A blue and yellow logo

Description automatically generated with low confidence

TheaterPlanner: Χρονοπρογραμματισμός Θέασης Θεατρικών Παραστάσεων.

Τσουτσούκας Αντώνιος   
dai19187

26 Ιουνίου 2023

Πανεπιστήμιο Μακεδονίας

Τμήμα εφαρμοσμένης πληροφορικής

Επιβλέπων καθηγητής: Ηλίας Σακελλαρίου

**ΠΕΡΙΛΗΨΗ**

Αυτή η περίληψη αποτελεί κομμάτι πτυχιακής εργασίας σχετικά με ένα σύστημα σχεδιασμένο για την επίλυση προβλημάτων γραμμικού προγραμματισμού, εστιάζοντας ιδιαίτερα στη βελτιστοποίηση της κατανομής χρονολογικών περιορισμών και των διαδικασιών λήψης αποφάσεων. Το σύστημα στοχεύει να αντιμετωπίσει τις προκλήσεις του πραγματικού κόσμου διατυπώνοντας προβλήματα γραμμικού προγραμματισμού και χρησιμοποιώντας τεχνικές βελτιστοποίησης, μεγιστοποιώντας την αντικειμενική συνάρτηση ενός μαθηματικού μοντέλου με στόχο την παρουσίαση ενός βέλτιστου αποτελέσματος. Παρακάτω παρουσιάζεται πως αντλείται ένα σύνολο θεατρικών παραστάσεων μέσω ιστότοπου και στην συνέχεια προσφέρεται στον χρήστη δίνοντας του την ικανότητα να κάνει τις επιλογές του σχετικά με την παρακολούθηση. Επιπλέον, μία φιλική προς το χρήστη διεπαφή προς τους χρήστες προσφέρεται δίνοντας την ικανότητες όπως η δημιουργίας λογαριασμού. Ακόμη, αναφερόμαστε για τον τρόπο ανάπτυξης της εφαρμογής αξιοποιώντας διάφορες τεχνολογίες και τεχνικές όπως για παράδειγμα η αξιοποίηση πλαίσιού κώδικα ή αρχιτεκτονικής πελάτη-διακομιστή. Τελικά, με λίγα λόγια, στην εργασία γίνεται μία προσπάθεια κατασκευής μίας διαδικτυακής εφαρμογής με δυνατότητα χρονοπρογραμματισμού θεατρικών παραστάσεων και κατασκευής βέλτιστου προγράμματος.

**Λέξεις κλειδιά**: προβλήματα γραμμικού προγραμματισμού, βελτιστοποίηση, κατανομή χρονολογιών περιορισμών, λήψη αποφάσεων, τεχνικές βελτιστοποίησης, αντικειμενική συνάρτηση, βέλτιστου αποτελέσματος, θεατρικές παραστάσεις, ιστότοπος, διεπαφή χρήστη, πλαίσιο κώδικα, αρχιτεκτονική πελάτη-διακομιστή, διαδικτυακή εφαρμογή.

**ABSTRACT**  
 This abstract is part of a thesis on a system designed to solve linear programming problems, focusing on the optimization of timing constraint distributions and decision processes. The system aims to address real-world challenges by formulating linear programming problems and using optimization techniques, maximizing the objective function of a mathematical model with the goal of presenting an optimal result. Below is how a set of plays is pulled through a website and then offered to the user giving them the ability to make their own choices about tracking. In addition, a user-friendly interface is offered to users by giving capabilities such as account creation. Furthermore, we refer to the way of developing the application by leveraging various technologies and techniques such as, for example, the use of framework code or client-server architecture. Finally, in a few words, in the work an attempt is made to build an online application with the possibility of scheduling theater performances and building an optimal program.

**Keywords**: linear programming problems, optimization, constraint time distribution, decision making, optimization techniques, objective function, optimal outcome, plays, website, user interface, code framework, client-server architecture, web application.

Πίνακας Περιεχομένων

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή 6

2. ΠΑΡΟΜΟΙΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ 8

2.1 ΣΥΣΤΗΜΑΤΑ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ 8

2.1.2 ΜΕΘΟΔΟΙ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ 9

2.1.3 Παράδειγμα παρόμοιου προβλήματος: «ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΟΛΙΤΙΣΤΙΚΩΝ ΓΕΓΟΝΟΤΩΝ» - MYvisitplanner GR - Εξατομικευμένο Σύστημα Σχεδιασμού Διαδρομών στον κλάδο του Τουρισμού 10

2.2 ΑΞΙΟΠΟΙΗΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΤΟ ΣΥΣΤΗΜΑ 11

2.2.1 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ 12

2.2.1.2 PYTHON 12

2.2.1.3 JavaScript 13

2.2.1.4 HTML 13

2.2.1.5 CSS 13

2.2.2 Πλαίσια – Web Frameworks 13

2.2.3 FLASK 14

2.2.4 ΠΑΡΟΜΟΙΑ FRAMEWORKS ΜΕ ΤΟ FLASK: 15

2.2.5 JINJA (TEMPLATE ENGINE) 15

2.2.6 ΕΠΙΠΡΌΣΘΕΤΕΣ ΕΠΕΚΤΑΣΕΙΣ ΓΙΑ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ 16

2.2.6.1 Επέκταση: 16

2.2.7 ΑΝΤΛΗΣΗ ΔΕΔΟΜΕΝΩΝ ΙΣΤΟΥ - WEB SCRAPE 17

2.2.8 Χρήση Curl 17

2.2.9 Επέκταση «beautifulsoup4» 17

2.2.10 Εργαλεία για χρονοπρογραμματισμό 18

2.2.10.1 Google OR-Tools 18

2.2.11 Εργαλεία παρόμοια με το google Or-Tools για χρονοπρογραμματισμό 18

3. ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΉΜΑΤΟΣ ΚΑΙ ΘΕΩΡΗΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ 19

3.1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΧΡΗΣΤΗ 19

3.1.1 Γενική μορφή απαιτήσεων 19

3.1.2 Απόδοση και επεξήγηση στον χρήστη 19

3.1.3 Δυνατότητες αλληλεπίδρασης με το σύστημα 20

3.2 ΑΝΤΛΗΣΗ ΤΩΝ ΑΠΑΡΑΙΤΗΤΩΝ ΠΛΗΡΟΦΟΡΙΩΝ 20

3.2.1 ΛΟΓΟΙ ΑΠΑΙΤΗΣΗΣ 20

3.2.2 ΑΞΙΟΠΟΙΗΣΗ ΑΠΑΙΤΗΣΗΣ ΣΤΟ ΣΥΣΤΗΜΑ 21

3.3 ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ 21

4. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΛΥΣΗΣ 23

4.1 Απαραίτητες βιβλιοθήκες ανάπτυξης – Απαιτήσεις συστήματος 23

4.2 Σχεδίαση διεπαφής χρήστη 24

4.2.1 ΑΚΟΛΟΥΘΙΑ ΟΘΟΝΩΝ ΔΙΕΠΑΦΗΣ 25

4.2.1.2 Welcome screen 25

4.2.1.3 Sign up 25

4.2.1.4 Login 26

4.2.1.5 Home 26

4.2.1.6 Η οθόνη του αποτελέσματος 27

4.3 ΣΧΕΔΙΑΣΗ ΚΩΔΙΚΑ ΠΕΛΑΤΗ – index.js 27

4.3.1 SELECT button 27

4.3.2 Submit button 28

4.4 Ανάλυση αρχιτεκτονικής συστήματος 29

4.4.1 Αρχιτεκτονική πελάτη-διακομιστή 29

4.4.2 ΓΕΝΙΚΗ ΔΟΜΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ 30

4.5 Σχεδίαση Βάσης Δεδομένων 31

4.5.1 Παρουσίαση των πινάκων της βάσης 31

4.5.2 Δομή αρχείων 32

4.5.3 φάκελος /website: 33

4.6 ΥΛΟΠΟΊΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΛΥΣΗ 33

4.6.2 ΜΑΘΗΜΑΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ 33

4.6.3 ΜΕΤΑΒΛΗΤΕΣ 34

4.6.4 Αντικειμενική συνάρτηση 34

4.6.5 Περιορισμοί 34

4.6.6 Προτεραιότητα παραστάσεων 34

4.6.7 Εν κατακλείδι 35

4.7 Λήψη και διαχείριση επιλεγμένων παραστάσεων 35

4.8 Επίλυση μέσω κώδικα 35

4.9 Τελική απάντηση 37

4.10 ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΠΡΟΕΚΥΨΑΝ ΚΑΤΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ 38

5. ΔΟΚΙΜΕΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ 41

5.1 ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΕ ΔΙΑΦΟΡΕΤΙΚΕΣ ΤΙΜΕΣ ΤΩΝ ΔΕΔΟΜΕΝΩΝ 41

5.1.2.2 Αναμενόμενο αποτέλεσμα 42

5.2 ΣΧΕΤΙΚΑ ΜΕ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ 45

6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΟΠΤΙΚΕΣ 47

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

[Εικόνα 1 “ Αρχείο απαιτήσεων” 24](#_Toc138678601)

[Εικόνα 2 “Welcome screen 25](#_Toc138678602)

[Εικόνα 3 "Sign up" 25](#_Toc138678603)

[Εικόνα 4 "Login" 26](#_Toc138678604)

[Εικόνα 5 "Home screen" 26](#_Toc138678605)

[Εικόνα 6 "The result" 27](#_Toc138678606)

[Εικόνα 7 “Console log map” 28](#_Toc138678607)

[Εικόνα 8 “SendMapToServer” 29](#_Toc138678608)

[Εικόνα 9 "Αρχιτεκτονική συστήματος" 30](#_Toc138678609)

[Εικόνα 10 "Οι πίνακες της βάσης δεδομένων" 32](#_Toc138678610)

Test No.1 41

Πίνακας 1 - Επιλεγμένα 41

Πίνακας 2 - Αποτέλεσμα 42

Test No.2 42

Πίνακας 3 - Επιλεγμένα 42

Πίνακας 4 - Αποτέλεσμα 43

Test No.3 43

Πίνακας 5 - Επιλεγμένα 43

Πίνακας 6 - Αποτέλεσμα 44

1. Εισαγωγή

Στον σημερινό κόσμο με τους γρήγορους ρυθμούς ζωής, η παρακολούθηση πολιτιστικών εκδηλώσεων όπως συναυλίες, θεατρικές παραστάσεις, εκθέσεις και φεστιβάλ έχει γίνει αναπόσπαστο μέρος της ζωής των ανθρώπων. Ωστόσο, με την αφθονία των επιλογών και τον περιορισμένο διαθέσιμο χρόνο, τα άτομα αντιμετωπίζουν συχνά το δύσκολο έργο της επιλογής και της οργάνωσης ενός εξατομικευμένου προγράμματος που μεγιστοποιεί την εμπειρία του ενδιαφέροντος τους. Αυτή η πρόκληση απαιτεί μια έξυπνη και αποτελεσματική λύση που μπορεί να βοηθήσει τους χρήστες να λαμβάνουν τεκμηριωμένες αποφάσεις σχετικά με ποιες παραστάσεις θα παρακολουθήσουν και πώς να βελτιστοποιήσουν το πρόγραμμά τους. Η πτυχιακή εργασία, αυτή, στοχεύει να αντιμετωπίσει αυτό το πρόβλημα αναπτύσσοντας μια διαδικτυακή εφαρμογή που χρησιμοποιεί τεχνικές βελτιστοποίησης για να δημιουργήσει ένα βέλτιστο πρόγραμμα για τους χρήστες που επιθυμούν να παρακολουθήσουν πολιτιστικές εκδηλώσεις. Η εφαρμογή επιτρέπει στους χρήστες να αναζητήσουν έναν ολοκληρωμένο κατάλογο παραστάσεων και τους παρέχει την ευελιξία να επιλέξουν όσες επιθυμούν. Αξιοποιώντας τη δύναμη της μαθηματικής βελτιστοποίησης, το σύστημα, με τις προτιμήσεις του χρήστη και τις προτεραιότητες που μπορεί και θέτει μπορεί να προτείνει ένα βέλτιστο πρόγραμμα σύμφωνα με τις μεμονωμένες προτιμήσεις του.

Το πρόγραμμα υιοθετεί μια προσέγγιση γραμμικού προγραμματισμού για την αντιμετώπιση του προβλήματος εμφάνισης ενός προγράμματος παραστάσεων. Σε κάθε εκδήλωση εκχωρείται βαθμολογία προτεραιότητας με βάση τη συμβολή του χρήστη. Στη συνέχεια, ο αλγόριθμος βελτιστοποίησης επιδιώκει να μεγιστοποιήσει τη συνολική βαθμολογία προτεραιότητας των συμμετεχόντων συμβάντων, τηρώντας ταυτόχρονα διάφορους περιορισμούς, συμπεριλαμβανομένης της διαθεσιμότητας χρόνου και την αποφυγή συγκρούσεων χρονοδιαγράμματος. Τα βασικά χαρακτηριστικά της εφαρμογής περιλαμβάνουν μια διαισθητική διεπαφή χρήστη που επιτρέπει στους χρήστες να περιηγούνται σε συμβάντα, να κοιτάνε ημερομηνίες και τοποθεσίες και να επισημαίνουν τις προτιμήσεις τους. Ο κώδικας της μηχανής βελτιστοποίησης λειτουργεί στα παρασκήνια για να δημιουργήσει ένα βέλτιστο πρόγραμμα που επιτυγχάνει μια ισορροπία μεταξύ των προτιμήσεων του χρήστη και των κατάλληλων προτεραιοτήτων.

Η συμβολή αυτής της εργασίας έγκειται στον σχεδιασμό και την ανάπτυξη μιας διαδικτυακής εφαρμογής που συνδυάζει τη δύναμη των τεχνικών βελτιστοποίησης με την επιλογή παραστάσεων με επίκεντρο τον χρήστη. Αξιοποιώντας τη μαθηματική βελτιστοποίηση, οι χρήστες μπορούν τώρα να επωφεληθούν από εξατομικευμένα προγράμματα εκδηλώσεων που βελτιστοποιούν την εμπειρία. Επιπλέον, αυτή η έρευνα παρέχει πληροφορίες για την πρακτική εφαρμογή των τεχνικών βελτιστοποίησης στον τομέα του σχεδιασμού εφαρμογών ιστού με εκδηλώσεις και ρίχνει φως στις δυνατότητες περαιτέρω βελτιώσεων, προσαρμογής για μελλοντικές ασχολίες.

Τα επόμενα κεφάλαια εμβαθύνουν στις λεπτομέρειες του προτεινόμενου μοντέλου βελτιστοποίησης, την υλοποίηση της διαδικτυακής εφαρμογής και την αξιολόγηση της απόδοσής της μέσω πραγματικών περιπτωσιολογικών μελετών. Στο κεφάλαιο 2 παρουσιάζονται κάποιες παρόμοιες τεχνικές με αυτές που χρησιμοποιήθηκαν στο πρόγραμμα καθώς και παρόμοιας λογικής συστήματα. Επιπλέον, αναλύονται οι βασικές τεχνολογίες που αξιοποιούνται με σκοπό την κάλυψη των αναγκών της εφαρμογής. Αναφέρονται, έννοιες κατάλληλες για την κατανόηση του συστήματος.

Στο τρίτο κεφάλαιο γίνεται η περιγραφή των απαιτήσεων του χρήστη και τις διαθέσιμες ικανότητες που του δίνονται. Επιπλέον γίνεται αναφορά για το πως η ιστοσελίδα μπορεί και αντλεί τα δεδομένα που απαιτούνται. Τέλος, γίνεται μία θεωρητική παρουσίαση του προβλήματος στην προσπάθεια κατανόησης της λύσης και των παραγόμενων αποτελεσμάτων.

Στο τέταρτο κεφάλαιο αναφερόμαστε στον σχεδιασμό του συστήματος και την υλοποίηση του. Γίνεται παρουσίαση του σχεδιασμού της εμφάνισης της διεπαφής χρήστη, η σχεδίαση του κώδικα του πελάτη, η αρχιτεκτονική της εφαρμογής, ο σχεδιασμός της βάσης των δεδομένων. Επιπρόσθετα βλέπουμε στο πρακτικό κομμάτι την λογική της ανάπτυξης του κώδικα, της δομής των αρχείων και την επίτευξη της λύσης. Αποτελεί το βασικότερο κεφάλαιο της εργασίας εμφανίζοντας όλα τα απαραίτητα κομμάτια της του συστήματος. Τέλος υπάρχουν κατάλληλα code snippet βοηθώντας στην περαιτέρω επεξήγηση του κώδικα.

Στο πέμπτο κεφάλαιο, γίνονται αναφορές σε παραδείγματα χρήσης που προέκυψαν από το σύστημα. Εμφανίζονται κάποια αποτελέσματα από τις δοκιμές που έγιναν.

Στο έκτο και τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα της εργασίας και γενικά της κατασκευής μιας εφαρμογής ιστού. Τέλος εμφανίζονται μελλοντικές πιθανές προσθήκες και πώς το σύστημα μπορεί να επεκταθεί.

1. ΠΑΡΟΜΟΙΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ

ΤΕΧΝΟΛΟΓΙΕΣ

# ΣΥΣΤΗΜΑΤΑ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η επίλυση των προβλημάτων χρονοπρογραμματισμού αποτελεί θεματική που μπορεί να απασχολήσει μεγάλο εύρος θεμάτων. Η ανάγκη οργάνωσης του χρόνου και η διαχείριση των διαθέσιμων δυνατών πόρων καθιστούν την εύρεση του καλύτερου δυνατού αποτελέσματος απόλυτα απαραίτητη. Έτσι λογικό είναι ότι υπάρχουν παρόμοια συστήματα με την εύρεση λύσεων όσον αφορά την παρουσίαση ενός βέλτιστου προγράμματος. Στην συγκεκριμένη εργασία, για παράδειγμα, γίνεται η προσπάθεια επίλυσης του προβλήματος της αξιοποίησης του χρόνου του χρήστη έχοντας απώτερο σκοπό την παρακολούθηση όσο το δυνατό περισσότερων θεατρικών παραστάσεων. Ωστόσο, υπάρχουν συστήματα που εφαρμόζονται σε θέματα παρόμοιου είδους. Μπορεί να μην είναι φαινομενικά ίδιες αλλά στην ανάλυση τους πολλές είναι οι ομοιότητες τους σχετικά με την εφαρμογή του μοντέλου του γραμμικού προγραμματισμού. Ο τρόπος που χρησιμοποιείται η λογική της επίλυσης τον περιορισμών είναι αρκετά κοντά με το παρόν πρόγραμμα.

