



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

**ΣΧΟΛΗ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΟΛΙΤΙΣΜΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΣ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

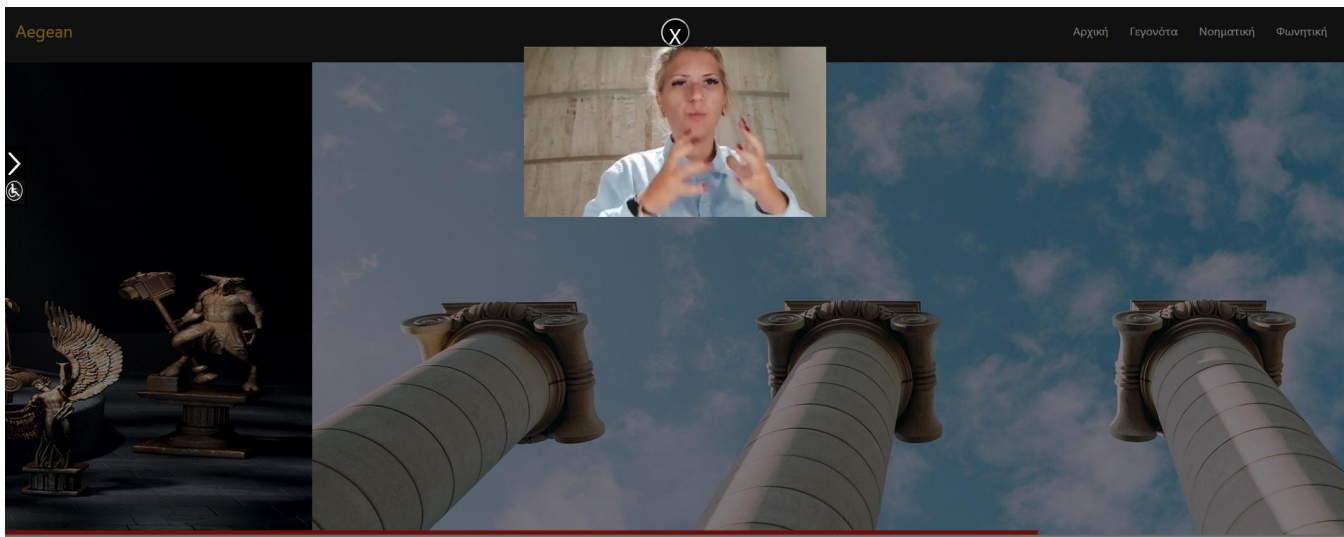
Εφαρμογή ενημέρωσης πολιτιστικών γεγονότων για ΑΜΕΑ

**Ανδρέας ΚΩΣΤΑΡΑΣ (1312018072)
Δημήτριος ΑΥΓΟΥΣΤΟΠΟΥΛΟΣ (1312018010)
Μάριος ΚΑΛΔΕΛΛΗΣ (1312018042)**

**Επίβλεψη:
Κωνσταντίνος ΚΩΤΗΣ, επικ. καθηγητής
Ακαδημαϊκό Έτος:
2021-2022**

ΜΥΤΙΛΗΝΗ | ΣΕΠΤΕΜΒΡΙΟΣ 2022

ΕΦΑΡΜΟΓΗ ΕΝΗΜΕΡΩΣΗΣ ΠΟΛΙΤΙΣΤΙΚΩΝ ΓΕΓΟΝΟΤΩΝ ΓΙΑ ΑΜΕΑ



Μέλη εξεταστικής επιτροπής:

1. Κώτης Κωνσταντίνος, επικ. καθηγητής (επιβλέπων)
2. Τσεκούρας Γεώργιος, καθηγητής (μέλος)
3. Κασαπάκης Βλάσιος, επικ. καθηγητής (μέλος)

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας εργασίας είναι η δημιουργία μιας εφαρμογής που αφορά την ενημέρωση των Ατόμων με Ειδικές Ανάγκες (ΑΜΕΑ) για πολιτιστικά γεγονότα. Τα πολιτιστικά γεγονότα είναι αποθηκευμένα σε μία βάση δεδομένων SQL και εμφανίζονται ανάλογα με τις απαιτήσεις του χρήστη. Για την αναζήτηση, ανάκτηση και παρουσίαση των γεγονότων υπάρχει συγκεκριμένο λεξιλόγιο, διαμορφωμένο με τις κατάλληλες ερωτήσεις προκειμένου ο χρήστης να έχει ξεκάθαρη εικόνα σχετικά με τις πληροφορίες που συλλέγει. Οι πληροφορίες αυτές παρέχονται με τη μορφή κειμένου, ήχου αλλά και βίντεο. Επιπροσθέτως, για τη καλύτερη δυνατή επικοινωνία του χρήστη με την εφαρμογή, υπάρχει τόσο το μοντέλο αναγνώρισης νοηματικής γλώσσας, όσο και το chatbot, που απαντάει στα ερωτήματα **Πού; Πότε;** και **Τι;** αναφορικά με τα γεγονότα.

Το τελικό αποτέλεσμα αυτής της εργασίας είναι η ανάπτυξη εφαρμογής που διευκολύνει όλους τους χρήστες με ειδικές ανάγκες (ΑΜΕΑ), να ενημερώνονται για τα πολιτιστικά δρώμενα της περιοχής δίνοντάς τους παράλληλα τη δυνατότητα να αλληλεπιδρούν μέσω της ελληνικής νοηματικής γλώσσας.

Λέξεις κλειδιά: ΑΜΕΑ, Πολιτιστικά Γεγονότα, Νοηματική Γλώσσα, Chatbot.

ABSTRACT

The aim of this project is to create an application that informs Persons with Disabilities (PWD) about cultural events. The cultural events are stored in an SQL database and retrieved according to the user's queries. For the search, retrieval and presentation of the events there is a specific vocabulary, formulated with appropriate questions in order for the user to have a clear picture about the information collected. This information is provided in the form of text, audio and video. In addition, for the efficient communication between the user and the application, the Greek sign language recognition model and a chatbot are provided, which answer questions related to the **Where? When?** and **What?** of cultural events. The outcome of this project is the development of a Web-based application that supports PWD to be informed about cultural events in Greece.

Keywords: PWD, Cultural Events, Sign Language, Chatbot.

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της παρούσας εργασίας αισθανόμαστε την υποχρέωση να ευχαριστήσουμε τον κύριο Κωνσταντίνο Κώτη, επίκουρο καθηγητή του Τμήματος Πολιτισμικής Τεχνολογίας και Επικοινωνίας τόσο για τη συμβολή του στην υλοποίησή όσο και για την εξαιρετική συνεργασία που είχαμε καθ' όλη τη διάρκειά της. Επιπλέον, θα θέλαμε να ευχαριστήσουμε την κυρία Γεωργία Μπακάλη, για τη πολύτιμη βοήθειά της σχετικά με τη δημιουργία των βίντεο στην ελληνική νοηματική γλώσσα. Τέλος, θα θέλαμε να ευχαριστήσουμε για την αμέριστη στήριξή τους, τις οικογένειές μας που βοήθησαν να ξεπεράσουμε κάθε δυσκολία προκειμένου να υλοποιηθεί η συγκεκριμένη πτυχιακή εργασία.

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1. Διάγραμμα εφαρμογής	13
Εικόνα 2. Διάγραμμα μοντέλου 1	15
Εικόνα 3. Διάγραμμα μοντέλου 2	16
Εικόνα 4. Διάγραμμα μοντέλου 3	16
Εικόνα 5. Διάγραμμα μοντέλου 4	17
Εικόνα 6. Καρτέλα Γεγονότα	19
Εικόνα 7. Ταξινόμηση	20
Εικόνα 8. Καρτέλα Νοηματική	20
Εικόνα 9. Αντικατοπτρισμός Κάμερας	21
Εικόνα 10. Αποτέλεσμα Νοηματικής	21
Εικόνα 11. Καρτέλα Φωνητική	22
Εικόνα 12. Αποτελέσματα Φωνητικής	22
Εικόνα 13. Ανακατεύθυνση Αρχικής	23
Εικόνα 14. Ποσοστό εγκυρότητας	24
Εικόνα 15. Rasa CLI	25
Εικόνα 16. Αρχείο NLU.yml	26
Εικόνα 17. Αρχείο actions.py	27
Εικόνα 18. Αρχείο domain.yml	28
Εικόνα 19. Αρχείο NLU.yml (2)	29
Εικόνα 20. Dockerfile	30
Εικόνα 21. Αρχείο docker-compose.yml	31
Εικόνα 22. Δοκιμή chatbot	31
Εικόνα 23. Κουμπιά Γεγονότα	32
Εικόνα 24. Κουμπιά Τοποθεσία	33
Εικόνα 25. Κουμπιά Είδη Γεγονότων	34
Εικόνα 26. Ερώτηση 1	36
Εικόνα 27. Ερώτηση 2	37
Εικόνα 28. Ερώτηση 3	37
Εικόνα 29. Ερώτηση 4	38
Εικόνα 30. Ερώτηση 5	38
Εικόνα 31. Ερώτηση 6	39
Εικόνα 32. Ερώτηση 7	39
Εικόνα 33. Ερώτηση 8	40
Εικόνα 34. Ερώτηση 9	40
Εικόνα 35. Ερώτηση 10	41

ΑΚΡΩΝΥΜΙΑ

AMEA	Άτομα με Ειδικές Ανάγκες
AI	Artificial Intelligence
API	Application Programming Interface
CLI	Command Line Interface
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
ID	Identity
NLU	Natural language Understanding
NPM	Node Package Manager
OpenCV	Open-Source Computer Vision Library
PHP	Hypertext Preprocessor
PWD	Persons With Disabilities
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
TF	TensorFlow

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	4
ABSTRACT	5
ΕΥΧΑΡΙΣΤΙΕΣ	6
ΛΙΣΤΑ ΕΙΚΟΝΩΝ.....	7
ΑΚΡΩΝΥΜΙΑ	8
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	9
1. ΕΙΣΑΓΩΓΗ.....	10
2. ΑΠΑΙΤΗΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	11
3. ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	13
3.1. Ανάπτυξη εφαρμογής με τις γλώσσες HTML,PHP και Javascript	14
3.1.1. Βιβλιοθήκες.....	14
3.2. Δημιουργία Βάσης Δεδομένων με χρήση της γλώσσας SQL	14
3.2.1. Structured Query Language (SQL)	14
3.3. Αρχικά Μοντέλα Μετατροπής της Πληροφορίας.....	15
3.3.1. Μοντέλο Μετατροπής από Κείμενο σε Ομιλία.....	15
3.3.2. Μοντέλο Μετατροπής από Ομιλία σε Κείμενο.....	16
3.3.3. Μοντέλο Μετατροπής από Κείμενο σε Νοηματική	16
3.3.4. Μοντέλο Μετατροπής από Νοηματική σε Κείμενο	17
3.4. Σχεδιασμός του μοντέλου αναγνώρισης νοηματικής γλώσσας και η βιβλιοθήκη OpenCV.....	17
3.5. Σχεδιασμός chatbot με Rasa Framework.....	18
3.5.1. RASA Framework.....	18
4. ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	19
4.1. Υλοποίηση του μοντέλου αναγνώρισης νοηματικής γλώσσας της τεχνολογίας Python -Tensorflow	23
4.2. Υλοποίηση chatbot της τεχνολογίας Python και Rasa Framework.....	25
4.2.1 Chatbot Buttons	32
4.3. Λογισμικό φιλοξενίας εφαρμογής.....	35
5. ΔΟΚΙΜΗ – ΕΛΕΓΧΟΣ – ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	36
6. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	42
ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ	43
ΠΑΡΑΡΤΗΜΑ.....	44

1. ΕΙΣΑΓΩΓΗ

Η επικοινωνία είναι μία πράξη ανταλλαγής πληροφοριών, ιδεών και συναισθημάτων. Για την επικοινωνία μεταξύ δύο ανθρώπων, και οι δύο απαιτείται να έχουν γνώση και κατανόηση μιας κοινής γλώσσας. Στην περίπτωση όμως των ΑΜΕΑ και συγκεκριμένα των κωφάλαλων, τα μέσα επικοινωνίας είναι διαφορετικά. Ο κουφός δεν έχει τη δυνατότητα να ακούσει και ο μουγγός δεν έχει τη δυνατότητα να μιλήσει. Έτσι, επικοινωνούν χρησιμοποιώντας τη νοηματική γλώσσα μεταξύ τους αλλά και με τους άλλους ανθρώπους. Όμως, δεν διαθέτουν όλοι τη γνώση και την κατανόηση της νοηματικής γλώσσας, πράγμα που καθιστά δύσκολη την επικοινωνία για τα άτομα αυτά. Η παρούσα εργασία στοχεύει στη δημιουργία μιας εφαρμογής πολιτιστικού περιεχομένου, που θα είναι ειδικά διαμορφωμένη προκειμένου το περιεχόμενό της να απευθύνεται σε ΑΜΕΑ.

Η εργασία πραγματοποιείται στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του τμήματος Πολιτισμικής Τεχνολογίας και Επικοινωνίας του Πανεπιστημίου Αιγαίου, κατά τη διάρκεια του ακαδημαϊκού έτους 2021-2022. Αντικείμενο της εργασίας είναι η δημιουργία ενός ‘ημερολογίου’ πολιτιστικών γεγονότων όπου η πληροφορία δίνεται με τη μορφή κειμένου, ήχου αλλά και βίντεο ανάλογα με τις απαιτήσεις του εκάστοτε χρήστη. Συγκεκριμένα, κρίνεται απαραίτητη - για την εκπλήρωση του παραπάνω σκοπού – τόσο η δημιουργία ενός μοντέλου αναγνώρισης νοηματικής γλώσσας όσο και ένα chatbot, που απαντάει σε ερωτήσεις που αφορούν τα πολιτιστικά αυτά γεγονότα.

2. ΑΠΑΙΤΗΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Στην παρούσα πτυχιακή εργασία στόχος είναι η δημιουργία μιας εφαρμογής πολιτιστικού περιεχομένου που απευθύνεται σε ΑΜΕΑ. Ειδικότερα, είναι αναγκαία - για την εκπλήρωση του παραπάνω σκοπού - τόσο η δημιουργία ενός μοντέλου αναγνώρισης νοηματικής γλώσσας όσο και ένα chatbot, που απαντάει σε ερωτήσεις που αφορούν τα πολιτιστικά αυτά γεγονότα.

