

Εφαρμογή Επαγγελματικής Δικτύωσης (τύπου LinkedIn)

Απο τους:

Τυπάλδος Παύλος Αποστολάτος 1115201800009

Γεώργιος Αχιλλέας Παπαγεωργίου 1115201800147

ΠΡΟΛΟΓΟΣ

Το project μας ακολουθεί κατα γράμμα τις οδηγίες της εκφώνησης και πληρεί τις προϋποθέσεις που έχουν τεθεί. Η μόνη διαφοροποίηση είναι στην front page όπου την σχεδιάσαμε με βάση το πρότυπο όλων των μοντέρνων social media. Όλες οι λειτουργίες είναι προσβάσιμες από το navbar μας. Η εφαρμογή δημιουργήθηκε με React στο front end ενώ η επικοινωνία γίνεται με HTTP requests μέσω μιας δεύτερης εφαρμογής REST API backend βασισμένη στο express του node.js. Η εφαρμογή αυτή είναι υπεύθυνη για την ανταλλαγή δεδομένων (από και προς) την MySQL database καθώς και εκτελεί διαδικαστικούς αλγορίθμους επεξεργασίας δεδομένων.

ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ

Αρχικά θα πρέπει να έχετε εγκατεστημένο το MySQL Server & Workbench. Έπειτα, για να γίνει επιτυχής σύνδεση με την node πρέπει ο user σας να είναι authorised με τον legacy 5.0 τρόπο. Αντί να ξανακάνετε setup τον υπολογιστή σας απλά εκτελέστε μια από τις 2 εντολές στο workbench:

```
CREATE USER 'foo'@'localhost' IDENTIFIED WITH mysql_native_password BY 'bar';  
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'MyNewPass';
```

```
17  
18  
19 const connection= mysql.createConnection({  
20   host: 'localhost',  
21   user : 'root',  
22   password : 'pavlos',  
23   database : 'linked_in',  
24   insecureAuth : false,  
25   multipleStatements: true  
26 });  
27
```

Ανάλογα αν θέλετε νέο χρήστη στο Workbench για την εκτέλεση του database του project. Ύστερα συμπληρώστε το username και τον κωδικό σας στο index.js του backend όπως ακριβώς βλέπετε στην εικόνα ή απλά βάλτε τα ίδια στοιχεία στις 2 παραπάνω εντολές mysql. Δυστυχώς αυτή η

διαδικασία δεν μπορεί να γίνει από εμάς όπως πολύ καλά γνωρίζετε αφού δεν έχουμε πρόσβαση στα στοιχεία σας.

Η μόνη πλατφόρμα που χρειάζεται για να εκτελεστεί η εργασία είναι η node.js και ο npm(συμπεριλαμβάνεται στο node) .Υπάρχει η δυνατότητα να βάλουμε την εργασία και τον intelij όπως είδαμε στο μάθημα αλλά και πάλι η node θα ήταν απαραίτητη(οδηγίες για το πως να το τρέξεις ακολουθούν).

Κατεβάστε την node από: <https://nodejs.org/en/download/>

Για περισσότερες οδηγίες:

[δείτε:https://phoenixnap.com/kb/install-node-js-npm-on-windows](https://phoenixnap.com/kb/install-node-js-npm-on-windows)

Επειδή το eclass δεν επιτρέπει να στείλουμε όλο το repository θα πρέπει να τρέξετε μερικές εντολές για να εγκαταστήσετε βιβλιοθήκες και dependencies της react. Μας δόθηκε οδηγία να μην στείλουμε τα node modules της react, αρα θα τα παράγετε ως εξής στο command line:

1. Npm i -g create-react-app
2. Create-react-app ergasia
3. στον φάκελο εργασία αντικαταστήστε τον φάκελο src με τον δικό μας
4. Npm install —save export-from-json
5. Npm install —save react-router-dom
6. Npm install —save styled-components

Ύστερα σε άλλο directory κάντε unzip το backend app μας που στάλθηκε ολόκληρο γιατί είναι μικρό σε χωρητικότητα. Τέλος κάντε import τα MySql δεδομένα από το .sql αρχείο στο παραδοτέο μας.

Στο workbench πηγαίνετε Server-> Data import-> import from self contained file δίνεται την τοποθεσία του αρχείου. Ίσως χρειαστεί να πατήσετε New στο πεδίο target default schema και να την ονομάσετε linked_in. Πατήστε import και θα σας δοθεί μήνυμα επιτυχίας.

ΤΡΟΠΟΣ ΕΚΤΕΛΕΣΗΣ

Ανοίγεται δύο terminals, ένα στο directory sqlserver και ένα στον φάκελο ergasia. Στο πρώτο εκτελέστε την εντολή node index.js, στο δεύτερο την npm start.

ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ ΕΦΑΡΜΟΓΗΣ

Για την εισαγωγή στην ιστοσελίδα χρειάζεται να κάνετε authenticate ως ένας από τους χρήστες της. Αυτό γίνεται με login και μπορείτε να δείτε το

email και τον κωδικό που χρειάζεται εκτυπώνοντας όλον τον πίνακα user στο workbench και να επιλέξετε έναν από αυτούς.

