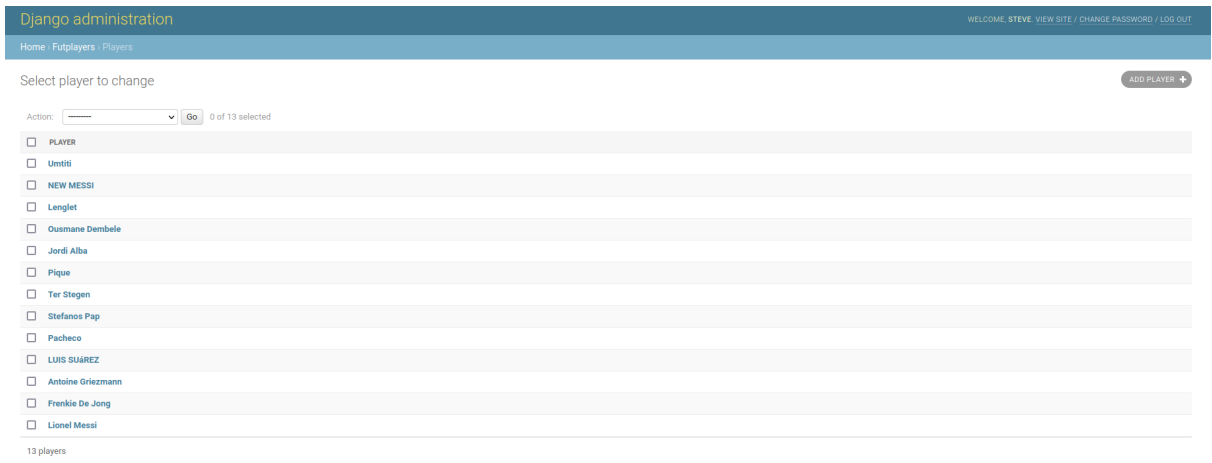
#### Site administration

AUTHENTICATION AND AUTHORIZATION	
<b>Groups</b>	<a href="#">+ Add</a> <a href="#">✎ Change</a>
<b>Users</b>	<a href="#">+ Add</a> <a href="#">✎ Change</a>
FUTPLAYERS	
<b>My teams</b>	<a href="#">+ Add</a> <a href="#">✎ Change</a>
<b>Nationalitys</b>	<a href="#">+ Add</a> <a href="#">✎ Change</a>
<b>Players</b>	<a href="#">+ Add</a> <a href="#">✎ Change</a>
<b>Teams</b>	<a href="#">+ Add</a> <a href="#">✎ Change</a>

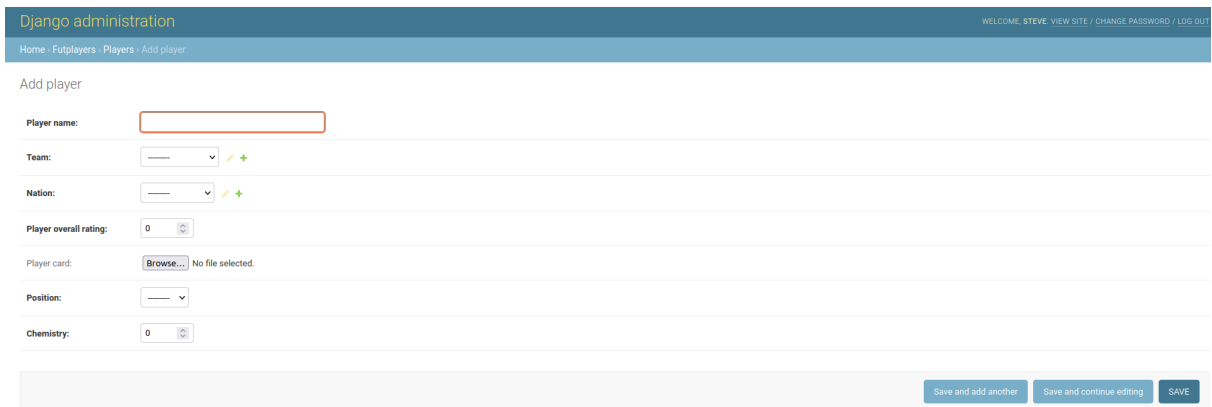
Σχήμα 1.27: Django Administration Home Page

Συγκεκριμένα μπορούμε να δούμε όλες τις εγγραφές στην βάση αλλά και να προσθέσουμε /αφαιρέσουμε εγγραφές ή να τροποποιήσουμε κάποια εγγραφή . Αυτό το εργαλείο είναι πολύ σημαντικό καθώς μας δίνει την δυνατότητα να διαχειριστούμε το back-end της εφαρμογής και να κάνουμε ότι test επιθυμούμε.





Σχήμα 1.28: Django Administration Page Model View



Σχήμα 1.29: Django Administration Page Edit Model View

## ΚΕΦΑΛΑΙΟ 3 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ

Σκοπός του συστήματος είναι να μπορεί ο χρήστης να δημιουργήσει την ομάδα της επιθυμίας του . Συνεπώς προσφέρεται η λειτουργία δημιουργίας μιας ομάδας όπου ο χρήστης μπορεί ανά πάσα στιγμή να προσθέσει ή να αφαιρέσει κάποιον παίκτη , να αλλάξει το σύστημα της ομάδας( οπότε αλλάζει και ο βαθμός συγγένειας κάθε παίκτη και η συνολική χημεία της ομάδα) και παράλληλα να βλέπει την συνολική χημεία και Overall rating της ομάδας. Το σύστημα ζητάει από τον χρήστη να τοποθετήσει όσους παίκτες σε όποια θέση επιθυμεί και με βάση το σύστημα που έχει επιλεγεί προτείνει τους κατάλληλους παίκτες ανά θέση. Αρχικά μέσω του look up table δημιουργείται μία δομή που αποτελείται από τους γείτονες του κάθε παίκτη που θα προτείνει το σύστημα. Με αυτόν τον τρόπο δημιουργείται ένα προσωρινό dataset που αποτελείται από παίκτες ίδιου πρωταθλήματος,ίδιας ομάδας ή ίδιας εθνικότητας με τους γειτονικούς παίκτες . Είναι λογικό ότι σε κάθε θέση θα θέλουμε να προσθέσουμε μόνο σχετικούς προς τους γύρω παίκτες , αφού πάντα ο στόχος μας είναι η χημεία να αυξηθεί .Στην περίπτωση που είναι ενεργοποιημένο το φίλτρο Target Overall Rating αφαιρούνται παίκτες που ξέρουμε ότι το overall rating είναι πολύ χαμηλό ή πολύ υψηλό για να φτάσουμε το συνολικό στόχο Overall Rating της ομάδας.