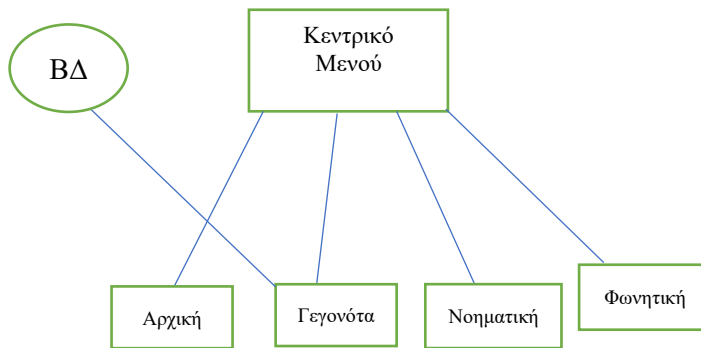
Κάθε χρήστης έχει τη δυνατότητα να αλληλεπιδρά με την εφαρμογή χρησιμοποιώντας τη νοηματική γλώσσα και τις φωνητικές εντολές αλλά και παράλληλα να λαμβάνει απαντήσεις με τον ίδιο ακριβώς τρόπο.

Απαιτήσεις:

1. Κατόπιν φόρτωσης της σελίδας θα εμφανίζεται ένα βίντεο στη νοηματική, στο οποίο θα παρουσιάζονται οι δυνατότητες που προσφέρονται στο χρήστη και παράλληλα θα ακούγεται ένα μήνυμα καλωσορίσματος σχετικά με αυτές.
2. Η εφαρμογή είναι δομημένη για τη διευκόλυνση του χρήστη γι' αυτό και θα υπάρχει ειδικό μενού (πάνω δεξιά) καθώς και το footer με καρτέλες, για το διαχωρισμό των δυνατοτήτων (Αρχική, Γεγονότα, Νοηματική, Φωνητική).
3. Μετά το κλείσιμο του βίντεο ο χρήστης μεταβαίνει στην **Αρχική** καρτέλα, όπου θα φαίνεται ένα slideshow με εικόνες πολιτιστικού περιεχομένου και στη συνέχεια αφού κάνει scroll down θα μπορεί να διαβάσει λίγα λόγια για τις δυνατότητες που παρέχονται.
4. Στην καρτέλα **Γεγονότα** ο χρήστης θα μπορεί να ενημερωθεί για τα πολιτιστικά δρώμενα με τη μορφή κειμένου αλλά και βίντεο στη νοηματική.
5. Στην ίδια καρτέλα θα υπάρχει ειδικό κουμπί ταξινόμησης προς διευκόλυνση του χρήστη. Η ταξινόμηση των γεγονότων θα μπορεί να γίνει ανά Τύπο, Ημερομηνία και Πόλη.
6. Για κάθε ξεχωριστό πολιτιστικό γεγονός θα υπάρχει ηχητική ενημέρωση πατώντας το σχετικό κουμπί.
7. Στην καρτέλα **Νοηματική** (εφόσον ο χρήστης έχει συνδεδεμένη την κάμερα μαζί με το μικρόφωνό του) θα εμφανίζεται αρχικά ένα μικρό παράθυρο διαλόγου στο φυλλομετρητή πάνω αριστερά, όπου θα ζητάει την άδεια χρήσης των παραπάνω και στη συνέχεια (αν επιτραπεί η χρήση της κάμερας) θα παρέχεται η δυνατότητα διάδρασης με την εφαρμογή, μέσω του μοντέλου αναγνώρισης νοηματικής γλώσσας. Επιπλέον, υπάρχει ειδικό μπλε κουμπί στα αριστερά που με το πάτημά του, εμφανίζεται ένα βίντεο επεξηγώντας στη νοηματική καθώς και εικόνες σχετικά με τη σωστή χρήση της κάμερας για αποτελεσματικότερη επικοινωνία.

8. Με το κουμπί στα αριστερά « **Reverse Camera**» θα δίνεται η δυνατότητα αντικατοπτρισμού της κάμερας στο χρήστη προκειμένου να γίνεται πιο εύκολη η αναγνώριση της νοηματικής γλώσσας από το μοντέλο.
9. Στην καρτέλα **Φωνητική** (εφόσον ο χρήστης έχει συνδεδεμένη την κάμερα μαζί με το μικρόφωνό του) θα εμφανίζεται αρχικά ένα μικρό παράθυρο διαλόγου στο φυλλομετρητή πάνω αριστερά, όπου ζητάει την άδεια χρήσης των παραπάνω και στη συνέχεια (αν επιτραπεί η χρήση του μικροφώνου) ο χρήστης θα μπορεί μέσω φωνητικών εντολών και ερωτήσεων να ενημερωθεί για τα πολιτιστικά γεγονότα. Επιπροσθέτως, η ύπαρξη του μπλε κουμπιού αριστερά « Φωνητική βοήθεια » (όπου με ηχητικό μήνυμα θα δίνονται οδηγίες σχετικά με τον τρόπο υποβολής της ερώτησης από τον χρήστη) κάνει τη διαδικασία ακόμα πιο εύκολη.
10. Οι χρήστες θα μπορούν να ανακατευθυνθούν μέσω της φωνητικής σε όποια καρτέλα επιθυμούν, απλά αναφέροντας το όνομα της καρτέλας π.χ. Αρχική.
11. Με τη φωνητική εντολή ‘καθάρισμα’ ο χρήστης θα μπορεί να σβήσει τους χαρακτήρες από το πεδίο αναζήτησης.
12. Στα αριστερά της σελίδας θα βρίσκεται το μενού « **Προσβασιμότητας** », (βελάκι άσπρο δεξί με μαύρο φόντο) όπου εκεί παρέχονται οι εξής δυνατότητες: Μεγέθυνση Γραμματοσειράς, Αντεστραμμένα Χρώματα και Φωνητική ανάγνωση του περιεχομένου της εκάστοτε σελίδας.
13. Στην κάτω δεξιά γωνία της σελίδας θα βρίσκεται το στρόγγυλο κουμπί για το chatbot, όπου με το πάτημα από το χρήστη θα είναι δυνατή η υποβολή ερωτημάτων αναφορικά με τα πολιτιστικά γεγονότα σε σχέση με το είδος, την ημερομηνία και τον τόπο.
14. Το chatbot θα έχει και κουμπιά σχετικά με τα παραπάνω (είδος, ημερομηνία, τόπος) για καλύτερη αναζήτηση γεγονότων του χρήστη σε περίπτωση γενικής ερώτησης.
15. Λειτουργία Back to Top, οι χρήστες δεν θα χρειάζεται να κάνουν scroll για να επιστρέψουν στο αρχικό πάνω μέρος της σελίδας.
16. Responsive design: Η πρόσβαση στη σελίδα θα γίνεται το ίδιο εύκολη τόσο από υπολογιστή όσο και από κινητό.
17. Οι δυνατότητες της Φωνητικής θα μπορούν να λειτουργούν μόνο από τον φυλλομετρητή Google Chrome τόσο σε υπολογιστή όσο και σε κινητό.
18. Η αποθήκευση των πολιτιστικών γεγονότων θα γίνεται σε βάση δεδομένων SQL.
19. Δεν θα απαιτείται κάποια προεγγραφή χρήστη.

3. ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ



Εικόνα 1. Διάγραμμα εφαρμογής

Αρχικά, η σχεδίαση και η δομή της εφαρμογής παρουσιάζεται στο παραπάνω διάγραμμα (Εικόνα 1). Η **Αρχική καρτέλα**, περιέχει ένα slideshow με εικόνες πολιτιστικού περιεχομένου καθώς και λίγα λόγια για τις δυνατότητες που παρέχονται. Συνεχίζοντας, στην καρτέλα **Γεγονότα** (τα γεγονότα είναι αποθηκευμένα σε **Βάση Δεδομένων**) ο χρήστης μπορεί να ενημερωθεί για τα πολιτιστικά δρώμενα με τη μορφή κειμένου αλλά και βίντεο στη νοηματική. Επιπλέον, στην καρτέλα **Νοηματική** παρέχεται η δυνατότητα διάδρασης με την εφαρμογή και επικοινωνίας μέσω της ελληνικής νοηματικής γλώσσας, με τη βοήθεια του μοντέλου αναγνώρισης νοηματικής γλώσσας. (Εφόσον ο χρήστης έχει συνδεδεμένη την κάμερα μαζί με το μικρόφωνό του) εμφανίζεται ένα μικρό παράθυρο διαλόγου στο φυλλομετρητή πάνω αριστερά (για την άδεια χρήσης της κάμερας). Τέλος, στην καρτέλα **Φωνητική** ο χρήστης μπορεί μέσω φωνητικών εντολών και ερωτήσεων να ενημερωθεί για τα πολιτιστικά γεγονότα. (Εφόσον έχει συνδεδεμένη την κάμερα μαζί με το μικρόφωνό του) εμφανίζεται ένα μικρό παράθυρο διαλόγου στο φυλλομετρητή πάνω αριστερά, (για την άδεια χρήσης του μικροφώνου).

3.1 Ανάπτυξη εφαρμογής με τις γλώσσες HTML, PHP και Javascript

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η γλώσσα HTML (HyperText Markup Language). Οι λόγοι που επιλέχθηκε είναι η εξοικείωση που έχουμε με τη συγκεκριμένη γλώσσα στη δημιουργία ιστοσελίδων αλλά και για την εύκολη διαχείρισή τους. Παρέχει πολλές δυνατότητες προσθέτοντας πίνακες, εικόνες, λίστες, αρχεία κ.α. Επιπροσθέτως, έχει πολύ υψηλή συμβατότητα από όλους τους φυλλομετρητές. Στον κώδικα της HTML προστέθηκε και μορφοποίηση CSS (Cascading Style Sheets) για λόγους ευχρηστίας αλλά και εμφάνισης της εφαρμογής. Ταυτόχρονα με την HTML αναπτύχθηκε κώδικας στις γλώσσες PHP (Hypertext Preprocessor) και JavaScript. Με τις PHP και JavaScript το περιεχόμενο γίνεται δυναμικό, δίχως να απαιτείται συνεχής ανανέωση.

3.1.1 Βιβλιοθήκες

Κατά τη δημιουργία της εφαρμογής, η χρήση βιβλιοθηκών είχε ως στόχο την ενσωμάτωση νέων αλλά και την βελτίωση υπάρχοντων λειτουργιών.

Έγινε χρήση της βιβλιοθήκης jQuery. Η jQuery επιτρέπει τη χρήση σχεδόν όλων των δυνατοτήτων της JavaScript, όπως τη διαμόρφωση HTML και CSS στοιχείων, προσθήκη animation και εφέ μαζί με αρκετές ακόμα εφαρμογές.

3.2 Δημιουργία Βάσης Δεδομένων με χρήση της γλώσσας SQL

Η δημιουργία μιας βάσης δεδομένων κρίθηκε απαραίτητη, διότι έπρεπε τα δεδομένα που αφορούν τα πολιτιστικά γεγονότα να είναι αποθηκευμένα κάπου με αποτέλεσμα, αφενός να είναι πιο εύκολα διαχειρίσιμα και αφετέρου να είναι προσβάσιμα, προκειμένου να ανακτηθούν. Χρησιμοποιήθηκε η SQL (Structured Query Language) λόγω της εξοικείωσής μας με αυτή.

3.2.1 Structured Query Language (SQL)

Η Structured Query Language είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για τη διαχείριση βάσεων δεδομένων και την εκτέλεση διαφόρων λειτουργιών στα δεδομένα τους. Η SQL χρησιμοποιείται τακτικά όχι μόνο από διαχειριστές βάσεων δεδομένων, αλλά από προγραμματιστές και αναλυτές που θέλουν να δημιουργήσουν και να εκτελέσουν αναλυτικά ερωτήματα [1].

3.3 Αρχικά Μοντέλα Μετατροπής της Πληροφορίας

Ο στόχος δημιουργίας της εφαρμογής είναι η διευκόλυνση των Ατόμων με Ειδικές Ανάγκες (ΑΜΕΑ) που θέλουν να ενημερωθούν για τα πολιτιστικά γεγονότα γι' αυτό και προσχεδιάστηκαν τέσσερα μοντέλα προκειμένου να ικανοποιηθούν όλες οι ανάγκες.

3.3.1 Μοντέλο Μετατροπής από Κείμενο σε Ομιλία



Εικόνα 2. Διάγραμμα μοντέλου 1

Όπως φαίνεται στην παραπάνω εικόνα (Εικόνα 2) το συγκεκριμένο μοντέλο μετατρέπει το κείμενο σε ομιλία, παρέχοντας τη δυνατότητα στο χρήστη να συλλέξει πληροφορίες για τα πολιτιστικά δρώμενα που τον αφορούν με τη μορφή ήχου μέσω ομιλίας.

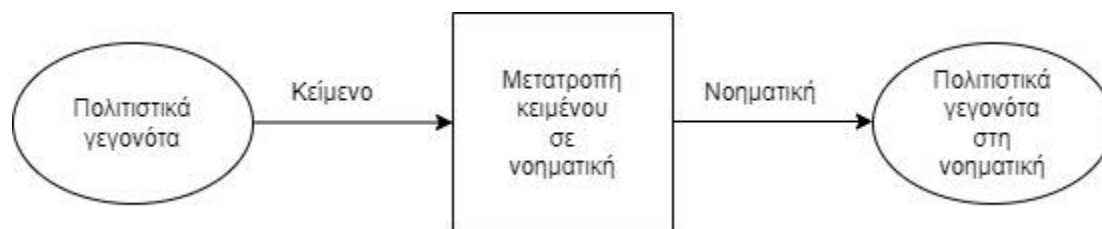
3.3.2 Μοντέλο Μετατροπής από Ομιλία σε Κείμενο



Εικόνα 3. Διάγραμμα μοντέλου 2

Όπως φαίνεται στην παραπάνω εικόνα (Εικόνα 3) το συγκεκριμένο μοντέλο μετατρέπει την ομιλία σε κείμενο, παρέχοντας τη δυνατότητα στο χρήστη να συλλέξει πληροφορίες για τα πολιτιστικά δρώμενα που τον αφορούν με τη μορφή κειμένου αφού μέσω φωνητικών εντολών ικανοποιείται το αίτημά του.

3.3.3 Μοντέλο Μετατροπής από Κείμενο σε Νοηματική



Εικόνα 4. Διάγραμμα μοντέλου 3

Όπως φαίνεται στην παραπάνω εικόνα (Εικόνα 4) το συγκεκριμένο μοντέλο μετατρέπει το κείμενο σε νοήματα, έτσι ώστε ο χρήστης να συλλέξει πληροφορίες για τα πολιτιστικά γεγονότα που τον αφορούν μέσω τη ελληνικής νοηματικής γλώσσας.