ΣΥΜΒΟΥΛΗ: το testing μας έγινε στους 2 πρώτους χρήστες panlos & George οπότε εκεί θα δείτε την πιο πολύ δραστηριότητα(posts, likes, notifications, φιλία(συνδεση) με άλλους χρήστες κλπ)

Αλλιώς μπορείτε να κάνετε sign up έναν νέο χρήστη και να αρχίσετε αυτές τις δραστηριότητες από την αρχή. Το sign up κάνει αυτόματο login
Έχουμε παραδοχή ότι αυτή η διαδικασία δεν χρησιμοποιεί SSL/TLS.

ΣΕΛΙΔΕΣ:

Στην **αρχική σελίδα** εμφανίζονται Posts δημοσιευμένα από τον ίδιο τον χρήστη και τους φίλους του. Στην υποσέλιδα "Αγγελίες" εμφανίζονται αγγελίες φίλων καθώς και αγγελίες από άγνωστους χρήστες αν τα θέματα(subjects) της αγγελίας έχουν κοινά με τα ενδιαφέροντα του χρήστη. Και στις 2 δυο σελίδες εμφανίζονται το πολύ 5 αναρτήσεις που προτείνει ο αλγόριθμος matrix factorization.

Σε αυτά τα post μπορεί να κάνει comment και like και στο page notifications του συγγραφέα θα υπάρχει ειδοποίηση για αυτο. Στο πάνω μέρος μπορείτε να ανεβάσετε τα δικά σας post

Στην σελίδα **δικτυο** ο χρήστης μπορεί να δει τους φίλους του και να μεταφερθεί στο userpage τους πατώντας πάνω στο ονομα τους. Φυσικά, δεν θα μπορούσε να λείπει και μια search bar που όσο πληκτρολογείς ένα όνομα σου εμφανίζει χρήστες που το όνομα τους ταιριάζει με αυτό που έχει εισαχθεί μέχρι στιγμής. Πατώντας σε ένα από αυτά τα ονόματα μεταφέρεσαι στο userpage τους.

Η σελίδα **userpage** έχει μια ιδιαιτερότητα σε σχέση με τα άλλα subpages που είναι διαθέσιμα μέσω της navbar. Σε αντίθεση με τα υπόλοιπα παίρνει το όρισμα του χρήστη που θέλουμε να δούμε μέσω του url. Αυτή η επιλογή έγινε λόγω του ότι πρέπει να φαίνεται το δημόσιο προφίλ του εκάστοτε χρήστη χωρίς να απαιτείται log in (όπως ακριβώς γίνεται σε όλα τα συγχρονα social media). Τέλος, σε αυτή η σελίδα υπάρχει η δυνατότητα για friend request του οποίου ο παραλήπτης το βλέπει στο notification page του.

Η σελίδα **messenger** περιέχει όλες τις υπάρχοντες συνομιλίες του χρήστη. Υπάρχει δυνατότητα συνομιλίας με άλλους χρήστες μέσω αναζήτησης τους στο search bar που λειτουργεί με τον ίδιο τρόπο με αυτή του δικτυου. Σε επίπεδο υλοποίησης έχουμε ένα table στο οποίο αποθηκεύουμε όλα τα μηνύματα και για να επιλέξουμε συνομιλίες μεταξύ ατόμων δημιουργούμε views. Αυτά τα παρουσιάζουμε με χρονολογική σειρά έτσι ώστε ο κάθε χρήστης να βλέπει τα νεότερα μηνύματα.

Αν κάποιος κάνει login με credentials admin(εκτυπώνοντας το user table στο workbench θα δείτε την στήλη admin που έχει τιμή 1 μόνο για έναν admin που έχουμε εγκαταστήσει)

Εκεί μπορείτε να κάνετε export τα στοιχεία των απλών χρηστών επιλέγοντας αυτούς που θέλετε (υπάρχουν επιλογές SELECT /UNSELECT ALL , JSON/XML). Δείτε ειδοποίηση του browser σας ότι κατέβηκε ένα αρχείο "download.json/xml"

ΕΙΚΟΝΕΣ

Η εφαρμογή μας δίνει τη δυνατότητα εικόνας προφίλ για κάθε χρήστη καθώς και για κάθε post. Η εκάστοτε εικόνα αποθηκεύεται τοπικά στον φάκελο uploads καθώς και στην data base μας.

ΣΧΟΛΙΑ ΣΕ ΕΠΙΠΕΔΟ ΥΛΟΠΟΙΗΣΗΣ

Αν κανείς θέλει να εξετάσει πως λειτουργεί και επιτυγχάνεται μια υποσελίδα της εφαρμογής αρκεί να πάει στο αρχείο όπου βρίσκεται το component της.

Η ροή της εφαρμογής αρχίζει από το App.js που μέσω του src/index.js σου εκτυπώνει ένα navigation bar αν είσαι logged in ή τα sign in ,signup κουμπιά αν δεν είσαι. Όταν πατάς αυτά τα κουμπιά το src/index.js σε μεταφέρει σε ένα sublink πχ localhost:3000/home ,localhost:3000/signin κλπ. Το app.js ορίζει για κάθε τέτοιο url τι component θα εκτελεστεί πχ στο localhost:3000/home θα εκτελεστεί το homepage.

