MOMA Project

Η παρούσα εργασία πραγματοποιήθηκε στα πλαίσια της τελικής εξέτασης του μαθήματος ΠΛΗ ΠΡΟ με αντικείμενο την γλώσσα προγραμματισμού python. Ζητούμενο της εργασίας ήταν η ανάπτυξη εφαρμογής σε γραφικό περιβάλλον. Αντικείμενο της εν λόγω εφαρμογής είναι η αποθήκευση δεδομένων της συλλογής έργων του Μουσείου Μοντέρνας Τέχνης της Νέας Υόρκης, σε μια βάση δεδομένων, και η παροχή της δυνατότητας στους χρήστες, για περιήγηση στη συλλογή, για αναζήτηση με πολλαπλά κριτήρια, για προσθήκη ή/και διαγραφή έργων, αλλά και εμφάνιση έργων με τυχαίο τρόπο. Όλα τα παραπάνω θα πραγματοποιούνται με χρήση της κατάλληλης διεπαφής. Ο κώδικας της εφαρμογής βρίσκεται σε αποθετήριο στο github στη διεύθυνση https://github.com/aris-sk/moma_final_project_eap.

Τα δεδομένα της συλλογής έργων του μουσείου προσφέρονται με άδεια CCA σε δυο αρχεία μορφής csv στο github, και συγκεκριμένα στη διεύθυνση https://github.com/MuseumofModernArt/collection. Τα δεδομένα, λοιπόν, αυτά χρησιμοποιήθηκαν για την κατασκευή δυο πινάκων σε βάση δεδομένων sqlite.

Βιβλιοθήκες

Για την άναπτυξη της εφαρμογής χρησιμοποιήθηκαν πολλαπλές βιβλιοθήκες τόσο εσωτερικές (που περιλαμβάνονται δηλαδή στη διανομή της Python 3.10.4) όσο και εξωτερικές. Συγκεκριμένα χρησιμοποιήθηκαν οι βιβλιοθήκες:

- **tkinter** (για την δημιουργία της γραφικής διεπαφής),
- **pandas** για τη διαχείριση δεδομών,
- **matplotlib** (για την κατασκευή και παρουσίαση γραφικών παραστάσεων),
- **PIL** (για τον χειρισμό και την απεικόνιση φωτογραφιών διαφόρων τύπων),
- **sqlite3** (για την κατασκευή και χρήση μια ελαφριάς βάσης δεδομένων που αποθηκέυει στην ουσία σε ένα αρχείο),

- **urllib** για την ανάκτηση των κατάλληλων δεδομένων από το διαδίκτυο,
- os για την προσβαση σε διαδρομές του συστήματος αρχείων και την εκτέλεση εργασιών όπως φόρτωση/αποθήκευση δεδομένων,
- **sqlwrapper** για queries στην βάση DBpedia, η οποία επιστρέφει δομημένο περιεχόμενο απο πληροφορίες της wikipedia,
- **qwikidata** για ανάκτηση του url της εικόνας του καλλιτέχνη με τη βοήθεια του api των wikidata,
- **hashlib** για την αποκωδικοποίηση και σχηματισμό του url της εικόνας του καλλιτέχνη από τα δεδομένα που επιστρέφει το **wikidata**,
- **wget** για λήψη των αρχείων csv από το github
- **random** για την παρουσίαση τυχαίου έργου από την βάση δεδομένων.

Η εφαρμογή

Η εφαρμογή αποτελείται από 4 αρχεία **(app.py, artists.py, artworks.py, database_functions.py)** η σημασία των οποίων θα αναλυθεί στη συνέχεια, ενώ δεδομένα απαραίτητα για τη λειτουργικότητά της, αποθηκεύονται στη διαδρομή **"./files"** και **"./files/image**s". Σε αδρές γραμμές σημειώνεται οτι με τη βοήθεια των *artists.py* και *artworks.py* δημιουργούνται αντικείμενα που απεικονίζουν καλλιτέχνες και έργα τους αντίστοιχα, βάσει αναζήτησης ή περιήγησης, ενώ στο module *database_functions.py* ορίζονται οι συναρτήσεις για τα διάφορα queries που απαιτούνται για την επικοινωνία με τη βάση δεδομένων.

Η εφαρμογή εκκινείται από το *app.py* με το οποίο και φορτώνονται οι απαραίτητες βιβλιοθήκες για την λειτουργικότητα της εφαρμογής και οι οποίες αναφέρθηκαν παραπάνω. Πρώτη διαδικασία που εκτελείται είναι ο έλεγχος αν υπάρχει στην τρέχουσα διαδρομή (την ίδια δηλαδή στην οποία βρίσκεται το *app.py*) βάση δεδομένων με το όνομα **"mydatabase.db"**. Καλείται γι' αυτό το λόγο η συνάρτηση *load_database* του module *database_functions.py*. Εαν δεν βρεθεί το συγκεκριμένο αρχείο τότε δημιουργείται εξ αρχής από τα δυο αρχεία csv με τα δεδομένα που παρέχονται στο github και τα οποία βρίσονται εντός της διαδρομής "./files", και τα οποία φορτώνονται σε dataframes με τη