### 3.1 Αλγόριθμος υπολογισμού αξιολόγησης της ομάδας

Σε περίπτωση που ζητηθεί από τον χρήστη κάποιο συγκεκριμένο όριο Rating θα πρέπει να φιλτράρουμε τους παίκτες με βάση το όριο αφαιρώντας τους από την λίστα με τους πιθανούς υποψηφίους. Πάντα είμαστε σε θέση να γνωρίζουμε τους πιθανούς συνδυασμούς εφόσον το Overall της ομάδας αποτελείται από τον μέσο όρο του ατομικού overall rating του κάθε παίκτη. Ακολουθεί παράδειγμα υπολογισμού όλων των συνδυασμών παικτών για ομάδα με rating 80 :

75	76	77	78	79	80	81	82	83
0	0	0	1	3	6	1	0	0
0	0	0	0	2	9	0	0	0
0	0	0	0	6	3	2	0	0
1	0	0	0	0	10	0	0	0
0	0	0	1	6	1	3	0	0
0	0	0	3	3	2	3	0	0
0	0	0	1	6	2	1	1	0
0	0	0	0	6	4	0	1	0
0	0	0	0	9	0	0	2	0
0	0	0	6	0	1	4	0	0
0	0	1	4	1	1	4	0	0
0	0	0	1	6	3	0	0	1
0	0	0	3	3	4	0	0	1
0	0	0	2	7	0	0	1	1
0	0	1	0	8	0	0	1	1
0	4	1	0	1	0	4	0	1

Σχήμα 1.30: Παράδειγμα αλγορίθμου υπολογισμού rating [5]

Για κάθε κενή θέση δημιουργείται μία λίστα από υποψήφιους παίκτες και στην συνέχεια υπολογίζεται η χημεία της ομάδας και το συνολικό κόστος για κάθε δυνατό συνδυασμό. Συνδυασμοί παικτών που δεν ικανοποιούν την συνθήκη

$$\text{Team Chemistry} \geq \text{Target Chemistry}$$

αφαιρούνται από την δομή . Η τελική δομή έχει την παρακάτω μορφή :

Combined Candidates
combination0( candidate0,candidate1,...candidateN,team_chemistry,team_price)
combination1( candidate0,candidate1,...candidateN,team_chemistry,team_price)
combination2( candidate0,candidate1,...candidateN,team_chemistry,team_price)
....
....
combinationN( candidate0,candidate1,...candidateN,team_chemistry,team_price)

Σχήμα 1.31: Combined candidates data structure

Η χημεία της ομάδας και η τιμή της για συνδυασμό παικτών αποτελούν τα δύο αντικρουόμενα κριτήρια πάνω στα οποία θα εφαρμοστούν οι αλγόριθμοι λήψης αποφάσεων που μελετήθηκαν.

### 3.2 Weighted sum model

Ο αλγόριθμος weighted sum model υπολογίζει το score βάση της παρακάτω εξίσωσης [6][7]

$$WSNscore = (C1+ C2+ .. + Cn ) \times 0.70 + Ctm \times 0.30 , \text{όπου}$$

$C1+ C2+ .. + Cn$  είναι το άθροισμα των τιμών των υποψήφιων παικτών και  $Ctm$  είναι η συνολική χημεία της ομάδας.Οι δύο τιμές πολλαπλασιάζονται με τα βάρη που έχουμε επιλέξει.

### 3.3 Weighted product model

Ο αλγόριθμος weighted product model χωρίζει τα κριτήρια σε beneficial τα οποία είναι αυτά που επιθυμούμε να είναι στην μέγιστη τιμή τους(χημεία της ομάδας) και

σε non-beneficial που είναι αυτά που επιθυμούμε να είναι στην ελάχιστη τιμή(τιμή).

Αρχικά τα δεδομένα κανονικοποιούνται βάση του παρακάτω κανόνα

- For beneficial attributes,  $X = x/x_{max}$
- For non-beneficial attributes,  $X = X_{min}/x$

και μετά υπολογίζεται το score για κάθε συνδυασμό όπως στον weighted sum model.[17] , [18]

### 3.4 VIKOR vs TOPSIS

Η μέθοδος VIKOR (Opricovic, 1998, Opricovic & Tzeng, 2004, 2007)[9] προσπαθεί να καταλήξει σε μία απόφαση για ένα πρόβλημα με αντικρουόμενα κριτήρια δίνοντας ιδιαίτερη έμφαση στον τρόπο που κατατάσει και επιλέγει μία λύση που θεωρεί ο αλγόριθμος ικανοποιητική(compromise solution). Βασικό χαρακτηριστικό της αποτελεί η εύρεση της ιδανικότερης λύσης με κύριο γνώμονα την απόσταση της υπάρχουσας λύσης από την βέλτιστη λύση . Για αυτό τον σκοπό χρησιμοποιείται η LLpp-μετρική με την μορφή μια συνάρτησης συνάθροισης . Βασικό πλεονέκτημα της VIKOR είναι ότι μας δίνεται η δυνατότητα να επιλέγουμε τον τρόπο που θέλουμε να επιλέξουμε τα κριτήρια θέτοντας άλλοτε ως βασικό κριτήριο την καλή επίδοση στο σύνολο των κριτηρίων ( $pp=1$ ) και άλλοτε την καλή επίδοση σε κάθε μεμονωμένο κριτήριο ( $pp = \infty$ ).

Έστω ότι έχουμε τον παρακάτω πίνακα αποφάσεων:

	$C_1$	$C_2$	...	$C_n$
$A_1$	$f_{11}$	$f_{12}$	...	$f_{1n}$
$A_2$	$f_{21}$	$f_{22}$	...	$f_{2n}$
...	...	...	...	...
$A_m$	$f_{m1}$	$f_{m2}$	...	$f_{mn}$

όπου  $A_1, A_2, \dots, A_m$  είναι οι διαθέσιμες εναλλακτικές,  $C_1, C_2, \dots, C_n$  είναι τα κριτήρια και  $f_{ij}$  είναι η επίδοση της εναλλακτικής  $A_i$  έναντι του κριτηρίου  $C_j$  με δεδομένο ότι το βάρος είναι τα ακόλουθα:

$$[w_1, w_2, \dots, w_n]$$

που καθορίζουν το πόσο σημαντικό είναι το κάθε κριτήριο και πρέπει να ισχύει ότι:

$$\sum_{j=1}^n w_j = 1 \text{ και } w_j \geq 0, \quad j = 1, \dots, n$$

Το μοντέλο της VIKOR ακολουθεί τα παρακάτω βήματα:

1. Υπολογισμός της καλύτερης και της χειρότερης επίδοσης για κάθε κριτήριο

- Αν το  $j$ -στο κριτήριο είναι ωφέλιμο ,τότε

$$f_j^* = \max_i f_{ij}, \quad f_j^- = \min_i f_{ij}$$

-Αν το κριτήριο είναι κοστοβόρο ,τότε

$$f_j^* = \min_i f_{ij}, \quad f_j^- = \max_i f_{ij}$$

2. Υπολογισμός των τιμών  $S_i$  και  $R_i$  ,  $i = 1, \dots, m$ .

$$S_i = \sum_{j=1}^n w_j \frac{|f_j^* - f_{ij}|}{|f_j^* - f_j^-|} \qquad R_i = \max_j w_j \frac{|f_j^* - f_{ij}|}{|f_j^* - f_j^-|}$$

3. Υπολογισμός των τιμών  $Q_i$ ,  $i = 1, \dots, m$

$$Q_i = v \frac{(S_i - S^*)}{(S^- - S^*)} + (1 - v) \frac{(R_i - R^*)}{(R^- - R^*)}$$

$$S^* = \min_i S_i, \quad S^- = \max_i S_i$$

$$R^* = \min_i R_i, \quad R^- = \max_i R_i$$

Η παράμετρος  $v \in \pi[0,1]$  και καθορίζει το πλεονέκτημα της στρατηγικής που αναφέραμε παραπάνω .

4. Διαμόρφωση τριών λιστων κατάταξης με βάση τις τιμές  $S$  ,  $R$  και  $Q$

5. Εύρεση της λύσης συμβιβασμού όταν ισχύουν οι παρακάτω προϋποθέσεις:

-C1 (Συγκριτικό Πλεονέκτημα):

$$Q(A'') - Q(A') \geq DQ$$

όπου  $A''$  είναι η εναλλακτική που είναι δεύτερη στην κατάταξη ως προς τα  $Q$

$$DQ = \frac{1}{m-1}$$

$m$ : αριθμός των εναλλακτικών

-C2 (Ευστάθεια κατά τη Λήψη Απόφασης): Η εναλλακτική  $AA'$  οφείλει να έχει την καλύτερη επίδοση (minimum) ως προς την τιμή  $SS$  ή/και  $RR$ .