Συστήματα επίλυσης προβλημάτων χρονοπρογραμματισμού έχουν εφαρμοστεί κατά καιρούς σε εφαρμογές πραγματικού κόσμου, που αφορούν πολιτιστικά γεγονότα. Αναπτυγμένα είτε από μικρούς οργανισμούς, ή μεγάλες επιχειρήσεις έχουν εισαχθεί λογισμικά λύσεων που χρησιμοποιούν διάφορες τεχνικές βελτιστοποίησης για να βρουν την καλύτερη λύση. Για παράδειγμα, μπορούν να χρησιμοποιούν ακέραιο προγραμματισμό για να βελτιστοποιήσουν τον προγραμματισμό συμβάντων και πόρων, όπως χώρους, προσωπικό και εξοπλισμό.. Συνολικά, αυτές οι λύσεις στοχεύουν να παρέχουν στους προγραμματιστές εκδηλώσεων εργαλεία για τη βελτίωση της αποτελεσματικότητας, την αύξηση των εσόδων και τη βελτίωση της εμπειρίας των συμμετεχόντων. Συγκεκριμένα, ένα παράδειγμα που λύνει η ίδια η Google για την επεξήγηση του εργαλείου της OR-Tools αποτελεί το πρόβλημα των εργαζομένων. Αφού η εργασία βασίστηκε σε αυτό το εργαλείο καλό είναι να αναλυθεί το αντίστοιχο παράδειγμα της εταιρείας. Για παράδειγμα, οργανισμοί όπου οι εργαζόμενοι δουλεύουν σε πολλές βάρδιες , διαφορετικές ημέρες έχουν την ανάγκη ώστε να δημιουργείται ένα πρόγραμμα κατάλληλο με αυτοματοποιημένο τρόπο , χωρίς να υπερβαίνει ανθρώπινος παράγοντας. Θεωρητικά τα προγράμματα κάποιοι από τους περιορισμούς αποτελούνται από έννοιες όπως :

· Οκτάωρη εργασία

· Οι εργαζόμενοι δεν μπορούν να δουλέψουν πάνω από μία βάρδια στη σειρά.

· Οφείλουν να διαιρούν τις συνολικές ώρες μέσα στην εβδομάδα.

Κάποια από αυτά είναι παραδείγματα περιορισμών για μία τέτοια περίπτωση προβλήματος χρονοπρογραμματισμού. Πιο συγκεκριμένα ένα πραγματικό σενάριο είναι οι νοσοκόμοι ενός νοσοκομείου και οι βάρδιες που θα κάνουν καθημερινά. Οπότε το σύστημα με τους παραπάνω περιορισμούς οφείλει να αποτυπώσει ένα αποτέλεσμα που ακολουθεί τα παρακάτω κριτήρια:

1. Τήρηση των παραπάνω περιορισμών.

2. Μεγιστοποίηση και πλήρης αξιοποίηση των πόρων (στο παράδειγμα νοσοκόμων).

3. Εύρεση βέλτιστου προγράμματος. [1]

## ΜΕΘΟΔΟΙ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Όταν αναφερόμαστε στην επίλυση των προβλημάτων αναφερόμαστε κυρίως στον τρόπο με τον οποίο λύνουμε το πρόβλημα αλλά και την χρήση των μαθηματικών μοντέλων αλγορίθμων ώστε από το πλήθος των δεδομένων να καταλήγουμε σε μία αποδεκτή λύση. Συγκεκριμένα στην εργασία χρησιμοποιούμε την τεχνική του **Mixed Integer Programming** (MIP) [2]. Ένας λύτης **Μικτού Ακέραιου Προγραμματισμού** (MIP) είναι ένα ισχυρό εργαλείο μαθηματικής βελτιστοποίησης που χρησιμοποιείται για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης όπου ορισμένες μεταβλητές περιορίζονται σε ακέραιες τιμές. Στο πλαίσιο των προβλημάτων προγραμματισμού, οι λύτες MIP μπορούν να χειριστούν διακριτές μεταβλητές απόφασης που αντιπροσωπεύουν δραστηριότητες, πόρους και γενικά δεδομένα. Διατυπώνοντας το πρόβλημα χρονοπρογραμματισμού ως μοντέλο μαθηματικής βελτιστοποίησης, αναζητούν μια βέλτιστη λύση εξερευνώντας τον χώρο εφικτής λύσης. Επίσης πέραν της λύσης χρονικών περιθωρίων, οι εφαρμογές τους καλύπτουν μία μεγάλη ποικιλία θεματικών όπως, της κατασκευής εφοδιαστικής αλυσίδας, των μεταφορών και της διαχείρισης έργων. Τέλος αυτά τα προβλήματα περιλαμβάνουν την κατανομή των πόρων και την αλληλουχία των δραστηριοτήτων για τη βελτιστοποίηση στόχων, όπως η ελαχιστοποίηση της απόδοσης ή η μεγιστοποίηση της χρήσης πόρων ή η τήρηση των περιορισμών προθεσμίας.

Οι λύτες MIP, σύμφωνα με τον ορισμό τους, χρησιμοποιούν έναν αλγόριθμο διακλάδωσης και δέσμευσης σε συνδυασμό με χαλάρωση γραμμικού προγραμματισμού για την αποτελεσματική αναζήτηση εφικτών λύσεων και την εύρεση της βέλτιστης λύσης. Ο αλγόριθμος διακλάδωσης και δέσμευσης χωρίζει συστηματικά τον χώρο αναζήτησης σε μικρότερα υποπροβλήματα, εξερευνώντας διαφορετικούς συνδυασμούς αποφάσεων. Το βήμα χαλάρωσης γραμμικού προγραμματισμού περιλαμβάνει τη χαλάρωση των ακέραιων περιορισμών του προβλήματος με σκοπό την επίλυση ενός προβλήματος. Αυτό παρέχει ένα χαμηλότερο όριο στην τιμή της βέλτιστης λύσης και βοηθά στην καθοδήγηση της διαδικασίας αναζήτησης.

Εκτός από τους λύτες MIP, πολλές παρόμοιες τεχνικές χρησιμοποιούνται για την αποτελεσματική επίλυση προβλημάτων προγραμματισμού. Αυτές οι τεχνικές περιλαμβάνουν:

**Προγραμματισμός περιορισμών** (Constraint Programming) [3]: Το CP είναι ένα δηλωτικό παράδειγμα προγραμματισμού που εστιάζει στη μοντελοποίηση και την επίλυση συνδυαστικών προβλημάτων χρησιμοποιώντας περιορισμούς. Στο πλαίσιο του προγραμματισμού, οι λύτες CP επιτρέπουν τη διατύπωση περιορισμών προγραμματισμού και βρίσκουν αυτόματα εφικτές λύσεις εξερευνώντας το δίκτυο περιορισμών.

**Τοπική αναζήτηση** [4]: Οι αλγόριθμοι τοπικής αναζήτησης ξεκινούν με μια αρχική λύση και τη βελτιώνουν επαναληπτικά κάνοντας μικρές τροποποιήσεις. Αυτές οι τροποποιήσεις καθοδηγούνται από μια δομή γειτονιάς που ορίζεται πάνω από το σύνολο των εφικτών λύσεων. Η τοπική αναζήτηση είναι ιδιαίτερα χρήσιμη για προβλήματα προγραμματισμού μεγάλης κλίμακας, όπου η εύρεση του βέλτιστου αποτελέσματος είναι δύσκολη.

**Ευρετικοί αλγόριθμοι** [5]: Οι ευρετικοί αλγόριθμοι είναι μέθοδοι βασισμένες σε κανόνες που λαμβάνουν τεκμηριωμένες αποφάσεις με βάση τη γνώση του συγκεκριμένου προβλήματος ή την τεχνογνωσία του τομέα. Αυτοί οι αλγόριθμοι στοχεύουν να βρουν γρήγορα λύσεις καλής ποιότητας, αν και μπορεί να μην εγγυώνται τη βέλτιστη λύση. Παραδείγματα ευρετικών αλγορίθμων για προγραμματισμό χρόνου, όπου οι εργασίες ή τα συμβάντα έχουν συγκεκριμένες προθεσμίες που πρέπει να τηρούνται. Δίνει προτεραιότητα στις εργασίες με βάση τις παλαιότερες προθεσμίες τους, διασφαλίζοντας ότι οι εργασίες με τις πλησιέστερες προθεσμίες προγραμματίζονται πρώτα. Αυτή η προσέγγιση στοχεύει στην ελαχιστοποίηση του αριθμού των χαμένων προθεσμιών και στη μεγιστοποίηση της συνολικής απόδοσης του συστήματος.

Αυτές οι τεχνικές, προσφέρουν διαφορετικούς συμβιβασμούς μεταξύ ποιότητας, λύσης και υπολογιστικής απόδοσης. Η επιλογή της καταλληλότερης τεχνικής εξαρτάται από τα συγκεκριμένα χαρακτηριστικά του προβλήματος προγραμματισμού, όπως το μέγεθος του προβλήματος, η πολυπλοκότητα και η διαθέσιμη γνώση για το συγκεκριμένο πρόβλημα. Συμπερασματικά, τα προβλήματα χρονοπρογραμματισμού είναι πολύπλοκες προκλήσεις βελτιστοποίησης που απαιτούν προηγμένες τεχνικές για την εύρεση βέλτιστων ή σχεδόν βέλτιστων λύσεων. Οι λύτες Μικτού Ακέραιου Προγραμματισμού (MIP), παρέχουν ισχυρά εργαλεία για την αποτελεσματική αντιμετώπιση προβλημάτων, ενεργώντας ως παράγοντας για την επιλογή χρήσης στο πρόγραμμα.

## Παράδειγμα παρόμοιου προβλήματος: «ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΟΛΙΤΙΣΤΙΚΩΝ ΓΕΓΟΝΟΤΩΝ» - MYvisitplanner GR - Εξατομικευμένο Σύστημα Σχεδιασμού Διαδρομών στον κλάδο του Τουρισμού

Αποτελεί ένα σύστημα προγραμματισμού ψυχαγωγικών ,πολιτιστικών και άλλων δραστηριοτήτων στην περιοχή της Βόρειας Ελλάδας. Βοηθώντας με προτάσεις για το πρόγραμμα που μπορούν να ακολουθήσουν οι επισκέπτες στην περιοχή. Οι δραστηριότητες αυτές κατηγοριοποιούνται σε κατηγορίες όπως ο τύπος της δραστηριότητας , η ιστορική εποχή , το προφίλ του χρήστη και η ηλικιακή ομάδα. Χαρακτηριστικά τους είναι το κόστος , η τοποθεσία , η διαθεσιμότητα αλλά και η διάρκεια. Το σύστημα έχει σαν σκοπό να κάνει προτάσεις στον χρήστη βάσει κριτήρια που έχει θέσει ο ίδιος για δεδομένα που τον ενδιαφέρουν περισσότερο (Διάρκεια ,χρόνος επίσκεψης κ.α.). Επιπλέον , με βάση το προτεινόμενο πλάνο ο χρήστης μπορεί να το επεξεργαστεί κατάλληλα ή και να ζητήσει ένα νέο. Ουσιαστικά, το σύστημα κάνει χρήση αλγορίθμου μηχανικής μάθησης με σκοπό την δυναμική αξιολόγηση των προτιμήσεων του χρήστη για καλύτερα αποτελέσματα στο μέλλον. [6].

# ΑΞΙΟΠΟΙΗΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΤΟ ΣΥΣΤΗΜΑ

Η ραγδαία ανάπτυξη των υποδομών του διαδικτύου σε συνδυασμό με τους περιηγητές ιστού, έχει δώσει τεράστια ώθηση στις εφαρμογές ιστού . Σε παρόμοιο ρυθμό και οι Τεχνολογίες ανάπτυξης λογισμικού έχουν φτάσει στο επίπεδο να μπορούν να υποστηρίξουν αποτελεσματικά την προσπάθεια κατασκευής μιας τέτοιας εφαρμογής. Για την κατασκευή μιας διαδικτυακής εφαρμογής πολλά είναι τα απαραίτητα εργαλεία για την ολοκλήρωση τους. Συγκεκριμένα, από την χρήση των γλωσσών που την αναπτύσσουν μέχρι και τις τεχνολογίες που την διαμορφώνουν σίγουρα αποτελούν τα βασικά θεμέλια του προγραμματισμού του συστήματος αυτού. Ο προγραμματιστής έχει θα πρέπει να επιλέξει όλα εκείνα τα κατάλληλα εργαλεία που θα εντάξει στην φαρέτρα του, με τον σκοπό της επίτευξης του στόχου του για την εύρεση της λύσης. Αυτό μπορεί να γίνει με διάφορους τρόπους. Στη συνέχεια παρουσιάζονται τα μέσα τα οποία δώσαν το τελική μορφή του συστήματος, αλλά και κάποιοι από τους λόγους που υπερίσχυσαν στην επιλογή αυτών.

Αρχικά απαραίτητη είναι η αναφορά της άντλησης των δεδομένων από ιστότοπους με ανάλυση του κώδικα HTML ή XML, κάνουμε λόγο για το Web Scrape [7]. Μία τεχνολογία η οποία μας επιτρέπει να τραβήξουμε τα διαθέσιμα δεδομένα από επιλεγμένες ιστοσελίδες τοποθετώντας τα στην δικιά μας. Υπάρχουν διάφορα πακέτα και τεχνολογίες που επιτυγχάνουν την συγκεκριμένη λειτουργία. Για παράδειγμα, η χρήση πλαισίων όπως το **Scrapy** [8], το οποίο είναι ένα πιο ολοκληρωμένο πλαίσιο απόξεσης ιστού για την Python. Το **Scrapy** παρέχει μια ολοκληρωμένη λύση, συμπεριλαμβανομένου του χειρισμού αιτημάτων, της ανάλυσης απαντήσεων και της διαχείρισης αγωγών εξαγωγής δεδομένων. Επιπλέον, ορισμένοι ιστότοποι προσφέρουν **API** (Διεπαφές Προγραμματισμού Εφαρμογών) [9] που επιτρέπουν στους προγραμματιστές να ανακτούν δεδομένα σε δομημένη μορφή χωρίς την ανάγκη απόξεσης ιστού. Τα API παρέχουν έναν τυποποιημένο τρόπο πρόσβασης και ανάκτησης συγκεκριμένων δεδομένων από τη βάση δεδομένων ενός ιστότοπου. Αυτή η προσέγγιση είναι συχνά πιο αξιόπιστη και αποτελεσματική για το web scraping, καθώς βασίζεται σε μια καλά καθορισμένη διεπαφή που παρέχεται από τον ίδιο τον ιστότοπο. Ωστόσο, όταν τα API δεν είναι διαθέσιμα ή δεν παρέχουν τα απαιτούμενα δεδομένα. Η βιβλιοθήκη που χρησιμοποιήθηκε τελικά ονομάζεται BeautifulSoup4 προσφέροντας μια ισχυρή και ευέλικτη λύση για την εξαγωγή δεδομένων από ιστότοπους. Το **BeautifulSoup** είναι μια βιβλιοθήκη Python που παρέχει έναν βολικό τρόπο εξαγωγής δεδομένων από έγγραφα HTML και XML. Κάνει την άντληση πληροφοριών ευκολότερη παρέχοντας ένα σύνολο λειτουργιών και μεθόδων για την πλοήγηση, την αναζήτηση και τον χειρισμό του αναλυμένου δέντρου HTML ή XML. Επίσης βοηθά στην εξαγωγή συγκεκριμένων στοιχείων, όπως κείμενο, συνδέσμους, πίνακες ή εικόνες, από ιστοσελίδες χρησιμοποιώντας διάφορες τεχνικές αναζήτησης και φιλτραρίσματος. Ακόμη, χρησιμοποιείται η βιβλιοθήκη αιτημάτων της Python για να στείλετε ένα αίτημα HTTP GET στην επιθυμητή διεύθυνση URL. Αυτό ανακτά το περιεχόμενο HTML της ιστοσελίδας. Αυτό αντιπροσωπεύει τη δομή δέντρου HTML που έχει αναλυθεί, η οποία μπορεί να πλοηγηθεί και να αναζητηθεί. Αυτό μπορεί να γίνει χρησιμοποιώντας ετικέτες, χαρακτηριστικά CSS ή άλλα μοτίβα αναζήτησης.

## ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Οι γλώσσες που χρησιμοποιήθηκαν για την ανάπτυξη και σχεδίαση της εφαρμογής.

### PYTHON

Η γλώσσα προγραμματισμού Python [10] αποτελεί μια από τις πιο πολυχρησιμοποιημένες και διαδεδομένες γλώσσες στον κόσμο. Η πολλαπλή χρηστικότητα της και η ευκολία είναι κάποιοι από τους λόγους που επιλέχθηκε σαν κύρια γλώσσα κατασκευής του προγράμματος. Με την κατάλληλη προσθήκη πακέτων και βιβλιοθηκών παρέχονται δυνατότητες για να καλυφθεί η οποιαδήποτε ανάγκη του συστήματος. Επιπλέον, αποδίδει την ικανότητα και σε τρίτους, μελετητές, να μπορούν να κατανοήσουν την κώδικα όντας ευανάγνωστη.

Λειτουργίες που προσφέρονται με την κατάλληλη χρήση πακέτων:

* Ως διαχειριστής του Διακομιστή(Server)
* Κατασκευή link
* Μοντελοποίηση και Διαχείριση βάσης
* Διαχείριση αιτημάτων HTTP
* Ταυτοποίηση σύνδεσης χρήστη
* Συλλογή δεδομένων από ιστοσελίδες(web scraping)
* Την μετάβαση των δεδομένων από σελίδα σε σελίδα
* Λύση του γραμμικού προβλήματος

Τα παραπάνω αποτελούν τις βασικότερες αιτίες που επιλέχθηκε η Python σε κύρια γλώσσα του συστήματος. Λειτουργίες που επιτεύχθηκαν με την προσθήκη των κατάλληλων βιβλιοθηκών(Βλέπε Ανάλυση του συστήματος και απαιτήσεις).