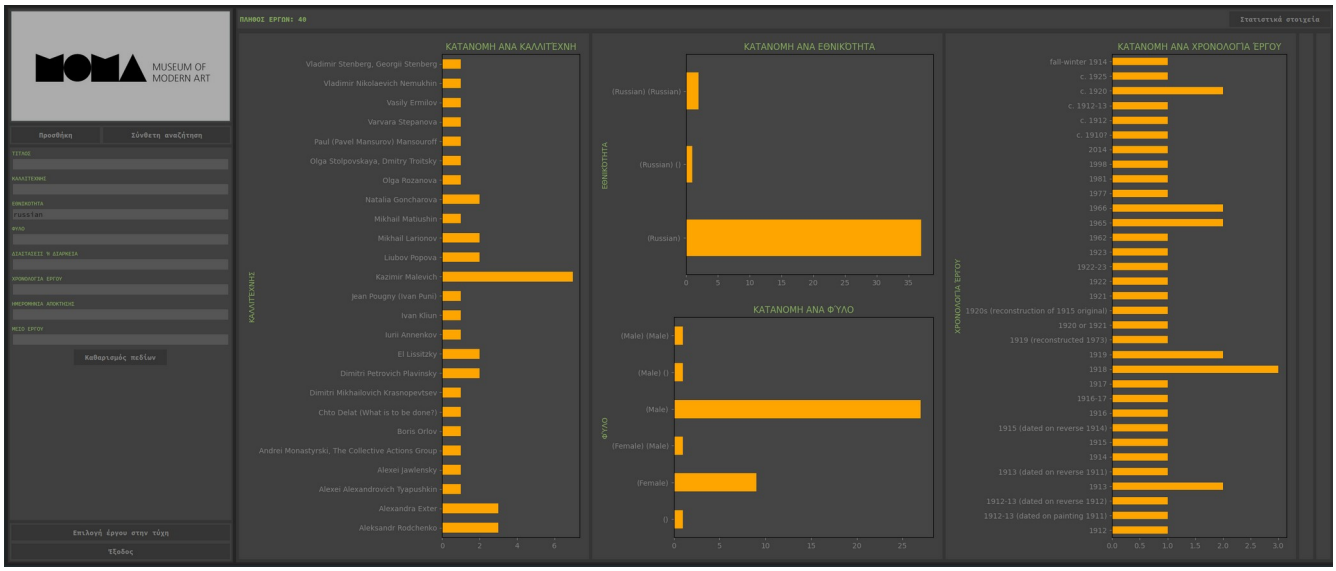
βιβλιοθήκη pandas. Η δημιουργία της βάσης δεδομένων γίνεται με τη βοήθεια ενος sql script (που υπάρχει στη διαδρομή "./files" και τη χρήση της βιβλιοθήκης sqlite. Θα μπορούσε να χρησιμοποιηθεί αποκλειστικά η βιβλιοθήκη pandas για την δημιουργία της βάσης δεδομένων, όμως δε δίνεται η δυνατότητα δια μέσω αυτής της οδού για ορισμό πρωτεύοντων κ ξένων κλειδιών. Έτσι επιλέχθηκε η χρήση του sql script και της sqlite για τη δημιουργία δυο κενών από δεδομένα πινάκων, *artists* και *artworks* με τις επιθυμητές στήλες και κλειδιά, και στη συνέχεια να φορτωθούν σε αυτούς τα δεδομένα που έχουμε ήδη σε dataframes. Ως πρωτεύοντα κλειδιά επιλέχθηκαν το *ObjectID* στον πίνακα *artworks* και το *ConstituentID* στον πίνακα *artists*, στα οποία δόθηκε και η ιδότητα του *autoincrement*. Παράλληλα λαμβάνεται μέριμνα για την εμφάνιση κατάλληλων μηνυμάτων σφάλματος σε τέτοια περίπτωση με τη δομή *try/except*.

Μετά τον έλεγχο ύπαρξης της βάσης δεδομένων, σειρά έχει η εμφάνιση της γραφικής διεπαφής. Η γραφική διεπαφή αναπτύχθηκε στην κλάση **MomaGuiApp**. Έτσι σε αυτή αρχικοποιούνται οι βασικές ιδιότητες της διεπαφής, όπως η γεωμετρία, ο τίτλος του βασικού παραθύρου, αλλά και επιμέρους ιδιότητες συγκεκριμένων γραφικών στοιχείων όπως buttons, labels, entry objects, μια γραμματοσειρά κ.λπ. Τέλος φορτώνεται σε ένα dataframe ένας πίνακας με όλα τα έργα που υπάρχουν στη βάση δεδομένων, κάνοντας το κατάλληλο ερώτημα στη βάση με την συνάρτηση *search_database()* του module *database_functions.py*. Να σημειωθεί εδώ πως έχουμε επιλέξει να εμφανίζονται σε αυτόν τον πίνακα οι στήλες του τίτλου έργου, του ονόματος του καλλιτέχνη, της εθνικότητάς του, του φύλου, των διαστάσεων του έργου, της χρονολογίας δημιουργίας και απόκτησής του, του μέσου του και τέλος του μοναδικού του objectid. Για την απεικόνιση τελικά και την είσοδο στην αρχική οθόνη της εφαρμογής καλείται η μέδοδος *widgets* της κλάσης *MomaGuiApp*, στην οποία ορίζονται και κατασκευάζονται σχεδόν όλα τα γραφικά στοιχεία.