3.3.4 Μοντέλο Μετατροπής από Νοηματική σε Κείμενο



Εικόνα 5. Διάγραμμα μοντέλου 4

Όπως φαίνεται στην παραπάνω εικόνα (Εικόνα 5) το συγκεκριμένο μοντέλο μετατρέπει τη νοηματική σε κείμενο, παρέχοντας τη δυνατότητα στο χρήστη να συλλέξει πληροφορίες για τα πολιτιστικά δρώμενα που τον αφορούν αφού μέσω του μοντέλου της αναγνώρισης των νοημάτων θα εξυπηρετηθεί άμεσα.

3.4 Σχεδιασμός του μοντέλου αναγνώρισης νοηματικής γλώσσας και η βιβλιοθήκη OpenCV

Το συγκεκριμένο μοντέλο είναι ένα σύστημα ανίχνευσης της νοηματικής γλώσσας σε πραγματικό χρόνο που αφορά την Ελληνική Νοηματική Γλώσσα. Για την απόκτηση δεδομένων, οι εικόνες συλλαμβάνονται από την Web κάμερα με τη χρήση Python και OpenCV.

Η βιβλιοθήκη OpenCV παρέχει λειτουργίες που έχουν ως κύριο στόχο την όραση υπολογιστών σε πραγματικό χρόνο. Επιταχύνει τη χρήση της μηχανικής μάθησης και παρέχει μια κοινή υποδομή για τις εφαρμογές που βασίζονται στην όραση υπολογιστών. Η βιβλιοθήκη OpenCV διαθέτει περισσότερες από 2500 λειτουργίες όρασης υπολογιστών και μηχανικής μάθησης καθώς και αλγορίθμους που μπορούν να χρησιμοποιηθούν για την ανίχνευση και αναγνώριση προσώπων, την αναγνώριση αντικειμένων, ταξινόμηση ανθρώπινων ενεργειών, παρακολούθηση των κινήσεων της κάμερας και των αντικειμένων, εξαγωγή τρισδιάστατων μοντέλων αντικειμένων και πολλά άλλα.

Το σύνολο δεδομένων που δημιουργήθηκε αποτελείται από σημάδια που αντιπροσωπεύουν λέξεις στην Ελληνική Νοηματική Γλώσσα, για κάθε λέξη, λαμβάνονται εικόνες για να διαμορφωθεί το σύνολο των δεδομένων. Οι εικόνες που καταγράφονται αποθηκεύονται σε αντίστοιχο φάκελο [2].

3.5 Σχεδιασμός chatbot με Rasa Framework

Το chatbot είναι ένας προσωπικός βοηθός βασισμένος στη διαλογική τεχνητή νοημοσύνη (conversational AI) που μπορεί να έχει διαδραστικές συνομιλίες με ανθρώπους. Μπορεί να βοηθήσει τους χρήστες να βρουν απαντήσεις στα ερωτήματά τους, να κάνουν δηλαδή μικρές εργασίες παρέχοντας πληροφορίες σχετικά με τις ερωτήσεις που θέτουν οι χρήστες, όπως για παράδειγμα, η ερώτηση μπορεί να αφορά τα πολιτιστικά γεγονότα που συμβαίνουν μια συγκεκριμένη χρονική περίοδο στην πόλη τους.

Τα ευφυή συστήματα chatbot είναι αναγνωρισμένα για την εφαρμογή ρομποτικών συστημάτων και της επεξεργασίας φυσικής γλώσσας. Καθώς λόγω της ανάπτυξης της φυσικής γλώσσας και των νευρωνικών δικτύων, η εφαρμογή της τεχνητής νοημοσύνης αυξάνεται στα συστήματα chatbot, τα οποία χρησιμοποιούνται συνήθως σε συστήματα διαλόγου για διάφορους πρακτικούς λόγους, όπως είναι η εξυπηρέτηση πελατών και η απόκτηση πληροφοριών.

3.5.1 RASA Framework

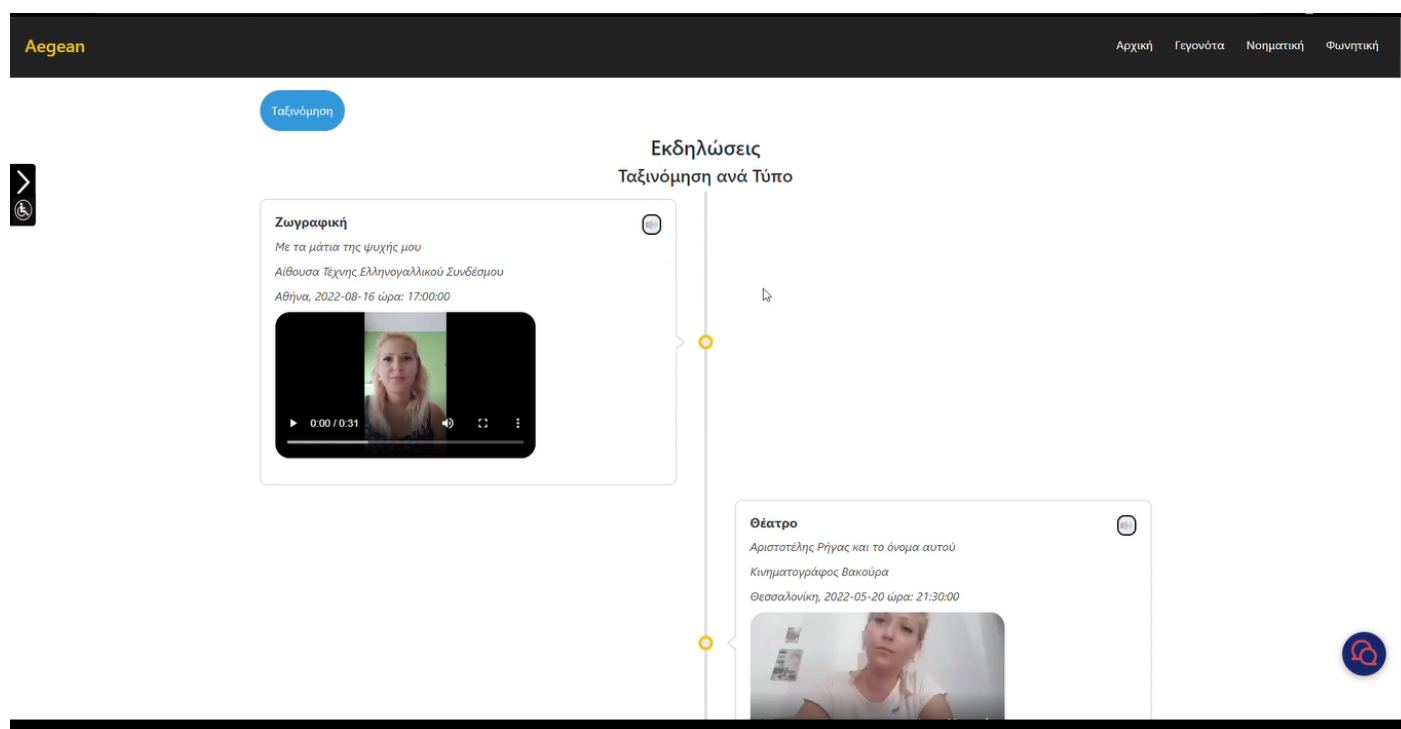
Το RASA είναι ένα πλαίσιο αλγορίθμων μηχανικής μάθησης που βοηθά στην κατασκευή ρομπότ συνομιλίας. Αυτός ο βοηθός που βασίζεται στην τεχνητή νοημοσύνη αρχικά, κατανοεί τα μηνύματα των χρηστών και τις προθέσεις τους, στη συνέχεια πραγματοποιεί συνομιλίες και τέλος δίνει απαντήσεις στον χρήστη με πληροφορίες/δεδομένα, έχοντας συνδεθεί με κανάλια μηνυμάτων και APIs. Έχει επίσης τη δυνατότητα να αποθηκεύει τις πληροφορίες του χρήστη που μπορούν να βοηθήσουν στην περαιτέρω εξέλιξη της συνομιλίας αλλά και να βοηθήσουν σε μελλοντικές συνομιλίες.

Το πλαίσιο μηχανικής μάθησης RASA αποτελείται από δύο κύριους άξονες. Το πρώτο είναι το Rasa NLU (Natural language Understanding) και το Rasa Core (διαχείριση διαλόγου), όπου το δεύτερο είναι το μοντέλο. Το μοντέλο NLU είναι ένας αλγόριθμος που βοηθάει στην εκπαίδευση του βοηθού μας προκειμένου να μαθαίνει τη σημασία των λέξεων και την πρόθεση πίσω από αυτές. Το Core είναι υπεύθυνο για την ανταπόκριση του βοηθού στις εισόδους του χρήστη και κατευθύνει τη συνομιλία μέχρι να ολοκληρωθούν οι ενέργειές του.

4. ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

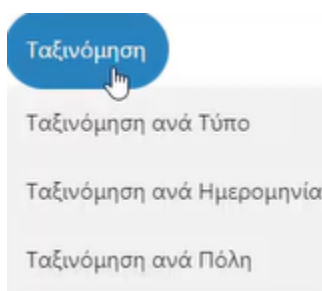
Παρακάτω, παρουσιάζεται η υλοποίηση ολόκληρης της εφαρμογής, αναλύοντας ειδικότερα την διαδικασία που ακολουθήθηκε. Οι ενότητες που ακολουθούν παρουσιάζουν την διαδικασία δημιουργίας της εφαρμογής, του μοντέλου αναγνώρισης νοηματικής γλώσσας αλλά και του chatbot. Τέλος, γίνεται αναφορά και στη δημοσίευση της εφαρμογής σε πάροχο φιλοξενίας ιστού.

Αρχικά, η εφαρμογή παρουσιάζει ένα ‘ημερολόγιο’ πολιτιστικών γεγονότων όπου η πληροφορία δίνεται με τη μορφή κειμένου, ήχου αλλά και βίντεο ανάλογα με τις απαιτήσεις του εκάστοτε χρήστη. Συγκεκριμένα, τα πολιτιστικά γεγονότα χωρίζονται - προς το παρόν - στις επιμέρους κατηγορίες: **Ζωγραφική, Θέατρο, Ομιλία, Σεμινάριο, Συναυλία** και **Φεστιβάλ**. Η δυνατότητα ηχητικής ενημέρωσης παρέχεται με το πάτημα σχετικού κουμπιού. Έτσι, ο χρήστης μπορεί να βρει εύκολα με βάση τα ενδιαφέροντά του, τα γεγονότα που συμβαίνουν στην πόλη του, όπως φαίνεται στο παρακάτω παράδειγμα (Εικόνα 6).



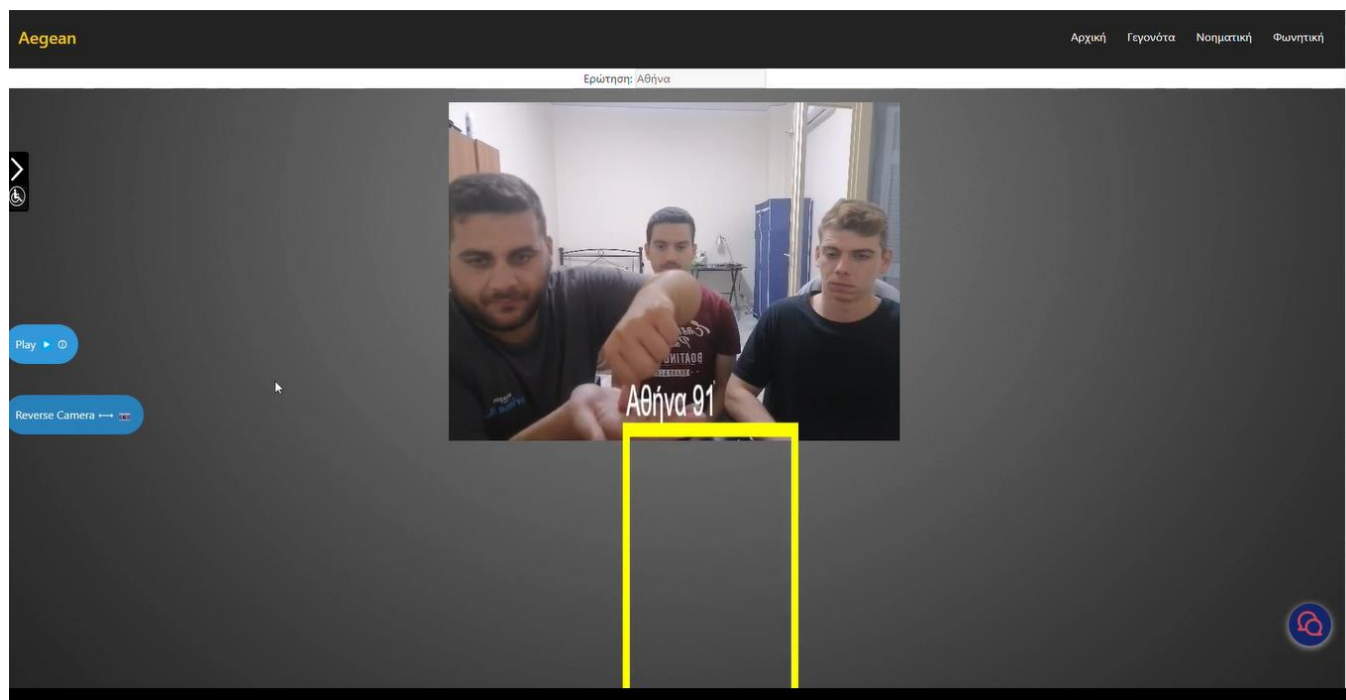
Εικόνα 6. Καρτέλα Γεγονότα

Στην καρτέλα **Γεγονότα** υπάρχει ειδικό κουμπί ταξινόμησης προς διευκόλυνση του χρήστη. Η ταξινόμηση των γεγονότων μπορεί να γίνει ανά Τύπο, Ημερομηνία και Πόλη (Εικόνα 7).