### JavaScript

Διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές που χρησιμοποιείται στο κομμάτι των φυλλομετρητών ιστού δίνοντας την ικανότητα αλληλεπίδρασης της ιστοσελίδας με τον χρήστη. [11]

### HTML

Ή αλλιώς Hypertext Markup Language ορίζεται ως η Κύρια γλώσσα σήμανσης για τις ιστοσελίδες. Προφανώς ανήκει στην μεριά του πελάτη (client) και είναι υπεύθυνη για την εμφάνιση των δεδομένων του συστήματος. [12]

### CSS

Ή και Cascading Style Sheets γλώσσα η οποία χρησιμοποιείται για την διαμόρφωση της εμφάνισης των εγγράφων που έχουν γραφτεί σε γλώσσα σήμανσης. [13]

## Πλαίσια – Web Frameworks

Τα πλαίσια αυτοματοποιούν την υλοποίηση πολλών εργασιών και δίνουν στους προγραμματιστές μια δομή για την ανάπτυξη εφαρμογών ιστού. Κάθε πλαίσιο συνοδεύεται από τη δική του συλλογή ενοτήτων ή πακέτων που μειώνουν σημαντικά τον χρόνο ανάπτυξης. Ένα πλαίσιο Python μπορεί να είναι είτε full-stack , micro ή ασύγχρονο. Πιο συγκεκριμένα, τα πλαίσια πλήρους στοίβας(full-stack) παρέχουν μια ολοκληρωμένη λύση για την ανάπτυξη εφαρμογών ιστού, προσφέροντας στοιχεία τόσο frontend όσο και backend. Περιλαμβάνουν εργαλεία και βιβλιοθήκες για το χειρισμό ολόκληρης της στοίβας εφαρμογών, συμπεριλαμβανομένων των διεπαφών χρήστη, της λογικής από την πλευρά του διακομιστή και των αλληλεπιδράσεων βάσης δεδομένων. Παραδείγματα πλαισίου πλήρους στοίβας περιλαμβάνουν τα Django (Python), Ruby on Rails (Ruby) και Laravel (PHP). Τα Microframeworks είναι ελαφριά πλαίσια web που εστιάζουν στην απλότητα και τον μινιμαλισμό. Συνήθως παρέχουν μόνο τα βασικά χαρακτηριστικά για τη δημιουργία εφαρμογών web, επιτρέποντας στους προγραμματιστές να έχουν περισσότερο έλεγχο και ευελιξία στη βάση κώδικα. Τα μικροπλαίσια προτιμώνται συχνά για μικρά έργα ή συγκεκριμένες περιπτώσεις χρήσης όπου ένα πλαίσιο πλήρους στοίβας μπορεί να είναι υπερβολικό. Παραδείγματα μικροπλαισίων περιλαμβάνουν τα Flask (Python), Sinatra (Ruby) και Express.js (JavaScript). Τα ασύγχρονα πλαίσια web έχουν σχεδιαστεί για να χειρίζονται υψηλά επίπεδα ταυτόχρονης χρήσης και να διαχειρίζονται αποτελεσματικά πολλαπλές ταυτόχρονες συνδέσεις. Αξιοποιούν τεχνικές ασύγχρονου προγραμματισμού, όπως I/O βάσει συμβάντων ή μη αποκλεισμού, για να μεγιστοποιήσουν την απόδοση και την επεκτασιμότητα. Τα ασύγχρονα πλαίσια είναι ιδιαίτερα κατάλληλα για εφαρμογές που απαιτούν ενημερώσεις σε πραγματικό χρόνο, ροή δεδομένων ή χειρισμό μεγάλου αριθμού ταυτόχρονων αιτημάτων. Παραδείγματα ασύγχρονων πλαισίων περιλαμβάνουν τα Tornado (Python), Node.js (JavaScript) και Vert.x (Java). Κάθε τύπος πλαισίου ιστού έχει τα δικά του δυνατά σημεία και είναι κατάλληλος για διαφορετικούς τύπους έργων και προσεγγίσεις ανάπτυξης.

Η επιλογή του σωστού πλαισίου εξαρτάται από παράγοντες όπως η πολυπλοκότητα της εφαρμογής, οι απαιτήσεις επεκτασιμότητας, οι προτιμήσεις προγραμματιστή και η γλώσσα προγραμματισμού που χρησιμοποιείται. Είναι σημαντικό να λάβετε υπόψη τις συγκεκριμένες ανάγκες του έργου σας και να αξιολογήσετε τα χαρακτηριστικά και τις δυνατότητες διαφορετικών πλαισίων πριν λάβετε μια απόφαση. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε το **Flask**. [14] [15] [16].

## FLASK

Το **Flask**, είναι ένα micro web πλαίσιο που παρέχει μινιμαλιστικές βασικές λειτουργίες για τη δημιουργία απλών εφαρμογών web. Είναι ελαφρύ και ευέλικτο, επιτρέποντας στους προγραμματιστές να προσθέσουν τη λειτουργικότητα που χρειάζονται μέσω **βιβλιοθηκών τρίτων**. Αυτό το καθιστά ιδανικό για τη δημιουργία μικρότερων, λιγότερο σύνθετων εφαρμογών. Το **Flask** ακολουθεί μια αρχιτεκτονική μοντέλου-προβολή-πρότυπο (MVT), όπου οι προβολές καθορίζουν τη λογική που χειρίζεται αιτήματα και απαντήσεις, τα πρότυπα ορίζουν τον τρόπο με τον οποίο παρουσιάζονται τα δεδομένα στο χρήστη και τα μοντέλα συνήθως χειρίζονται μέσω βιβλιοθηκών τρίτων.

Συγκεκριμένα:

* Το **Flask** χρησιμοποιεί τα επονομαζόμενα σχεδιαγράμματα (αγγλικός όρος: Blueprints ή αλλιώς Προσχέδια) για την δημιουργία στοιχείων εφαρμογής , δίνοντας δομή και λειτουργικότητα σε επαναχρησιμοποιήσιμα στοιχεία . Τα προσχέδια απλοποιούν τον τρόπο με τον οποίο λειτουργούν εφαρμογές παρέχοντας κεντρικό μέσο για επεκτάσεις που ενδέχεται να γίνουν στο μέλλον. Τα Προσχέδια μοιάζουν με **Flask** εφαρμογή αλλά στην πραγματικότητα δεν είναι. Είναι τρόπος κατασκευής, επέκτασης, διαμόρφωσης της εφαρμογής. Φίλτρα προτύπων , στατικά αρχεία , πρότυπα , άλλα βοηθητικά προγράμματα , κάποια από τα προτερήματα των σχεδιαγραμμάτων.
* Μπορεί να καταχωρηθεί ένα σχέδιο σε μια εφαρμογή σε ένα πρόθεμα (prefix) URL ή και έναν υπό τομέα. Κάτι το οποίο έχει ως αποτέλεσμα να δίνει όνομα στις απαραίτητες λειτουργίες στον ιστότοπο. Οι παράμετροι στο πρόθεμα υπό τομέα διεύθυνσης URL γίνονται κοινά ορίσματα προβολής (με προεπιλογές) σε όλες τις συναρτήσεις προβολής στο σχεδιάγραμμα.
* Διαχείριση και αποστολή αιτημάτων Πρωτοκόλλων Μεταφοράς Υπερκειμένου Επικοινωνίας (HTTP).

## ΠΑΡΟΜΟΙΑ FRAMEWORKS ΜΕ ΤΟ FLASK:

Πέραν του FLASK υπάρχει μεγάλη ποικιλία και παρόμοιες τεχνολογίες σαν αυτή. Αρχικά, το **Django** [17]είναι ένα πλαίσιο ιστού Python υψηλού επιπέδου που ακολουθεί το αρχιτεκτονικό μοτίβο Model-View-Controller (MVC). Παρέχει ένα σύνολο εργαλείων, βιβλιοθηκών και συμβάσεων για να βοηθήσει τους προγραμματιστές να δημιουργήσουν εφαρμογές Ιστού γρήγορα και αποτελεσματικά. Ο κύριος στόχος του Django είναι να απλοποιήσει τη διαδικασία ανάπτυξης ιστού παρέχοντας προκατασκευασμένα στοιχεία και προωθώντας βέλτιστες πρακτικές. Επιπλέον, έχει μια μεγάλη και ενεργή κοινότητα, η οποία συμβάλλει στο εκτεταμένο οικοσύστημά του με πακέτα και βιβλιοθήκες τρίτων. Χρησιμοποιείται ευρέως σε διάφορους τομείς και έχει υιοθετηθεί από πολλούς δημοφιλείς ιστότοπους και οργανισμούς. Συνολικά, η εστίαση του Django στην απλότητα, την επεκτασιμότητα και την τήρηση των βέλτιστων πρακτικών το καθιστά δημοφιλή επιλογή για την ανάπτυξη ισχυρών εφαρμογών ιστού στην Python.

**Pyramid**: Ένα ευέλικτο και ελαφρύ πλαίσιο web που επιτρέπει την επιλογή στοιχείων που χρειάζεστε για να δημιουργηθεί η εφαρμογή. Προσφέρει πολλές προηγμένες δυνατότητες, όπως πολιτικές ασφαλείας, δημιουργία URL και υποστήριξη cookie. [18]

**Bottle**: Το Bottle είναι ένα άλλο ελαφρύ πλαίσιο web που είναι παρόμοιο με το Flask. Έχει σχεδιαστεί για να είναι εύκολο στη χρήση και εύκολο στην εκμάθηση. Έρχεται με ενσωματωμένη υποστήριξη για πρότυπα, δρομολόγηση και διαχείριση αιτημάτων. [19]

## JINJA (TEMPLATE ENGINE)

To **Jinja** είναι μια μηχανή προτύπων Ιστού για τη γλώσσα προγραμματισμού Python. Χρησιμοποιείται για την σύνδεση του χρήστη με το σύστημα προσφέροντας την δυνατότητα αλληλεπίδρασης .Είναι μια γλώσσα προτύπου βασιζόμενη σε κείμενο και μπορεί να χρησιμοποιηθεί σε γλώσσα σήμανσης, όπως και έγινε. Είναι προεπιλεγμένη μηχανή προτύπων του **Flask** οπότε έδωσε μεγάλη ευκολία στην διαμόρφωση της εμφάνισής των δεδομένων. Δηλαδή, την σύνδεση και την αποστολή των δεδομένων μεταξύ front και back end. Ως προς την ανάπτυξη η κληρονομικότητα κομματιών html , είναι που την καθιστούν χρήσιμη αλλά και βολική. Ως προς την σύνταξη , γίνεται χρήση αγκυλών στις οποίες γράφεται κώδικας python όπως μεταβλητές, if statement, for loop ,κλπ .

Παράδειγμα: {%for i in range 10 %} , {%if condition%}, {{event.name}}

Με την χρήση του python flask framework είναι επόμενο στις περισσότερες περιπτώσεις να χρησιμοποιηθεί η μηχανή προτύπων Jinja. [20]

## ΕΠΙΠΡΌΣΘΕΤΕΣ ΕΠΕΚΤΑΣΕΙΣ ΓΙΑ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Το σύνολο των επεκτάσεων που έχουν προστεθεί στο πρόγραμμα για την κάλυψη των αναγκών των διαφόρων λειτουργιών του. Παρακάτω παρουσιάζονται οι τεχνολογίες που αξιοποιήθηκαν με σκοπό την προσφορά ενός ολοκληρωμένου και αυτόνομου συστήματος. Πιο αναλυτικά έχουμε:

### Επέκταση:

* **Εισόδου χρήστη**: Με την χρήση της βιβλιοθήκης για την είσοδο του χρήστη (flask\_login). Υπεύθυνη επέκταση για την ταυτοποίηση τις λίστας χρηστών.
* **Κρυπτογράφηση κωδικού χρήστη** , επέκταση Werkzeug.security: Αποτελεί μέθοδο για την υπεύθυνη για την κρυπτογράφηση του κωδικού του εκάστοτε χρήστη. [21]
* **Για την δημιουργία βάσης** (flask\_sqlalchemy) [22] : Απαραίτητη επέκταση για την δημιουργία και συντήρηση των δεδομένων στο σύστημα. Παρέχει λειτουργίες όπως :

1. Δημιουργία Βάσης δεδομένων, εάν δεν υπάρχει.

2. Ερωτήματα σε Δομημένη Σειριακή Γλώσσα (SQL). Υπεύθυνη για την δημιουργία , διαγραφή αντικειμένων από και προς πίνακες.

3. Αναζήτηση πινάκων.

* **Ημερομηνίες και τωρινή ώρα :** Διαχείριση σε δεδομένα τύπου ημερομηνιών και χρήση τωρινής ημερομηνίας.
* **Για τον προγραμματισμό ανανέωσης της βάσης :**(apscheduler.schedulers. flask\_apscheduler/ APScheduler) : Το ‘APScheduler είναι μια κλάση από το πακέτο python 'flask\_APScheduler', το οποίο επιτρέπει τον προγραμματισμό εργασιών που θα εκτελούνται σε συγκεκριμένες ώρες ή διαστήματα. Μπορεί να χρησιμοποιηθεί για εργασίες όπως η περιοδική ενημέρωση μιας βάσης δεδομένων ή η αποστολή ειδοποιήσεων μέσω email. Ο προγραμματιστής εκτελείται στο παρασκήνιο της εφαρμογής ,και μεταβλητές όπως συχνότητα , ώρα καθορίζονται ώστε μια συγκεκριμένη εργασία να εκτελείται συγκεκριμένη στιγμή (Παράδειγμα: Κάθε 24 ώρες εκτέλεση μεθόδου ανανέωσης της Βάσης). Συνολικά, το BackgroundScheduler είναι ένα ισχυρό εργαλείο για την αυτοματοποίηση εργασιών στην εφαρμογή σας και χρησιμοποιείται συνήθως σε εφαρμογές Ιστού και άλλο λογισμικό από την πλευρά του διακομιστή.
* **Τέλος, έχει γίνει προσθήκη της επέκτασης «orTools.linear\_solver» :** Μία επέκταση που χρησιμοποιείται για την Λύση του προβλήματος που λύνει το πρόγραμμα , «χρονοπρογραμματισμός Πολιτιστικών Γεγονότων» . (Βλέπε «Μοντελοποίηση του προβλήματος και Υλοποίηση»).

## ΑΝΤΛΗΣΗ ΔΕΔΟΜΕΝΩΝ ΙΣΤΟΥ - WEB SCRAPE

Για την άντληση των δεδομένων, το σύστημα χρειάζεται μια αυτοματοποιημένη και γρήγορη μέθοδο. Σε αυτό το σημείο δεν εξηγείται πως λειτουργεί το κομμάτι του κώδικα με την υλοποίηση (βλέπε κεφάλαιο Υλοποίηση) αλλά την τεχνολογία για την χειραγώγηση των δεδομένων ώστε να είναι αξιοποιήσιμα από το σύστημα.

I. beautifulsoup4

II. Curl

Είναι 2 τεχνολογίες που αξιοποιούνται για να επιτευχθεί το αποτέλεσμα την απόκτησης τον δεδομένων από άλλες ιστοσελίδες. Πιο συγκεκριμένα , ξεκινώντας από το «Curl». [23] [24]

## Χρήση Curl

Αποτελεί μία βιβλιοθήκη, που συνδυαστικά με την γραμμή εντολών της κονσόλας του διακομιστή επιτυγχάνεται η μεταφορά δεδομένων, χρησιμοποιώντας διάφορα πρωτόκολλα δικτύου. Έχει ένα μεγάλο αριθμό υποστηριζόμενων πρωτόκολλων σχεδιασμένο έτσι ώστε να λειτουργεί χωρίς αλληλεπίδραση του χρήστη. Αναλυτικότερα μπορεί να στείλει σαν αίτημα μια μορφοποιημένη σειρά δεδομένων σε έναν συγκεκριμένο στόχο ή και στόχους με αποτέλεσμα την άντληση δεδομένων.

Παρακάτω κάποια από τα προτερήματα της τεχνολογίας αυτής :

Πολλαπλές διευθύνσεις URL

Τα ονόματα χρηστών και οι κωδικοί πρόσβασης υποστηρίζονται

Υποστήριξη IPv6

Επανάληψη λήψης

Ακολουθίες διευθύνσεων URL

Υποστήριξη Win32

Μεγάλη υποστήριξη αρχείων

TFTP

SCP upload / download

SFTP upload / download

HTTP Proxy

Πεδία HTTP

Ακολουθία ανακατευθύνσεων HTTP

## Επέκταση «beautifulsoup4»

Η συγκεκριμένη επέκταση χρησιμοποιείται με σκοπό διαχείριση αδόμητων δεδομένων . Με άλλα λόγια διαθέτει μεθόδους κατάλληλες ώστε να μπορεί να λάβει ένα κείμενο γραμμένο σε γλώσσα σήμανσης και να επιστρέφει σαν αποτέλεσμα δεδομένα. Δεδομένα τα οποία στην συνέχεια είναι διαθέσιμα για να επεξεργαστούν. Χρήσιμα στοιχεία :

* Ιδιωματισμοί για πλοήγηση αναζήτηση , ανάλυση.
* Τροποποίηση δέντρου ανάλυσης και διεξαγωγή τον χρήσιμών στοιχείων.
* Διάφορες στρατηγικές ανάλυσης.

Προτερήματα που με την σωστή χρήση τους προσφέρουν ταχύτητα , ευελιξία και χρηστικότητα.

## Εργαλεία για χρονοπρογραμματισμό

### Google OR-Tools

Το **Google** **OR-Tools** [25] είναι μια σουίτα λογισμικού βελτιστοποίησης ανοιχτού κώδικα που αναπτύχθηκε από την Google. Παρέχει ένα σύνολο εργαλείων για προβλήματα βελτιστοποίησης, συμπεριλαμβανομένου του μαθηματικού προγραμματισμού, του προγραμματισμού περιορισμών και της τοπικής αναζήτησης. Χρησιμοποιήθηκε στο πρόγραμμα σαν εργαλείο για την εύρεση της βέλτιστης λύσης στο πρόβλημα του χρονοπρογραμματισμού των παραστάσεων. (Βλέπε υλοποίηση για λεπτομέρειες).

## Εργαλεία παρόμοια με το google Or-Tools για χρονοπρογραμματισμό

1. **PuLP**: Το PuLP είναι μια βιβλιοθήκη μοντελοποίησης γραμμικού προγραμματισμού ανοιχτού κώδικα στην Python. Παρέχει μια διαισθητική σύνταξη για τον ορισμό γραμμικών προγραμμάτων και υποστηρίζει μια ποικιλία λύσεων, συμπεριλαμβανομένων των CBC, CPLEX, GLPK και άλλων. [26]

2. **SCIP**: Το SCIP είναι ένας λύτης βελτιστοποίησης ανοιχτού κώδικα για προγραμματισμό μικτών ακεραίων και προγραμματισμό περιορισμών. Παρέχει ένα ευρύ φάσμα χαρακτηριστικών, συμπεριλαμβανομένου γραμμικού και μη γραμμικού προγραμματισμού, διάδοσης περιορισμών και σύσφιξης δεσμών. Το SCIP είναι γραμμένο σε C++ και προσφέρει ένα API που μπορεί να χρησιμοποιηθεί σε διάφορες γλώσσες προγραμματισμού, συμπεριλαμβανομένων των Python και C++. [27]

1. ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΉΜΑΤΟΣ ΚΑΙ ΘΕΩΡΗΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

# ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΧΡΗΣΤΗ

## Γενική μορφή απαιτήσεων

Αναφερόμαστε στην αλληλοεπίδραση του χρήστη με το σύστημα ως μία σύνδεση απαιτήσεων για την ολοκληρωμένη εμπειρία του χρήστη. Περιλαμβάνει τον εντοπισμό, την κατανόηση και την τεκμηρίωση των αναγκών, των προσδοκιών και των περιορισμών των χρηστών-στόχων. Ο στόχος της ανάλυσης απαιτήσεων χρήστη είναι να διασφαλίσει ότι το τελικό προϊόν ανταποκρίνεται στις προσδοκίες των χρηστών, παρέχει αξία και αντιμετωπίζει αποτελεσματικά τα συγκεκριμένα σημεία πιθανού error. Στην εργασία οι δυνατότητες και οι απαιτήσεις του χρήστη έχουν να κάνουν κυρίως με κάποιες από τις αναμενόμενες ενέργειες που προσφέρουν οι περισσότερες σύγχρονες ιστοσελίδες. Πιο συγκεκριμένα, αναφερόμαστε στην δυνατότητα δημιουργία λογαριασμών, στην επιλογή των διαθέσιμων παραστάσεων και την δυνατότητα ελέγχου του αποτελέσματος. Αυτές είναι οι βασικότερες απαιτήσεις που οδήγησαν και στην κατασκευή του κατάλληλου user interface.