Η πρώτη οθόνη αποτελείται από ένα sidebar στα αριστερά, και ένα κεντρικό frame στο οποίο και απεικονίζεται κάθε στιγμή, είτε ο πίνακας με το σύνολο των έργων της βάσης, είτε το αποτέλεσμα που έχει επιστραφεί μετά από ένα ερώτημα για κάποια αναζήτηση. Στο sidebar εμφανίζεται το λογότυπο του Μουσείου, αφου φορτωθεί από την διαδρομή "./files/images". Επιπλέον κάτω από το λογότυπο εμφανίζεται ένα thumbnail που παρουσιάζει κάθε στιγμή μια μικρή απεικόνιση του έργου

το οποίο είναι επιλεγμένο στον πίνακα του κεντρικού frame μαζί με τον τίτλο του και τον καλλιτέχνη, και δυο πλαίσια με κουμπιά, το καθένα από τα οποία επιτελούν κάθε μια από τις δυνατότητες που πρόσφερονται στο χρήστη από την εφαρμογή. Συγκεκριμένα υπάρχει πάνω από τη μικρογραφία του έργου, ένα ζευγάρι buttons, το ένα για προσθήκη καινούριου έργου και το δεύτερο για αναζήτηση με πολλαπλά κριτήρια στη βάση δεδομένων. Κάτω από τη μικρογραφία υπάρχει σειρά από buttons, για εμφάνιση της καρτέλας του επιλεγμένου έργου, της καρτέλας του δημιουργού του επιλεγμένου έργου, επεξεργασία και ενημέρωση των στοιχείων του, διαγραφή του από τη βάση δεδομένων, επιλογής ενός έργου στην τύχη και τέλος για τερματισμό της εφαρμογής. Στο κεντρικό frame όπως αναφέρθηκε απεικονίζεται πίνακας των έργων που φέρει επιλέον οριζόντιο και κατακόρυφο scrollbar για ευκολότερη περιήγηση, ενώ επιπρόσθετα έχουν ορισθεί και bindings σε πλήκτρα του πληκτρολογίου για περιήγηση με αυτό (βέλη για πλοήγηση, return για επιλογή, q επιστροφή). Υπάρχει η δυνατότητα για ταξινόμηση των γραμμών του πίνακα με αύξουσα ή φθίνουσα σειρά με κλικ στον τίτλο οποιασδήποτε στήλης. Στο ανώτερο σημείο του κεντρικού frame υπάρχει ετικέτα που αναγράφει κάθε στιγμή το σύνολο των έργων που απεικονίζει ο πίνακας, που να θυμίσουμε είναι είτε το σύνολο της βάσης είτε το αποτέλεσμα αναζήτησης του χρήστη. Επίσης υπάρχει button που οδηγεί σε οθόνη απεικόνισης στατιστικών πινάκων επί των στοιχείων του πίνακα.

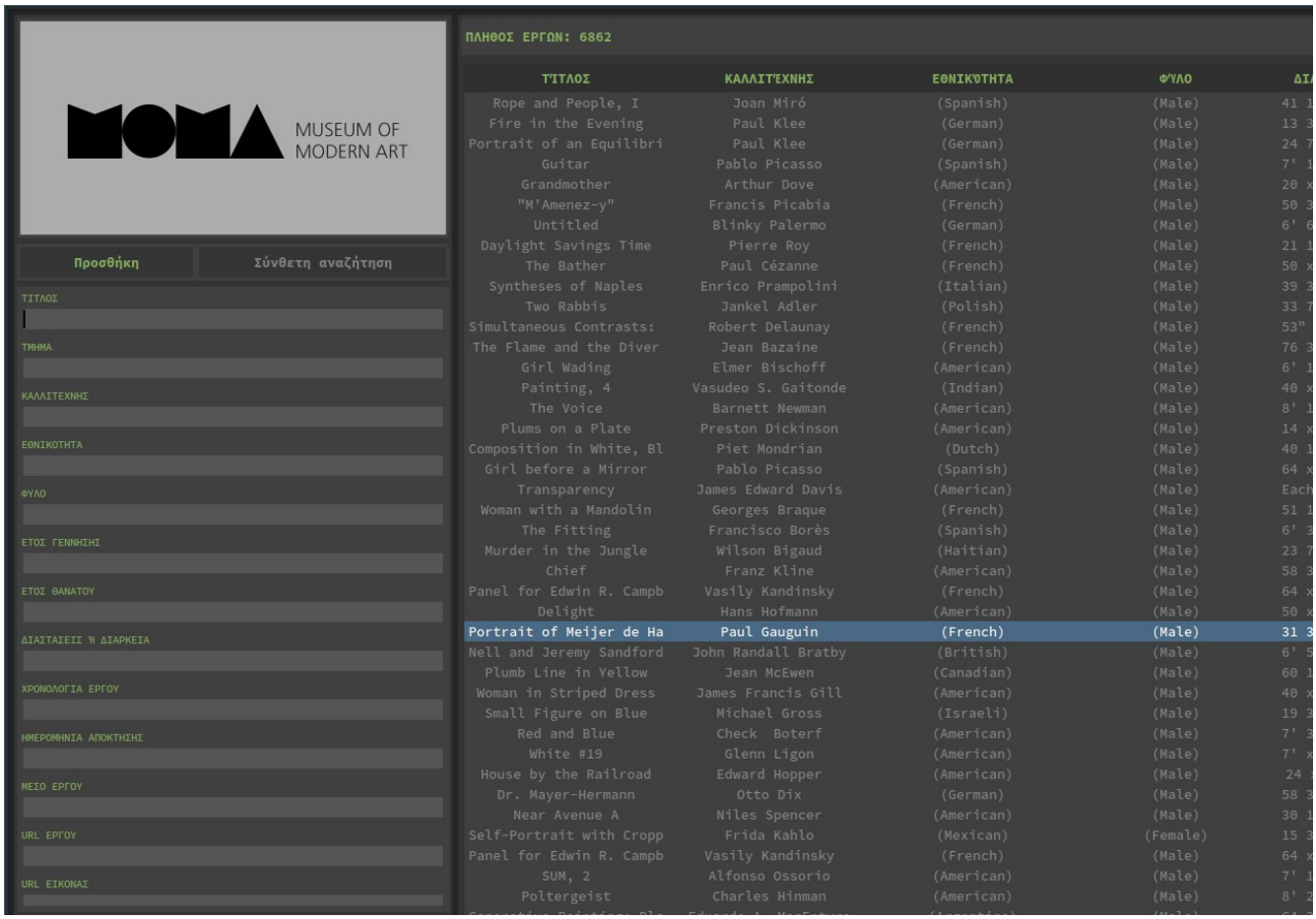
Εικόνα 1: Η πρώτη οθόνη μετά το άνοιγμα της εφαρμογής