Εικόνα 7. Ταξινόμηση

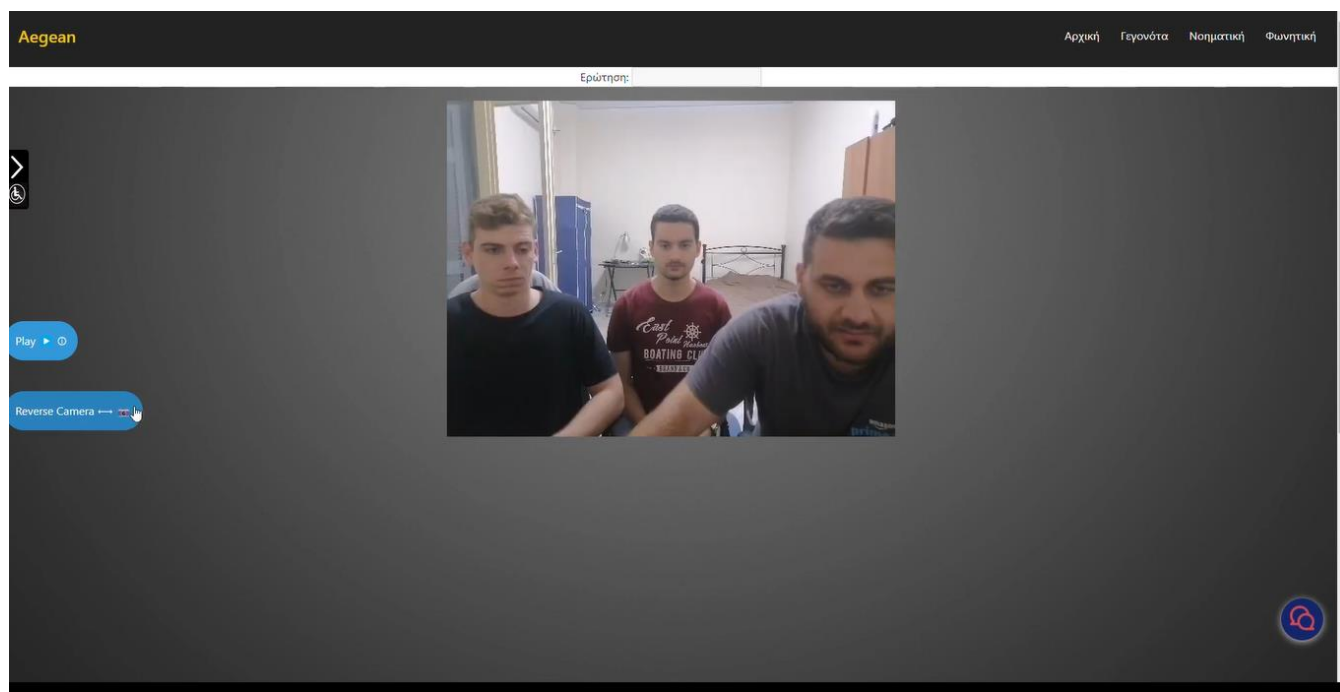
Στη συνέχεια, μέσω του μοντέλου αναγνώρισης νοηματικής γλώσσας της τεχνολογίας Python Tensorflow Object-Detection, παρέχεται η δυνατότητα ενημέρωσης του χρήστη μέσω της νοηματικής γλώσσας, αφού με το άνοιγμα της κάμεράς του (και άδειας χρήσης από τον φυλλομετρητή) στέλνοντας νοήματα να λαμβάνει και τις αντίστοιχες απαντήσεις (Εικόνα 8).



Εικόνα 8. Καρτέλα Νοηματική

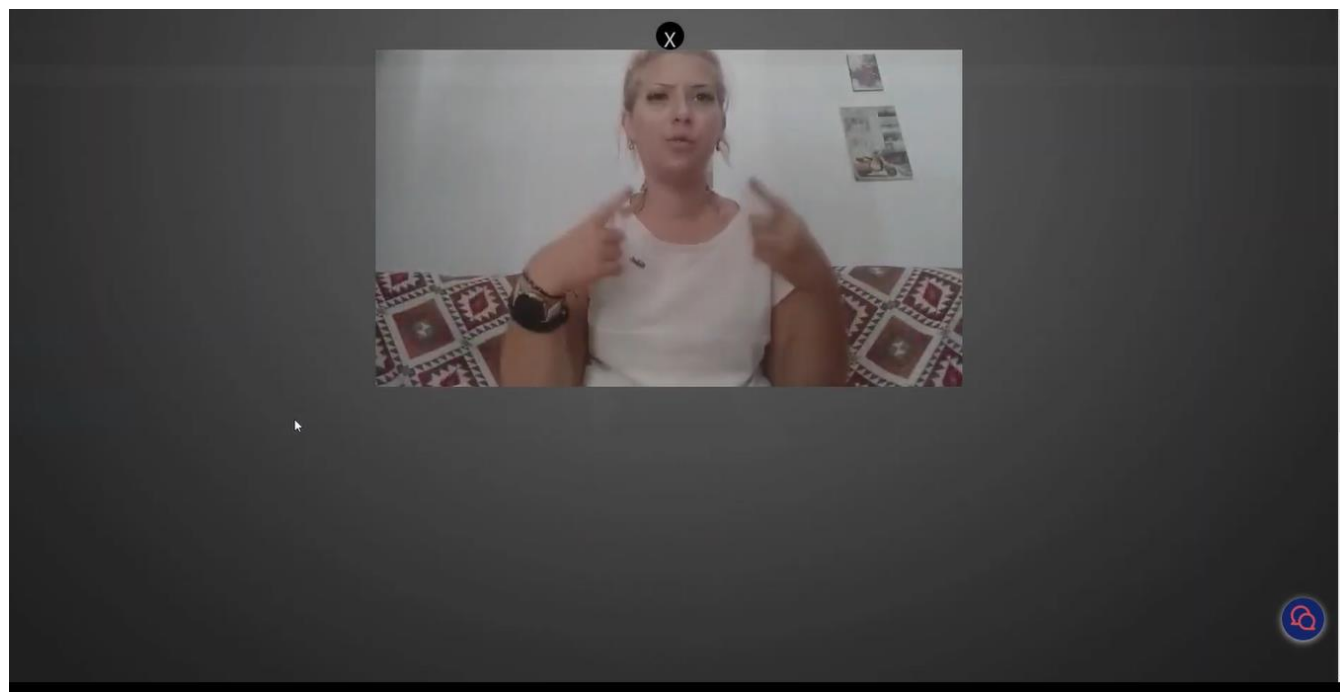
Εφαρμογή ενημέρωσης πολιτιστικών γεγονότων για AMEA

Για την καλύτερη δυνατή αναγνώριση της νοηματικής γλώσσας από το μοντέλο δίνεται η δυνατότητα αντικατοπτρισμού της κάμερας όπως φαίνεται παρακάτω (Εικόνα 9):



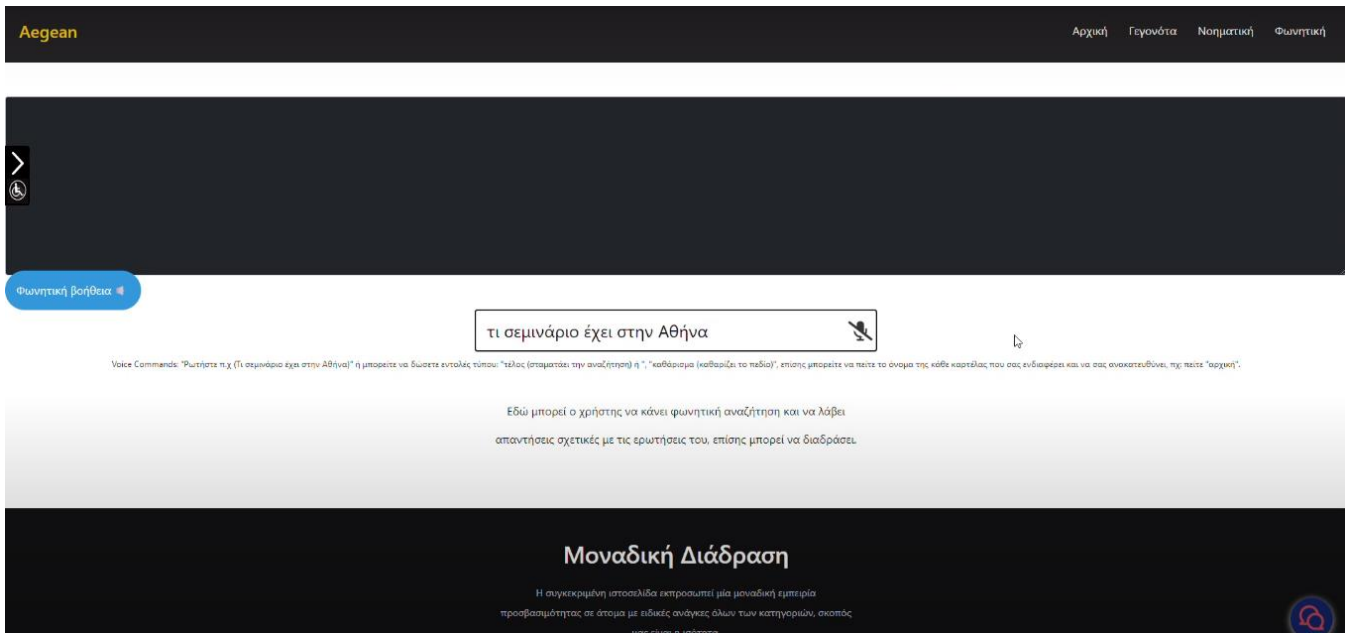
Εικόνα 9. Αντικατοπτρισμός κάμερας

Με την αναγνώριση της νοηματικής γλώσσας, η απάντηση στο αίτημα του χρήστη γίνεται με τον ίδιο τρόπο (Εικόνα 10).



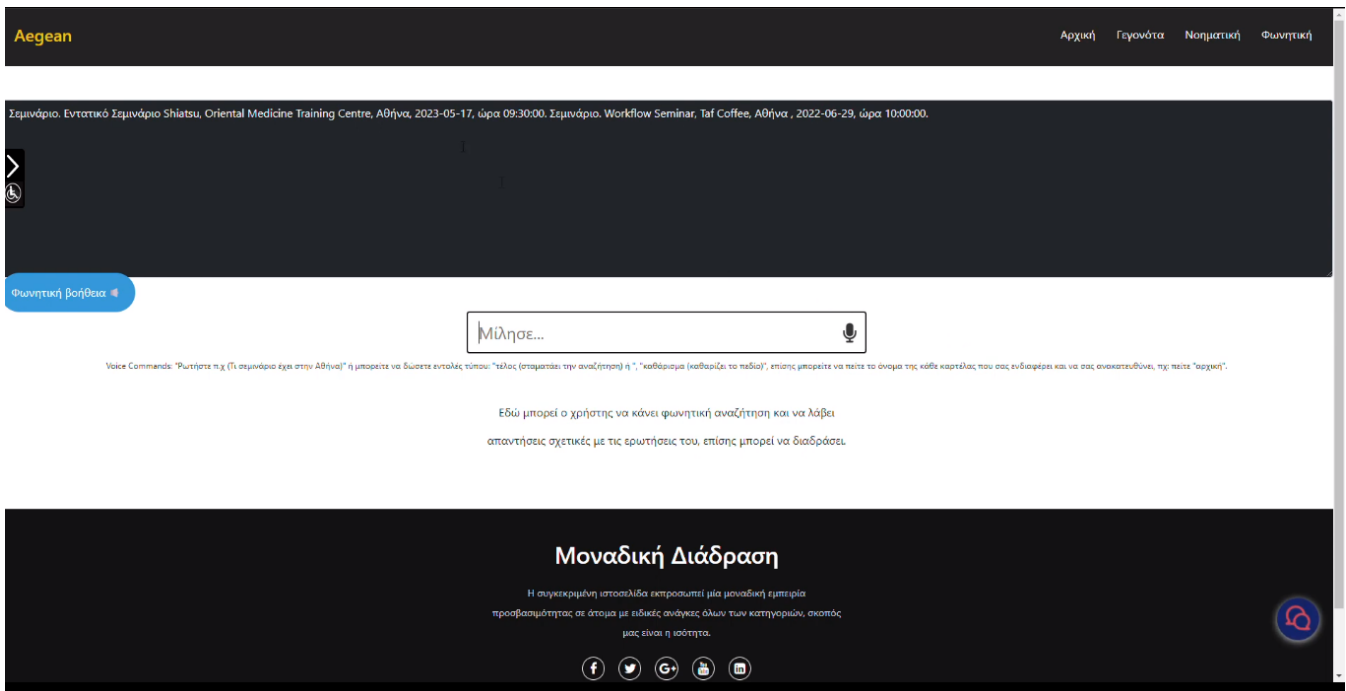
Εικόνα 10. Αποτέλεσμα Νοηματικής

Επιπροσθέτως, από την καρτέλα **Φωνητική** και με το άνοιγμα του μικροφώνου στο πεδίο αναζήτησης, ο χρήστης μπορεί μέσω φωνητικών εντολών και ερωτήσεων να ενημερωθεί για τα πολιτιστικά γεγονότα (Εικόνα 11).



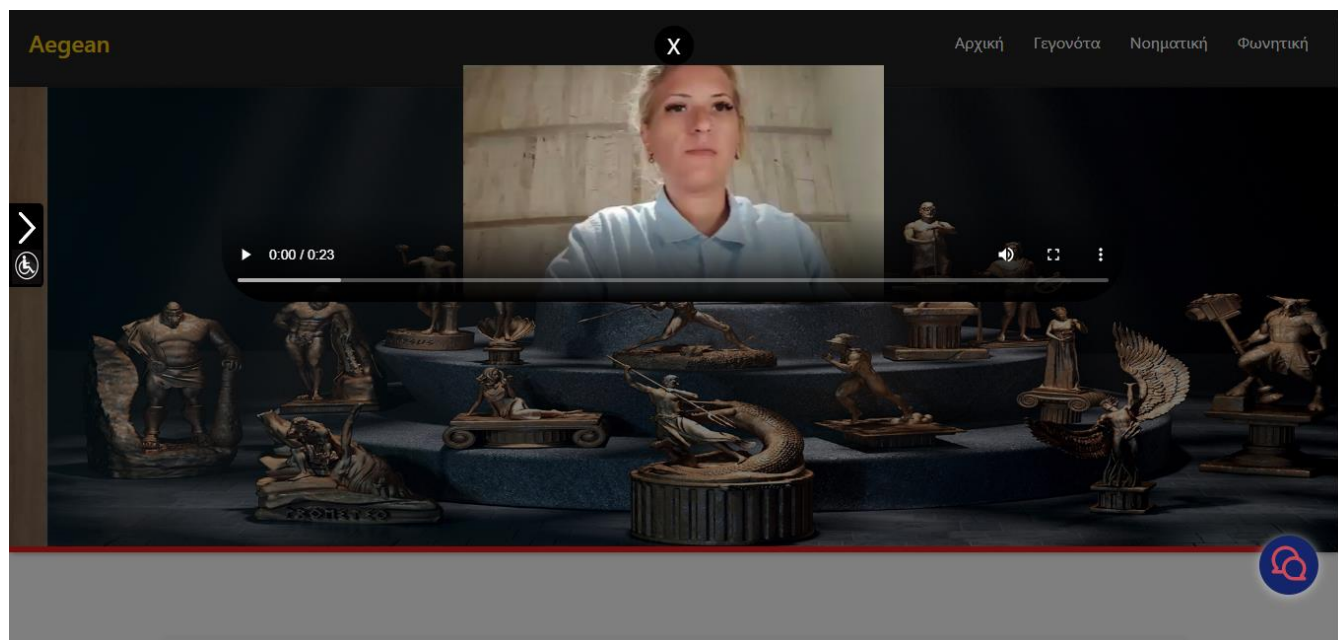
Εικόνα 11. Καρτέλα Φωνητική

Αφού τελειώσει η ερώτηση, με τη λέξη τέλος ξεκινάει η αναζήτηση για τα πολιτιστικά γεγονότα. Οι απαντήσεις είναι με φωνητική αλλά και με κείμενο, όπως φαίνεται στην εικόνα (Εικόνα 12):



Εικόνα 12. Αποτελέσματα Φωνητικής

Τέλος, είναι εφικτή και η εναλλαγή των καρτελών μέσω της φωνητικής καθώς υπαγορεύοντας το όνομα της εκάστοτε καρτέλας θα γίνει ανακατεύθυνση π.χ. Αρχική (Εικόνα 13).



