# ΤΕΧΝΟΛΟΓΙΕΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΙΚΤΥΟΥ 2022

Αλίκη Τσαμοπούλου Δαμιανή:

Τίτλος: Εφαρμογή Ηλεκτρονικών Δημοπρασιών στον Παγκόσμιο Ιστό.

Στόχος αυτής της εργασίας είναι να αναπτυχθεί μια εφαρμογή επαγγελματικής δικτύωσης.Η εφαρμογη αυτη θα υλοποιηθεί βάσει της αρχιτεκτονικης web browser / web server και κάθε χρήστης να έχει πρόσβαση σε αυτή μέσω του φυλλομετρητη παγκόσμιου ιστού.

Οι τεχνολογίες που χρησιμοποιήθηκαν ήταν το fast api για το back-end και η βιβλιοθήκη React τη Javascript

# **INSTALLATION**

react version 18.2.0

npm version 8.11.0

cd to backend/

python3 main.py

cd to frontend

npm start

ή

./start.sh για το ssl

visit localhost:3000

**login credentials**

admin: admin:admin

user: 123:123

FRONTEND

Για το front-end χρησιμοποιήθηκε η βιβλιοθηκη ReactJS της Javascript, το πακέτο material UI για την μορφοποίηση καθώς και το πακέτο axios για την κλήση τον API.

Η κλήση των API γίνεται ασύγχρονα.

**Σχεδιαστικές αποφάσεις**

Παραδοχές που έκανα είναι ότι κατά την δημιουργία νέου ποστ αν ο χρήστης θέλει να ανεβάσει βίντεο θα γίνετε με λινκ μέσα από το youtube καθώς και στις φωτογραφίες ή στην ήχο στην βάση θα αποθηκεύετε ένα string με το path του ήχου ή της φωτογραφίας.

Κατά την σύνδεση στο Σιτέ ο σερβερ μου επιστρέφει το token και το user role αν είναι επιτυχής η σύνδεση. Τα στοιχεία αυτά τα αποθηκεύω σαν cookies ώστε να μην είναι απαραίτητη η επανασύνδεση μετά από κάθε page refresh και τα κάνω κάνω αυτόματα expire μετά από 1 χρόνο ή τα διαγράφω κατά την αποσύνδεση του χρήστη.

**Προβλήματα**

Το call που μου φέρνει όλα τα post για να το προβάλλω στην αρχική οθόνη απαιτεί το token του user αν όμως ο χρήστης δεν έχει συνδεθει δεν έχω post να του προβάλλω οπότε μένει στο default που περιμένει να τα φορτώσει.

Κάποια μυνήματα σφάλματος στο console γίνονται λόγω κάποιον catch στις κλήσεις των api και λόγω κάποιων warnings απο components του material UI.

BACKEND

Για το backend χρησιμοποιήθηκε SQLAlchemy και alembic για την βάση δεδομένων, fastAPI για API calls, pydantic καθώς και χρήση της NLTK βιβλιοθήκης για normalization γλώσσας για την αναζήτηση ελεύθερου κειμένου.

Το αρχείο main.py εμπεριέχει όλα τα API calls, με την λογικη των συναρτήσεων να υπάρχει στο logic.py αρχείο. Επίσης υπάρχουν βοηθητικές συναρτήσεις στο utils.py (οι οποίες καλούνται από συναρτήσεις του logic αρχείου ως επί το πλείστον). Τέλος, το αρχείο crud.py περιέχει τις core συναρτήσεις που αφορούν οποιαδήποτε αλληλεπίδραση με την βάση.

Το αρχείο schemas.py εμπεριέχει την μορφοποίηση των στοιχείων που ανταλλάσσονται με το frontend

(οι ανταλλαγές γίνονται πάντα\* σε μορφη json).

Στο αρχείο models.py υπάρχουν οι για τον σχεδιασμο της βάσης δεδομένων (εκ των οποίων μπορεί κανείς να έχει μια αναπαράσταση της βάσης)

Συμπληρωματικά, στο errors.py αρχείο αναγράφονται όλα τα μηνύματα error που τυπώνονται σε περίπτωση error handling.

Τα αρχεία alembic.ini, sql\_app.db καθώς και ο φάκελος alembic είναι autogenerated αρχεία των εργαλείων

**Τι δεν υλοποιήθηκε**

1)Tο πλαίσιο εισόδου κειμένου για την αναζήτηση επαγγελματιών

2)Η εμφάνιση των αγγελιών γίνεται τόσο βάση των δεξιοτήτων που έχει δηλώσει ο χρήστης στα προσωπικά στοιχεία του

3) Τα bonus ερωτήματα

**Προβλήματα**

Το κυριότερο πρόβλημα αφορά την εξαγωγή του αρχείου ως XML καθώς υπάρχει θέμα κατα την μετατροπή σε XML όπου εμφανίζεται το εξής error κατα την μετατροπή απο dict σε XML:

AttributeError: module 'collections' has no attribute 'iterable'

Το error εμφανίζεται σε συναρτήσεις της βιβλιοθήκης και δεν βρέθηκε κάποια λύση σε αυτό.

Το αρχείο wrapper.py το οποίο υπάρχει στον φάκελο, υποστηρίζει την εξαγωγή ενος χρήστη ως XML, αλλά η απόπειρα εξαγωγής οδηγεί σε αυτο το σφάλμα