Εικόνα 2: Στατιστικά που παράγονται για πίνακες απο καλλιτέχνες που κατάγονται από την Ρωσία

Όπως αναφέρθηκε κάθε button επιτελεί μια από τις δυνατότητες που προσφέρονται στον χρήστη. Το button 'Προσθήκη' ανοίγει ακριβώς από κάτω του, και αφού αποκρυφθούν η μικρογραφία του έργου και κάποια κουμπιά από κάτω της, εμφανίζονται κάποια πεδία για την καταχώρηση στοιχείων και την προσθήκη νέου έργου. Τα πεδία τα οποία μπορεί να καταχωρήσει ο χρήστης κατά την προσθήκη είναι τα παρακάτω: Title, Department, Artist, Nationality, Gender, BeginDate, EndDate, Dimensions, Date, DateAcquired, Medium, URL, ThumbnailURL, ConstituentID. Στο σημείο αυτό να αναφερθεί οτι η εφαρμογή έχει χτιστεί γύρω από την έννοια του έργου και απο εκεί και μετά το κάθε έργο ανήκει και σε κάποιον ή κάποιους καλλιτέχνες. Αυτό σημαίνει οτι δίνεται η δυνατότητα προσθήκης έργου, και στην περίπτωση που δεν υπάρχει άλλο έργο από τον ίδιο καλλιτέχνη τότε προστίθεται και ο καλλιτέχνης, ενώ δε δίνεται η δυνατότητα προσθήκης μόνο καλλιτέχνη χωρίς απαραίτητα αυτός να συνδεθεί με κάποιο έργο, καθώς θεωρήσαμε πως δεν έχει κάποιο νόημα αυτό για ένα μουσείο. Ένα μουσείο διαθέτει έργα που εκθέτει, και όχι καλλιτέχνες. Επομένως με την εισαγωγή των στοιχείων στα entry boxes και την επιβεβαίωση προστίθεται ένα έργο στη βάση δεδομένων, με κλήση της συνάρτησης *insert* του module *database_functions.py*. Η συνάρτηση αυτή, αρχικά εξετάζει την ύπαρξη άλλου έργου του ίδιου καλλιτέχνη, αν δηλαδή υπάρχει ήδη ο καλλιτέχνης. Αν ναι, τότε αποδίδεται στο έργο το ίδιο *ConstituentID* που σχετίζεται με τον καλλιτέχνη και ενημερώνεται και η καταχώρησή του στον πίνακα *artists* της βάσης, ειδάλλως του αποδίδεται το αμέσως

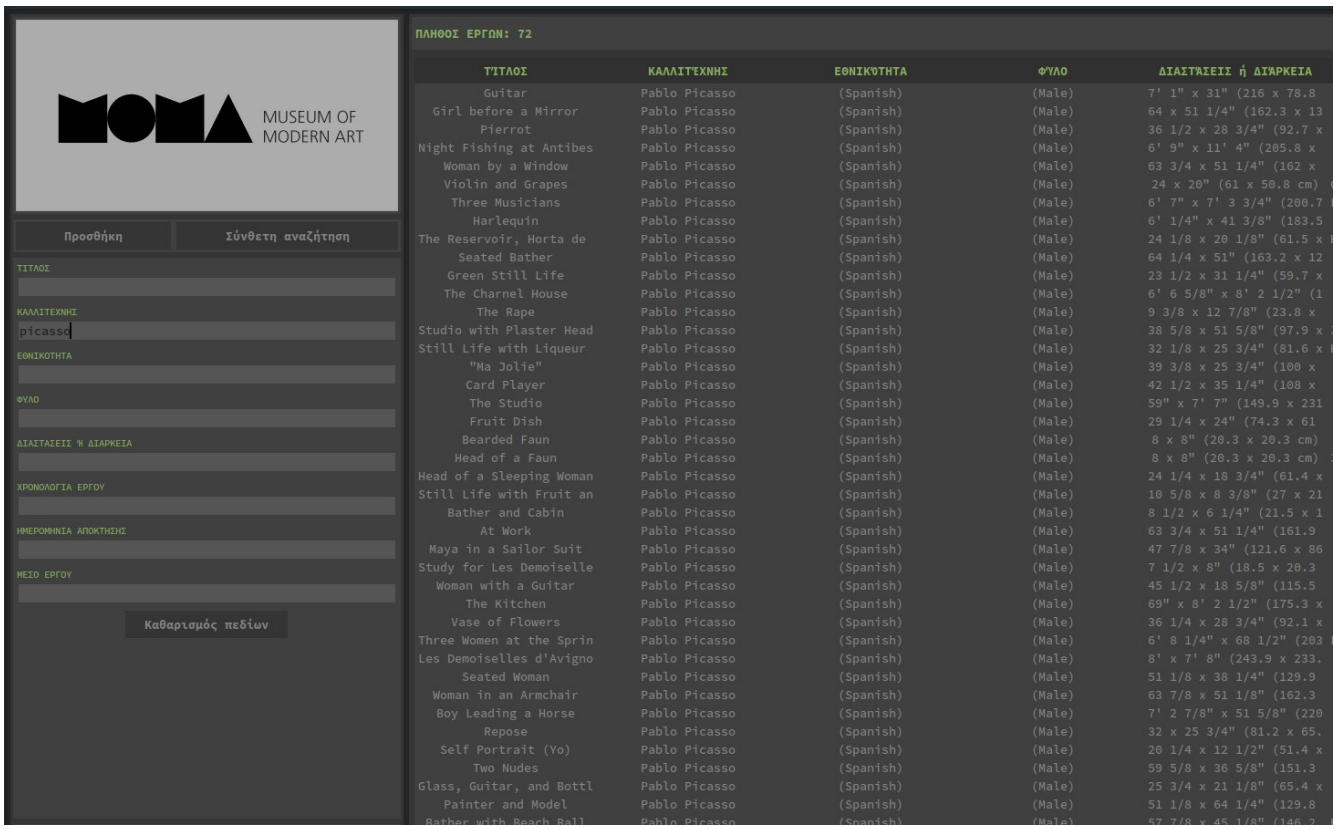
μεγαλύτερο κατά 1. Ειδικά για την περίπτωση που ο καλλιτέχνης υπάρχει ήδη στον πίνακα *artists* της βάσης δεδομένων λαμβάνονται υπόψην και κάποιες επιπλέον λεπτομέρειες για αποφυγή λαθών. Έτσι εαν κατά την εισαγωγή ενός έργου δεν έχουν συμπληρωθεί κάποια πεδία που αφορούν τον καλλιτέχνη, αυτά δεν λαμβάνονται υπόψην ώστε να μην αντικαταστήσουν από λάθος ή παράλλειψη, ή άγνοια του εκάστοτε χρήστη στοιχεία που υπάρχουν ήδη καταχωρηθεί στη βάση δεδομένων για τον καλλιτέχνη. Τέλος για λόγους ομοιόμορφης εμφάνισης μετατρέπονται τα στοιχεία σε μορφή *str.Title* (Πρώτος χαρακτήρας κεφαλαίος και τα υπόλοιπα πεζά).