Εικόνα 13. Ανακατεύθυνση Αρχικής

4.1 Υλοποίηση του μοντέλου αναγνώρισης νοηματικής γλώσσας της τεχνολογίας Python - TensorFlow

Το συγκεκριμένο σύστημα παρουσιάζει την ανάπτυξη ενός ανιχνευτή νοηματικής γλώσσας σε πραγματικό χρόνο με τη χρήση ενός TensorFlow API (Application Programming Interface) ανιχνευτή αντικειμένων και την εκπαίδευσή του μέσω της μάθησης μεταφοράς για το υπάρχον σύνολο δεδομένων [3].

Μετά την απόκτηση δεδομένων, δημιουργείται ένας επισημασμένος χάρτης, ο οποίος είναι μια αναπαράσταση όλων των αντικειμένων εντός του μοντέλου, δηλαδή περιέχει την ετικέτα κάθε σημείου (ελληνική λέξη) μαζί με το id (identity) τους. Οι εγγραφές TF (TensorFlow) των δεδομένων εκπαίδευσης και των δεδομένων δοκιμής στη συνέχεια δημιουργούνται με τη χρήση της `generate_tfrecord`, η οποία χρησιμοποιείται για την εκπαίδευση της ανίχνευσης αντικειμένων TensorFlow API.

Η εγγραφή TF είναι η δυαδική μορφή αποθήκευσης του TensorFlow. Η χρήση δυαδικών αρχείων για την αποθήκευση των δεδομένων επηρεάζει σημαντικά την απόδοση της εισαγωγής και κατά συνέπεια τον χρόνο εκπαίδευσης του μοντέλου. Καταλαμβάνει λιγότερο χώρο, αντιγράφει γρήγορα και μπορεί να διαβαστεί

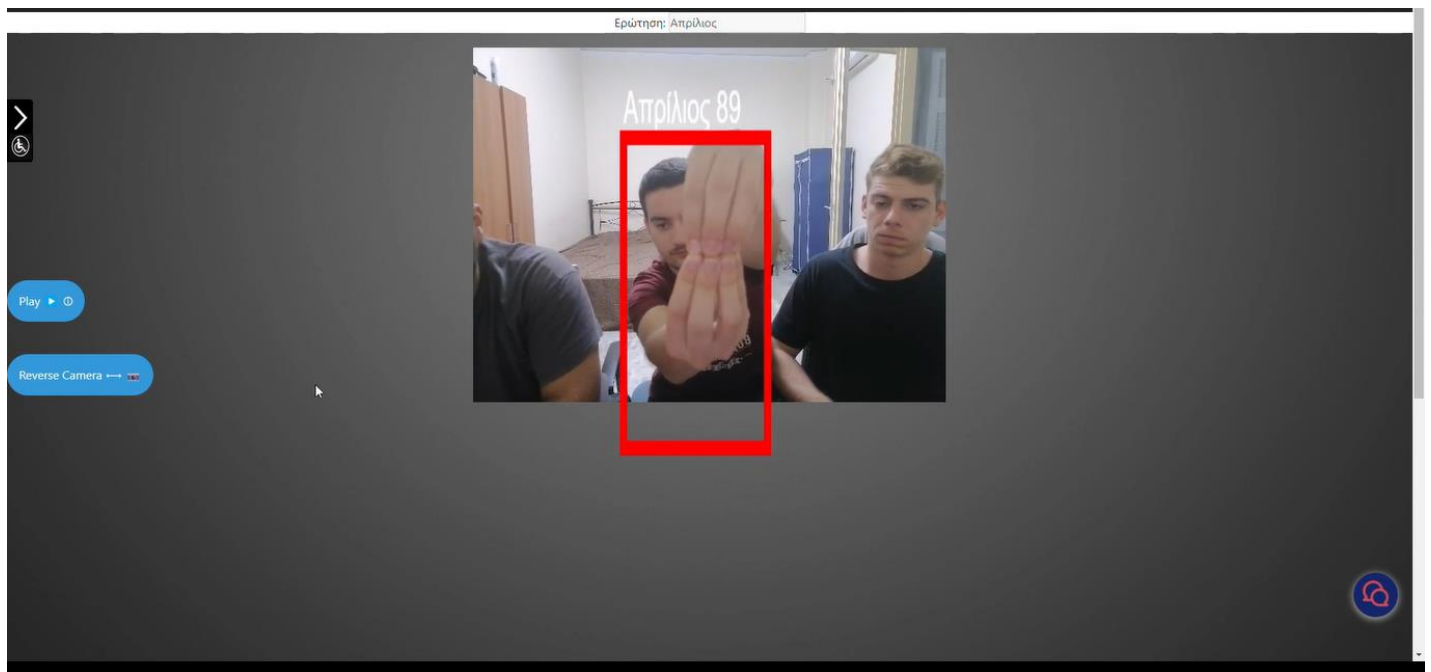
αποτελεσματικά από το δίσκο.

Το πλαίσιο ανοικτού κώδικα, TensorFlow object detection API καθιστά εύκολη την εκπαίδευση και την ανάπτυξη ενός μοντέλου ανίχνευσης αντικειμένων. Η διαμόρφωση του προ-εκπαιδευμένου μοντέλου ρυθμίζεται και στη συνέχεια ενημερώνεται για τη μάθηση μεταφοράς και για την εκπαίδευσή του από το σύνολο δεδομένων που δημιουργήθηκε. Μετά τη ρύθμιση και την ενημέρωσή του, το μοντέλο εκπαιδεύτηκε σε 18000 βήματα. Για τη δημιουργία και το ανέβασμα του μοντέλου (API) χρησιμοποιήθηκε το λογισμικό npm-package (Node Package Manager) και η γλώσσα της React.

Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο έχει κάποιες απώλειες όπως απώλεια ταξινόμησης, απώλεια κανονικοποίησης και απώλεια εντοπισμού. Επιπλέον, το εκπαιδευμένο μοντέλο φορτώνεται από το τελευταίο σημείο ελέγχου το οποίο δημιουργείται κατά την εκπαίδευση. Με αυτόν τον τρόπο ολοκληρώνεται η διαδικασία δημιουργίας του μοντέλου και είναι έτοιμο για χρήση.

Η ανίχνευση των νοημάτων γίνεται σε πραγματικό χρόνο και πάλι με τη χρήση της OpenCV και της Web κάμερας. Το σύστημα ανιχνεύει τα νοήματα σε πραγματικό χρόνο και μεταφράζει τι σημαίνει κάθε χειρονομία στα ελληνικά, όπως φαίνεται στην παρακάτω εικόνα.

Το σύστημα δοκιμάζεται σε πραγματικό χρόνο όπου μέσω ενδείξεων ενημερώνει το χρήστη για το ποσοστό εγκυρότητας του κάθε νοήματος (ελληνική λέξη), δηλαδή πόσο σίγουρο είναι το σύστημα για την αναγνώριση ενός νοήματος (ελληνικής λέξης) (Εικόνα 14) [4].

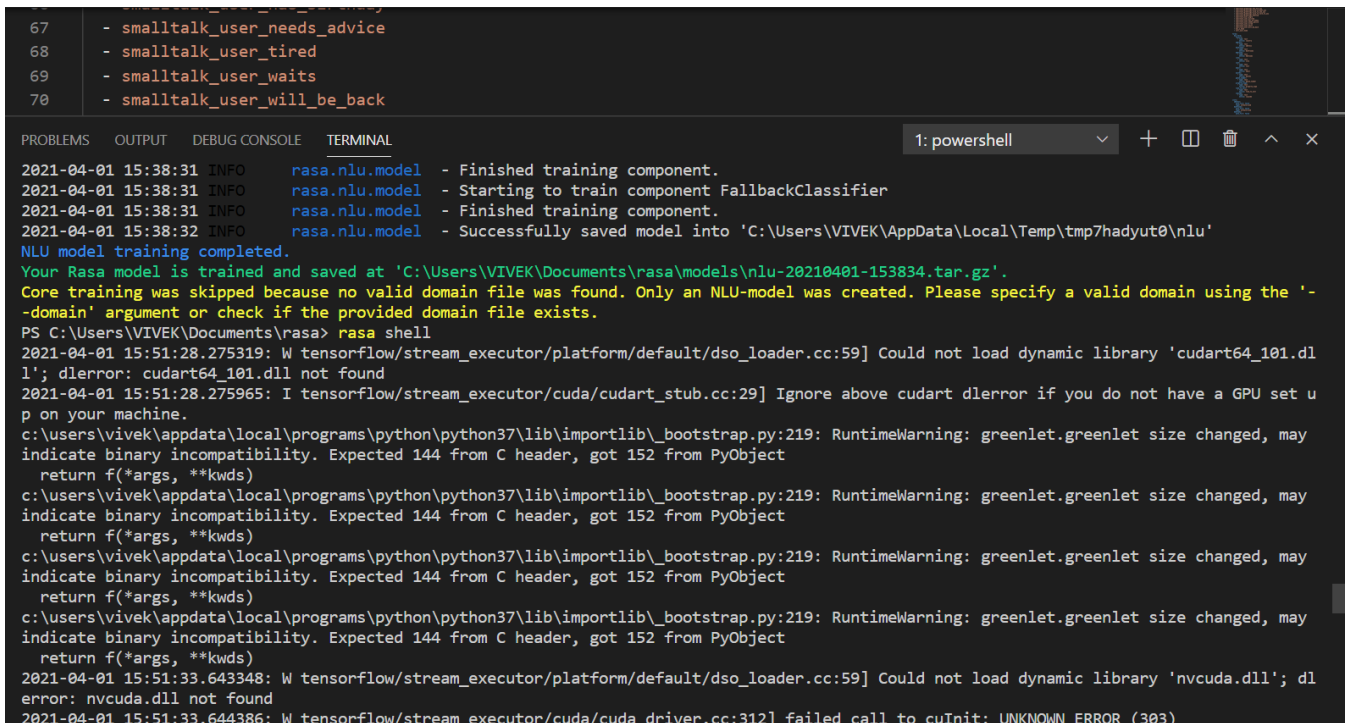


Εικόνα 14. Ποσοστό εγκυρότητας

4.2 Υλοποίηση chatbot της τεχνολογίας Python και Rasa Framework

1. Εγκατάσταση του Rasa και της Python αλλά και ρύθμιση του έργου:

Σε αυτό το βήμα γίνεται όλη η απαραίτητη εγκατάσταση, διαμόρφωση και οι εισαγωγές βιβλιοθηκών. Στη συνέχεια, με τη χρήση του Rasa CLI (Command Line Interface) ρυθμίζεται το έργο (Εικόνα 15).



```

67 - smalltalk_user_needs_advice
68 - smalltalk_user_tired
69 - smalltalk_user_waits
70 - smalltalk_user_will_be_back

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell
2021-04-01 15:38:31 INFO rasa.nlu.model - Finished training component.
2021-04-01 15:38:31 INFO rasa.nlu.model - Starting to train component FallbackClassifier
2021-04-01 15:38:31 INFO rasa.nlu.model - Finished training component.
2021-04-01 15:38:32 INFO rasa.nlu.model - Successfully saved model into 'C:\Users\VIVEK\AppData\Local\Temp\tmp7hadyut0\nlu'
NLU model training completed.
Your Rasa model is trained and saved at 'C:\Users\VIVEK\Documents\rasa\models\nlu-20210401-153834.tar.gz'.
Core training was skipped because no valid domain file was found. Only an NLU-model was created. Please specify a valid domain using the '-domain' argument or check if the provided domain file exists.
PS C:\Users\VIVEK\Documents\rasa> rasa shell
2021-04-01 15:51:28.275319: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cudart64_101.dll'; dlderror: cudart64_101.dll not found
2021-04-01 15:51:28.275965: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
c:\users\vivek\appdata\local\programs\python\python37\lib\importlib\_bootstrap.py:219: RuntimeWarning: greenlet.greenlet size changed, may indicate binary incompatibility. Expected 144 from C header, got 152 from PyObject
return f(*args, **kwargs)
c:\users\vivek\appdata\local\programs\python\python37\lib\importlib\_bootstrap.py:219: RuntimeWarning: greenlet.greenlet size changed, may indicate binary incompatibility. Expected 144 from C header, got 152 from PyObject
return f(*args, **kwargs)
c:\users\vivek\appdata\local\programs\python\python37\lib\importlib\_bootstrap.py:219: RuntimeWarning: greenlet.greenlet size changed, may indicate binary incompatibility. Expected 144 from C header, got 152 from PyObject
return f(*args, **kwargs)
c:\users\vivek\appdata\local\programs\python\python37\lib\importlib\_bootstrap.py:219: RuntimeWarning: greenlet.greenlet size changed, may indicate binary incompatibility. Expected 144 from C header, got 152 from PyObject
return f(*args, **kwargs)
2021-04-01 15:51:33.643348: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2021-04-01 15:51:33.644386: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)

```

Εικόνα 15. Rasa CLI

2. Χρήση του συνόλου δεδομένων για την εκπαίδευση του μοντέλου:

Σε αυτό το βήμα χρησιμοποιείται το σύνολο δεδομένων για την εκπαίδευση του μοντέλου. Γίνεται εισαγωγή των δεδομένων εκπαίδευσης με τη μορφή οντοτήτων σε συγκεκριμένο αρχείο που ονομάζεται NLU.yml (Εικόνα 16).

```
data > ! nlu.yml
1  version: "3.1"
2
3  nlu:
4    - intent: greet
5      - examples: |
6        - γειά
7        - γειά σου
8        - γειά σας
9        - καλημέρα
10       - καλησπέρα
11       - καλημέρα σας
12       - καλησπέρα σας
13       - χαίρεται
14       - την καλησπέρα μου
15     -
16    - intent: goodbye
17      - examples: |
18        - τα λέμε
19        - τα λέμε μετά
20        - τα λέμε αργότερα
21        - αντίο
22        - αντίο σας
23        - καλή νύχτα
24        - καλή νύχτα σας
25        - καλή σας νύχτα
26        - καλή σας μέρα
27        - εις το επανιδείν
28        - καλή συνέχεια
29    -
30    - intent: affirm
31      - examples: |
32        - ναι
33        - αμέ
34        - βεβαίως
35        - μάλιστα
36        - φυσικά
37        - σωστά
38        - ok
39        - ok
40    -
41    - intent: deny
42      - examples: |
43        - όχι
44        - ποτέ
45        - με τίποτα
46        - δεν νομίζω
47        - δεν το νομίζω
48        - σε καμία περίπτωση
```

Εικόνα 16. Αρχείο NLU.yml

3. Προσαρμοσμένες ενέργειες/μορφές και δεδομένα τομέα:

Σε αυτό το βήμα χρησιμοποιείται η Python για τη δημιουργία προσαρμοσμένων ενεργειών.