Γενικά σε μία εφαρμογή ιστού υπάρχουν συγκεκριμένες παράμετροι που την καθιστούν χρηστική. Είναι αυτές που θα απαιτηθούν από τους χρήστες και θα τους κάνουν να θέλουν να την χρησιμοποιήσουν. Διεξοδικά μερικές από τις παραμέτρους αυτές είναι ο απλός σχεδιασμός της σελίδας, η καλή και γρήγορη ανταπόκριση των στοιχείων, η γρήγορη επεξεργασία και εμφάνιση των δεδομένων και τελικά η παρουσίαση των αναμενόμενων αποτελεσμάτων. Ο μέσος χρήστης τις περισσότερες φορές θέλει να αλληλοεπιδράσει με ένα σύστημα το οποίο αρχικά εμφανισιακά θα τον διευκολύνει ενώ παράλληλα θα του δοθεί εύκολα και γρήγορα η πληροφορία που θέλει. Τέλος, το σύστημα οφείλει να είναι άμεσο και γρήγορο προσφέροντας μία καλή εμπειρία χρήσης. [28]

## Απόδοση και επεξήγηση στον χρήστη

Το πρόγραμμα στη προσπάθεια εγκλιματισμού του χρήστη στον τρόπο λειτουργίας της πτυχιακής εργασίας τον καλωσορίζει σε ένα αρχικό welcome screen. Αυτό αναφέρει κάποιες βασικές πληροφορίες για την μετέπειτα ροή του συστήματος. Με άλλα λόγια, γίνεται η πρώτη αναφορά σχετικά με το θέμα της ιστοσελίδας προϊδεάζοντας τον επισκεπτόμενο σχετικά με αυτή. Επιπλέον τον προτρέπει στην κατασκευή ενός λογαριασμού διατηρώντας κάποιων πληροφοριών στην προσπάθεια προσφοράς μιας ολοκληρωμένης εμπειρίας. Έτσι του δίνεται η επιλογή δημιουργίας, sign up, ενός νέου λογαριασμού ως νέος user ή εάν έχει ήδη δημιουργήσει account, να κάνει log in. Αναφορικά, στην ένταξη των στοιχείων του λογαριασμού υπάρχουν συγκεκριμένες απαιτήσεις για τα πεδία που πρέπει να συμπληρωθούν. Κάποια από αυτά για παράδειγμα αποτελούν το email θα πρέπει να είναι στο κατάλληλο format, ο κωδικός να έχει πάνω από 7 γράμματα. Το πρόγραμμα παρουσιάζει στον χρήστη μηνύματα βοηθώντας τον για την συμπλήρωση αυτών των στοιχείων.

## Δυνατότητες αλληλεπίδρασης με το σύστημα

Το πρόγραμμα έχει σαν απώτερο σκοπό την παρουσίαση ενός βέλτιστου αποτελέσματος βάσει των επιλογών του χρήστη. Έτσι, οι επιλογές αυτές θα πρέπει να γίνουν. Διαφορετικά δεν θα υπάρξει κάποιο αποτέλεσμα αφού ο κώδικας δεν θα έχει τροφοδοτηθεί με τα κατάλληλα δεδομένα ώστε να προσπαθήσει να λύσει το πρόβλημα. Επομένως δίνεται η ευκαιρία στον χρήστη μέσω των κουμπιών Select να επιλέξει όσες από τις διαθέσιμες παραστάσεις επιθυμεί να παρακολουθήσει, βάσει των προτιμήσεων του, της διάθεσης του και των προσφερόμενων ημερομηνιών της καθεμιάς. Αφού επιλέξει όσες από αυτές του ταιριάζουν έχει να κάνει ακόμα ένα πράγμα για την τελική παρουσίαση του προγράμματος. Αυτό είναι η ενεργοποίηση του λύτη των περιορισμών, πατώντας δηλαδή το κουμπί Submit. Οπότε, το πρόβλημα λύνεται και το αποτέλεσμα εμφανίζεται στην τελική οθόνη. Ουσιαστικά ελευθερία που δίνεται στο κάθε χρήστη είναι η εκάστοτε επιλογή από το σύνολο των παραστάσεων. Συνολικά αυτά που μπορεί να κάνει είναι η δημιουργία ενός λογαριασμού sign up, η είσοδος του log in/log out και η επιλογή των παραστάσεων που έχει στην διάθεση του προς παρακολούθηση. Τέλος, μία ακόμη επιλογή που υπάρχει στο navigation bar είναι η επιλογή του profile του έχοντας την ικανότητα να ξανά δει το τελευταίο διαθέσιμο πρόγραμμα που του δημιουργήθηκε, όντας αποθηκευμένο, σε περίπτωση που επιθυμεί.

# ΑΝΤΛΗΣΗ ΤΩΝ ΑΠΑΡΑΙΤΗΤΩΝ ΠΛΗΡΟΦΟΡΙΩΝ

## ΛΟΓΟΙ ΑΠΑΙΤΗΣΗΣ

Ο σχεδιασμός και η λογική της εφαρμογής απαίτησαν την αυτόματη άντληση και παρουσίαση των δεδομένων. Με κάποιον τρόπο, δηλαδή, η βάση των δεδομένων να ανανεώνεται με τις παραστάσεις που υπάρχουν την κάθε ημέρα, καθιστώντας τα δεδομένα του προγράμματος επίκαιρα. Οπότε ο χρήστης θα γνωρίζει ότι οι διαθέσιμες εκδηλώσεις που του εμφανίζονται, πρώτον, έχουν ανανεωθεί σίγουρα το τελευταίο εικοσιτετράωρο και, δεύτερον, ότι είναι προσβάσιμες για το κοινό. Επομένως, για την επίτευξη μιας τέτοιας απαίτησης έχοντας λάβει υπόψη τα παραπάνω καταλήγουμε στην τεχνική του Web Scrape.

## ΑΞΙΟΠΟΙΗΣΗ ΑΠΑΙΤΗΣΗΣ ΣΤΟ ΣΥΣΤΗΜΑ

Στο σύστημα όπως προαναφέρθηκε η τεχνολογία του web scrape είναι απαραίτητη για την τροφοδότηση της βάσης των δεδομένων. Συγκεκριμένα οι παραστάσεις που εμφανίζονται αντλούνται από την ιστοσελίδα [29]

Ιστότοπος ο οποίος κρίθηκε κατάλληλος και αξιόπιστος αφού προσφέρει μία μεγάλη ποικιλία παραστάσεων στα διάφορα θέατρα της πόλης. Έτσι, αφού ληφθούν τα δεδομένα που χρειάζονται αυτό που μένει είναι η προσαρμογή τους στο σύστημα. Δεδομένα τα οποία αντιπροσωπεύουν τα χαρακτηριστικά της κάθε παράστασης όπως το όνομα, την εικόνα εξωφύλλου της, την τοποθεσία του θεάτρου, την περιγραφή της, τις διαθέσιμες ημερομηνίες που θα παίζεται και τέλος κάποιες επιπλέον πληροφορίες σχετικά με αυτή (πιθανόν τιμές, διάρκεια). Όλα αυτά τα χαρακτηριστικά αντλούνται με την αναζήτηση στο δέντρο HTML και πιο αναλυτικά μέσω των Div Tag που τους έχουν ανατεθεί. Με άλλα λόγια, έχει γίνει μία πολύ διεξοδική ανάλυση του πηγαίου κώδικα σήμανσης της ιστοσελίδας. Στη συνέχεια, αφού καταφέρουμε και αντλήσουμε τα παραπάνω, το σύστημα είναι υπεύθυνο για την ένταξη τους στην βάση των δεδομένων ως πεδία των παραστάσεων και πιο συγκεκριμένα στην ένταξη του κατάλληλου πίνακα. Οπότε τελικά, αυτά αποθηκεύονται στην βάση και είναι καθίστανται ικανά να εμφανιστούν στην βασική σελίδα που περιέχει το σύνολο όλων αυτών των παραστάσεων.

# ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Στην επόμενη παράγραφο εξηγούμε την θεωρητική σκοπιά που περιγράφει το πρόβλημα της εργασίας καθώς και τον τρόπο με τον οποίο επιτεύχθηκε να βρεθεί μια ικανοποιητική λύση. Αναφερόμαστε, με άλλα λόγια, στους συντελεστές που το ορίζουν και του δίνουν δομή. Στη συγκεκριμένη περίπτωση η εύρεση ενός ιδανικού προγράμματος θα προκύψει ως λύση με στόχο ο χρήστης να μπορέσει να παρευρεθεί σε όσο το δυνατόν περισσότερες παραστάσεις γίνεται. Λύνεται με αλλά λόγια ένα πρόβλημα χρονοπρογραμματισμού για θέαση θεατρικών παραστάσεων αποσκοπώντας να παρευρεθεί σε όσες περισσότερες από αυτές. Ωστόσο, υπάρχουν συγκεκριμένες παράμετροι οι οποίες εξ ‘αρχής πρέπει να οριστούν ώστε το πρόβλημα να αντιστοιχίζεται στην πραγματικότητα ενώ παράλληλα θα έχει κάποια εφικτή διευθέτηση. Αρχικά θεωρώντας ότι ο κάθε άνθρωπος δεν έχει ολόκληρη την ημέρα στην διάθεση του καθώς και ότι οι παραστάσεις πολλές φόρες βρίσκονται σε διαφορετικά θέατρα σε διαφορετικές περιοχές στην πόλη, καταλήγουμε στον πρώτο κύριο περιορισμό ότι κάθε ημερομηνία θα μπορεί να έχει όχι παραπάνω από μια παραστάσεις την ημέρα. Αυτό προφανώς έχει να κάνει, θεωρητικά, με τον χρόνο που χρειάζεται κανείς για να πάει από το ένα θέατρο στο άλλο αλλά και ότι μέσα στην κάθε ημέρα δεν αφιερώνει όλο τον ελεύθερο χρόνο του στη θέαση. Φυσικά κάθε παράσταση θεωρούμε ότι μπορεί να παρακολουθηθεί μια και μόνο φορά αφού δεν είναι λογικό κάποιος θεατής να παρευρεθεί 2 φορές στην ίδια. Επιπρόσθετα πηγαίνοντας στο σημείο της άντλησης των πληροφοριών για την συμπλήρωση της βάσης δεδομένων, παραστάσεις που δεν τους αντιστοιχίζονται ημερομηνίες, διαγράφονται, όπως και παραστάσεις με περασμένες ημερομηνίες από την τωρινή δεν είναι λογικό να εμφανίζονται σαν διαθέσιμες.

Ένα επιπλέον κομμάτι που προσδίδει τόσο λειτουργικότητα όσο και πρακτικότητα είναι εκείνο των **προτεραιοτήτων** των παραστάσεων ή αλλιώς **priority**. Η δυνατότητα της επιλογής του χρήστη στην απόδοση βαρύτητας στη εκάστοτε παράσταση, δίνοντας της περισσότερη έμφαση και προτρέποντας το σύστημα να την επιλέξει στο αποτέλεσμα. Ουσιαστικά αποτελεί έναν αριθμό ο οποίος κυμαίνεται από το 1 έως το 3, που 1 είναι το λιγότερο (και προεπιλεγμένο) και 3 το περισσότερο, με αποτέλεσμα οι βασικότερες (σύμφωνα με τον χρήστη πάντα) εκδηλώσεις να εμφανίζονται στο πρόγραμμα του χρήστη. Πρακτικά, όταν οι ημερομηνίες δύο ή και περισσότερων παραστάσεων συμπίπτουν τότε επιλέγεται εκείνη με το μεγαλύτερο priority.

Αυτό που μένει να μελετηθεί είναι η ίδια η λύση του προβλήματος. Όταν αναφερόμαστε στην εύρεση του βέλτιστου προγράμματος εννοούμε μια λύση η οποία αρχικά θα ικανοποιεί τους βασικούς περιορισμούς που αναφέρθηκαν και μετ' έπειτα να είναι η βέλτιστη. Για την εύρεση του ζητούμενου αυτού οι μέθοδοι σε προγραμματιστικά περιβάλλοντα είναι σχεδόν αμέτρητοι. Συγκεκριμένα, ο γραμμικός προγραμματισμός σίγουρα αποτελεί μία από τις καλύτερες αλλά και ταχύτερες μεθόδους. Έτσι εάν θέσουμε τις κατάλληλες μεταβλητές και τα κατάλληλα όρια καταλήγουμε σε ένα μοντέλο επίλυσης που έχει να κάνει με άθροισμα ακεραίων τιμών δυαδικών μεταβλητών απόφασης. Δηλαδή μοντελοποιούμε και λύνουμε ένα σύνολο περιορισμών αθροισμάτων, στοιχείων ως μία βασική προσέγγιση της λύσης. Η μοντελοποίηση της υλοποίησης του προβλήματος είναι αυτή που βοηθάει στην καλύτερη κατανόηση των λειτουργιών του προβλήματος.

1. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΛΥΣΗΣ

# Απαραίτητες βιβλιοθήκες ανάπτυξης – Απαιτήσεις συστήματος

Πληθώρα η χρήση των τεχνολογιών και των πακέτων που απαιτήθηκαν με σκοπό την επίτευξη των απαραίτητων ικανοτήτων της εφαρμογής. Συγκεκριμένα ως Requirements εννοούμε εκείνες τις βιβλιοθήκες-Πακέτα που αξιοποιήθηκαν για την σύνταξη και την σχεδίαση του κώδικα με σκοπό την επίτευξη των διαφόρων λειτουργιών του συστήματος.

Στο κομμάτι του πηγαίου κώδικα υπάρχει αρχείο επονομαζόμενο ¨Requirements.txt¨ το οποίο εμπεριέχει αναλυτικά τα ονόματα των πακέτων αλλά και τον αριθμό των εκδόσεων που χρησιμοποιήθηκαν εκείνη την δεδομένη στιγμή. Με άλλα λόγια υπάρχει ένας κατάλογος που βοηθάει τον προγραμματιστή να ξέρει τι χρησιμοποίησε για να φέρει εις πέρας την εκάστοτε απαιτούμενη διεργασία.

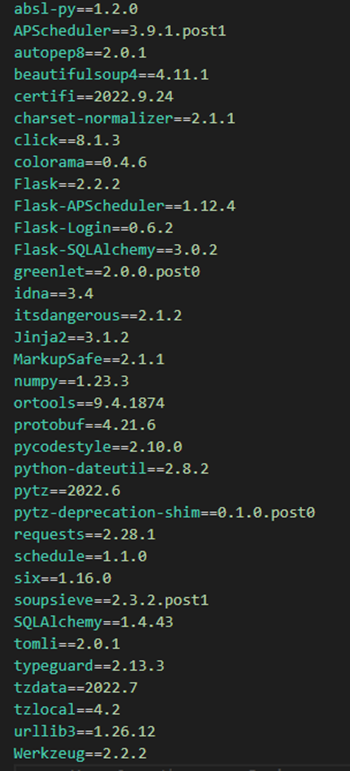
Στη συγκεκριμένη εφαρμογή τα απαιτούμενα πακέτα ποικίλλουν αφού υπάρχουν ανάγκες όπως: επικοινωνία του χρήστη με τον διακομιστή , του συστήματος με την Βάση , κρυπτογράφηση κωδικών του κάθε χρήστη κ.α.

Τα βασικότερα πακέτα που χρησιμοποιήθηκαν :

* Flask , -login, -sqlalchemy ,apscheduler
* Jinja2
* Werkzeug
* beautifulsoup4
* Google Or-Tools

Τα παρακάτω είναι απαραίτητα για την εκτέλεση του κώδικα και αν δεν υπάρχουν κατεβασμένα στο σύστημα , η εφαρμογή δεν θα μπορεί να τρέξει εμφανίζοντας σφάλματα. Επίσης , σε περίπτωση που κάποιος εξωτερικός θα ήθελε να τρέξει και να δοκιμάσει την εφαρμογή το “Requirements.txt” είναι το αρχείο που θα του δείξει επακριβώς τα εργαλεία που θα χρειαστεί.

Πιο συγκεκριμένα οι ακριβείς απαιτήσεις του συστήματος σε αρχείο τύπου text. Με τους αριθμούς των εκδόσεων που αξιοποιήθηκαν.



Εικόνα “ Αρχείο απαιτήσεων”

# Σχεδίαση διεπαφής χρήστη

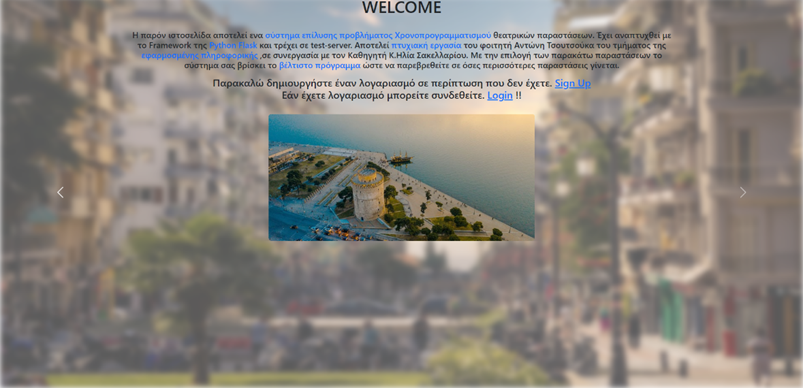
Η **διεπαφή** χρήστη ή αλλιώς **User Interface** [30] είναι εκείνη που προσφέρει την εικόνα στον χρήστη αλλά και την δυνατότητα αλληλεπίδρασης με αυτή. Έχει να κάνει με τη μεριά του client. Δηλαδή τι του εμφανίζεται και τι request στέλνεται στο πρόγραμμα αλληλοεπιδρώντας με αυτό. Για την σχεδίαση της ιστοσελίδας η γλώσσα σήμανσής html αξιοποιήθηκε σε συνδυασμό με το styling της γλώσσας **CSS**. Επιπλέον, με σκοπό την διατήρηση μιας οπτικής αρμονίας το **bootstrap** λειτούργησε ως κυρίαρχη δομική βιβλιοθήκη ώστε πολλά από τα elements που υπάρχουν να έχουν κοινά χαρακτηριστικά και να μην χρειάζεται περεταίρω προσαγόρευση.