Εικόνα 3: Προσθήκη έργου

Με ανάλογο τρόπο, με κλικ στο button 'Σύνθετη αναζήτηση', εμφανίζονται μια σειρά από entry boxes με τα οποία ο χρήστης μπορεί να κάνει αναζήτηση με πολλαπλά κριτήρια. Κάθε φορά που εισάγεται κάποιος χαρακτήρας σε κάποιο από τα πεδία, δίνεται εντολή για ένα νέο query στη βάση δεδομένων, και επιστρέφεται ο πίνακας με τα αποτελέσματα που αντιστιχούν στα στοιχεία που έχει εισάγει ο χρήστης.

Υπάρχει δηλαδή μια άμεση δυναμική αλληλεπίδραση με τη βάση δεδομένων. Βεβαίως όταν ο χρήστης επιλέξει ένα έργο από τον πίνακα που έχει επιστραφεί, αμέσως ανοίγει η μικρογραφία του καθώς και τα υπόλοιπα buttons, στο frame του sidebar. Η λειτουργικότητα της αναζήτησης υλοποιείται με τη συνάρτηση *search_database* του module *database_functions.py*.



Εικόνα 4: Σύνθετη αναζήτηση (παράδειγμα για Picasso)

Κάτω από τη μικρογραφία υπάρχουν τα κουμπιά: 'Καρτέλα έργου', 'Καρτέλα καλλιτέχνη', 'Ενημέρωση στοιχείων έργου', 'Διαγραφή έργου', 'Επιλογή έργου στην τύχη', 'Έξοδος'. Η 'Καρτέλα έργου' ανοίγει στο κεντρικό frame και στη θέση του πίνακα με τα έργα, ένα νέο frame με τα ιδιαίτερα χαρακτηριστικά του έργου και δυο buttons στο τέλος, το ένα ανοίγει στο κεντρικό frame μεγενθυμένη την φωτογραφία του έργου, και το άλλο επιστρέφει στον πίνακα με τα έργα. Να σημειωθεί έδω πως τα στοιχεία που παρουσιάζονται στην καρτέλα του έργου πρόερχονται από τον πίνακα *artworks* της βάσης δεδομένων και ειναι αυτά που σχετίζονται με το έργο. Τυχόν στοιχεία που βρίσκονται στον ίδιο πίνακα και σχετίζονται με τον καλλιτέχνη (πχ φύλλο κ.λπ.)