Αυτές οι προσαρμοσμένες ενέργειες θα βοηθήσουν το chatbot να ανταποκριθεί σε ένα αίτημα του χρήστη. Αυτές οι προσαρμοσμένες ενέργειες αναφέρονται σε ένα αρχείο που ονομάζεται actions.py (Εικόνα 17).

```

61 ..... if (latEntitType=="Seminar"):
62 .....
63 .....     dispatcher.utter_message("Κατηγορία: {}".format(latEntitType))
64 .....     sqltypevar='Σεμινάριο'
65 .....     sqlQuery="SELECT * FROM events WHERE type='%s'" % (sqltypevar)
66 .....
67 ..... elif (latEntitType=="Speech"):
68 .....
69 .....     dispatcher.utter_message("Κατηγορία: {}".format(latEntitType))
70 .....     sqltypevar='Ομιλία'
71 .....     sqlQuery="SELECT * FROM events WHERE type='%s'" % (sqltypevar)
72 .....
73 ..... elif (latEntitType=="Theatre"):
74 .....
75 .....     dispatcher.utter_message("Κατηγορία: {}".format(latEntitType))
76 .....     sqltypevar='Θέατρο'
77 .....     sqlQuery="SELECT * FROM events WHERE type='%s'" % (sqltypevar)
78 .....
79 ..... elif (latEntitType=="Painting"):
80 .....
81 .....     dispatcher.utter_message("Κατηγορία: {}".format(latEntitType))
82 .....     sqltypevar='Ζωγραφική'
83 .....     sqlQuery="SELECT * FROM events WHERE type='%s'" % (sqltypevar)
84 .....
85 ..... elif (latEntitType=="Concert"):
86 .....
87 .....     dispatcher.utter_message("Κατηγορία: {}".format(latEntitType))
88 .....     sqltypevar='Συναυλία'
89 .....     sqlQuery="SELECT * FROM events WHERE type='%s'" % (sqltypevar)
90 .....
91 ..... elif (latEntitType=="Festival"):
92 .....
93 .....     dispatcher.utter_message("Κατηγορία: {}".format(latEntitType))
94 .....     sqltypevar='Φεστιβάλ'
95 .....     sqlQuery="SELECT * FROM events WHERE type='%s'" % (sqltypevar)
96 .....
97 .....
98 ..... elif (latEntitValue=="Αθήνα" or latEntitValue=="αθήνα"):
99 .....
100 .....     dispatcher.utter_message("Keyword Τοποθεσίας: {}".format(latEntitValue))
101 .....     sqlcityvar='Αθήνα'
102 .....     sqlQuery="SELECT * FROM events WHERE city='%s'" % (sqlcityvar)
103 .....
104 ..... elif (latEntitValue=="Θεσσαλονίκη" or latEntitValue=="θεσσαλονίκη"):
105 .....
106 .....     dispatcher.utter_message("Location Τοποθεσίας: {}".format(latEntitValue))
107 .....     sqlcityvar='Θεσσαλονίκη'
108 .....     sqlQuery="SELECT * FROM events WHERE city='%s'" % (sqlcityvar)
109 .....
110 ..... elif (latEntitValue=="Πάτρα" or latEntitValue=="πάτρα"):
111 .....
112 .....     dispatcher.utter_message("Location Τοποθεσίας: {}".format(latEntitValue))
113 .....     sqlcityvar='Πάτρα'
114 .....     sqlQuery="SELECT * FROM events WHERE city='%s'" % (sqlcityvar)

```

Εικόνα 17. Αρχείο actions.py

Έχουμε επίσης ένα αρχείο που ονομάζεται domain.yml. Αυτό χρησιμοποιείται για την αναφορά του domain και τα μεταδεδομένα που σχετίζονται με το domain για το chatbot (Εικόνα 18).

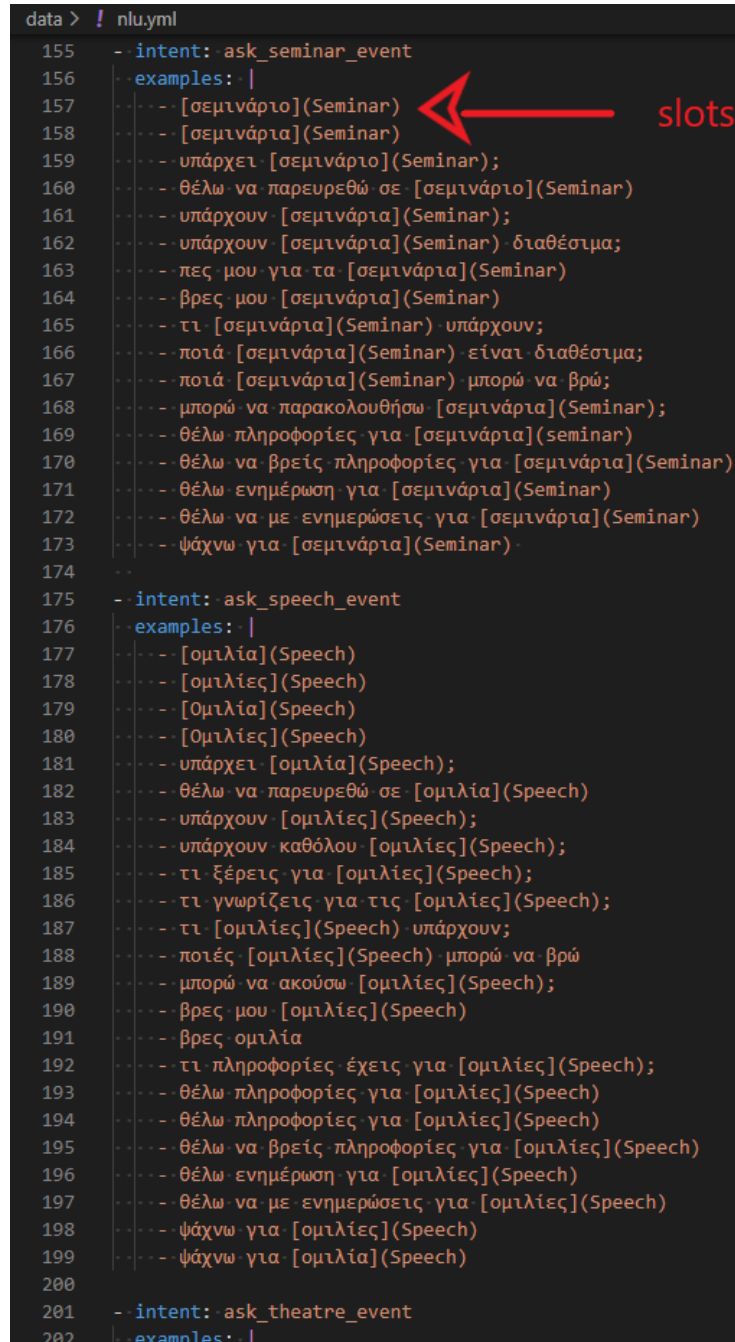
```
! domain.yml
109
110 utter_button_categories:
111   --text: "Παρακαλώ επέλεξε εκδήλωση με βάση τις κατηγορίες"
112   --buttons:
113     --title: "Τοποθεσία"
114     --payload: "/by_location"
115     --title: "Τύπος"
116     --payload: "/by_type"
117     --title: "Ημερομηνία"
118     --payload: "/by_date"
119     --title: "Έτος"
120     --payload: "/by_year"
121
122 #buttons-with-slot-value
123 utter_button_locationEvents:
124   --text: "Επέλεξε την επιθυμητή τοποθεσία"
125   --buttons:
126     --title: "Αθήνα"
127     --payload: "/ask_athens_event{{\"Location\":\"Αθήνα\"}}'"
128     --title: "Θεσσαλονίκη"
129     --payload: "/ask_thessaloniki_event{{\"Location\":\"Θεσσαλονίκη\"}}'"
130     --title: "Πάτρα"
131     --payload: "/ask_patras_event{{\"Location\":\"Πάτρα\"}}'"
132
133
134 utter_button_typeEvents:
135   --text: "Επέλεξε τον τύπο εκδήλωσης που σε ενδιαφέρει"
136   --buttons:
137     --title: "Σεμινάριο"
138     --payload: "/ask_seminar_event{{\"Seminar\":\"σεμινάριο\"}}'"
139     --title: "Ομιλία"
140     --payload: "/ask_speech_event{{\"Speech\":\"ομιλία\"}}'"
141     --title: "Ζωγραφική"
142     --payload: "/ask_painting_event{{\"Painting\":\"ζωγραφική\"}}'"
143     --title: "Θέατρο"
144     --payload: "/ask_theatre_event{{\"Theatre\":\"θέατρο\"}}'"
145     --title: "Συναυλία"
146     --payload: "/ask_concert_event{{\"Concert\":\"συναυλία\"}}'"
147     --title: "Φεστιβάλ"
148     --payload: "/ask_festival_event{{\"Festival\":\"φεστιβάλ\"}}'"
149
150 utter_button_yearEvents:
151   --text: "Επέλεξε το επιθυμητό έτος"
152   --buttons:
153     --title: "2022"
154     --payload: "/ask_year_event{{\"Year\":\"2022\"}}'"
155     --title: "2023"
156     --payload: "/ask_year_event{{\"Year\":\"2023\"}}'"

```

Εικόνα 18. Αρχείο domain.yml

4. Ενσωμάτωση με το API και τη βάση δεδομένων:

Όταν λαμβάνεται ερώτημα ή είσοδος μέσω RASA από τον τελικό χρήστη, προβλέπει τις τιμές των οντοτήτων και των προθέσεων από το μήνυμα, όλος αυτός ο χειρισμός γίνεται από το RASA NLU (Εικόνα 19).



```
data > ! nlu.yml
155 --intent: ask_seminar_event
156 --examples: |
157     [σεμινάριο](Seminar)
158     [σεμινάρια](Seminar)
159     υπάρχει [σεμινάριο](Seminar);
160     θέλω να παρευρεθώ σε [σεμινάριο](Seminar)
161     υπάρχουν [σεμινάρια](Seminar);
162     υπάρχουν [σεμινάρια](Seminar) διαθέσιμα;
163     πες μου για τα [σεμινάρια](Seminar)
164     βρες μου [σεμινάρια](Seminar)
165     τι [σεμινάρια](Seminar) υπάρχουν;
166     ποιά [σεμινάρια](Seminar) είναι διαθέσιμα;
167     ποιά [σεμινάρια](Seminar) μπορώ να βρώ;
168     μπορώ να παρακολουθήσω [σεμινάρια](Seminar);
169     θέλω πληροφορίες για [σεμινάρια](seminar)
170     θέλω να βρείς πληροφορίες για [σεμινάρια](Seminar)
171     θέλω ενημέρωση για [σεμινάρια](Seminar)
172     θέλω να με ενημερώσεις για [σεμινάρια](Seminar)
173     ψάχνω για [σεμινάρια](Seminar)
174 --
175 --intent: ask_speech_event
176 --examples: |
177     [ομιλία](Speech)
178     [ομιλίες](Speech)
179     [Ομιλία](Speech)
180     [Ομιλίες](Speech)
181     υπάρχει [ομιλία](Speech);
182     θέλω να παρευρεθώ σε [ομιλία](Speech)
183     υπάρχουν [ομιλίες](Speech);
184     υπάρχουν καθόλου [ομιλίες](Speech);
185     τι ξέρεις για [ομιλίες](Speech);
186     τι γνωρίζεις για τις [ομιλίες](Speech);
187     τι [ομιλίες](Speech) υπάρχουν;
188     ποιές [ομιλίες](Speech) μπορώ να βρώ
189     μπορώ να ακούσω [ομιλίες](Speech);
190     βρες μου [ομιλίες](Speech)
191     βρες ομιλία
192     τι πληροφορίες έχεις για [ομιλίες](Speech);
193     θέλω πληροφορίες για [ομιλίες](Speech)
194     θέλω πληροφορίες για [ομιλίες](Speech)
195     θέλω να βρείς πληροφορίες για [ομιλίες](Speech)
196     θέλω ενημέρωση για [ομιλίες](Speech)
197     θέλω να με ενημερώσεις για [ομιλίες](Speech)
198     ψάχνω για [ομιλίες](Speech)
199     ψάχνω για [ομιλία](Speech)
200 --
201 --intent: ask_theatre_event
202 --examples: |
```

Εικόνα 19. Αρχείο NLU.yml (2)

5. Διαμόρφωση Dockerfile (Okteto):

Εδώ δημιουργείται το Dockerfile (docker-compose.yml) προκειμένου να ρυθμιστούν οι υπηρεσίες του chatbot. (Εικόνα 20). Στη συνέχεια, με μία μόνο εντολή, ξεκινούν όλες οι υπηρεσίες από τη διαμόρφωσή αυτού του αρχείου. Δηλαδή, η εκτέλεση των ενεργειών γίνεται αυτόματα από έναν ‘εικονικό’ χρήστη (Εικόνα 21).

```
Dockerfile > ...
1  FROM python:3.7.3-stretch AS BASE
2
3  RUN apt-get update \
4  |>&& apt-get --assume-yes --no-install-recommends install \
5  |> build-essential \
6  |> curl \
7  |> git \
8  |> jq \
9  |> libgomp1 \
10 |> vim
11
12 WORKDIR /app
13
14 # upgrade pip version
15 RUN pip install --no-cache-dir --upgrade pip
16
17 RUN pip install rasa==3.1.0
18 RUN pip install mysql
19 RUN pip install mysql-connector-python==8.0.28
20
21 ADD config.yml config.yml
22 ADD domain.yml domain.yml
23 ADD credentials.yml credentials.yml
24 ADD endpoints.yml endpoints.yml
```