Το APP δίνει πληροφορίες στα components μέσω των props και αυτά δίνουν πίσω πληροφορίες στο app μέσω συναρτήσεων που περνιούνται στα props. Είναι η μέθοδος state lifting της react.

Συχνά θα δείτε την συνάρτηση componentDidMount που αρχικοποιεί το state/πληροφορίες ενός component με δεδομένα από την βάση δεδομένων.

Η βάση δεδομένων κατα 90% τρέχει τετριμμένα mysql queries με ορίσματα από το front end που διαβάζει με το http url καθώς και τρέχει τον αλγόριθμο matrix factorisation για τον εκάστοτε συνδεδεμένο χρήστη.

Αυτά τα queries και fetch στην express είναι ασύγχρονα οπότε συχνά θα βλέπετε await που κάνει την εφαρμογή να 'περιμένει' , όσο ορίζουν τα promises τους, την mysql να φέρει τα δεδομένα.

Matrix Factorization

Η εκφώνηση υποδεικνύει την δημιουργία ενός αλγορίθμου matrix factorization με τα 3 latent features (likes, comments και προβολές) .Η σειρά σημαντικότητας είναι comments > likes > προβολές. Τα πρώτα 2 είναι ήδη μέρος της εφαρμογής και για το bonus εφαρμόσαμε το 3ο.

Ο αριθμός των προβολών μετριέται με το να αυξάνονται κατά 1 οι προβολές του post ακριβώς πριν επιστραφούν (από το back end) και οι

συνολικές προβολές ενός χρήστη αυξάνονται κατά n αφού του προβλήθηκαν n posts. Αναγνωρίζουμε ότι αυτός δεν είναι ο βέλτιστος τρόπος μέτρησης των προβολών αλλά κάθε άλλος ήταν πέρα από τις δυνάμεις μας και τον χρόνο που μας δόθηκε (εξάλλου δεν είμαστε senior full stack developers ακόμα).

Όταν φορτώνεται το home page μέσω του backend γίνεται αναζήτηση συνιστώμενων posts για τον συνδεδεμένο χρήστη.

Για αυτό τον σκοπό φτιάχνουμε πίνακες P , Q & R .

- Ο P έχει 3 στήλες και δείχνει για κάθε χρήστη(σε κάθε γραμμή) πόσα likes, comments και προβολές έχει κάνει. Το πρόγραμμά μας έχει το optimisation ότι έχει μόνο 1 γραμμή γιατί μας ενδιαφέρει να βρούμε recommendations για μόνο 1 χρήστη έτσι ώστε να σώσουμε computation time στον πολλαπλασιασμό πινάκων. Η γνώμη των άλλων χρηστών δίνει αντίκτυπο μέσω του πίνακα Q .
- Ο Q έχει n στήλες για n posts και 3 γραμμές για τα 3 latent features. Πάλι μέσω data management μαθαίνουμε πόσα likes comments και προβολές έχει ένα post.
- Το στοιχείο R_{ij} του R δείχνει το πως πιστεύουμε ότι ο χρήστης i αξιολογεί το post j . Στην περίπτωση που το post j είναι ένα από αυτά που θα εμφανίζονταν έτσι και αλλιώς στην homepage του χρήστη, δεν το λαμβάνουμε υπόψη στον υπολογισμό. Σε κάθε άλλη περίπτωση το R_{ij} κηρύσσεται άγνωστο, δηλαδή 0 στον πίνακα (Αν το βάζαμε null ή NaN δε θα ετρεχαν οι πράξεις καθώς και ορίσαμε ελαχιστη γνωστή αξιολογήγη $=0.001$).

Αυτοί οι 3 πίνακες περνάνε σε μια standard matrix factorisation function που μέσω πολλαπλασιασμού των P και Q υπολογίζουν την απόκλιση (error*). Έπειτα, σε κάθε επανάληψη ο πειραματικός πίνακας R προσεγγίζει όλο και περισσότερο στον πραγματικό R που υπολογίζεται μέσω του data management και της φόρμουλας μας (εξου και το error*). Με λίγα λόγια εκεί που ο παλιός R είχε 0 (άγνωστη τιμή), τώρα περιέχει μια τιμή αξιολόγησης. Παίρνουμε τα post με τις 5 καλύτερες άγνωστες (αλλά υπολογισμένες) αξιολογήσεις και τα επιστρέφουμε στο homepage.

Για την αξιολόγηση του αλγορίθμου με τεχνητά dataset όπως ζητάει η εκφώνηση (δηλαδή ο αλγόριθμος να τρεξει με αυτούς τους 3 πίνακες με τυχαίες τιμές), μπορείτε να κάνετε ένα http request στο backend πηγαίνοντας στο url

http://localhost:4000/matrix_factorization?user_id=1&random=1

Το user_id δεν έχει σημασία αφού οι πίνακες είναι τυχαίοι. Έτσι θα μπορείτε να συγκρίνετε τον πραγματικό και τον πειραματικό R !