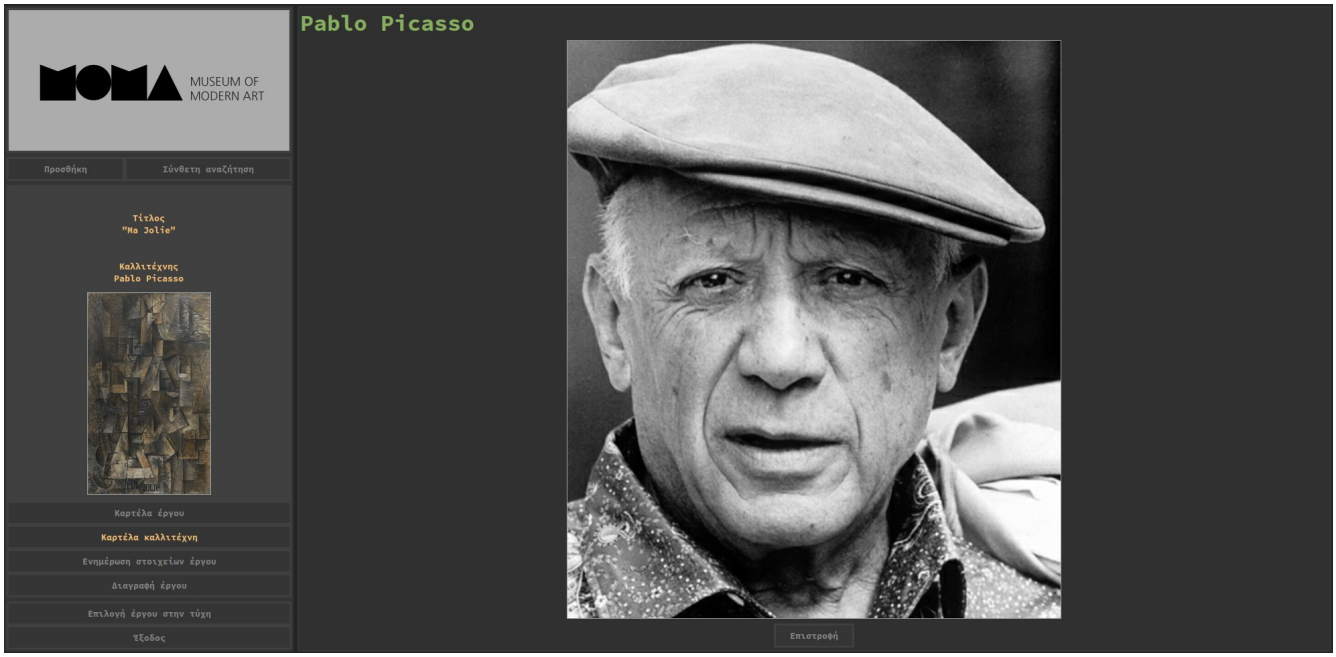
εμφανίζονται στην καρτέλα του καλλιτέχνη. Επίσης εμφανίζονται μόνο τα πεδία για τα οποία υπάρχει τιμή και όχι τα κενά, ενώ οι τίτλοι των πεδίων έχουν μεταφραστεί στα ελληνικά βάση ενος λεξικού μου αρχικοποιήθηκε στην *init* της κλάσης της γραφικής διεπαφής (*MomaGuiApp*). Η φωτογραφία του έργου μεγενθύνεται διατηρώντας το aspect ratio λαμβάνοντας υπόψην το μέγεθος του frame τη δεδομένη στιγμή. Τέλος έχουν δημιουργηθεί και εδώ key bindings για αλληλεπίδραση και με το πληκτρολόγιο (return -> άνοιγμα φωτογραφίας, q -> επιστροφή στον πίνακα έργων). Η συνάρτηση που υλοποιεί την καρτέλα έργου είναι η *create_artwork_tab*.

Εικόνα 5: Η καρτέλα έργου για το έργο 'Ma Jolie'

Εικόνα 6: Η προβολή του έργου 'Ma Jolie'

Το button 'Καρτέλα καλλιτέχνη' σε αντιστοιχία με αυτό της 'Καρτέλας έργου' ανοίγει στο κεντρικό frame μια καρτέλα με τα ιδιαίτερα στοιχεία του καλλιτέχνη. Σε αυτή την καρτέλα παρουσιάζονται στοιχεία από τον πίνακα artists της βάσης δεδομένων, ενώ οι τίτλοι των πεδίων εμφανίζονται και εδώ στα ελληνικά με τη βοήθεια του λεξικού που αναφέρθηκε προηγουμένως. Επιπλέον εδώ εμφανίζεται και μια σύντομη βιογραφία του καλλιτέχνη, που έχει ανακτηθεί από την dpedia, και σε ένα listbox παραθέτονται τα υπόλοιπα έργα του ίδιου καλλιτέχνη που υπάρχουν στο μουσείο. Όπως και στην καρτέλα έργου έτσι και εδώ εμφανίζονται πεδία για τα οποία έχουν δοθεί στοιχεία και όχι τα κενά. Η επιλογή ενός άλλου έργου του ίδιου καλλιτέχνη από το listbox οδηγεί στην καρτέλα έργου του. Παράλληλα υπάρχουν στο κάτω μέρος δυο buttons το ένα ανοίγει τη φωτογραφία του καλλιτέχνη εφόσον υπάρχει, ενώ το δεύτερο επιστρέφει στον πίνακα των έργων. Ομοίως με την καρτέλα έργου η φωτογραφία του καλλιτέχνη μεγενθύνεται λαμβάνοντας υπόψην το μέγεθος του frame τη δεδομένη στιγμή. Τέλος και εδώ λειτουργούν τα ίδια bindings με την καρτέλα έργου για άνοιγμα της φωτογραφίας και επιστροφή στον πίνακα των έργων. Η συνάρτηση που υλοποιεί την καρτέλα καλλιτέχνη είναι η *create_artist_tab*.