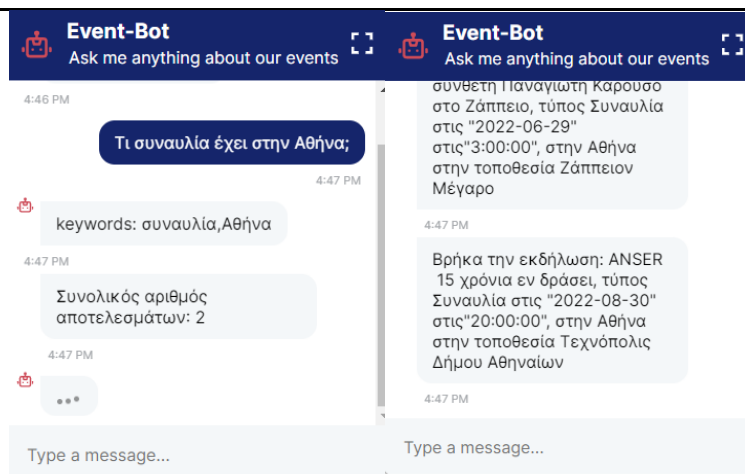
Εικόνα 20. Dockerfile

```
docker-compose.yml
1  version: '3.4'
2  services:
3
4
5  -- rasa-server:
6  -- image: rasa-bot:latest
7  -- working_dir: /app
8  -- build: "./"
9  -- restart: always
10 -- volumes:
11 -- -- ./actions:/app/actions
12 -- -- ./data:/app/data
13 -- command: bash -c "rm -rf models/* && rasa train && rasa run --enable-api -p 5006 --cors "*" --debug"
14 -- ports:
15 -- -- '5007:5006'
16 -- public: true
17 -- networks:
18 -- -- all
19
20 -- rasa-actions-server:
21 -- image: rasa-bot:latest
22 -- working_dir: /app
23 -- build: "./"
24 -- restart: always
25 -- volumes:
26 -- -- ./actions:/app/actions
27 -- command: ["rasa", "run", "actions"]
28 -- ports:
29 -- -- '5055:5055'
30 -- networks:
31 -- -- all
32
33
34
35 networks:
36 -- all:
37 -- driver: bridge
38 -- driver_opts:
39 -- -- com.docker.network.enable_ipv6: "true"
```

Εικόνα 21. Αρχείο docker-compose.yml

6. Αξιολόγηση/δοκιμή του μοντέλου:

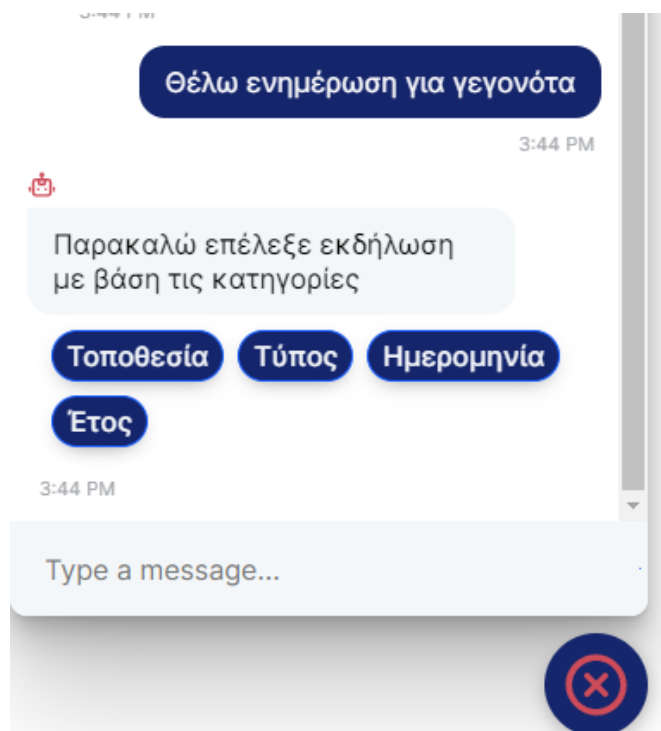
Αυτό είναι το βήμα όπου δοκιμάζεται το μοντέλο με τα δεδομένα δοκιμής. Η δοκιμή μπορεί να γίνει με τη χρήση του CLI. Κατά τη δοκιμή απλώς παίρνονται δειγματικά δεδομένα και δίνονται ως είσοδο στο μοντέλο για να διαπιστωθεί αν λαμβάνεται η αναμενόμενη έξοδος και με ποια ακρίβεια (Εικόνα 22) [5].



Εικόνα 22. Δοκιμή chatbot

4.2.1 Chatbot Buttons

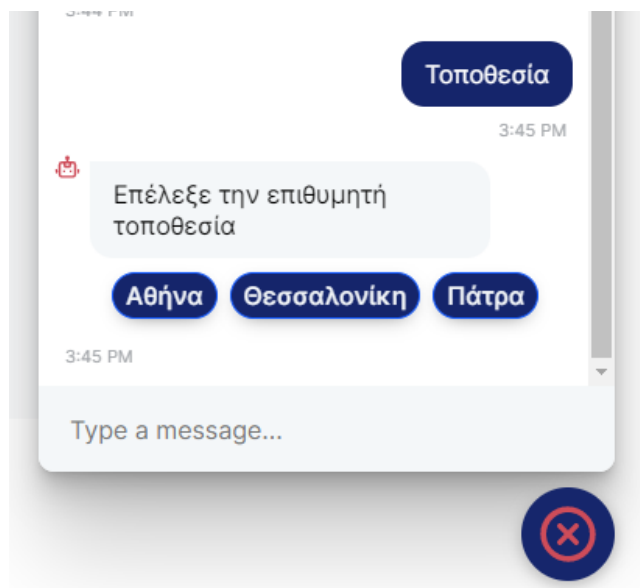
Με σκοπό την καλύτερη δυνατή εύρεση πολιτιστικών γεγονότων, δημιουργήθηκαν στο chatbot κουμπιά τα οποία δίνουν στο χρήστη κάποιες επιλογές σε περίπτωση που έχει κάνει μία γενική αναζήτηση σε γεγονότα όπως η παρακάτω (Εικόνα 23):



Εικόνα 23. Κουμπιά Γεγονότα

Όπως φαίνεται στην παραπάνω εικόνα (Εικόνα 23) μετά από αίτημα του χρήστη για εύρεση γεγονότων, το chatbot του προσφέρει τέσσερις επιλογές προκειμένου να κάνει μια πιο στοχευμένη αναζήτηση με βάση την Τοποθεσία, τον Τύπο, την Ημερομηνία αλλά και το έτος των πολιτιστικών γεγονότων.

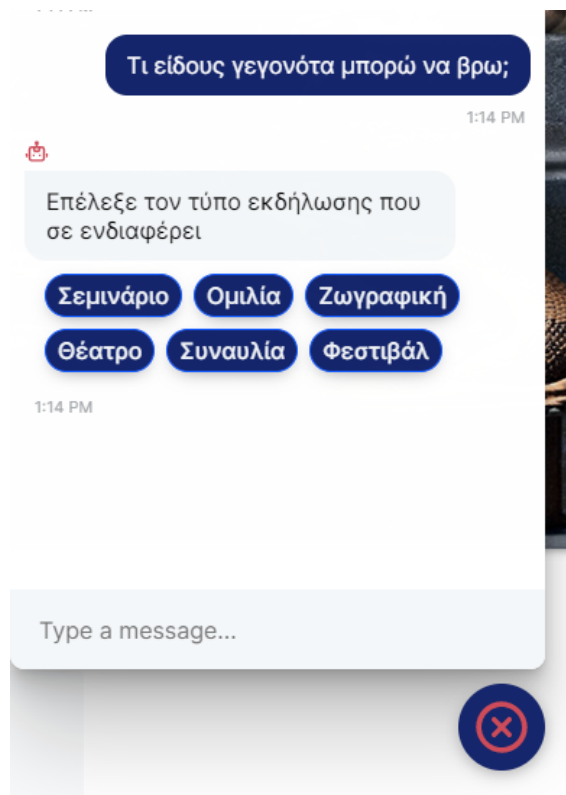
Έπειτα, μετά το πάτημα του κουμπιού Τοποθεσία καθώς και όλων των παραπάνω αντίστοιχα εμφανίζονται νέες επιλογές με σκοπό να προχωρήσει η αναζήτηση (Εικόνα 24):



Εικόνα 24. Κουμπιά Τοποθεσία

Μόλις ο χρήστης κάνει την επιλογή του σχετικά με την τοποθεσία στην προκειμένη περίπτωση θα του εμφανιστούν πολιτιστικά γεγονότα που συμβαίνουν στην τοποθεσία της επιλογής του.

Ο χρήστης μπορεί ακόμα να κάνει γενική ερώτηση σχετικά με το είδος των γεγονότων και να του εμφανιστούν τα εξής κουμπιά (Εικόνα 25):



Εικόνα 25. Κουμπιά Είδη Γεγονότων

Συνεπώς, το chatbot δίνει πρόσθετες επιλογές και για το είδος παρακινώντας το χρήστη να επιλέξει σύμφωνα με τα ενδιαφέροντα του, μία από τις διαθέσιμες επιλογές.

4.3 Λογισμικό φιλοξενίας εφαρμογής

Για τη φιλοξενία της εφαρμογής χρησιμοποιήθηκε το δωρεάν λογισμικό **Alwaysdata** λόγω ευκολότερης διαχείρισης της βάσης δεδομένων με τη δυνατότητα remote καθώς και για τον απεριόριστο χρόνο φιλοξενίας που παρέχει.

Η **Alwaysdata** είναι ένας ευρωπαϊκός πάροχος φιλοξενίας που προσφέρει διαφορετικά πακέτα σε κοινόχρηστη πλατφόρμα ή αποκλειστικούς διαχειριζόμενους διακομιστές. Παρέχει πολλές δυνατότητες, όπως αυτόματα αντίγραφα ασφαλείας, δωρεάν πιστοποιητικά SSL (Secure Sockets Layer), πρόσβαση SSH (Secure Shell) κ.λ.π.

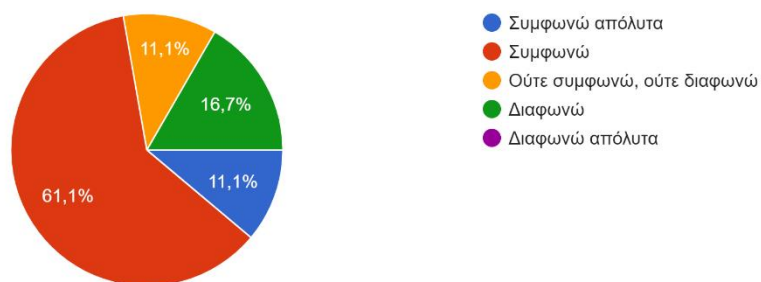
Ένα δωρεάν κοινόχρηστο πακέτο φιλοξενίας είναι διαθέσιμο για μεγέθη κάτω από 100MB χώρο στο δίσκο. Διαφορετικά, τα πακέτα ξεκινούν από 8€/μήνα [6].

Η ηλεκτρονική διεύθυνση της εφαρμογής είναι: <https://ptuxiakh.alwaysdata.net/index.php>

5. ΔΟΚΙΜΗ – ΕΛΕΓΧΟΣ – ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή που υλοποιήθηκε, ελέγχθηκε και δοκιμάστηκε σε κάθε βήμα κατά τη διάρκεια δημιουργίας της μέχρι και την τελική της μορφή. Για τη διαδικασία αξιολόγησης της εφαρμογής δημιουργήθηκε ένα ερωτηματολόγιο απευθυνόμενο σε Άτομα με Ειδικές Ανάγκες (ΑΜΕΑ) ηλικίας 18-55 ετών, όπου και υπήρξαν 18 συμμετέχοντες. Το ερωτηματολόγιο έχει να κάνει σχετικά με την ευχρηστία του συστήματος αλλά και της αξιοποίησης του από τους χρήστες.

Η εφαρμογή ήταν εύκολη στη χρήση:
18 απαντήσεις

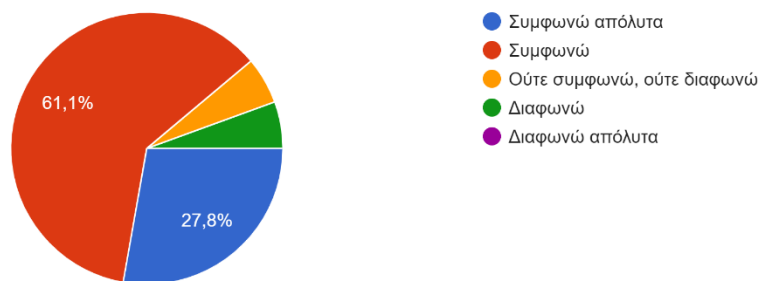


Εικόνα 26. Ερώτηση 1

Στην ερώτηση (Εικόνα 26) αν η εφαρμογή ήταν εύκολη στη χρήση οι απαντήσεις στην πλειονότητά τους ήταν θετικές με 61,1%. Ενώ το 16,7% ήταν αρνητικές, με το άλλο 22,2% να μοιράζεται εξ ημισείας σε 11,1% ανάμεσα σε αυτούς που συμφωνούν απόλυτα και σε εκείνους που ούτε συμφωνούν, ούτε διαφωνούν.

Οι λειτουργίες της εφαρμογής ήταν χρήσιμες:

18 απαντήσεις

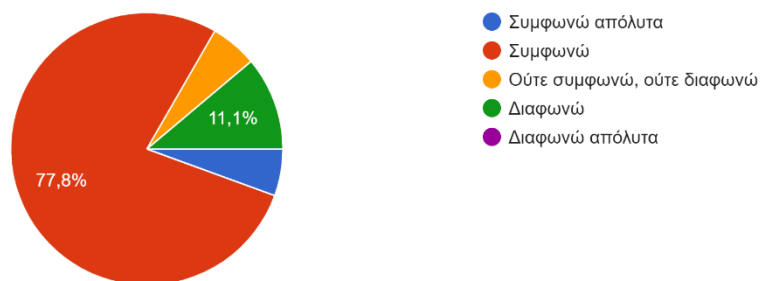


Εικόνα 27. Ερώτηση 2

Στην ερώτηση (Εικόνα 27) αν οι λειτουργίες της εφαρμογής ήταν χρήσιμες, υπήρξαν αρνητικές απαντήσεις σε ποσοστό της τάξης του 5,6%. Το σύνολο των απαντήσεων είναι θετικές, με ένα ποσοστό 27,8% να συμφωνεί απόλυτα.