Ως προς την δομή της σχεδίασης στο δέντρο των φακέλων διατηρείται φάκελος /templates που μέσα έχει όλες τις κύριες οθόνες html. Ενώ για τα στατικά αρχεία τύπου φωτογραφίες, το styling CSS αλλά και ο κώδικας client “index.js” (προσδίδει λειτουργικότητα στα κουμπιά). Συγκεκριμένα για τον φάκελο /templates βλέπουμε τα html αρχεία: base, welcomescreen, home, login, signup, userevents, result. Καθένα από το αυτά προσφέρει την εικόνα που περιγράφεται από το όνομα, αντίστοιχα . Για παράδειγμα, το home είναι η βασική οθόνη με την παραστάσεις, το login η οθόνη που κάνει login ο χρήστης, το signup αξιοποιείται για την εγγραφή του χρήστη.

## ΑΚΟΛΟΥΘΙΑ ΟΘΟΝΩΝ ΔΙΕΠΑΦΗΣ

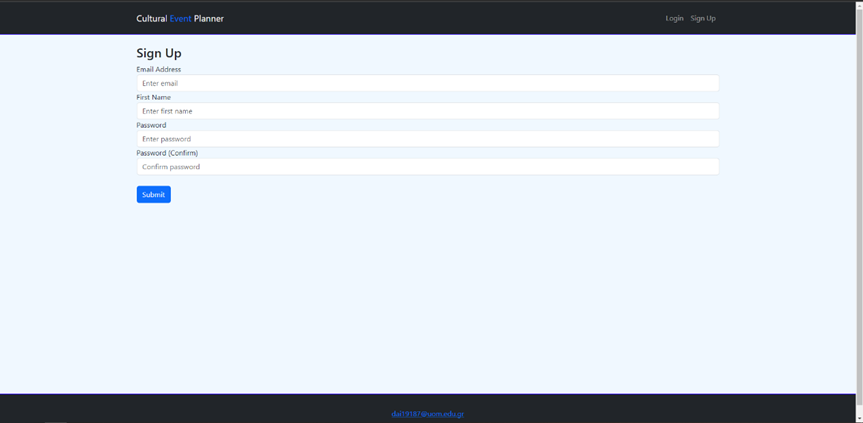
Η ροή που ακολουθείται στο πρόγραμμα από την είσοδο του χρήστη μέχρι την εμφάνιση του τελικού αποτελέσματος. Αρχικά, η πρωταρχική οθόνη που καλωσορίζει τον χρήστη είναι το welcome screen, η οποία τον παροτρύνει είτε στην δημιουργία ενός λογαριασμού (μετάβαση σε signup) είτε στην είσοδο του, εάν υπάρχει ήδη λογαριασμός (μετάβαση σε login).

### Welcome screen



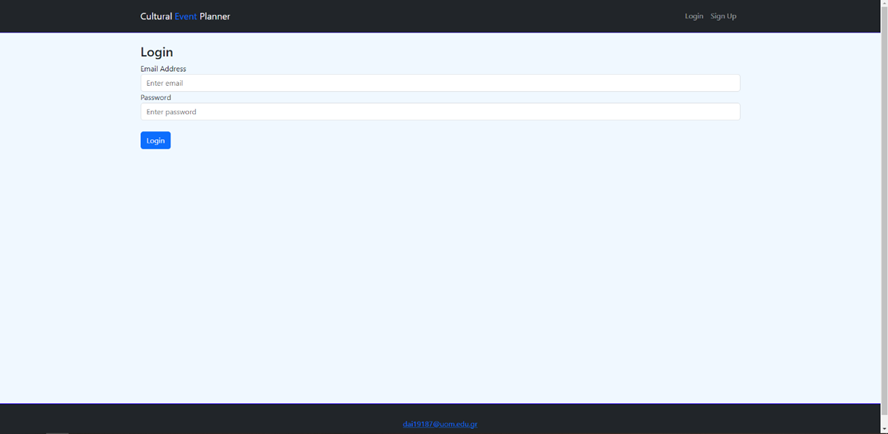
Εικόνα “Welcome screen

### Sign up

****

Εικόνα "Sign up"

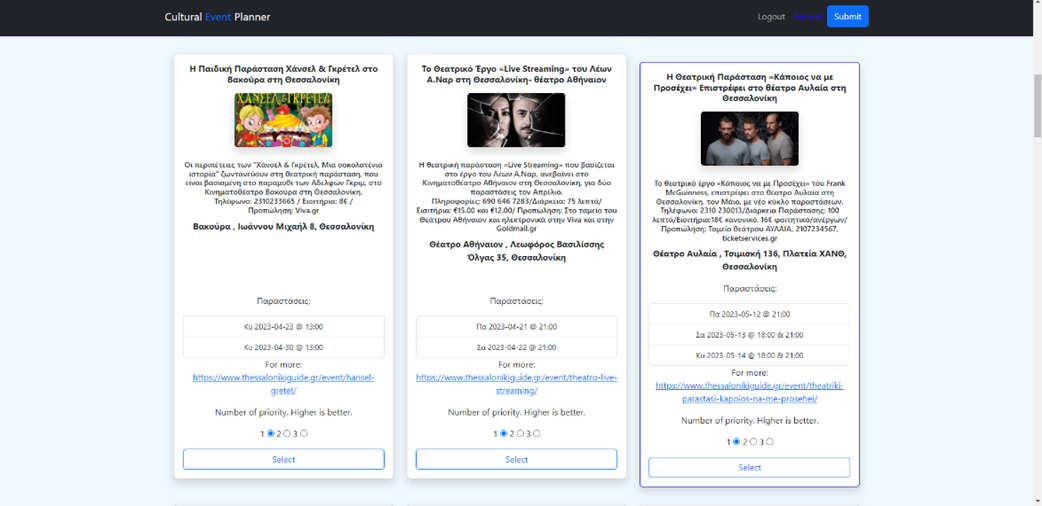
### Login

****

Εικόνα "Login"

Αφού συνδεθεί, και η βάση έχει τροφοδοτηθεί με τα απαραίτητα στοιχεία των παραστάσεων, το home screen είναι εκεί που δίνεται η δυνατότητα στον χρήστη να επιλέξει, με το κουμπί **select**, τις παραστάσεις όπου θέλει να παραβρεθεί.

### Home

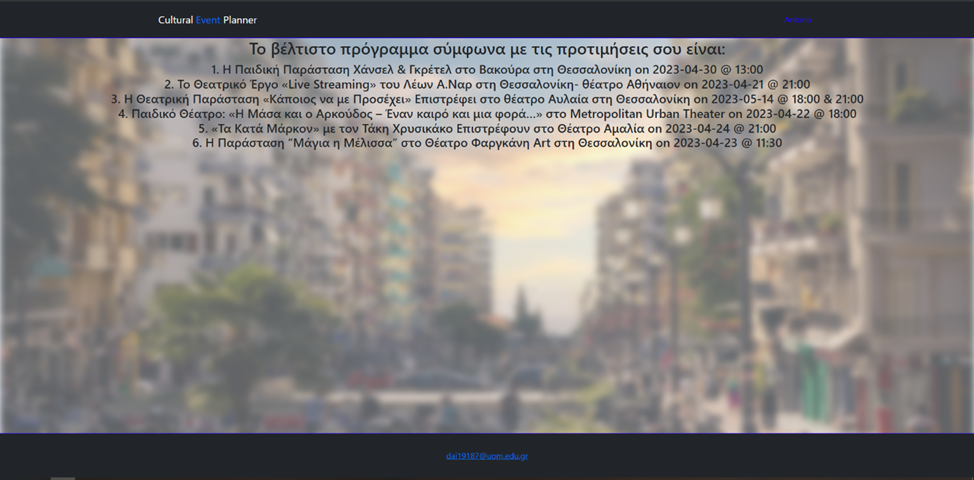


Εικόνα "Home screen"

Έτσι επιπρόσθετα έχουμε, το navigation bar και το σύνολο των εκδηλώσεων. Το **nav-bar** είναι κατασκευασμένο στο αρχείο base.html το οποίο με την λειτουργία της κληρονομικότητας που παρέχεται από το **Jinja** μπορεί και εμφανίζεται στο home. Αφού επιλεχθούν οι κατάλληλες παραστάσεις και το αντίστοιχο priority (από 1 έως 3) το κουμπί **Submit** είναι εκείνο που θα στείλει αντίστοιχο μήνυμα στον διακομιστή για να αρχίσει την αναζήτηση του βέλτιστου χρονοπρογράμματος των παραστάσεων.

Έτσι τελικά γίνεται και η εμφάνιση του αποτελέσματος στην οθόνη result παρέχοντας στον χρήστη την τελική λύση. Να σημειωθεί ότι αυτό αποθηκεύεται στην βάση οπότε ο χρήστης το έχει στην διάθεση του ανά πάσα στιγμή.

### Η οθόνη του αποτελέσματος



Εικόνα 6 "The result"

Με το “Cultural Event Planner” ο χρήστης επιστρέφει στην αρχική οθόνη οπότε η διαδικασία μπορεί να επαναληφθεί.

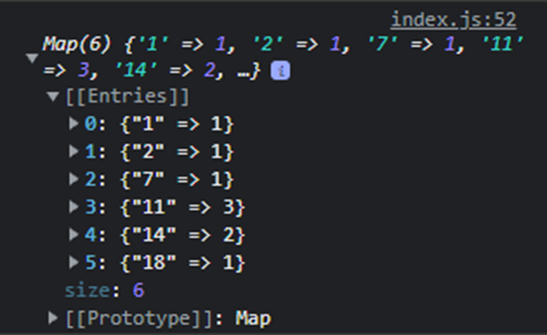
# ΣΧΕΔΙΑΣΗ ΚΩΔΙΚΑ ΠΕΛΑΤΗ – index.js

Πέραν του φακέλου με τα αρχεία σήμανσης html, αυτό που αποδίδει την λειτουργικότητα των κουμπιών που υπάρχουν στο σύστημα είναι το **index.js**. Ο κώδικας που τρέχει από τον πελάτη και είναι υπεύθυνος για λειτουργίες όπως μεταφορά των επιλεγμένων εκδηλώσεων από τον πελάτη στον διακομιστή, εμφάνιση κατάλληλων μηνυμάτων στην κονσόλα, όπως και άλλα. Η βιβλιοθήκη του JQuery μπορεί και αποδίδει λειτουργίες που χρησιμοποιούνται στην συνέχεια. [31]

## SELECT button

Αρχικοποίηση της κάθε κάρτας παράστασης με ένα id έχει σαν αποτέλεσμα και το κάθε αντίστοιχο κουμπί **select** να έχει εκείνο το id. Έτσι λοιπόν, ο κώδικας JavaScript αντλεί όλα τα id των κουμπιών select. Τα id τα οποία κατ’ επέκταση ανήκουν στα id των παραστάσεων της βάσης. Αφού πατηθεί κάποιο από αυτά το index.js παίρνει το id αυτού του κουμπιού καθώς και το priority (1-3) που υπάρχουν στην καρτέλα της εκάστοτε παράστασης και τα αποδίδει σε μία μεταβλητή Map.

Ο ‘Χάρτης’ στον κώδικα του πελάτη αποτελεί μια δομή δεδομένων όπου κάθε στοιχείο αποτελείται από ένα κλειδί(key) αντιστοιχισμένο σε μία τιμή(value). Το κλειδί θα πρέπει να είναι μοναδικό ενώ η τιμή όχι. Έτσι, όταν πατηθεί το κουμπί select ο κώδικας αποθηκεύει τον συγκεκριμένο συνδυασμό που δημιουργήθηκε με id,priority\_number. Το key ‘δείχνει’ σε ένα value που παίρνει τιμές από 1 έως 3. Αυτές οι τιμές αντιπροσωπεύουν το priority που δίνει ο χρήστης στην συγκεκριμένη παράσταση. Όσο πιο μεγάλο τόσο μεγαλύτερη είναι η πιθανότητα να υπάρχει στο αποτέλεσμα. Αυτό έγινε με σκοπό την προσθήκη έμφασης σε κάποιες παραστάσεις έτσι ώστε Αν μία ημερομηνία μιας παράστασης συμπίπτει με μία άλλη να επιλεχθεί εκείνη με το υψηλότερο priority.



Εικόνα “Console log map”

Τέλος, για λόγους λειτουργικότητας και χρηστικότητας το index.js όταν πατηθεί ένα κουμπί αλλάζει το εσωτερικό του text από select σε selected, βοηθώντας τον χρήστη να γνωρίζει ποιες παραστάσεις έχει επιλέξει. Αν εκείνο το κουμπί ξανά πατηθεί το σύστημα διαγράφει την επιλογή του χρήστη και το text αλλάζει πάλι σε select.

## Submit button

Ο κώδικας index.js διακατέχει την λειτουργία για ακόμη ένα ακόμα κουμπί, το **Submit**. Εκείνο που θα στείλει τον πίνακα map στον διακομιστή για να μπορέσει να επεξεργαστεί. Με την κατάλληλη μέθοδο το index.js έχει ένα function “sendmaptoServer()”. Αυτή η μέθοδος μέσω request κάνει Post στην σελίδα του result τον πίνακα map. Η αποστολή γίνεται κάνοντας μετατροπή με σειριοποίηση (**serialize**) του χάρτη σε δομή JSON. Οπότε τελικά, το σύστημα όταν βρεθεί στην σελίδα result δέχεται ένα αίτημα POST από το index με τον πίνακα JSON[map], ενώ έχει σαν σκοπό την μετατροπή της σειριοποιημένης μεταβλητής. [32]



Εικόνα “SendMapToServer”

# Ανάλυση αρχιτεκτονικής συστήματος

Στην κατανόηση του τρόπου λειτουργίας της εφαρμογής η ανάλυση της βασικής αρχιτεκτονικής είναι αναγκαία. Συντελεί απαραίτητο κομμάτι στην ανάπτυξη λογισμικού βοηθώντας τον προγραμματιστή να έχει ένα πρότυπο που θα ακολουθήσει. Η διάταξη που εφαρμόζεται στην εκάστοτε εφαρμογή είναι αυτή που καθορίζει την μεταφορά των δεδομένων, την σύνδεση των διάφορων κομματιών του κώδικα, την διατήρηση της συνέχειας στην λειτουργία της εφαρμογής. Στον κόσμο των διαδικτυακών εφαρμογών οι κατασκευαστικές μέθοδοι ποικίλουν, οπότε ανάλογα με τις απαιτήσεις του κάθε σεναρίου η επιλογή τους διαφέρει.

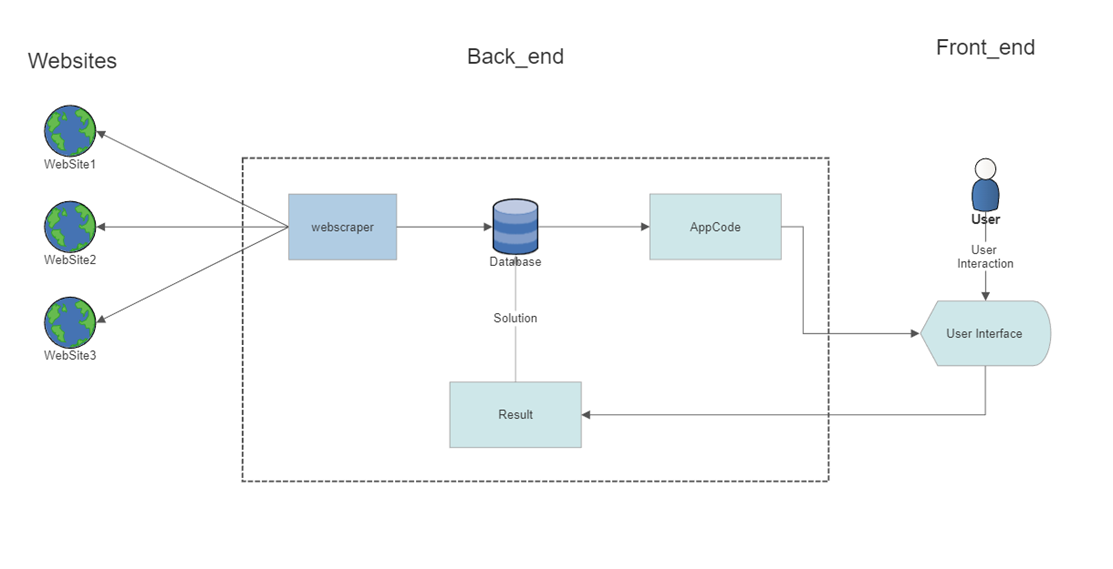
## Αρχιτεκτονική πελάτη-διακομιστή

Το πρόγραμμα χωρίζεται σε δύο ομάδες δρομολογήσεων αυτή που βλέπει ο χρήστης μέσω του φυλλομετρητή (front\_end) και αυτή που δεν βλέπει, αλλά είναι υπεύθυνη για την διαχείριση των δεδομένων (back\_end) . Αυτού του τύπου μοντέλο ονομάζεται μοντέλο του **πελάτη-διακομιστή.** [33][34]

Η **αρχιτεκτονική πελάτη-διακομιστή** **(client-server architecture)** αναφέρεται σε ένα σύστημα που φιλοξενεί, παραδίδει και διαχειρίζεται τους περισσότερους πόρους και υπηρεσίες που ζητά ο πελάτης. Σε αυτό το μοντέλο, όλα τα αιτήματα και οι υπηρεσίες παραδίδονται μέσω δικτύου, ενώ αναφέρεται επίσης ως μοντέλο υπολογιστών δικτύου ή δίκτυο διακομιστή πελάτη. Επιπρόσθετα, διαχωρίζει εργασίες και φόρτους εργασίας μεταξύ πελατών και διακομιστών που βρίσκονται στο ίδιο σύστημα ή συνδέονται σε ένα δίκτυο υπολογιστών. Τυπικά διαθέτει σταθμούς εργασίας πολλαπλών χρηστών, υπολογιστές ή άλλες συσκευές, συνδεδεμένους σε έναν κεντρικό διακομιστή μέσω σύνδεσης στο Διαδίκτυο ή άλλου δικτύου. Συνοπτικά, ο πελάτης στέλνει το αίτημά **HTTP** του μέσω μιας συσκευής συνδεδεμένη στο δίκτυο ενώ στη συνέχεια ο διακομιστής δικτύου αποδέχεται και επεξεργάζεται το αίτημα του χρήστη. Τελικά, ο διακομιστής παραδίδει την απάντηση στον πελάτη.