Αν κάποια από τις παραπάνω συνθήκες δεν ισχύει τότε:

- Αν δεν ικανοποιείται η C2 τότε σαν λύση συμβιβασμού έχουμε το σύνολο

$$\{A', A''\}$$

- Αν δεν ικανοποιείται η C1 τότε σαν λύση συμβιβασμού έχουμε το σύνολο

$$\{A', A'', \dots, A^{(M)}\}$$

όπου  $Q(A^{(M)}) - Q(A') < DQ$

Δηλαδή θεωρούμε σαν λύση τις εναλλακτικές που βρίσκονται στις  $M$  καλύτερες θέσεις της λίστας κατάταξης ως προς  $Q$

Η μέθοδος Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [8] των Hwang and Yoon (1981) είναι μία μέθοδος αντιστάθμισης που

προσπαθεί να βρει την λύση που έχει την μικρότερη γεωμετρική απόσταση από την ιδανική λύση και παράλληλα την μεγαλύτερη απόσταση από την λιγότερο ιδανική λύση. Η TOPSIS διαμορφώνει και κανονικοποιεί τον πίνακα απόφασης που περιέχει εναλλακτικές και κριτήρια , υπολογίζει τον σταθμισμένο πίνακα απόφασης και προσδιορίζει την θετική και αρνητική λύση στην διαδικασία καθορισμού της τελικής κατάταξης. η TOPSIS επεκτάθηκε από τον Chen (2000), μέσω της εισαγωγής της τριγωνικής μεθόδου για τον υπολογισμό της απόστασης μεταξύ δύο τριγωνικών ασαφών αριθμών, διαμορφώνοντας έτσι τη μέθοδο Fuzzy TOPSIS, η οποία επεκτάθηκε περαιτέρω για να διαχειρίζεται διαφορετικούς τύπους δεδομένων (Chen and Tsao, 2008 και Chen and Lee, 2010).[14] , [15] , [16]

Το μοντέλο της TOPSIS ακολουθεί τα παρακάτω βήματα:

1. Σχεδιασμός πίνακα απόφασης A

$$A = \begin{bmatrix} x_{11} & x_{1n} \\ x_{m1} & x_{mn} \end{bmatrix}$$

όπου  $A_1, A_2, \dots, A_m$  είναι οι διαθέσιμες εναλλακτικές,  $C_1, C_2, \dots, C_n$  είναι τα κριτήρια και  $X_{ij}$  είναι η επίδοση της εναλλακτικής  $A_i$  έναντι του κριτηρίου  $C_j$  .

2. Υπολογισμός του κανονικοποιημένου πίνακα απόφαση R

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{t=1}^m x_{tj}^2}}$$

όπου το  $r_{ij}$  αναπαριστά την κανονικοποιημένη επίδοση της εναλλακτικής  $A_i$  έναντι του κριτηρίου  $C_j$

3. Υπολογισμός του σταθμισμένου κανονικοποιημένου πίνακα P , πολλαπλασιάζοντας τον κανονικοποιημένο πίνακα R με τα αντίστοιχα βάρη. Το διάνυσμα των βαρών  $W = [w_1, w_2, \dots, w_n]$  και πρέπει να ισχύει:

$$\sum_{j=1}^{j=n} w_j = 1$$



Η σταθμισμένη κανονικοποιημένη αξία  $p_{ij}$  υπολογίζεται από τον τύπο:

$$p_{ij} = w_j \times r_{ij}$$

4. Προσδιορισμός των διανυσμάτων της θετικής ιδεατής λύσης  $PP^+$  και της αρνητικής ιδεατής λύσης  $PP^-$ , υπολογίζοντας τις θετικές και αρνητικές ιδεατές λύσεις για κάθε κριτήριο:

$$P^+ = (p_1^+, p_2^+, \dots, p_n^+)$$

$$P^- = (p_1^-, p_2^-, \dots, p_n^-)$$

$$p_i^+ = \{(\max p_{ij}, j \in J) \text{ or } (\min p_{ij}, j \in J')\}$$

$$p_i^- = \{(\min p_{ij}, j \in J) \text{ or } (\max p_{ij}, j \in J')\}$$

όπου το  $J$  αναπαριστά κριτήρια θετικής επίδρασης και το  $J'$  τα κριτήρια αρνητικής επίδρασης

5. Υπολογισμός της απόστασης κάθε εναλλακτικής από την θετική ιδεατή λύση και την αρνητική ιδεατή λύση :

$$S_i^- = \sqrt{\sum_{j=1}^n (p_{ij} - p_j^-)^2}$$

6. Υπολογισμός της σχετικής εγγύτητας  $D_i$  στην θετική ιδεατή λύση για κάθε εναλλακτική  $A_i$  :

$$D_i = \frac{S_i^-}{S_i^+ + S_i^-}$$

Αξίζει να σημειωθεί ότι η Behavioural Topsis είναι σε θέση να καταλάβει την σχέση μεταξύ της απόστασης από την ιδεατή και την μη ιδεατή λύση θεωρώντας την ιδεατή ως απώλεια ευκαιρίας και την μη ιδεατή ως το όφελος από την λήψη της λύσης  $A_i$  αντί της ιδεατής . Έτσι υπολογίζεται ο λόγος αποφυγής της απώλειας ως:

$$\lambda = \frac{\text{Change in Gain}}{\text{Change in Loss}} = \frac{\Delta S^-}{\Delta S^+}$$

όπου η συμπεριφορά είναι ριψοκίνδυνη για  $\lambda < 1$ , ουδέτερη για  $\lambda = 1$ , και συντηρητική ως προς τον κίνδυνο για  $\lambda > 1$ . Τέλος, η συνάρτηση αξίας για κάθε εναλλακτική δίνεται βάσει της οποίας ταξινομούνται οι εναλλακτικές

$$V = S^- - \lambda S^+$$

Οι μέθοδοι MCDM VIKOR και TOPSIS βασίζονται και οι δύο σε μια αθροιστική συνάρτηση που αντιπροσωπεύει την «εγγύτητα στο ιδανικό».[10][11] Η μέθοδος VIKOR εισάγει τον δείκτη κατάταξης με βάση το συγκεκριμένο μέτρο «εγγύτητας» στην ιδανική λύση. Αντίθετα, η βασική αρχή της μεθόδου TOPSIS είναι ότι η επιλεγμένη εναλλακτική πρέπει να έχει τη «μικρότερη απόσταση» από την ιδανική λύση και την «πιο απομακρυσμένη απόσταση» από την «αρνητική-ιδανική» λύση. Η μέθοδος TOPSIS εισάγει δύο σημεία «αναφοράς», αλλά δεν λαμβάνει υπόψη τη σχετική σημασία των αποστάσεων από αυτά τα σημεία. Αυτές οι δύο μέθοδοι MCDM χρησιμοποιούν διαφορετικά είδη κανονικοποίησης για την εξάλειψη των μονάδων συναρτήσεων κριτηρίου, ενώ η μέθοδος VIKOR χρησιμοποιεί γραμμική κανονικοποίηση, η μέθοδος TOPSIS χρησιμοποιεί κανονικοποίηση διανύσματος. Η κανονικοποιημένη τιμή στη μέθοδο VIKOR δεν εξαρτάται από τη μονάδα αξιολόγησης μιας συνάρτησης κριτηρίου, ενώ οι κανονικοποιημένες τιμές με κανονικοποίηση διανύσματος στη μέθοδο TOPSIS μπορεί να εξαρτώνται από τη μονάδα αξιολόγησης. Η μέθοδος VIKOR είναι πιο κατάλληλη για την εφαρμογή μας γιατί δεν εξαρτάται από τη μονάδα αξιολόγησης των κριτηρίων και τα κριτήριά μας ποικίλλουν. Ωστόσο η μέθοδος Weighted product model είναι η σαφής και πιο άμεση επομένως επιλέχθηκε ως η επικρατέστερη.[12]