Οι λειτουργίες στην εφαρμογή ήταν καλά ενσωματωμένες:

18 απαντήσεις

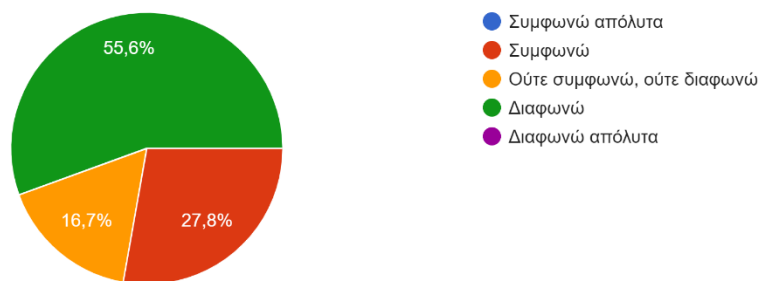


Εικόνα 28. Ερώτηση 3

Στην 3^η ερώτηση (Εικόνα 28) και στο αν ήταν καλά ενσωματωμένες οι λειτουργίες στην εφαρμογή. Από τη μία, το 77,8% συμφώνησε ότι ήταν καλά ενσωματωμένες. Από την άλλη, μόλις το 11,1% διαφώνησε.

Χρειαζόταν η συνδρομή κάποιου προσώπου προκειμένου να χρησιμοποιηθεί η εφαρμογή:

18 απαντήσεις

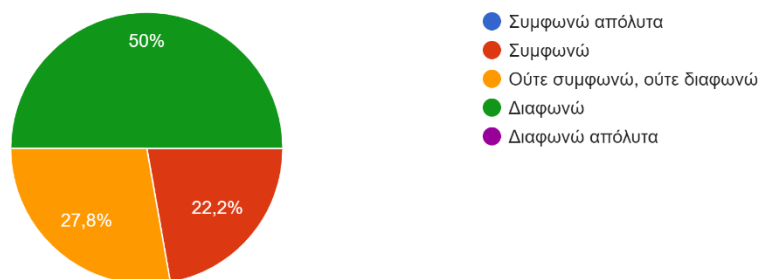


Εικόνα 29. Ερώτηση 4

Στο αν χρειαζόταν η συνδρομή κάποιου προσώπου προκειμένου να χρησιμοποιηθεί η εφαρμογή (Εικόνα 29), το 55,6% διαφώνησε ενώ το 27,8% συμφώνησε ότι χρειαζόταν. Το 16,7% ούτε συμφώνησε, ούτε διαφώνησε.

Για να χρησιμοποιηθεί η εφαρμογή χρειάζονται πολλές γνώσεις:

18 απαντήσεις

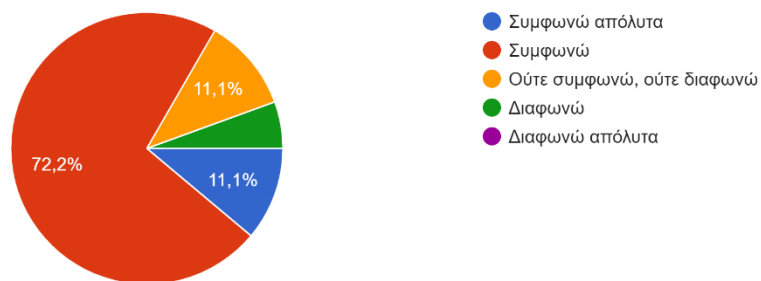


Εικόνα 30. Ερώτηση 5

Στην ερώτηση (Εικόνα 30) αν χρειάζονται πολλές γνώσεις για να χρησιμοποιηθεί η εφαρμογή, το 50% απάντησε πως δεν χρειάζονται ιδιαίτερες γνώσεις. Το 22% υποστήριξε ότι χρειάζονται γνώσεις ενώ το 27,8% δεν συμφώνησε αλλά ούτε και διαφώνησε.

Οι οδηγίες για τη χρησιμοποίηση της εφαρμογής ήταν κατατοπιστικές:

18 απαντήσεις

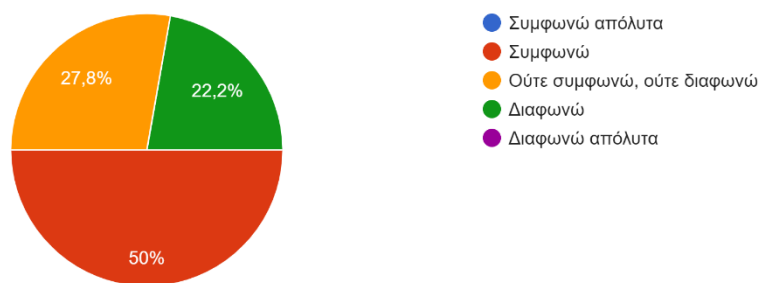


Εικόνα 31. Ερώτηση 6

Στο αν οι οδηγίες για τη χρησιμοποίηση της εφαρμογής ήταν κατατοπιστικές (Εικόνα 31), σε μεγάλο ποσοστό της τάξης του 72,2% οι απαντήσεις ήταν θετικές ενώ οι αρνητικές ήταν μόλις 5,6%. Από 11,1% συμφώνησαν απόλυτα αλλά και ούτε συμφώνησαν, ούτε διαφώνησαν.

Η εφαρμογή βοήθησε στην εκμάθηση νέων πραγμάτων:

18 απαντήσεις

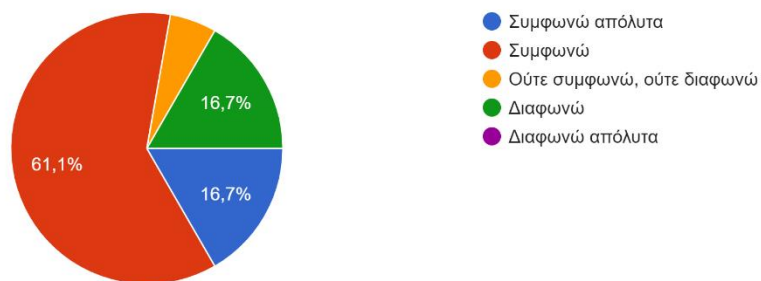


Εικόνα 32. Ερώτηση 7

Στην 7^η ερώτηση (Εικόνα 32) και στο αν η εφαρμογή βοήθησε στην εκμάθηση νέων πραγμάτων. Από τη μία, το 50% συμφώνησε ότι βοήθησε. Από την άλλη, το 22,2% διαφώνησε ενώ το 27,8% ούτε συμφώνησε, ούτε διαφώνησε.

Οι απαντήσεις που δόθηκαν από το chatbot ήταν ακριβείς:

18 απαντήσεις

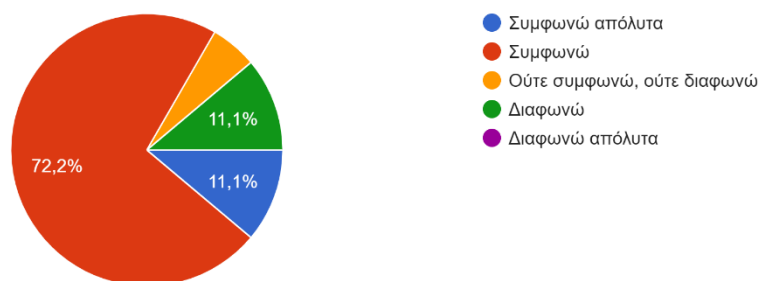


Εικόνα 33. Ερώτηση 8

Στην ερώτηση (Εικόνα 33) για το αν οι απαντήσεις που έλαβαν οι χρήστες από το chatbot ήταν ακριβείς. Το 61,1% συμφώνησε για την ακρίβεια των απαντήσεων καθώς και το 16,7% συμφώνησε απόλυτα. Το άλλο 16,7% διαφώνησε, ενώ ένα μικρό ποσοστό ούτε συμφώνησε, ούτε διαφώνησε.

Η χρήση της εφαρμογής ήταν εύκολη τόσο από κινητό όσο και από υπολογιστή:

18 απαντήσεις

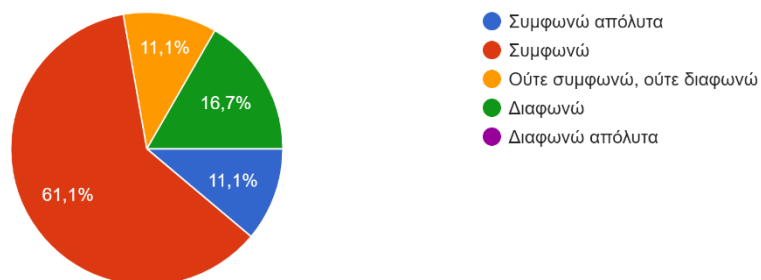


Εικόνα 34. Ερώτηση 9

Στην 9^η ερώτηση (Εικόνα 34) και στο αν η χρήση της εφαρμογής ήταν εύκολη τόσο από κινητό όσο και από υπολογιστή. Από τη μία, το 83,3% (και μάλιστα το 11,1% απόλυτα) συμφώνησε ότι ήταν εύκολη και στα δύο. Από την άλλη, το άλλο 11,1% διαφώνησε.

Η χρήση της νοηματικής γλώσσας επιτρέπει μέσω της εφαρμογής, την εύκολη ενημέρωση για τα πολιτιστικά γεγονότα:

18 απαντήσεις



Εικόνα 35. Ερώτηση 10

Στην τελευταία ερώτηση (Εικόνα 35) και στο αν η χρήση της νοηματικής γλώσσας επιτρέπει μέσω της εφαρμογής, την εύκολη ενημέρωση για τα πολιτιστικά γεγονότα. Το 61,1% απάντησε ότι επιτρέπει την εύκολη ενημέρωση. Από την άλλη, το 16,7% διαφώνησε, ενώ το 11,1% μεν συμφώνησε απόλυτα αλλά υπήρξε αντίστοιχο ποσοστό που ούτε συμφώνησε, ούτε διαφώνησε.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Κατά τη διάρκεια εκπόνησης της συγκεκριμένης πτυχιακής εργασίας, κύριος στόχος ήταν η δημιουργία ενός ‘ημερολογίου’ πολιτιστικών γεγονότων όπου η πληροφορία δίνεται με τη μορφή κειμένου, ήχου αλλά και βίντεο ανάλογα με τις απαιτήσεις του εκάστοτε χρήστη. Ειδικότερα, τα πολιτιστικά γεγονότα χωρίζονται - προς το παρόν - στις επιμέρους κατηγορίες: **Ζωγραφική, Θέατρο, Ομιλία, Σεμινάριο, Συναυλία και Φεστιβάλ**. Έτσι, ο χρήστης θα μπορεί να βρει με βάση τα ενδιαφέροντά του εύκολα, τα γεγονότα που συμβαίνουν στην πόλη του.

Στη συνέχεια, μέσω του μοντέλου αναγνώρισης νοηματικής γλώσσας της τεχνολογίας, παρέχεται η δυνατότητα ενημέρωσης του χρήστη μέσω της νοηματικής γλώσσας, αφού με το άνοιγμα της κάμεράς του, στέλνοντας νοήματα μπορεί να λαμβάνει και τις αντίστοιχες απαντήσεις. Επιπλέον, με το άνοιγμα του μικροφώνου του μπορεί μέσω φωνητικών ερωτήσεων και εντολών να δέχεται τις απαντήσεις με τη μορφή ήχου. Τέλος, η ύπαρξη του chatbot, διευκολύνει ακόμα περισσότερο το χρήστη, καθώς έχει διαμορφωθεί ειδικό λεξιλόγιο που απαντάει στα ερωτήματα **Πού; Πότε; Τι;** αναφορικά με τα γεγονότα.

Στις μελλοντικές εργασίες που θα μπορούσαν να εξελιχθούν με την παρούσα εφαρμογή εμπεριέχονται τα παρακάτω:

- Επέκταση των τύπων των πολιτιστικών γεγονότων, προσθέτοντας νέες κατηγορίες όπως π.χ. αθλητικά κ.λ.π.
- Πλήρης υποστήριξη των λειτουργιών φωνητικής της εφαρμογής και σε άλλους φυλλομετρητές εκτός από το Google Chrome που έχει ήδη γίνει.
- Υποστήριξη TTS και STT στο chatbot.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] Loshin, P. (2022). Structured Query Language (SQL) Διαθέσιμο στο: <https://www.techtarget.com/searchdatamanagement/definition/SQL> (Ημ. επισκ.: 30 Αυγούστου 2022).
- [2] About - OpenCV. (no date). Διαθέσιμο στο: <https://opencv.org/about/> (Ημ. επισκ.: 2 Σεπτεμβρίου 2022).
- [3] Transfer learning and fine-tuning. (no date). Διαθέσιμο στο: https://www.tensorflow.org/tutorials/images/transfer_learning (Ημ. επισκ.: 3 Σεπτεμβρίου 2022).
- [4] Srivastava,S et al. (2021). Sign Language Recognition System using TensorFlow Object Detection API Διαθέσιμο στο: <https://arxiv.org/ftp/arxiv/papers/2201/2201.01486.pdf> (Ημ. επισκ.: 8 Σεπτεμβρίου 2022).
- [5] Bafna,S et al. (2021). Chat-Bot using Rasa and Python Spacing Διαθέσιμο στο: <https://www.irjet.net/archives/V8/i6/IRJET-V8I6683.pdf> (Ημ. επισκ.: 7 Σεπτεμβρίου 2022).
- [6] Alwaysdata (no date). Διαθέσιμο στο: <https://learn.getgrav.org/15/webserver-hosting/shared/alwaysdata/> (Ημ. επισκ.: 30 Αυγούστου 2022).

ΠΑΡΑΡΤΗΜΑ

Πηγαίος κώδικας από την εφαρμογή ενημέρωσης πολιτιστικών γεγονότων για AMEA στο Github:

<https://github.com/roottoor1234/ptuxiakh-web-app>

https://github.com/dimitrisAvgoustopoulos/RasaChatbot_MySQL_db?fbclid=IwAR0TdcfbIVKGhqBry0bMuMOebBiqSkY3VKOd1JHGNSALoJOCflsizFBdnd8