## ΓΕΝΙΚΗ ΔΟΜΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η πρώτη λειτουργία που πρέπει να εκτελεστεί από το πρόγραμμα είναι η τροφοδότηση της βάσης με τις διαθέσιμες παραστάσεις. Εκείνες, δηλαδή, που μόλις συνδεθεί, ο χρήστης, θα μπορεί να δει και να επιλέξει. Αυτό γίνεται με τον κώδικα του webscraper.py. Από επιλεγμένη ιστοσελίδα με παραστάσεις και με την μέθοδο της απόξεσης ιστού, τα δεδομένα συλλέγονται και τοποθετούνται στην βάση. Η βάση έχει συγκεκριμένη δομή και κάθε στοιχείο ανήκει σε ένα πεδίο εξυπηρετώντας κάποιον σκοπό. Αφού τελειώσει η άντληση των παραμέτρων της παράστασης (Όνομα, ημερομηνία, τοποθεσία, URL εικόνας, κ.α.) τα υπόλοιπα κομμάτια του κώδικα αναλαμβάνουν για την δομημένη εμφάνιση των εκδηλώσεων. Εκεί είναι που ο χρήστης μπορεί να αλληλοεπιδράσει ουσιαστικά με το σύστημα και να κάνει τις επιλογές του διαλέγοντας τις παραστάσεις που θέλει να παρακολουθήσει. Στη συνέχεια, οι επιλογές αυτές πάλι μεταφέρονται στο σημείο της επεξεργασίας από το back\_end για την αναζήτηση του βέλτιστου αποτελέσματος και τελικά στην λύση του προβλήματος του χρονοπρογραμματισμού. Έτσι, αφού αποθηκευτεί σε κάποιον πίνακα, το τελικό αποτέλεσμα με την κατάλληλη σύνδεση back\_end και front\_end εμφανίζεται στον χρήστη. Τέλος, υπάρχει η δυνατότητα να το ξαναδεί αφού «μπει» στο προφίλ του.

Παρακάτω στην εικόνα 2 απεικονίζεται η αρχιτεκτονική (client-server) της εφαρμογής.

Εικόνα "Αρχιτεκτονική συστήματος"

# Σχεδίαση Βάσης Δεδομένων

Η βάση των δεδομένων αποτελεί αναπόσπαστο κομμάτι για την ανάπτυξη του προγράμματος. Εκεί είναι όπου αποθηκεύονται όλα τα δεδομένα με σκοπό την επεξεργασία ή την εμφάνιση τους. Η σωστή **μοντελοποίηση** μιας τέτοιας οντότητας έχει βασικούς κανόνες και όρια. Μια σωστά **σχεδιασμένη** **βάση δεδομένων** παρέχει πρόσβαση σε ενημερωμένη και ακριβή πληροφορία. Επειδή μια σωστή σχεδίαση είναι απαραίτητη για την επίτευξη των στόχων της εφαρμογής με μια βάση δεδομένων, έχει νόημα να επενδυθεί ο κατάλληλος χρόνος. Θα πρέπει να καταλήξουμε με μια βάση δεδομένων που ανταποκρίνεται στις ανάγκες και μπορεί εύκολα να εξυπηρετεί την κάθε ανάγκη.

Η διατήρησή κατάλληλων πινάκων, αρχικά, είναι ένας από τους πρωτεύον στόχους της σχεδίασης. Τι και πόσοι πίνακες θα χρησιμοποιηθούν. Παράλληλα πρέπει να γίνεται ο σωστός ορισμός των πεδίων όπως ο καθορισμός των πρωτεύων κλειδιών των πινάκων που χρησιμοποιούνται. Σίγουρα οι διπλότυπες πληροφορίες (ονομάζονται επίσης περιττά δεδομένα) είναι περιττές αφού σπαταλούν χώρο και αυξάνουν την πιθανότητα σφαλμάτων και ασυνεπειών. Επιπλέον, πρέπει να γίνεται πλήρης χρήση των πληροφοριών. Αυτά και αρκετά ακόμη αποτελούν τα χαρακτηριστικά που συντελούν μία καλά σχεδιασμένη Βάση Δεδομένων.

Άλλοι κανόνες που ακολουθήθηκαν για την σχεδίαση της Βάσης είναι να:

1. Διαιρεί τις πληροφορίες σας σε πίνακες βάσει θέματος για να μειώσει τα πλεονάζοντα δεδομένα.

2. Παρέχει πρόσβαση στις πληροφορίες που απαιτούνται για την ένωση των δεδομένων των πινάκων, εάν χρειάζεται.

3. Βοηθά στην υποστήριξη και τη διασφάλιση της ακρίβειας και της ακεραιότητας των πληροφοριών.

4. Εξυπηρετεί τις ανάγκες επεξεργασίας και αναφοράς δεδομένων. [35]

## Παρουσίαση των πινάκων της βάσης

Με γνώμονα τα παραπάνω η τελική μορφή της Βάσης είναι αυτή που χρησιμοποιείται και στην εφαρμογή. Υπάρχουν πίνακες που εξυπηρετούν διαφορετικούς σκοπούς. Οπότε στην επεξήγηση των δομών αυτών έχουμε:

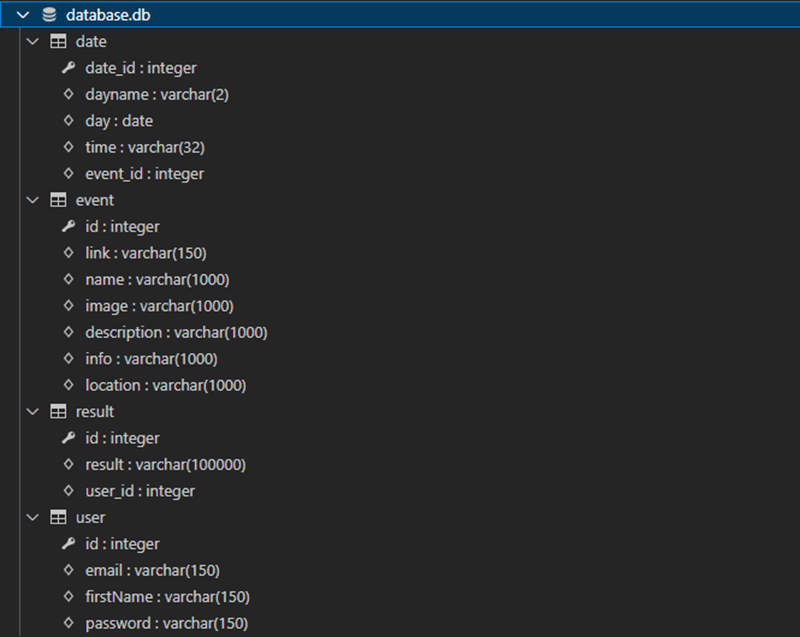
Αρχικά, ο πίνακας **User** ο οποίος διατηρεί το σύνολο των χρηστών που συνδέθηκαν στην εφαρμογή. Κάθε User αποτελείται από ένα πρωτεύον κλειδί id, email, όνομα username, κωδικό (ο οποίος κρυπτογραφείται από το σύστημα με την χρήση της μεθόδου sha256).

Ο πίνακας **Event** διακατέχει όλες τις παραστάσεις που αντλούνται από τις ιστοσελίδες. Είναι χωρισμένος σε πεδία : id(key), link, name, image, description, info, location.

Ο πίνακας **Date** ο οποίος έχει τις ημερομηνίες της κάθε παράστασης. Ο συγκεκριμένος πίνακας είναι συσχετισμένος και εξαρτώμενος από τον πίνακα Event και έχει διαχωριστεί για πρακτικούς λόγους. Τα πεδία του date\_id, dayname, day, time, event\_id. Κάθε Event μπορεί να έχει πολλά Date αλλά κάθε Date να ανήκει σε μόνο ένα Event.

Τέλος, ο πίνακας **Result** είναι αυτός που αποθηκεύεται το τελικό αποτέλεσμα. Πεδία id,result, user\_id(δευτερεύον κλειδί). Είναι εξαρτημένος με τον πίνακα User αφού κάθε user αποθηκεύει ένα αποτέλεσμα σε περίπτωση που θέλει να το δει σε κάποια άλλη χρονική στιγμή.

Στη βάση **database.db** (η οποία βρίσκεται στον υποφάκελο /instance) γράφονται SQL ερωτήματα μέσω της FLASK\_SQLALCHEMY υπεύθυνη για την σύνδεση του προγράμματος με την βάση. Έχει λειτουργίες όπως η δημιουργία πινάκων, διαγραφή στοιχείων και πολλά ακόμη μόνο με κώδικα python, ικανότητα πολύ χρηστική για τον προγραμματιστή. Τέλος, και η μοντελοποίηση έγινε σε python αρχείο (models.py).



Εικόνα 10 "Οι πίνακες της βάσης δεδομένων"

## Δομή αρχείων

Ο πηγαίος κώδικας με βάση την αρχιτεκτονική του συστήματος χωρίζεται σε υποφακέλους. Κάθε ένας από αυτούς χωρίζει τις διαθέσιμες λειτουργίες με σκοπό την διευκόλυνση ανάλυσης του κώδικα. Πιο συγκεκριμένα, στο αρχικό Pathing βλέπουμε :

Τα **αρχεία**:

· **Main.py**: Υπεύθυνο για να ξεκινάει τον (test) **server** του **flask** και να αρχίσει να τρέχει την εφαρμογή.

· **README.md**: Αποτελεί μία περιγραφική επεξήγηση του προγράμματος, σε περίπτωση που κάποιος ενδιαφερόμενος θέλει να αναλύσει τον κώδικα.

· **Requirements.txt**: Το αρχείο με τις απαραίτητες βιβλιοθήκες που χρησιμοποιεί ο κώδικας ώστε να λειτουργήσει το πρόγραμμα.

· **blank.txt** : Δοκιμαστικό αρχείο

Έπειτα υπάρχουν οι **φάκελοι** που συντελούν τον ουσιαστικό **πηγαίο κώδικα**:

· **/Instance:** Η κεντρική αποθήκη, η Βάση που χρησιμοποιείται (**database.db**). Χωρισμένη από το υπόλοιπο πρόγραμμα αφού αποτελεί μία διαφορετική οντότητα για το σύστημα.

· **/Website**

## φάκελος /website:

Εδώ είναι το σημείο που υπάρχει το μεγαλύτερο κομμάτι του κώδικα τόσο για το **back\_end** όσο και για το **front\_end.** Θέτοντας βασικό θεμέλιο για την δομή του συστήματος ώστε να υπάρχει μία λογική συνέχεια του κώδικα. Αρχικά, υπάρχουν τα βασικά συστατικά (**components**) που φέρουν εις πέρας τις κύριες λειτουργίες του συστήματος. Λειτουργίες όπως: η άντληση των δεδομένων, η αυθεντικοποίηση του χρήστη, η κατασκευή προτύπων URL για την χρήση των συνδέσμων, η μοντελοποίηση της Βάσης και, φυσικά, η επίλυση του προβλήματος χρονοπρογραμματισμού.

Στην ουσία, το back\_end ως πληροφοριακό σύστημα έχει σκοπό την τροφοδότηση των δεδομένων προς το κομμάτι του χρήστη .Έτσι, με την παρούσα αρχιτεκτονική, και τη χρήση του **Flask** επιτυγχάνεται η διαχείριση και μεταφορά των δεδομένων μεταξύ κώδικα, όπως και η διαχείριση των αιτημάτων που προκύπτουν ανά σελίδα.

# ΥΛΟΠΟΊΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΛΥΣΗ

Όσων αφορά τις λεπτομέρειες σχετικά με τον κώδικα που υλοποιεί την λύση του προβλήματος. Η υλοποίηση περιλαμβάνει μεθόδους για την σωστή διαχείριση των http αιτημάτων, την ανάγνωση και διαχωρισμό των επιλεγμένων παραστάσεων, την μοντελοποίηση του μοντέλου της λύσης αλλά και την τελική βέλτιστη απάντηση.

## ΜΑΘΗΜΑΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Για να περιγράψουμε τα παραπάνω με πιο συγκεκριμένο και δόκιμο τρόπο η μαθηματική ανάλυση του μοντέλου του προβλήματος θα αποτελέσει το μεταβατικό μέσω από το θεωρητικό υπόβαθρο σε μία πιο πρακτική προσέγγιση. Αναλυτικότερα το πρόβλημα εξηγείται ως ένα σύνολο περιορισμών, συντελεστών και αθροισμάτων στην εφαρμογή ενός μαθηματικού μοντέλου γραμμικού προγραμματισμού.

## ΜΕΤΑΒΛΗΤΕΣ

Ας υποθέσουμε ότι έχουμε ένα σύνολο γεγονότων X = {1, 2, ..., n} και ένα σύνολο ημερομηνιών D = {1, 2, ..., m}. Ορίζοντας δύο δυαδικές μεταβλητές απόφασης, οι οποίες δηλαδή λαμβάνουν τιμές 0 ή 1 εάν υπάρχουν στο αποτέλεσμα ή όχι. Η πρώτη μεταβλητή αποτελεί το σύνολο των όλων των διαθέσιμων ημερομηνιών, D[i] είναι αυτή που θα χρησιμοποιηθεί αργότερα για τον περιορισμό της απαίτησης της διαφοράς τιμής των ημερομηνιών. Επίσης, εννοείται ότι έχει απαιτηθεί εξαρχής η D να είναι πάντα ‘>=’ της συγκεκριμένης ημερομηνίας που εκτελείται το πρόγραμμα. Σαν δεύτερη μεταβλητή, ας πούμε E, ορίζουμε τον συνδυασμό των διαθέσιμων παραστάσεων και ημερομηνιών που τους ανήκουν. Ή αλλιώς Ε[i][j]= [X[i],D[j]] . Μεταβλητή η οποία αξιοποιείται τόσο στο κομμάτι των περιορισμών όσο και στην εύρεση της αντικειμενικής συνάρτησης. Τέλος υπάρχει και μία συντελεστής βαρύτητας, αναφέρεται ως priority P, και χρησιμοποιείται για την απόδοση προτεραιότητας στις παραστάσεις. Το σύνολο των τιμών του priority κυμαίνεται από 1 έως 3.

## Αντικειμενική συνάρτηση

Η αντικειμενική συνάρτηση είναι να μεγιστοποιήσει τον συνολικό αριθμό των συμβάντων που παρακολουθεί ο χρήστης. Αυτό μπορεί να αναπαρασταθεί ως:

Μεγιστοποίηση της συνάρτησης :

Z = \* p

## Περιορισμοί

Η παραπάνω συνάρτηση υπόκεινται στους ακόλουθους περιορισμούς:

1. Κάθε εκδήλωση μπορεί να προγραμματιστεί μόνο μία φορά:

<= 1 ,για όλα τα i=1,...,n

2. Κάθε ημερομηνία μπορεί να φιλοξενήσει μόνο ένα συμβάν:

<= 1 ,για όλα τα j=1,...,m

## Προτεραιότητα παραστάσεων

Ορισμένα συμβάντα μπορεί να έχουν προτεραιότητα έναντι άλλων. Εάν ένα συμβάν i έχει υψηλότερη προτεραιότητα από ένα άλλο συμβάν k, και έρχεται σε σύγκρουση χρονικά, τότε θα πρέπει να προγραμματιστεί στο χρονικό διάστημα:

E[i][j] >= E[k][j] ,για όλα τα j=1...m

Το priority συγκαταλέγεται στην αντικειμενική συνάρτηση δίνοντας ‘βαρύτητα’ στις εκδηλώσεις που έχουν τον μεγαλύτερο αριθμό προτεραιότητας.

## Εν κατακλείδι

Οι παραπάνω περιορισμοί διασφαλίζουν ότι κάθε συμβάν προγραμματίζεται ακριβώς μία φορά, κάθε χρονική περίοδος καταλαμβάνεται από το πολύ ένα συμβάν και ότι τα συμβάντα υψηλότερης προτεραιότητας προγραμματίζονται έναντι συμβάντων χαμηλότερης προτεραιότητας σε περίπτωση σύγκρουσης. Η αντικειμενική συνάρτηση διασφαλίζει ότι ο συνολικός αριθμός των συμβάντων που παρακολουθεί ο χρήστης μεγιστοποιείται. Η βέλτιστη λύση που λαμβάνεται από την επίλυση γραμμικού προγραμματισμού μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός χρονοδιαγράμματος συμβάντων που θα παρακολουθήσει ο χρήστης.

Αυτό το πρόβλημα μπορεί να λυθεί χρησιμοποιώντας μια ποικιλία λύσεων γραμμικού προγραμματισμού, όπως το IBM ILOG CPLEX Optimization Studio ή το GNU Linear Programming Kit (GLPK). [36] [37] [38]

Ωστόσο με το **Google OR-Tools** και συγκεκριμένα ο solver **CBC\_MIXED\_INTEGER\_PROGRAMMING** της βιβλιοθήκης **pywraplp** είναι εξίσου κατάλληλα εργαλεία για τέτοιου είδους λύσεις. [39] [40]

# Λήψη και διαχείριση επιλεγμένων παραστάσεων

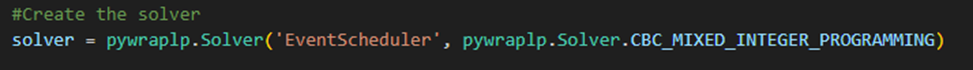
Όπως έχει προαναφερθεί τα σύστημα, αφού γίνει η επιλογή των παραστάσεων, κώδικα με σκοπό της τροφοδότησης του μοντέλου επίλυσης. Συγκεκριμένα, Post request στέλνονται από την κεντρική σελίδα των παραστάσεων στην σελίδα της εμφάνισης του τελικού αποτελέσματος. Μαζί με αυτά, ο Χάρτης Map που έχει δημιουργηθεί από τον κώδικα του πελάτη, το index.js δηλαδή, και ουσιαστικά είναι αυτός που εμπεριέχει αυτά τα δεδομένα. Έτσι στην συνέχεια, αφού ο κώδικας python του back\_end (result.py) δεχθεί τον χάρτη, μετά την διαχείριση του Http αιτήματος, βρίσκεται στην κατάλληλη θέση για την επαρκή παροχή δεδομένων στους πίνακες που θα συσχετιστούν με τις μεταβλητές του προβλήματος.

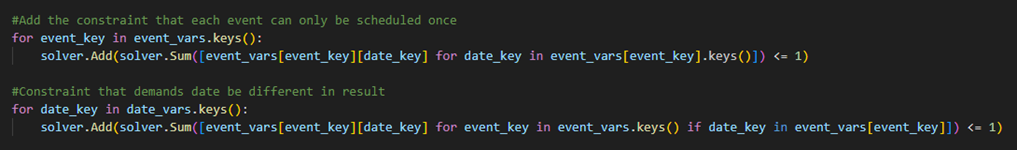
Πρακτικά, κατασκευάζονται δύο πίνακες με την χρήση του χάρτη: ο **selected\_array**, δηλαδή οι παραστάσεις που επιλέχθηκαν, (Παράδειγμα: event\_1, event\_3, event\_20,…) και ο **Date\_array**, δηλαδή οι ημερομηνίες των παραστάσεων αυτών. Ο τελευταίος είναι πίνακας εξαρτώμενος που συσχετίζεται μέσω event\_id.