Στην δική μας περίπτωση έχουμε κρατήσει τα βάρη ως Price = 0.7 και Chemistry = 0.3 για να μπορούμε να βγάλουμε σαφή συμπεράσματα. Ακολουθεί ένα μικρό παράδειγμα για το πως δουλεύει ο αλγόριθμος:

	<b>Candidate 0 Price</b>	<b>Candidate 1 Price</b>	<b>Team Chemistry</b>
<b>Weight</b>	<b>0.35</b>	<b>0.35</b>	<b>.30</b>
	500(min)	1000	<b>80</b>
	2000	300	70
	5000	200(min)	90(max)

Σχήμα 1.32: Παράδειγμα WPM

### Κανονικοποίηση

	<b>Candidate 0 Price</b>	<b>Candidate 1 Price</b>	<b>Team Chemistry</b>
<b>Weight</b>	<b>0.35</b>	<b>0.35</b>	<b>.30</b>
	500/500	200/1000	<b>80/90</b>
	500/2000	200/300	70/90
	500/5000	200/200	90/90

Σχήμα 1.32: Κανονικοποίηση

	<b>Candidate 0 Price</b>	<b>Candidate 1 Price</b>	<b>Team Chemistry</b>
<b>Weight</b>	<b>0.35</b>	<b>0.35</b>	<b>.30</b>
	1	0.2	<b>0.88</b>
	0.2	0.66	0.77
	0.1	1	1

Σχήμα 1.33: Κανονικοποίηση(2)

## Υπολογισμός των δυνάμεων των παραμέτρων

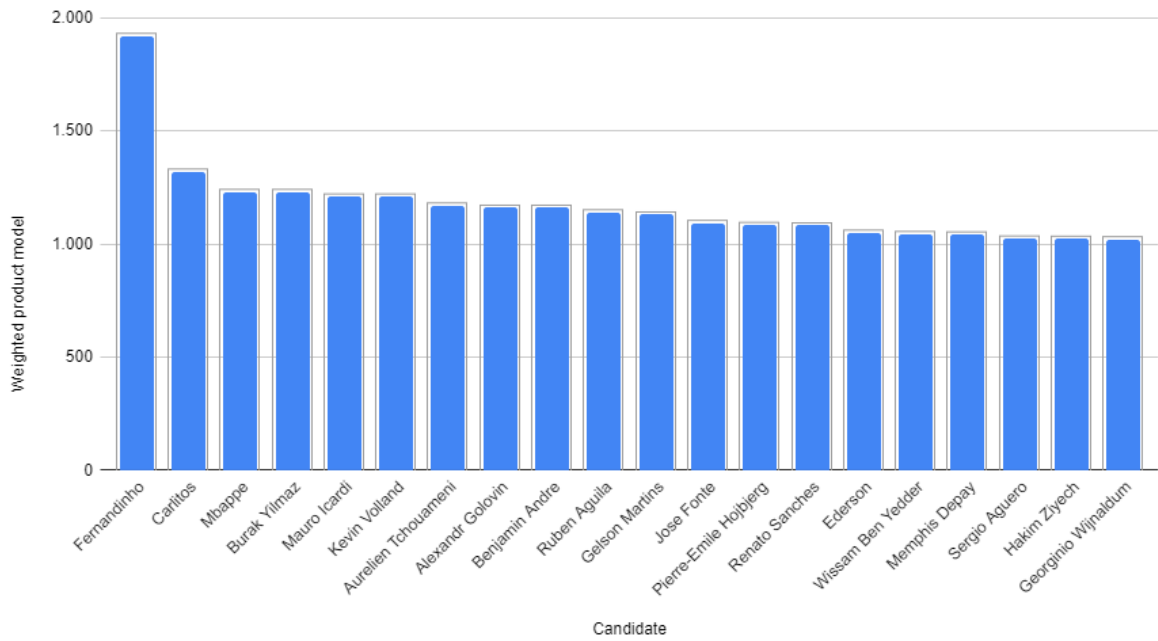
	<b>Candidate 0 Price</b>	<b>Candidate 1 Price</b>	<b>Team Chemistry</b>	<b>Total Score</b>
<b>Weight</b>	<b>0.35</b>	<b>0.35</b>	<b>.30</b>	
	$1^{0.35}$	$0.2^{0.35}$	$0.88^{0.30}$	2.2166
	$0.2^{0.35}$	$0.66^{0.35}$	$0.77^{0.30}$	2.3585
	$0.1^{0.35}$	$1^{0.35}$	$1^{0.30}$	<b>2.4466</b>

Σχήμα 1.34: Υπολογισμός των δυνάμεων των παραμέτρων

### 3.5 Αποτελέσματα και συμπεράσματα

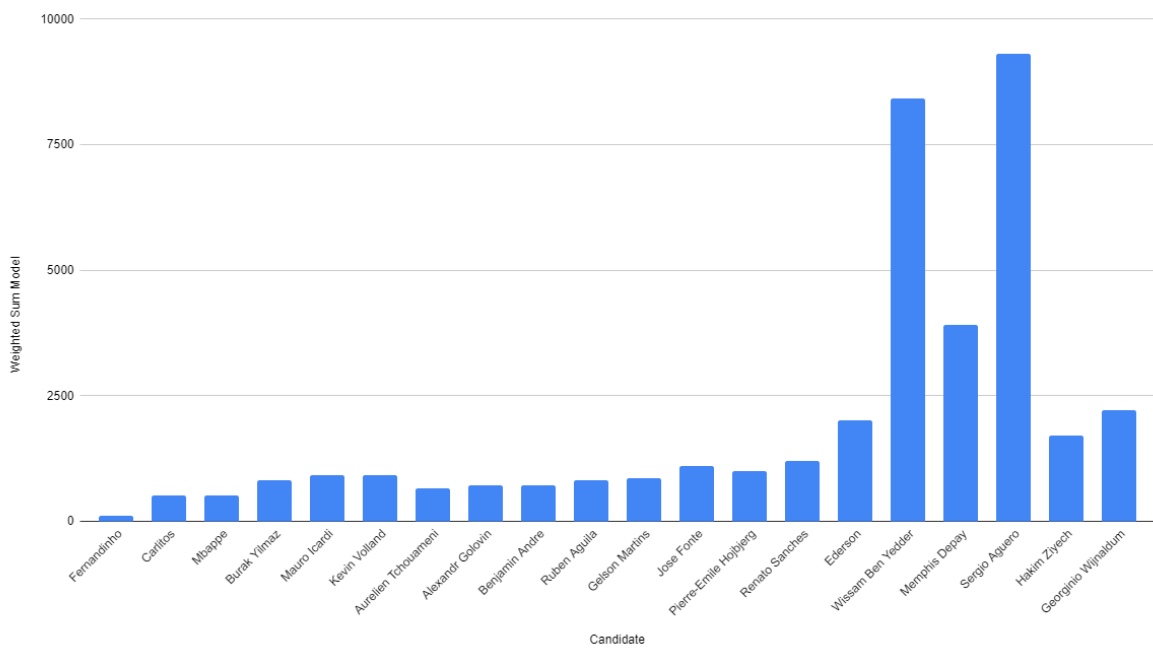
Για την επαλήθευση και την εγκυρότητα του αλγορίθμου πραγματοποιήθηκαν πολλά τεστ. Παρακάτω παραθέτουμε τα αποτελέσματα από δύο συγκεκριμένα πειράματα. Στα δύο πειράματα εφαρμόστηκαν οι αλγόριθμοι weighted product model και weighted sum model. Στο πρώτο παράδειγμα ο χρήστης δηλώνει δέκα παίκτες και ζητάει από το σύστημα να του βρει την ιδανική επιλογή. Στο συγκεκριμένο πείραμα βλέπουμε ότι για μία επιλογή οι δύο αλγόριθμοι έχουν τα ίδια αποτελέσματα κάτι που ήταν αναμενόμενο αφού όσο περισσότερες είναι οι επιλογές τόσο μεγαλύτερη ακρίβεια έχει ο weighted product model έναντι του weighted sum model.