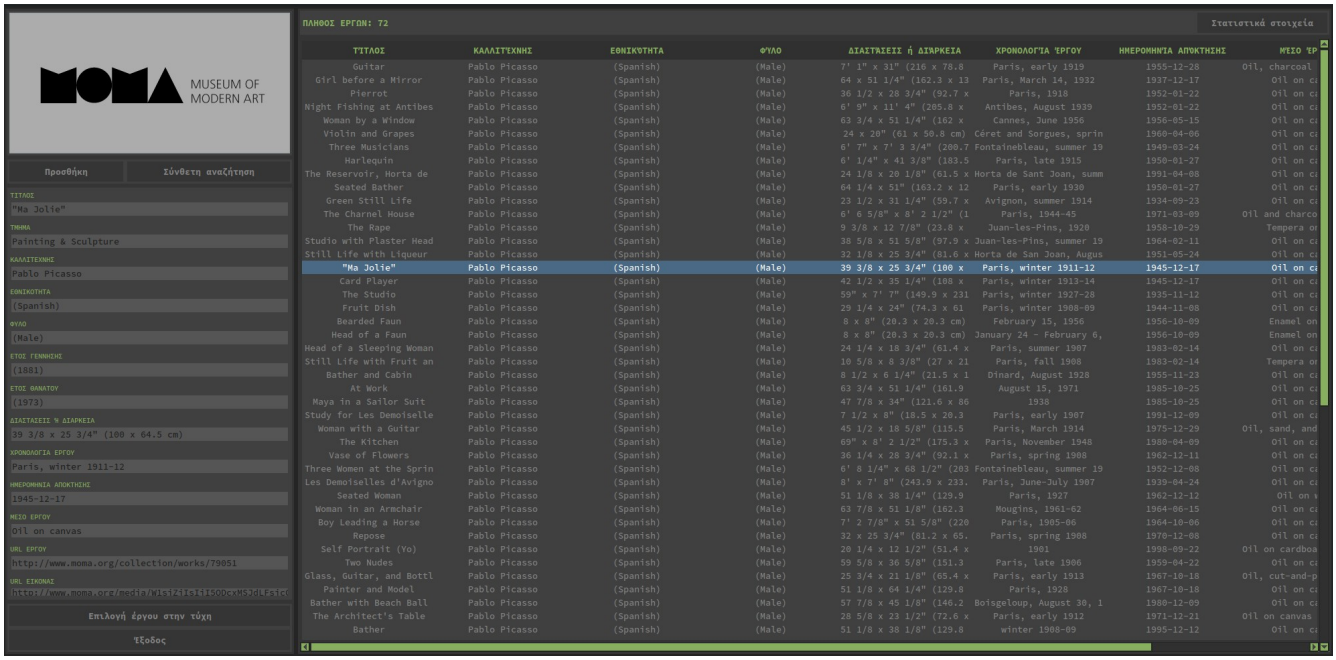
Εικόνα 7: Η καρτέλα καλλιτέχνη για τον Pablo Picasso



Εικόνα 8: Προβολή της φωτογραφίας του Pablo Picasso

Το κουμπί 'Ενημέρωση στοιχείων έργου' προσφέρει τη δυνατότητα για επεξεργασία των επιμέρους πληροφοριών του κάθε έργου και αποθήκευση ξανά στη βάση. Για το σκοπό αυτό ανοίγουν στο sidebar τα ίδια πεδία που ανοίγουν και κατά την προσθήκη ενός έργου, όμως αυτή τη φορά είναι συμπληρωμένα με τις πληροφορίες που έχουν ήδη καταχωρηθεί στη βάση. Στην περίπτωση που ενημερωθούν στοιχεία που

σχετίζονται με τον καλλιτέχνη του έργου τότε αυτά ενημερώνονται και για όλα τα υπόλοιπα έργα του ίδιου καλλιτέχνη. Η λειτουργικότητα της ενημέρωσης υλοποιείται με τη συνάρτηση *update* του module *database_functions.py*.



Εικόνα 9: Ενημέρωση στοιχείων έργου

Όπως είναι ευνόητο το button 'Διαγραφή έργου' διαγράφει το έργο το οποίο ειναι επιλεγμένο στον πίνακα ή είναι ανοιγμένο στην καρτέλα του, βάσει του *objectid* του, με τη συνάρτηση *delete_from_artworks* του module *database_functions.py*.

Με το κουμπί 'Επιλογή έργου στην τύχη' καλείται η *select_random_artwork*, επιλέγεται ένα τυχαίο *objectid* από τα έργα της βάσης δεδομένων και ανοίγει η καρτέλα για το συγκεκριμένο έργο. Το button 'Έξοδος' οδηγεί σε τερματισμό της εφαρμογής.

Αποτελέσματα – Γνωστοί περιορισμοί

Γενικά η εφαρμογή λειτουργεί ομαλά και με τον αναμενόμενο τρόπο (δοκιμασμένη σε Linux, Windows). Κάποια γνωστά προβλήματα που δημιουργούνται και που οφείλονται στον αρχικό κατάλογο των έργων που παρέχεται στα csv αρχεία στο github, είναι με κάποια έργα που αποδίδονται/ανήκουν σε ομάδα καλλιτεχνών. Σε τέτοιες περιπτώσεις είναι αδύνατη η αντιστοίχιση σε έναν μόνο καλλιτέχνη και επομένως η δημιουργία της καρτέλας του, γεγονός που οδηγεί σε σφάλματα και τερματισμό. Δηλαδή αυτή η δυσλειτουργία οφείλεται στον τρόπο καταχώρησης των αρχικών datasets, και στα οποία δεν έγινε κάποιος καθαρισμός από εμάς. Τέλος σε κάποιες περιπτώσεις δεν παρατίθεται σύνδεσμος με την φωτογραφία του έργου, ή είναι λανθασμένος ή μη ενημερωμένος. Σε αυτές τις περιπτώσεις η φωτογραφία που εμφανίζει η εφαρμογή είναι μια επιλεγμένη γενική φωτογραφία που παρέχεται στη διαδρομή './files/images' (no_image.jpg).