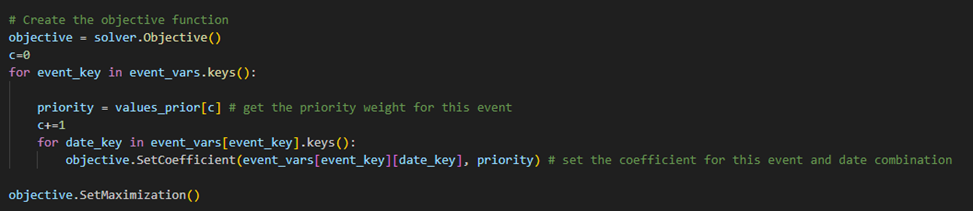
Άρα τελικά οι πίνακες αυτοί συσχετίζονται με τις μεταβλητές του προβλήματος **event\_vars**, **date\_vars** οι οποίες ουσιαστικά διατηρούν τα δεδομένα εκείνα που είναι χρήσιμα για την τροφοδότηση του προβλήματος. Ο κώδικας δημιουργεί μεταβλητές απόφασης για κάθε συνδυασμό συμβάντος, ημερομηνίας και το πρόγραμμα περνάει στο τελικό στάδιο κατασκευής και λύσης του ζητήματος.

# Επίλυση μέσω κώδικα

Ο στόχος του κώδικα είναι να βρει ένα χρονοδιάγραμμα που αντιστοιχίζει κάθε συμβάν σε μία μόνο ημερομηνία, με την επιφύλαξη ορισμένων περιορισμών. Πρωταρχικά, ο solver αποτελεί το εργαλείο που είναι υπεύθυνο για την ένταξη περιορισμών και την βελτιστοποίηση της αντικειμενικής συνάρτησης. Σαν solver χρησιμοποιείται από την βιβλιοθήκη του Google OR-Tools ο CBC\_MIXED\_INTEGER\_PROGRAMMING. COIN-OR Branch-and-Cut solver πιο αναλυτικά, κατάλληλος για λύση προβλημάτων γραμμικού προγραμματισμού, έχοντας πολλαπλές εφαρμογές σε πολλαπλά συστήματα μοντελοποίησης, πακέτα κ.λπ. Στην συγκεκριμένη περίπτωση περιλαμβάνει όλες τις απαραίτητες μεθόδους που απαιτούνται για την αναζήτηση της βέλτιστης λύσης με περιορισμούς.

 Τα βασικότερα βήματα για την επίλυση του προβλήματος χρονοπρογραμματισμού των παραστάσεων. Πρώτον γίνεται η απόδοση περιορισμών, όντας πρόβλημα περιορισμών. Σαν πρώτος περιορισμός αποτελεί ότι κάθε event μπορεί να υπάρχει το **πολύ μία φορά** στην απάντηση. Μέσα σε ένα for loop, λοιπόν, που αποδίδει τιμές βάσει των κλειδιών των επιλεγμένων παραστάσεων της μεταβλητής event\_vars. Με άλλα λόγια για κάθε event που ανήκει στο event\_vars.keys(), εντάσσουμε στον solver το άθροισμα των παραστάσεων για κάθε πιθανή ημερομηνία, έχοντας την απαίτηση να μην υπερβαίνει το 1. Εάν το άθροισμα είναι 1, σημαίνει ότι το συμβάν έχει εκχωρηθεί σε μία ακριβώς ημερομηνία. Εάν είναι 0, σημαίνει ότι το συμβάν δεν έχει εκχωρηθεί σε καμία ημερομηνία. εάν είναι μεγαλύτερο από 1, σημαίνει ότι το συμβάν έχει εκχωρηθεί σε πολλές ημερομηνίες, γεγονός που παραβιάζει τον περιορισμό. Το δεύτερο σύνολο περιορισμών διασφαλίζει ότι σε κάθε ημερομηνία έχει εκχωρηθεί το πολύ ένα συμβάν. Αυτό γίνεται αθροίζοντας τις μεταβλητές απόφασης για κάθε ημερομηνία και διασφαλίζοντας ότι το άθροισμα είναι μικρότερο ή ίσο με 1. Εάν το άθροισμα είναι 1, σημαίνει ότι υπάρχει ακριβώς ένα γεγονός που έχει εκχωρηθεί στην ημερομηνία. Εάν είναι 0, σημαίνει ότι δεν υπάρχουν συμβάντα που έχουν εκχωρηθεί στην ημερομηνία. Εάν είναι μεγαλύτερο από 1, σημαίνει ότι έχουν εκχωρηθεί πολλά συμβάντα στην ημερομηνία, γεγονός που παραβιάζει τον περιορισμό.

 Τέλος, ο κώδικας ορίζει μια αντικειμενική συνάρτηση που μεγιστοποιεί τον αριθμό των συμβάντων που έχουν εκχωρηθεί σε μια ημερομηνία. Αυτό γίνεται προσθέτοντας τις μεταβλητές απόφασης για κάθε συνδυασμό συμβάντος και ημερομηνίας και μεγιστοποιώντας το άθροισμα. Αυτό διασφαλίζει ότι ο λύτης θα δώσει προτεραιότητα στην ανάθεση συμβάντων σε ημερομηνίες όπου είναι δυνατόν.

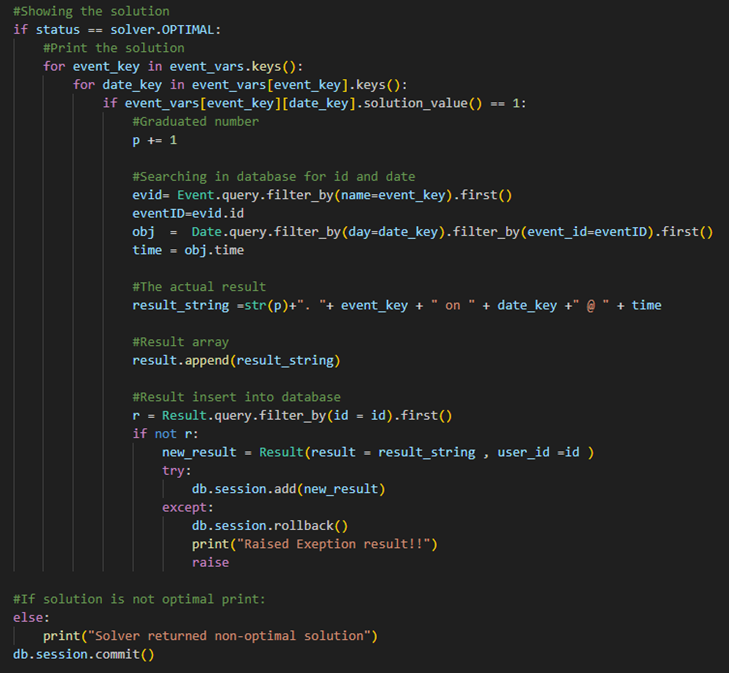
 Όσων αφορά την προτεραιότητα των παραστάσεων η μέθοδος SetCoefficient() είναι μια μέθοδος που χρησιμοποιείται στη μαθηματική βελτιστοποίηση και τον γραμμικό προγραμματισμό, που συνήθως εφαρμόζεται σε βιβλιοθήκες λύσεων όπως το Google OR-Tools ή το PuLP. Ορίζει τον συντελεστή μιας μεταβλητής απόφασης σε έναν γραμμικό περιορισμό, ορίζοντας τη σχέση μεταξύ των μεταβλητών απόφασης και της αντικειμενικής συνάρτησης. Με άλλα λόγια, η SetCoefficient() καθορίζει το βάρος μιας μεταβλητής απόφασης σε μια εξίσωση περιορισμού, η οποία χρησιμοποιείται για τον υπολογισμό της αντικειμενικής τιμής του προβλήματος βελτιστοποίησης. Επιτρέπει στον χρήστη να ορίσει τους συντελεστές των μεταβλητών απόφασης σε ένα πρόβλημα μαθηματικού προγραμματισμού με ευέλικτο και αποτελεσματικό τρόπο. Μόλις οριστούν οι περιορισμοί και η αντικειμενική συνάρτηση, ο κώδικας λύνει το πρόβλημα χρησιμοποιώντας έναν λύτη. [41]



Εάν βρεθεί μια βέλτιστη λύση, ο κωδικός εκτυπώνει το χρονοδιάγραμμα, υποδεικνύοντας ποιο συμβάν έχει εκχωρηθεί σε ποια ημερομηνία. Εάν δεν βρεθεί η βέλτιστη λύση, ο κώδικας εκτυπώνει ένα μήνυμα σφάλματος.

# Τελική απάντηση

Για την εμφάνιση του αποτελέσματος το κύριο κριτήριο είναι ένα, να βρεθεί η βέλτιστη λύση , αν βρεθεί, γίνεται η αποθήκευση αυτής ως ένα σύνολο string στην βάση των δεδομένων. Κάτι το οποίο εξυπηρετεί σκοπό για την καλύτερη εμπειρία του χρήστη. Επεξηγώντας, αφού αποθηκεύεται το αποτέλεσμα, ο πελάτης το έχει στην διάθεση του τόσο κατά την στιγμιαία παρουσίαση του όσο και σε κάποιο δεύτερο χρόνο. Υπάρχει λοιπόν στην διάθεση σου της εισαγωγής στο προφίλ του, σε οποιαδήποτε φάση, ώστε να δει το διαθέσιμο του πρόγραμμα. Η λύση αναλυτικότερα, δημιουργείται όταν ο solver την κρίνει βέλτιστη (optimal) status == solver.OPTIMAL και εν συνεχεία τα δεδομένα εντάσσονται στην βάση μόνο όταν το solution\_value ισούται με 1. Όταν δηλαδή η μεταβλητή απόφασης του συνδυασμού παράσταση-ημερομηνία θεωρείται αποδεκτή. Έτσι τελικά θα παρουσιαστεί ένα σύνολο συνδιασμών (παράσταση – ημερομηνία – ώρα) σε μία σειρά. Πάντα διαφορετικές ημερομηνίες για τις διαφορετικές παραστάσεις. Ο χρήστης είναι πλέον σε θέση να ακολουθήσει το πρόγραμμα που του προέκυψε ως αποτέλεσμα των επιλογών του.



# ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΠΡΟΕΚΥΨΑΝ ΚΑΤΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Στη ανάπτυξη ενός συστήματος τα προβλήματα που μπορούν να προκύψουν είναι πολλά. Όταν υπάρχει ένας προκαθορισμένος στόχος η επίτευξη αυτού προϋποθέτει αρκετά πράγματα. Γνώσεις, δοκιμές, επαναλήψεις εκτέλεση κώδικα και γενικά αφιέρωση αρκετού χρόνου για την εμπέδωση ενός αποδεκτού αποτελέσματος. Έχοντας λοιπόν ως βασικό σκοπό την επίλυση ενός προβλήματος με το εργαλείο του Google Or-Tools ως πρωταρχικό πρόβλημα που προκύπτει είναι η διαδικασία της απόφασης για την μοντελοποίηση του προβλήματος. Πώς δηλαδή, ενώ ακόμα η κατασκευή του κώδικα δεν έχει αρχίσει καν, θα χρησιμοποιηθούν τα κατάλληλα εργαλεία με την κατάλληλη σειρά ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα. Πρόβλημα λοιπόν στην επιλογή των τεχνικών και την σκοπιά της προσέγγισης της λύσης ώστε με τα δεδομένα που δίνονται να επιστρέφεται ένα θεμιτό αποτέλεσμα.

Αφού αποφασιστεί ένας βασικός γνώμονας για την υλοποίηση, προκύπτουν θέματα όσων αφορά την δομή των δεδομένων και την διατήρηση της ακεραιότητας. Τι και ποια από τα δεδομένα της βάσης που διακατέχουμε είμαστε σε θέση να χρησιμοποιήσουμε. Τι ουσιαστικά χρειάζεται και τι όχι, ενώ η διαμόρφωση αυτών ίσως μπορέσει να βοηθήσει περαιτέρω ή και να προκαλέσει περισσότερα προβλήματα. Πρακτικά, η ένταξη και η μεταφορά των μεταβλητών από το ένα κομμάτι του συστήματος σε ένα άλλο μπορεί να δημιουργήσει θέματα αφού όταν εμπλέκονται διαφορετικά στοιχεία και γλώσσες κώδικα θα πρέπει, ό,τι γίνεται να γίνεται με μεγάλη προσοχή. Αλλιώς, errors προκύπτουν αποτρέποντας την σωστή, λειτουργική απόδοση του συστήματος. Όταν για παράδειγμα κάποια από τα δεδομένα μεταφέρονται από το front\_end στο back\_end, δέχονται επεξεργασία και στην συνέχεια ξανά στο front\_end είναι επόμενο πολλές φορές ο προγραμματιστής να δυσκολευτεί ώστε τα στοιχεία να διατηρούνται στην ακέραια μορφή του και τα αποτελέσματα που δίνονται να είναι λογικά. Όλα τα παραπάνω παρατηρούνται αφού κάθε γλώσσα προγραμματισμού έχει διαφορές στην πλαισίωση των μεταβλητών, στον τρόπο αποθήκευσης της προσωρινής μνήμης αλλά και την λειτουργία των διαφόρων μεθόδων αποδίδοντας την απαίτηση πέραν την περισσότερης προσοχής και της επίλυσης ζητημάτων συντακτικών ή και λογικών λαθών.

Σχετικά με την ίδια την επίλυση του ζητήματος του χρονοπρογραμματισμού. Για την διεκπεραίωση του επιθυμητού βέλτιστου αποτελέσματος αρχικά πρέπει να γίνεται η ικανοποίηση των περιορισμών. Στα πρώτα στάδια την προσπάθειας της λύσης είναι κάτι που δεν ολοκλήρωνε την λειτουργία αυτή. Με άλλα λόγια κάποια από τα προβλήματα του παρατηρήθηκαν ήταν ότι οι παραστάσεις που επιλέγονταν σε συνδυασμό με τις ημερομηνίες δεν ήταν σωστές. Δηλαδή είτε είχαν ημερομηνίες που δεν τους ανήκαν είτε οι ημερομηνίες που εμφανίζονταν δεν ήταν όλες διαφορετικές η μία της άλλης. Μπορεί επίσης κάποιες εκδηλώσεις ενώ είχαν επιλεχθεί να μην εμφανίζονταν στο τελικό αποτέλεσμα αφού το πρόγραμμα τα θεωρούσε μη αποδεκτά. Στη συνέχεια αφού προστέθηκε και το κομμάτι των priorities των παραστάσεων πρόβλημα αποτέλεσε ότι ενώ κάποια παράσταση είχε μεγαλύτερο αριθμό από κάποια άλλη τελικά δεν ήταν αυτή που θα έμπαινε στο αποτέλεσμα. Ή το σύστημα πολλές φορές ήταν σαν να αγνοεί αυτές τις επιλογές. Τέλος, η εμφάνιση του αποτελέσματος δεν ήταν πάντα η βέλτιστη κάτι το οποίο ακύρωνε ολόκληρη την απόπειρα επίλυσης του προβλήματος με γραμμικό προγραμματισμό αφού ο κύριος στόχος του προγράμματος είναι αυτός

Κάποια από τα επιπλέον θέματα ήταν τα πολλαπλά error. Σχετικά με την βάση των δεδομένων, καθώς οι εγγραφές δεν δέχονταν πάντα τα σωστά attributes που απαιτούνταν για τον εκάστοτε πίνακα ή υπήρχαν αρκετά conflicts (συγκρούσεις) μεταξύ των εγγραφών αυτών. Ακόμη η αποστολή των αιτημάτων από την μία σελίδα στην άλλη πολλές φορές προκαλούσαν προβλήματα και errors, τύπου 505 κ.α., με αποτέλεσμα να απαιτηθεί πιο λεπτομερής διαχείριση αυτών.

Γενικά ως διαδικτυακή εφαρμογή η εξάλειψή όλων των πιθανών προβλημάτων είναι απαραίτητη έτσι ώστε να υπάρχει μία ολοκληρωμένη εμπειρία προς τον χρήστη. Επιπλέον θέλουμε το πρόγραμμα μας να αποδίδει στο βέλτιστο και να μην προκύπτουν βασικά θέματα για την διεκπεραίωση της λειτουργίας. Έτσι καλό είναι όλα αυτά που προκύπτουν κατά την ανάπτυξη να εξαλειφθούν κάνοντας ένα καλό και αποτελεσματικό debugging. Οπότε, έπειτα από πολλαπλές δοκιμές και προσπάθειες αφού η αποσφαλμάτωση ολοκληρώθηκε και έγινε ουσιαστική μεθοδική διαδικασία εύρεσης και εξάλειψης σφαλμάτων κώδικα το πρόγραμμα έφτασε σε ένα ικανοποιητικό αποτέλεσμα. Που τα μαζικά και μεγάλα error εξαλείφθηκαν ενώ υπάρχει μία λογική συνέχεια των οθονών χωρίς περαιτέρω θέματα. Τέλος, σημαντικό είναι ότι και λογικά το σύστημα λειτουργεί σύμφωνα με τις προδιαγραφές. Παράλληλα η λύση του προγράμματος του Χρονοπρογραμματισμού των θεατρικών παραστάσεων είναι η αναμενόμενη, προσφέροντας πάντα το βέλτιστο και καλύτερο δυνατό αποτέλεσμα στον χρήστη, απροβλημάτιστα και σε μικρό χρονικό διάστημα χωρίς την παρουσίαση σφαλμάτων είτε λογικών είτε λειτουργικών.

1. ΔΟΚΙΜΕΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

# **ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΕ ΔΙΑΦΟΡΕΤΙΚΕΣ ΤΙΜΕΣ ΤΩΝ ΔΕΔΟΜΕΝΩΝ**

Τα εκάστοτε αποτελέσματα που προκύπτουν κάθε φορά θα πρέπει κυρίως να συμβαδίζουν με τα αναμενόμενα. Οι δοκιμές αφορούν την εξέταση αν επιτεύχθηκε το αναμενόμενο, καθώς και αν η λύση δίνει το σύστημα είναι όντως και η βέλτιστη. Η ανάλυση των μεταβλητών τροφοδότησης σε συνδυασμό με το τελικό αποτέλεσμα είναι απαραίτητο να γίνει για την διεκπεραίωση της επιβεβαίωσης της αξιοπιστίας του συστήματος. Έτσι οφείλει να παρουσιαστεί ένα σύνολο διαφορετικών τιμών εξυπηρετώντας τον σκοπό αυτό. Άρα παρακάτω, παρατηραθέτουμε 3 use case με τα πραγματικά δεδομένα του προγράμματος, επιλέγοντας κάθε φορά έναν διαφορετικό συνδυασμό παραστάσεων και προτεραιοτήτων, για να δούμε αν και τα αποτελέσματα που δίνονται είναι τα αναμενόμενα.