Weighted product model έναντι Candidate

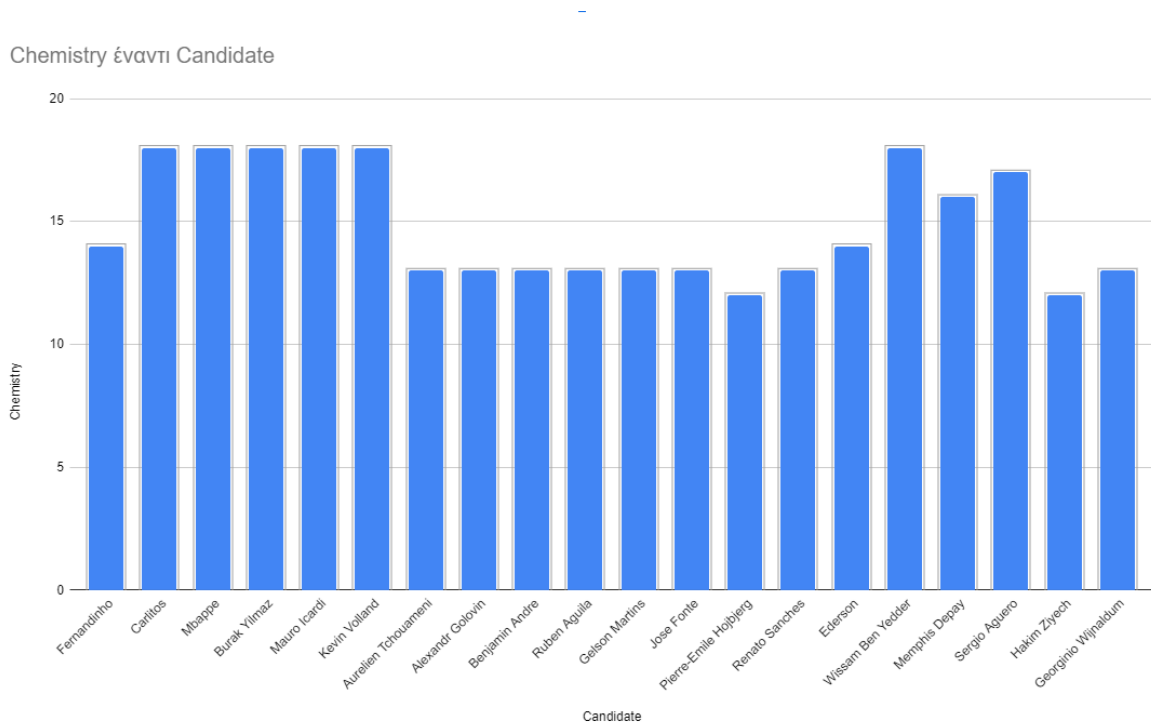


Σχήμα 1.35: Example 1 Weighted Product Model Results

Weighted Sum Model έναντι Candidate

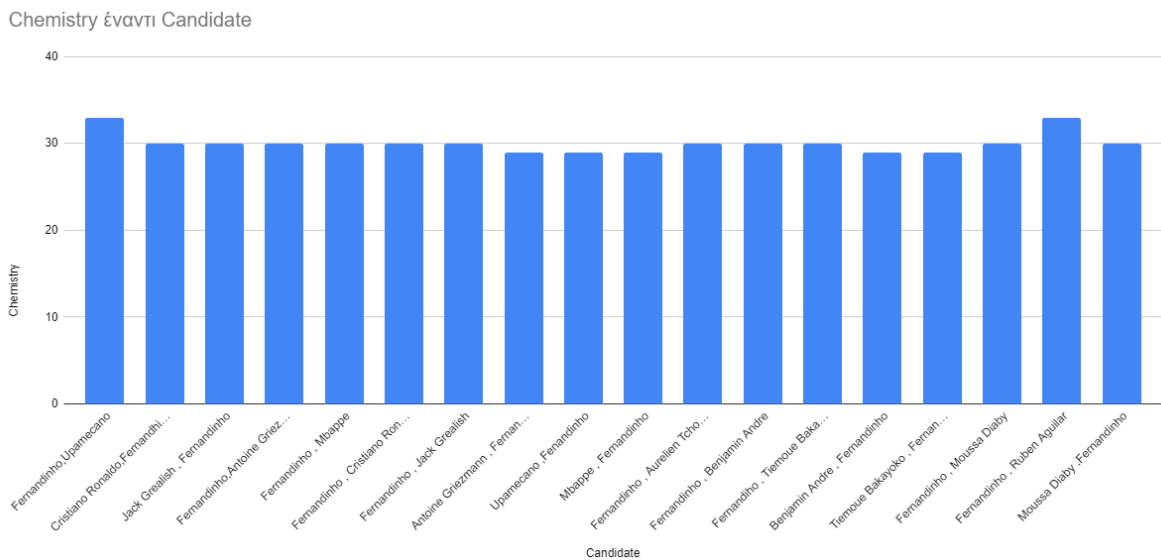


Σχήμα 1.36: Example 1 Weighted Sum Model Results



Σχήμα 1.37: Example 1 Results

Στο δεύτερο παράδειγμα ο χρήστης δηλώνει εννέα παίκτες και το σύστημα του προτείνει ζεύγη από βέλτιστες λύσεις . Τα κορυφαία ζευγάρια είναι παρόμοια και για τους δύο αλγόριθμους ωστόσο παρατηρούμε ότι το ranking διαφέρει και πιο συγκεκριμένα ο weighted product model καταλαβαίνει καλύτερα τα αντικρουόμενα συμφέροντα.Για αρχή παρατηρούμε την συνολική χημεία της ομάδας όταν επιλέγεται το κάθε ζευγάρι:



Σχήμα 1.38: Example 2 Chemistry Results

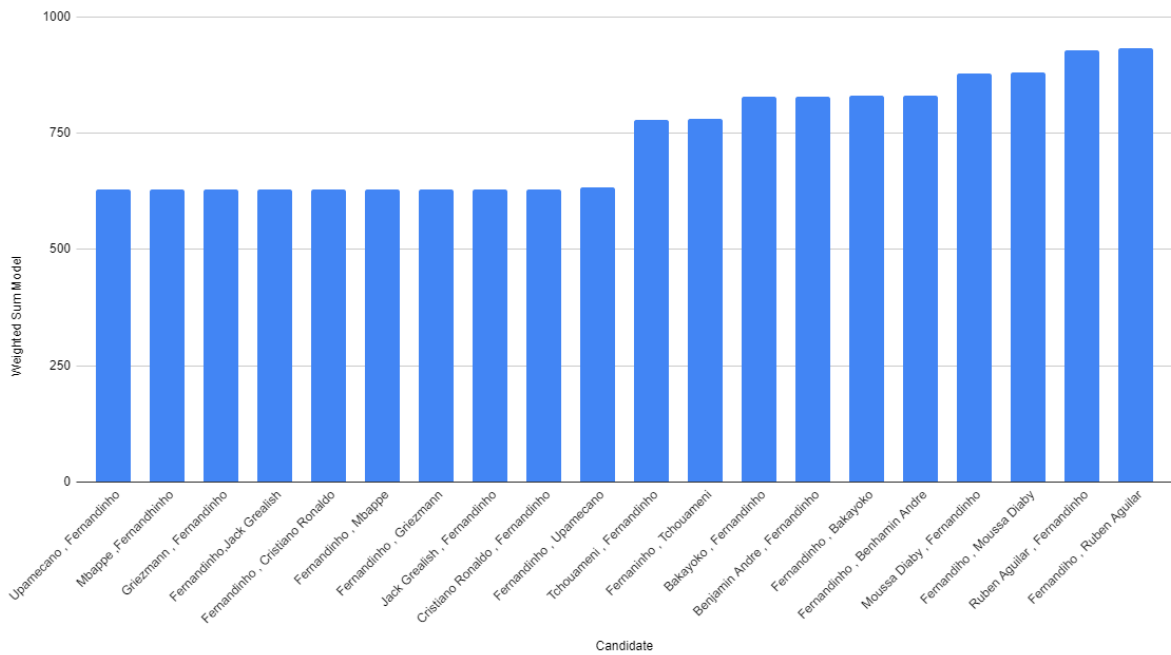
Στην συνέχεια παρατηρώντας τα αποτελέσματα των δύο αλγορίθμων βλέπουμε ότι ο weighted sum model δεν μπορεί να αντιληφθεί τις βέλτιστες περιπτώσεις με αποτέλεσμα το ranking να μην είναι σωστό . Ας δούμε πιο προσεκτικά τους παρακάτω πίνακες.Εύκολα γίνεται αντιληπτό ότι ο WPM παίρνει την σωστή απόφαση και θέτει το ranking των επιλογων όπως το περιμέναμε, δηλαδή στις πρώτες θέσεις βρίσκονται ζευγάρια με χαμηλή τιμή και υψηλό chemistry . Αντίθετα το ranking του WSM επηρεάζεται κυρίως από το price λόγω τη μονάδας μέτρησης του . Η έλλειψη κανονικοποίησης δεν μας δίνει την δυνατότητα να φιλτράρουμε τα δεδομένα οπότε ο WSM δεν κρίνεται αρκετά ικανός για να αποφασίσει για προβλήματα με αντικρουόμενα κριτήρια .Έτσι βλέπουμε ζευγάρια όπως το ( Mbarre , Fernandinho) με συνολική τιμή 600 και χημεία 29 να βρίσκονται στην δεύτερη θέση και πιο ψηλά από ζευγάρια όπως το Fernandinho,Uramecano με συνολική τιμή 600 και χημεία 33,το οποίο αποτελεί την βέλτιστη λύση στο συγκεκριμένο data set.

Candidate	Price	Chemistry	WPM Score	WSM Score	WPM Rank	WSM Rank
Fernandinho,Upamecano	600	33	1.911	629	1	10
Cristiano Ronaldo,Fernandinho	600	30	1.885	630	2	9
Jack Grealish , Fernandinho	600	30	1.885	630	3	8
Fernandinho,Antoine Griezmann	600	30	1.885	629	4	7
Fernandinho , Mbappe	600	30	1.885	630	5	6
Fernandinho , Cristiano Ronaldo	600	30	1.885	630	6	5
Fernandinho , Jack Grealish	600	30	1.885	630	7	4
Antoine Griezmann , Fernandinho	600	29	1.876	630	8	3
Upamecano ,Fernandinho	600	29	1.876	633	9	1
Mbappe , Fernandinho	600	29	1.876	629	10	2
Fernandinho , Aurelien Tchouameni	750	30	1.740	780	11	12
Fernandinho , Benjamin Andre	800	30	1.730	830	12	16
Fernandinho , Tiemoue Bakayoko	800	30	1.730	830	13	15
Benjamin Andre , Fernandinho	800	29	1.694	829	14	14
Tiemoue Bakayoko , Fernandinho	800	29	1.694	829	15	13
Fernandinho , Moussa Diaby	850	30	1.669	880	16	18
Fernandinho , Ruben Aguilar	900	33	1.669	933	17	20
Moussa Diaby ,Fernandinho	850	30	1.660	879	18	17