Ανάπτυξη – Συνεισφορά ομάδας

Το project ξεκίνησε με τη συμμετοχή 4 φοιτητών (ΚΥΡΑΤΖΙΔΟΥ ΚΩΝΣΤΑΝΤΙΝΑ, ΝΤΟΥΡΛΑΡΗΣ ΣΤΑΜΑΤΙΟΣ ΑΝΑΡΓΥΡΟΣ, ΚΙΟΥΣΗΣ ΔΗΜΗΤΡΙΟΣ και ΣΚΑΛΙΑΠΑΣ ΧΡΗΣΤΟΣ-ΑΡΙΣΤΕΙΔΗΣ). Σχεδόν αμέσως υπήρξε η αποχώρηση του ΚΙΟΥΣΗ ΔΗΜΗΤΡΙΟΥ και λίγο μετά ακολούθησε και η ΚΥΡΑΤΖΙΔΟΥ ΚΩΝΣΤΑΝΤΙΝΑ, ενώ σχεδόν τρεις εβδομάδες πριν την κατάθεση του project αποχώρησε και ο ΝΤΟΥΡΛΑΡΗΣ ΣΤΑΜΑΤΙΟΣ ΑΝΑΡΓΥΡΟΣ, με αποτέλεσμα πλέον μόνο εναπομείναν μέλος της τέταρτης ομάδας να είμαι εγώ (ΣΚΑΛΙΑΠΑΣ ΧΡΗΣΤΟΣ-ΑΡΙΣΤΕΙΔΗΣ). Και αυτό χωρίς να υπάρχει το παραμικρό πρόβλημα επικοινωνίας (έχουν γίνει πολλές συναντήσεις), και για λόγους που έχουν να κάνουν με τον διαθέσιμο χρόνο του καθενός κυρίως. Συνεπώς το τελικό αποτέλεσμα έχει αναπτυχθεί από εμένα (ΣΚΑΛΙΑΠΑΣ ΧΡΗΣΤΟΣ-ΑΡΙΣΤΕΙΔΗΣ) στο σύνολό του. Σε επίπεδο ιδεών γύρω από την γενική είκονα της εφαρμογής του project και τη μορφή της δομής της έχουν συνεισφέρει όλοι ανεξαιρέτως, αναλόγως και του χρόνου παραμονής τους στο project. Σε επίπεδο υλοποίησης η εφαρμογή έχει αναπτυχθεί αποκλειστικά από εμένα. Ο λόγος δεν είναι άλλος παρά από το οτι ξεκίνησα να δουλεύω απ τη πλευρά μου συνολικά για να είμαι έτοιμος να έχω κατι λειτουργικό τουλάχιστον, στην περίπτωση που κάτι δεν πάει

καλά με την συννενόηση ή την οργάνωσής μας. Φυσικά και είχαν μοιραστεί αρμοδιότητες στους τρεις τουλάχιστον που έμειναν στο project το περισσότερο διάστημα, χωρίς όμως να έχει κάποιος κάτι έτοιμο, για τους λόγους του ο καθένας.

Πηγές

Για την ανάπτυξη της εφαρμογής και ανά κατηγορία, έχουν χρησιμοποιηθεί ως βοήθεια και έχουν αντληθεί πληροφορίες από τις παρακάτω πηγές:

- tkinter
  - Προχωρημένος προγραμματισμός με Python - https://mathesis.cup.gr/courses/course-v1:ComputerScience+CS1.2+21E/course/
  - Python - GUI Programming (Tkinter) - https://www.tutorialspoint.com/python/python_gui_programming.htm
  - Tkinter - https://www.pythontutorial.net/tkinter/
  - Tkinter video course - https://www.youtube.com/watch?v=YXPyB4XeYLA
- pandas
  - official pandas documentation - https://pandas.pydata.org/docs/
  - pandas tutorial - https://www.w3schools.com/python/pandas/default.asp
  - Pandas video tutorials - https://www.youtube.com/playlist?list=PL-osiE80TeTsWmV9i9c58mdDCSskIFdDS
- matplotlib
  - Matplotlib 3.5.2 documentation (official) - https://matplotlib.org/stable/index.html
  - Matplotlib Pyplot - https://www.w3schools.com/python/matplotlib_pyplot.asp

    - Visualization with Matplotlib - https://jakevdp.github.io/PythonDataScienceHandbook/04.00-introduction-to-matplotlib.html
    - Python Plotting With Matplotlib (Guide) - https://realpython.com/python-matplotlib-guide/
    - Matplotlib cheat sheet - https://www.datacamp.com/cheat-sheet/matplotlib-cheat-sheet-plotting-in-python
    - Matplotlib video tutorial - https://www.youtube.com/watch?v=wB9C0Mz9gSo
- sqlite3
    - SQLite Databases With Python - Full video Course - https://www.youtube.com/watch?v=byHcYRpMgI4
    - SQLite Tutorial - https://www.sqlitetutorial.net/
    - SQLite - Python - https://www.tutorialspoint.com/sqlite/sqlite_python.htm

Παράλληλα και για πολύ ειδικά σημεία έχω ανατρέξει στο stackoverflow (https://stackoverflow.com/questions/tagged/python), και κάθε μια από αυτές τις περιπτώσεις αναφέρονται μέσα στον κώδικα σε σχόλια.

Τέλος μια από τις καλύτερες πηγές στο διαδίκτυο, ίσως η καλύτερη πηγή, τόσο από πλευράς περιεχομένου, διδασκαλίας είναι το κανάλι του Δημήτρη Ψούνη στο youtube στο οποίο προσφέρονται μια σειρά από playlists γύρω από τα βασικά της python, αλλά και διάφορες απαραίτητες βιβλιοθήκες. Η βοήθεια του ήταν πολύτιμη:

- https://www.youtube.com/c/%CE%94%CE%B7%CE%BC%CE%AE%CF%84%CF%81%CE%B7%CF%82%CE%A8%CE%BF%CF%8D%CE%BD%CE%B7%CF%82/playlists?view=50&sort=dd&shelf_id=6