## Test No.1

### Πίνακας 1 - Επιλεγμένα

|  |  |  |
| --- | --- | --- |
| ΠΑΡΑΣΤΑΣΗ | ΗΜΕΡΟΜΗΝΙΕΣ | ΠΡΟΤΕΡΑΙΟΤΗΤΑ |
| Ε3 | 19/05  20/05  22/05 | 1 |
| Ε4 | 20/05  27/05 | 1 |
| Ε6 | 24/05  25/05 | 1 |
| Ε8 | 29/05  30/05  31/05 | 1 |

Επεξηγηματικά, η 3η παράσταση είναι η Ε3 (δεν δίνεται όνομα για λόγους ευχρηστίας) και οι διαθέσιμες ημερομηνίες της είναι 19,20,22 του Μαΐου, ενώ η προτεραιότητα που της δόθηκε αναλογεί σε 1. Το ίδιο ισχύει και για τις άλλες αντιστοιχίζοντας τις τιμές.

### Αναμενόμενο αποτέλεσμα

Παρατηρείται ότι η παράσταση Ε3 και Ε4 συμπίπτουν με κοινή ημερομηνία επομένως η μία ή καμία θα διεκδικήσει το 20/05. Οι υπόλοιπες δεν έχουν κάποια ημερομηνία που να είναι ίδια με κάποια άλλη, επομένως θα επιλεχθεί μία από αυτές.

### Πίνακας 2 - Αποτέλεσμα

|  |  |
| --- | --- |
| ΠΑΡΑΣΤΑΣΗ | ΗΜΕΡΟΜΗΝΙΑ |
| Ε3 | 22/05 |
| Ε4 | 27/05 |
| Ε6 | 25/05 |
| Ε8 | 23/05 |

Επομένως, το αποτέλεσμα που προκύπτει είναι σωστό και αναμενόμενο.

## Test No.2

### Πίνακας 3 - Επιλεγμένα

|  |  |  |
| --- | --- | --- |
| ΠΑΡΑΣΤΑΣΗ | ΗΜΕΡΟΜΗΝΙΕΣ | ΠΡΟΤΕΡΑΙΟΤΗΤΑ |
| Ε11 | 24/06 | 1 |
| Ε13 | 27/06 | 3 |
| Ε10 | 02/06  03/06  04/06 | 2 |

### Πίνακας 4 - Αποτέλεσμα

|  |  |
| --- | --- |
| ΠΑΡΑΣΤΑΣΗ | ΗΜΕΡΟΜΗΝΙΑ |
| Ε10 | 04/06 |
| Ε11 | 24/06 |
| Ε13 | 27/06 |

## Test No.3

### Πίνακας 5 - Επιλεγμένα

|  |  |  |
| --- | --- | --- |
| ΠΑΡΑΣΤΑΣΗ | ΗΜΕΡΟΜΗΝΙΑ | ΠΡΟΤΕΡΑΙΟΤΗΤΑ |
| Ε11 | 24/06 | 3 |
| Ε7 | 24/06  25/06 | 2 |
| Ε20 | 23/06  24/06 | 2 |

### Πίνακας 6 - Αποτέλεσμα

|  |  |
| --- | --- |
| ΠΑΡΑΣΤΑΣΗ | ΗΜΕΡΟΜΗΝΙΑ |
| Ε7 | 25/06 |
| Ε11 | 24/06 |
| Ε20 | 23/06 |

# Παραδείγματα με screenshot

## Οι επιλεγμένες παραστάσεις ως εμφάνιση

A screenshot of a computer

Description automatically generated with medium confidence

## Αποτέλεσμα που εμφανίζεται στον χρήστη

A screen shot of a computer

Description automatically generated with low confidence

## Οι επιλεγμένες παραστάσεις σαν μεταβλητή JavaScript

A black rectangular object with a yellow border

Description automatically generated with low confidence

## Το αποτέλεσμα εμφανισμένο στο terminal

     ['1. Παιδικό Θέατρο: «Η Μάσα και ο Αρκούδος – Έναν καιρό και μια φορά…» στο Metropolitan Urban Theater on 2023-04-22 @ 18:00\n', '2. Η Παράσταση “Μάγια η Μέλισσα” στο Θέατρο Φαργκάνη Art στη Θεσσαλονίκη on 2023-04-30 @ 11:30\n', '3. Το Θεατρικό Έργο «Φαίδρας Άτοπος» στο Θέατρο Αθήναιον στη Θεσσαλονίκη on 2023-05-06 @ 21:00\n', '4. Η Θεατρική Κωμωδία «Παρασκευή & 13» στο Θέατρο Σοφούλη στη Θεσσαλονίκη on 2023-05-24 @ 21:00\n']

# ΣΧΕΤΙΚΑ ΜΕ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Καταληκτικά παρατηρείται ο πρωταρχικός στόχος της επίλυσης του ζητήματος επιτυγχάνεται. Η μεγιστοποίηση του αριθμού των παραστάσεων με διαφορετικές ημερομηνίες. Όπως φαίνεται και στις δοκιμές που γίναν, πρώτον όλες οι παραστάσεις που επιλέχθηκαν είναι στο αποτέλεσμα και δεύτερον κυριαρχούν αυτές με την υψηλότερη προτεραιότητα. Για παράδειγμα στο **Test NO3** η παράσταση Ε11 έχει μονάχα μία διαθέσιμη ημερομηνία, 24/06 και οι άλλες δύο παραστάσεις 7 και 20 έχουν διαθέσιμες και αυτές αυτή την ημερομηνία. Ωστόσο αφού η Ε11 έχει priority = 3 θα πει ότι σίγουρα θα πρέπει να είναι στην λύση εφόσον οι άλλες δύο έχουν χαμηλότερο (< 3). Έτσι τελικά οι Ε7, Ε20 επιλέγονται με τις άλλες κατάλληλες ημερομηνίες που διαθέτουν με αποτέλεσμα και η Ε11 να υπάρχει στην λύση. Σε περίπτωση που κάποια εκδήλωση έχει date ίδιο ακριβώς με κάποια άλλη και έχουν και ίδιο priority τότε επιλέγετε μόνο μία. Γενικά το πρόγραμμα προσπαθεί να εντάξει στο αποτέλεσμα όσο το δυνατόν περισσότερες παραστάσεις γίνεται αλλά αυτό με βάση τους περιορισμούς που έχουν τεθεί δεν γίνεται πάντα αφού υπάρχουν περιπτώσεις που για κάποια Ε δεν υπάρχουν Η που να τους απομένουν διαθέσιμες με αποτέλεσμα να μην εντάσσονται στην λύση. Ο Χρήστης σε κάθε περίπτωση θα λάβει το βέλτιστο αποτέλεσμα.

1. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΟΠΤΙΚΕΣ

Στη σημερινή εποχή ο ελεύθερος χρόνος που απομένει στους ανθρώπους σε καθημερινή βάση είναι ελάχιστος. Εργασιακές, οικογενειακές, κοινωνικές υποχρεώσεις και άλλα είναι κάποιοι από τους λόγους που έχουν δημιουργήσει μία τέτοια έλλειψη. Η ανάγκη λοιπόν για ψυχαγωγία στον κλάδο της αξιοποίησης του ελάχιστου χρόνου έχει αποτελέσει ζητούμενο μεγάλης αξίας για την ψυχική και πνευματική υγεία του κάθε ατόμου. Οι θεατρικές παραστάσεις-εκδηλώσεις απαρτίζουν έναν από τους τρόπους διαφυγής από τα προβλήματα του σύγχρονου τρόπου ζωής προσφέροντας αξιόλογα και ποιοτικά χαρακτηριστικά ψυχαγωγίας. Ωστόσο, το να βρει ο καθένας τις παραστάσεις που θα ταυτίζονται με τα αρέσω του, ενώ θα μπορεί να δημιουργήσει ένα πρόγραμμα ώστε να μπορέσει να αξιοποιήσει τον περιορισμένο αυτό χρόνο μέσα στις επόμενες ημέρες, αποτελεί ένα πρόβλημα που απαιτεί και λύση. Εκεί είναι που η ιδέα της κατασκευής του συστήματος θα ταιριάξει απίστευτα με αυτές τις ανάγκες. Η ανάγκη του χρονοπρογραμματισμού, ουσιαστικά, είναι αυτή που αποτέλεσε το κίνητρο για την κατασκευή της ιστοσελίδας “TheaterPlanning”. Ένα σύστημα το οποίο μπορεί να προσφέρει στους χρήστες αυτά τα οποία χρειάζονται για να κάνουν τα πλάνα τους πραγματικότητα.

Το TheaterPlanning (το όνομα της ιστοσελίδας) έχει την δυνατότητα προσφοράς στους χρήστες την απαλλαγή της προσπάθειας κατασκευής προγράμματος. Με την χρήση διαφόρων τεχνολογιών και του γραμμικού προγραμματισμού, το πρόγραμμα έχει την ικανότητα της δημιουργίας και παρουσίασης του βέλτιστου προγράμματος από μία μεγάλη πληθώρα διαθέσιμων παραστάσεων για την πόλη της Θεσσαλονίκης. Έτσι, δίνεται η επιλογή στους ενδιαφερόμενους να επιλέξουν σε πόσες και σε ποιες από τις εμφανιζόμενες θα ήθελαν να παν, προσφέροντας αποτελεσματικά και γρήγορα ένα ουσιαστικό plan ημερομηνιών που αν ακολουθηθεί θα αποδώσει την καλύτερη δυνατή εμπειρία ψυχαγωγίας, με έναν τρόπο που μόνο τα θέατρα μπορούν να προσφέρουν. Επιπλέον, όσον αφορά το σύνολο των εκδηλώσεων που εμφανίζονται, δίνονται οι απαραίτητες πληροφορίες για την καθεμία. Όνομα, ημερομηνίες, τοποθεσία, περιγραφή παράστασης, τιμές αλλά και αρκετά ακόμη καθιστώντας πηγή διάθεσης δεδομένων για την διευκόλυνση της λήψης της απόφασης του εκάστοτε εάν θα την συμπεριλάβει στο πρόγραμμα του ή όχι.

Ο χρονοπρογραμματισμός εκδηλώσεων δεν αποτέλεσε εύκολη διαδικασία αφού πολλές ήταν οι δυσκολίες, όπως η ικανοποίηση των περιορισμών, η εύρεση της καλύτερης λύσης και η μεγιστοποίηση της αντικειμενικής συνάρτησης. Παρόλα αυτά μετά από πολύωρες προσπάθειες ο κώδικας έφτασε σε στο επιθυμητό επίπεδο με την αποτροπή των bugs, και τον περιορισμό των περιττών σημείων του κώδικα. Προσφέροντας έτσι την προσδοκώμενη αξιοπιστία του συστήματος. Επομένως αφού έγιναν όλα τα προ απαιτούμενα και η εκτέλεση ολοκληρώνεται όπως υπολογίστηκε με τα αποτελέσματα που απαιτήθηκαν, μπορεί κανείς να πει ότι το πρόγραμμα ολοκληρώθηκε. Τέλος, όλα τα κείμενα για την απόκτηση βασικών γνώσεων, εργαλεία, τεχνολογίες και παρεμφερή αξιοποιήθηκαν σε κλίμα προσπάθειας κατασκευής μίας πλήρους εργασίας ενός φοιτητή της εφαρμοσμένης πληροφορικής.

Μία διαδικτυακή εφαρμογή σαν αυτή ποτέ δεν μπορούμε να πούμε ότι θα σταματήσει να αναπτύσσεται. Πάντα υπάρχουν νέες ιδέες, καινοτομίες ώστε να την ανανεώσουν ή να την κάνουν ακόμα πιο χρηστική. Για παράδειγμα, στο σύστημα εμφανίζονται παραστάσεις οι οποίες αντλούνται από website. Θα μπορούσε αρχικά, να προστεθούν ακόμα περισσότερα site καλύπτοντας ακόμη μεγαλύτερη πληθώρα παραστάσεων. Επιπλέον, πέρα των παραστάσεων, εκδηλώσεις γενικά θα είχαν μία αντάξια θέση στο cardboard του home page. Η Θεσσαλονίκη τόσο στο ιστορικό κομμάτι όσο και στην σύγχρονη πλέον πολυπολιτισμική κοινωνία με ποικιλία στους λαούς, έχει αρχίσει να παρουσιάζει μεγάλο πολιτιστικό ενδιαφέρον με πληθώρα εκδηλώσεων. Έτσι, όλες αυτές θα ήταν εύλογο να μπορέσουν να ενταχθούν στο σύστημα ως επιπλέον γεγονότα για τους ενδιαφερόμενους. Ακόμη, το κομμάτι της επιχειρηματικότητας είναι ακόμη ένας κλάδος που έχει αναπτυχθεί. Διεθνή συνέδρια, σεμινάρια οργανώνονται συνεχώς προτρέποντας τον νέο κόσμο να τα παρακολουθήσει. Πράγμα το οποίο εννοείται πως θα μπορούσε εξίσου να ενταχθεί στο Planning σχετικά με την οργάνωση της παρακολούθησης πολλών από αυτών που αναφέρθηκαν. Συμπερασματικά, παρατηρείται πως πέραν των θεατρικών παραστάσεων υπάρχουν πολλοί τομείς που θα μπορούσε να επεκταθεί το σύστημα, γεγονός που σίγουρα δίνει περιθώρια επέκτασης του ορίζοντα της εφαρμογής, με μία μελλοντική εργασία ανάπτυξης και προσθήκης των παραπάνω. Από την στιγμή που το σύστημα αποτελεί βασικό θεμέλιο σαν πρόγραμμα με περιθώρια προσθήκης νέων μεθόδων και τεχνικών

# ΒΙΒΛΙΟΓΡΑΦΙΑ

|  |  |
| --- | --- |
| [1] | “Google Or-Tools Employee Scheduling,” Google, [Online]. Available: https://developers.google.com/optimization/scheduling/employee\_scheduling. |
| [2] | L. A. Wolsey, “Mixed Integer Programming,” 2008. |
| [3] | F. . Grandoni and G. F. Italiano, “Algorithms and constraint programming,” , 2006. [Online]. Available: http://people.idsia.ch/~grandoni/pubblicazioni/gi06cp.pdf. [Accessed 12 6 2023]. |
| [4] | F. R. T. S. Pedro Meseguer, Handbook of Constraint Programming, 2006. |
| [5] | N. Kokash, “An introduction to heuristic algorithms”. |
| [6] | I. S. A. A. &. K. A. Ioannis Refanidis, Hellenic Conference on Artificial Intelligence, A. Likas, Ed., 2014, p. 615–629. |
| [7] | “Wikipedia/webscrape,” [Online]. Available: https://en.wikipedia.org/wiki/Web\_scraping. |
| [8] | “Scrapy,” [Online]. Available: https://scrapy.org/. |
| [9] | “API,” [Online]. Available: https://en.wikipedia.org/wiki/API. |
| [10] | “Python,” [Online]. Available: https://www.python.org/. |
| [11] | “Wikipedia/Javascript,” [Online]. Available: https://el.wikipedia.org/wiki/JavaScript. |
| [12] | “Wikipedia/Html,” [Online]. Available: https://el.wikipedia.org/wiki/HTML. |
| [13] | “Wikipedia/CSS,” [Online]. Available: https://en.wikipedia.org/wiki/CSS. |
| [14] | D. H. Curie, “Analysis on Web Frameworks,” *Journal of Physics: Conference Series,* pp. 1-7, 2019. |
| [15] | “Wikipedia/Flask,” [Online]. Available: https://en.wikipedia.org/wiki/Flask\_(web\_framework). |
| [16] | “Flask,” [Online]. Available: https://flask.palletsprojects.com/en/2.2.x/. |
| [17] | “Djangoproject,” [Online]. Available: https://www.djangoproject.com/. |
| [18] | “Pyramid,” [Online]. Available: https://trypyramid.com/. |
| [19] | “Bottle,” [Online]. Available: https://bottlepy.org/docs/dev/. |
| [20] | “Wikipedia/Jinja,” [Online]. Available: https://en.wikipedia.org/wiki/Jinja\_(template\_engine). |
| [21] | “werkzeug,” [Online]. Available: https://werkzeug.palletsprojects.com/en/2.3.x/. |
| [22] | “Flask-SqlAlchemy,” [Online]. Available: https://flask-sqlalchemy.palletsprojects.com/en/3.0.x/. |
| [23] | “Beautiful Soup Documentation¶,” [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. |
| [24] | “cURL,” [Online]. Available: https://en.wikipedia.org/wiki/CURL. |
| [25] | “Google Or-Tools,” [Online]. Available: https://developers.google.com/optimization/introduction/python#optimization. |
| [26] | “PuLP,” [Online]. Available: https://pypi.org/project/PuLP/. |
| [27] | “SCIP - Solving Constraint Integer Programs,” [Online]. Available: https://www.scipopt.org/. |
| [28] | S. Fredrickson, “Top 5 User Expectations Your Website Must Meet,” *Flexpro Technology Solutions,* 2019. |
| [29] | [Online]. Available: https://www.thessalonikiguide.gr/events/theatro/. |
| [30] | V. D. S. B. Ping Zhang, Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers, IEEE, 1999. |
| [31] | “jQuery,” OpenJS Foundation and jQuery contributors, [Online]. Available: https://jquery.com/. |
| [32] | S. Sriparasa, JavaScript and JSON Essentials, 2013. |
| [33] | P. P. a. C. D, Enabling high-level application development for the Internet of Things, 1-May-2015, pp. 62-84. |
| [34] | H. Oluwatosin, “IOSR Journal of Computer Engineering (IOSR-JCE),” *researchgate.net,* vol. 16, pp. 67-71, 2014. |
| [35] | M. Team, “Βασικές αρχές σχεδίασης βάσης δεδομένων”. |
| [36] | “IBM ILOG CP OPTIMIZER,” [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-cp-optimizer. |
| [37] | “Wikipedia/GNU,” [Online]. Available: https://en.wikipedia.org/wiki/GNU\_Linear\_Programming\_Kit. |
| [38] | “GLPK (GNU Linear Programming Kit),” [Online]. Available: https://www.gnu.org/software/glpk/. |
| [39] | “pywraplp,” [Online]. Available: https://google.github.io/or-tools/python/ortools/linear\_solver/pywraplp.html. |
| [40] | “COIN-OR Branch-and-Cut solver,” [Online]. Available: https://github.com/coin-or/Cbc. |
| [41] | “SetCoefficient Documentation,” [Online]. Available: https://google.github.io/or-tools/cpp/classoperations\_\_research\_1\_1\_m\_p\_objective.html#a2def997791a2a5119c3502aa68c34181. |