Σχήμα 1.39: Example 2 Overview

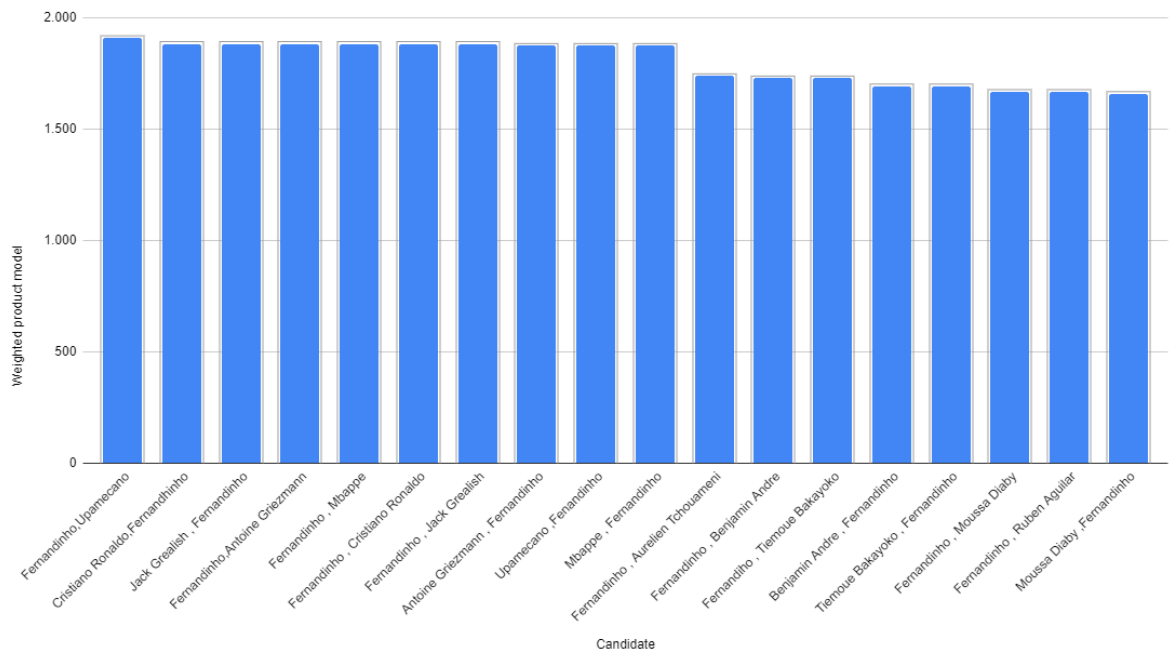


Weighted Sum Model έναντι Candidate

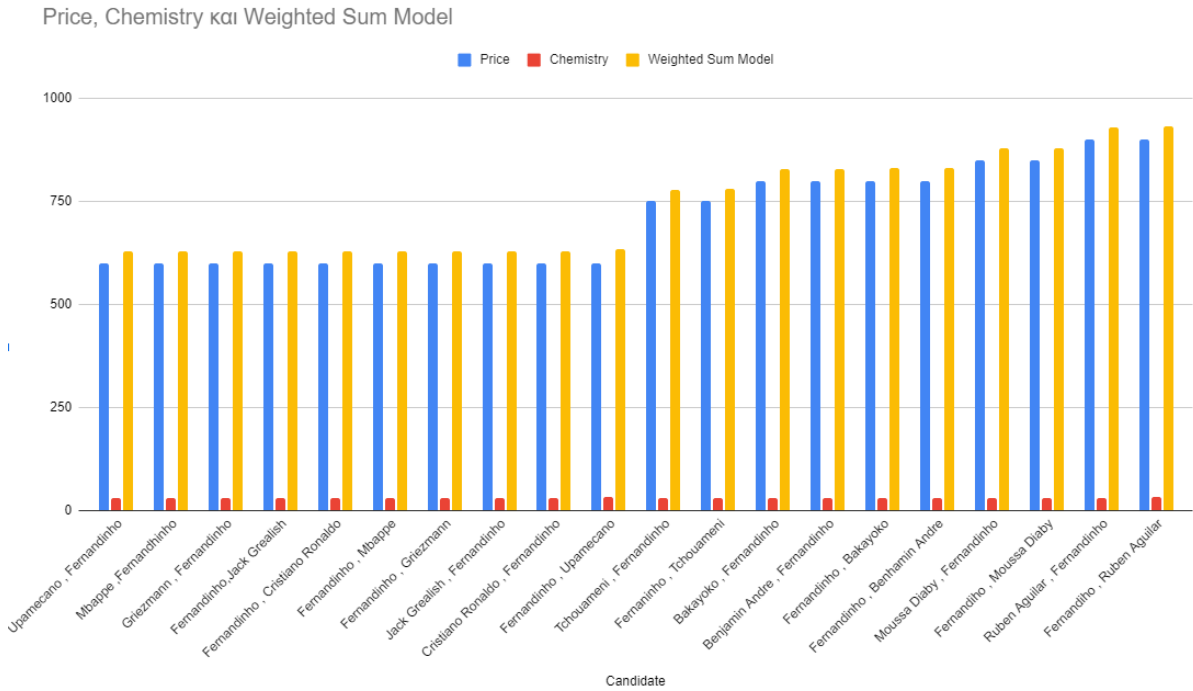


Σχήμα 1.40: Example 2 Weighted Sum Model Results

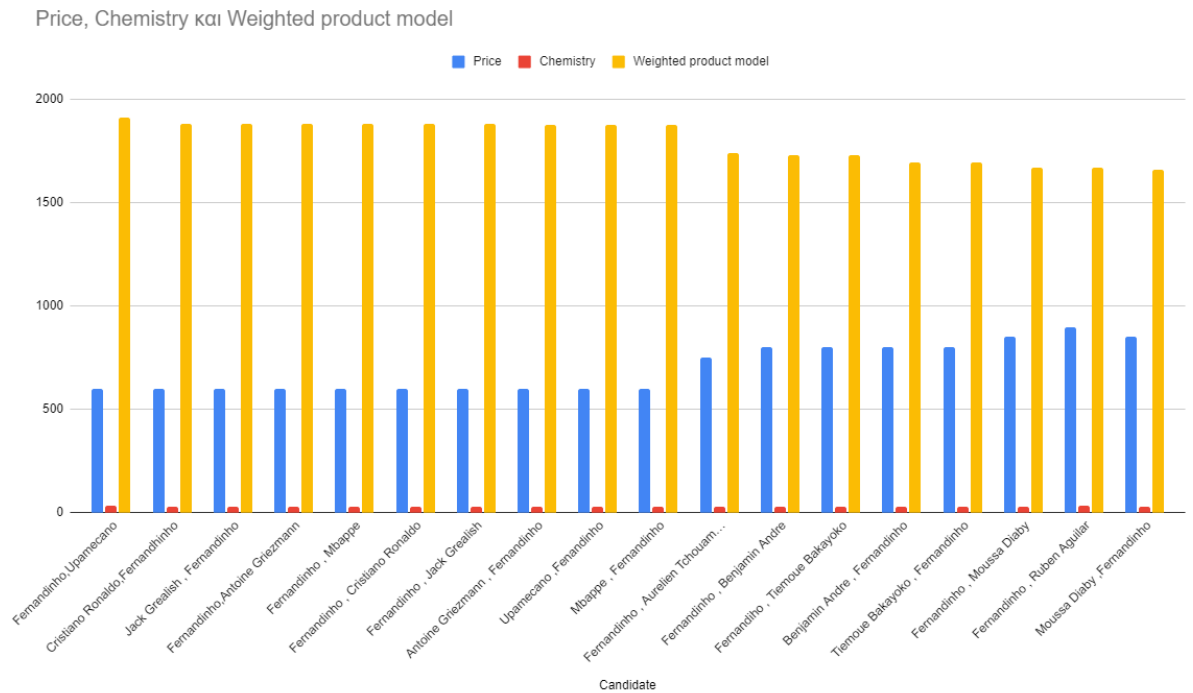
Weighted product model έναντι Candidate



Σχήμα 1.41: Example 2 Weighted Product Model Results



Σχήμα 1.42: Example 2 Weighted Sum Model, Price, Chemistry Results

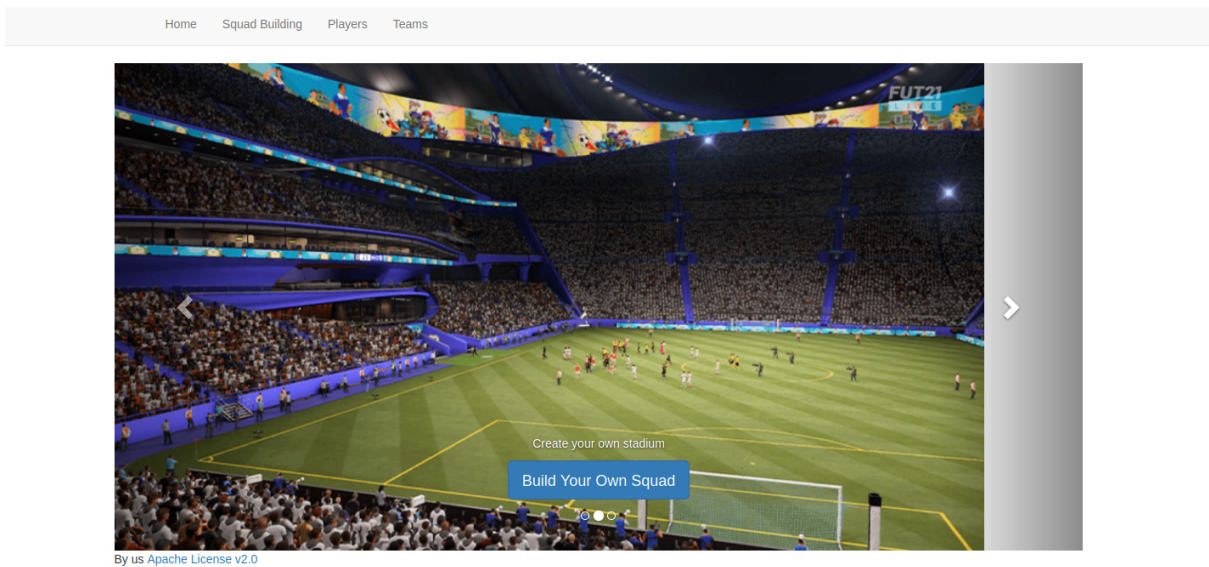


Σχήμα 1.43: Example 2 Weighted Product Model, Price, Chemistry Results

## ΚΕΦΑΛΑΙΟ 4 Η ΕΦΑΡΜΟΓΗ

Η αρχική οθόνη της ιστοσελίδας περιέχει ένα carousel slider menu μέσω του οποίου ο χρήστης μπορεί να περιηγηθεί στις επιμέρους σελίδες.

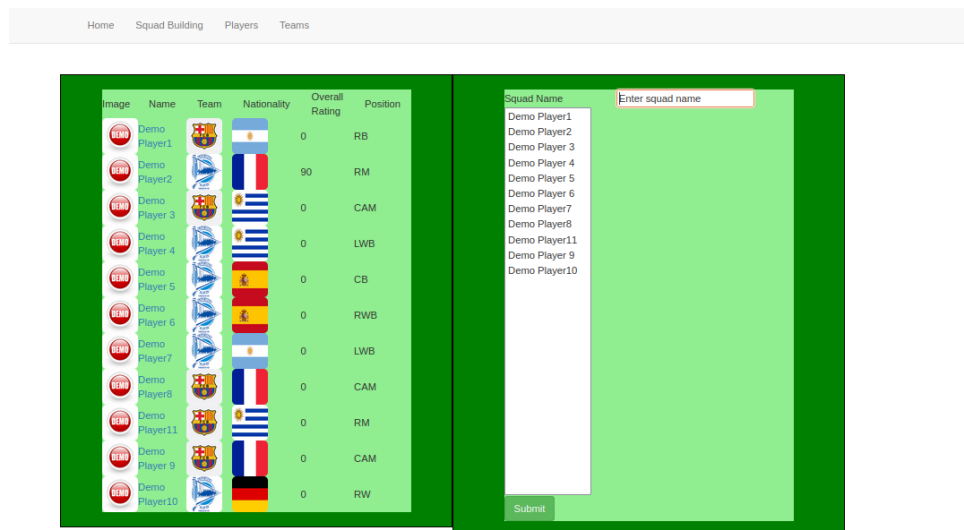
- Squad building
- Players
- Teams



Σχήμα 1.44: App Home Page

## 4.1 Squad building

























Δίνει την δυνατότητα στον χρήστη να δημιουργήσει την δική του ομάδα διαλέγοντας του παίκτες που επιθυμεί και δίνοντας στην ομάδα το όνομα της αρεσκείας του. Στο δεξί κουτί παρουσιάζονται όλοι οι διαθέσιμοι παίκτες ενώ στο δεξί κουτί ο χρήστης επιλέγει τους παίκτες και συμπληρώνει το όνομα της νέας ομάδας






Σχήμα 1.45: App Squad Building Page

## 4.2 Players

Όλοι οι διαθέσιμοι παίκτες που υπάρχουν στην βάση δεδομένων . Επιλέγοντας έναν παίκτη προχωράμε στην ατομική σελίδα του κάθε παίκτη που περιέχει παραπάνω λεπτομέρειες.

Image	Name	Team	Nationality	Overall Rating	Position
	Demo Player1			0	RB
	Demo Player2			90	RM
	Demo Player 3			0	CAM
	Demo Player 4			0	LWB
	Demo Player 5			0	CB
	Demo Player 6			0	RWB
	Demo Player7			0	LWB
	Demo Player8			0	CAM

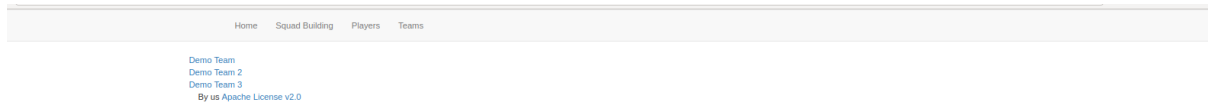
Σχήμα 1.46: App All Players Page

Home	Squad Building	Players	Teams
Name	Demo Player7		
Team			
Nationality			
Overall Rating	0		
Position	LWB		
By us Apache License v2.0			

Σχήμα 1.47: App Player Personal Page

### 4.3 Teams

Κάθε νέα ομάδα που δημιουργείται προστίθεται στην βάση στον πίνακα Teams . Δίνεται η δυνατότητα στον χρήστη αρχικά να δει όλες τις ομάδες που έχουν δημιουργηθεί και στη συνέχεια να εξετάσει λεπτομερώς όποια ομάδα επιθυμεί. Οι πληροφορίες που παρέχονται είναι chemistry, overall και οι παίκτες της ομάδας.



Σχήμα 1.48: Teams view



Σχήμα 1.49: Team view

## ΚΕΦΑΛΑΙΟ 5 ΕΠΙΛΟΓΟΣ

Η εργασία επικεντρώθηκε στην μελέτη των αλγορίθμων λήψης αποφάσεων με πολλαπλά κριτήρια καθώς και στην δημιουργία του αλγορίθμου εύρεσης των υποψηφίων ο οποίος φιλτράρει τα δεδομένα και τα μορφοποιεί κατάλληλα ώστε να μπορεί να εφαρμοστεί ο αλγόριθμος Weighted Product Model που επιλέχθηκε. Όπως είναι λογικό υπάρχουν πτυχές που υπάρχει αρκετός χώρος για βελτίωση. Κάποιες από τις πτυχές τις αναλύω στις παρακάτω παραγράφους. Οι τιμές των παικτών μεταβάλλονται δυναμικά με το πέρασμα του χρόνου . Ιδανικά θα έπρεπε να υπάρχει ένας server ο οποίος θα λειτουργεί σαν crawler και θα κοιτάει συνεχώς τις μεταβολές των τιμών όλων των παικτών και θα ανανεώνει τις εγγραφές στην βάση δεδομένων . Η υλοποίηση δεν έχει πραγματοποιηθεί στα πλαίσια αυτής της εργασίας αλλά θα ήταν απαραίτητη σε ενδεχόμενο που το λογισμικό περνούσε σε φάση παραγωγής. Αξίζει να σημειωθεί ότι υπάρχουν API που σου δίνουν όλες αυτές τις δυνατότητες αλλά το κόστος ενσωμάτωσης τους είναι απαγορευτικό ,τουλάχιστον για τους σκοπούς της εργασίας.Επίσης θα μπορούσε να επανασχεδιαστεί ο τρόπος εύρεσης των πιθανών υποψηφίων . Ο παρών τρόπος είναι κυρίως brute force κάτι που ενδεχομένως να προκαλέσει πρόβλημα κλιμάκωσης σε ενδεχόμενο που το λογισμικό περνούσε σε φάση παραγωγής, καθώς το σύστημα θα απαιτούσε μεγάλη υπολογιστική ισχύ για να εξυπηρετήσει αιτήματα πολλών χρηστών. Θα μπορούσε να εφαρμοστεί κάποιο μοντέλο γραφημάτων και με την βοήθεια της τεχνητής νοημοσύνης το σύστημα να είναι σε θέση να προβλέπει από μόνο του τις λύσεις ή για αρχή να μπορούσε να περιορίσει το dataset στο οποίο θα εφαρμοστεί ο υπάρχων τρόπος εύρεσης των υποψηφίων. Τέλος θα μπορούσε ιδανικά να γίνει ένας επανασχεδιασμός στο UI/UX μέρος του λογισμικού και να γίνει ενσωμάτωση κάποιου γνωστού framework όπως η React .Η δημιουργία mobile εφαρμογής θα μπορούσε να υλοποιηθεί στο πλαίσιο αυτού του επανασχεδιασμού σε σχετικά μικρό χρόνο υλοποίησης εφόσον το API από την πλευρά του server υπάρχει ήδη υλοποιημένο.

## BIBΛΙΟΓΡΑΦΙΑ

- [1] Django Tutorial.
- [2] Official Django Documentation.
- [3] Official Fifa Ultimate Team Website.
- [4] Chemistry Calculation Algorithm , fifauteam.com.
- [5] Team Rating Calculation Algorithm ,fifauteam.com.
- [6] Weighted Sum Model , geeksforgeeks.org
- [7] Weighted Sum Model , wikipedia.org.
- [8] Topsis Algorithm, wikipedia.org.
- [9] Vikor Algorithm , wikipedia.org
- [10] Multi-criteria Decision Analysis for Supporting the Selection of Engineering Materials in Product Design (Second Edition).
- [11] Comparison of Multi Criteria Decision Making Algorithms for Ranking Cloud J.Ruby Annette1 , Aisha Banu and P.Subash Chandran.
- [12] Algorithms and Models for Dynamic Multiple Criteria Decision Making Debora Di Caprio , Fransisco Javier Santos Arteaga.
- [13] Django Rest Framework , django-rest-framework.org.
- [14] Chen , C. T. (2000). Extensions of the Topsis for group decision -making under a fuzzy environment. Fuzzy sets and systems, 114(1), 1-9.
- [15] Chen , S. M. , & Lee , L.W.(2010). Fuzzy multiple attributes group decision making based on the interval type-2 TOPSIS method. Expert systems with applications, 37(4),2790-2798.
- [16] Chen, T. Y. , & Tsao, C. Y. (2008). The interval-valued fuzzy TOPSIS method and experimental analysis . Fuzzy sets and systems , 159(11) , 1410-1428.
- [17] Weighted Product Model, geeksforgeeks.org.
- [18] Weighted Product Model , wikipedia.org.