**WorkNow: Ανάπτυξη ιστοσελίδας κοινωνικής δικτύωσης με ενσωματωμένο σύστημα προτάσεων με βάση αλληλεπιδράσεις χρηστών**

**Πίνακας Περιεχομένων**

1. [Εισαγωγή](#%CE%B5%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE)
2. [Σχεδιαστικές Αποφάσεις](#%CF%83%CF%87%CE%B5%CE%B4%CE%B9%CE%B1%CF)
   1. [Επιλογή Τεχνολογιών](#%CE%B5%CF%80%CE%B9%CE%BB%CE%BF%CE%B3%CE)
   2. [Δομή Βάσης Δεδομένων](#%CE%B4%CE%BF%CE%BC%CE%AE-%CE%B2%CE%AC%C)
   3. [Αρχιτεκτονική Εφαρμογής](#%CE%B1%CF%81%CF%87%CE%B9%CF%84%CE%B5%CE)
   4. Ασφάλεια και Κρυπτογράφηση
3. [Παραδοχές και Περιορισμοί](#%CF%80%CE%B1%CF%81%CE%B1%CE%B4%CE%BF%CF)
4. [Λεπτομέρειες Υλοποίησης](#%CE%BB%CE%B5%CF%80%CF%84%CE%BF%CE%BC%CE)
   1. [Σχεδιασμός Διεπαφής Χρήστη](#%CF%83%CF%87%CE%B5%CE%B4%CE%B9%CE%B1%CF)
   2. [Υλοποίηση Συστήματος Προτάσεων](#%CF%85%CE%BB%CE%BF%CF%80%CE%BF%CE%AF%CE)
   3. [Διαχείριση Αλληλεπιδράσεων](#%CE%B4%CE%B9%CE%B1%CF%87%CE%B5%CE%AF%CF)
5. [Εγκατάσταση και Εκτέλεση](#%CE%B5%CE%B3%CE%BA%CE%B1%CF%84%CE%AC%CF)
   1. [Οδηγίες Εγκατάστασης](#%CE%BF%CE%B4%CE%B7%CE%B3%CE%AF%CE%B5%CF)
   2. [Οδηγίες Εκτέλεσης](#%CE%BF%CE%B4%CE%B7%CE%B3%CE%AF%CE%B5%CF)
6. [Επίλογος](#%CE%B5%CF%80%CE%AF%CE%BB%CE%BF%CE%B3%CE)
7. [Βιβλιογραφία](#%CE%B2%CE%B9%CE%B2%CE%BB%CE%B9%CE%BF%CE)

**1. Εισαγωγή**

Ο στόχος αυτής της εργασίας ήταν η ανάπτυξη μιας ιστοσελίδας για την κοινωνική δικτύωση επαγγελματιών με ενσωματωμένο ένα σύστημα προτάσεων για χρήστες, βασισμένο σε αλληλεπιδράσεις όπως likes, σχόλια και ενδιαφέρον για άρθρα και αγγελίες εργασίας. Το σύστημα χρησιμοποιεί έναν αλγόριθμο Matrix Factorization για την ανάλυση των αλληλεπιδράσεων και τη δημιουργία εξατομικευμένων προτάσεων. Τα επόμενα κεφάλαια περιγράφουν τις σχεδιαστικές αποφάσεις, τις παραδοχές, την υλοποίηση της εφαρμογής, καθώς και τις διαδικασίες εγκατάστασης και εκτέλεσης.

**2. Σχεδιαστικές Αποφάσεις**

**2.1 Επιλογή Τεχνολογιών**

Για την ανάπτυξη της εφαρμογής επέλεξα τις εξής τεχνολογίες:

* **Backend:**

1. **Node.js**: Χρησιμοποιήθηκε για την ανάπτυξη του backend, προσφέροντας ευελιξία και ταχύτητα στην ανάπτυξη της εφαρμογής.
2. **Εxpress.js**: Για τη δημιουργία του web server και τη διαχείριση των routes.
3. **Path**: Για τη διαχείριση διαδρομών αρχείων.
4. **Mongoose**: Για την αλληλεπίδραση με τη MongoDB.
5. **Body-parser**: Για την ανάλυση των δεδομένων που στέλνονται μέσω HTTP POST αιτημάτων.
6. **Bcrypt**: Για την κρυπτογράφηση των κωδικών πρόσβασης των χρηστών.
7. **Cors**: Για τη διαχείριση αιτημάτων από διαφορετικά domains.
8. **Https**: Για τη δημιουργία ενός HTTPS server.
9. **Socket.io**: Για την ενσωμάτωση real-time επικοινωνίας μέσω WebSockets.
10. **Multer**: Για την επεξεργασία αρχείων που ανεβαίνουν μέσω των φορμών.
11. **Fs**: Για τη διαχείριση αρχείων στο σύστημα.
12. **Nodemailer**: Για την αποστολή emails για επαναφορά password.
13. **Crypto**: Για τη δημιουργία τυχαίων αριθμών και τη διαχείριση κρυπτογραφικών λειτουργιών.
14. **Express-session**: Για τη διαχείριση συνδέσεων χρηστών.
15. **Connect-mongo**: Για την αποθήκευση συνδέσεων στη MongoDB.
16. **Child\_process** (μέσω spawnSync): Για την εκτέλεση scripts και εντολών από το Node.js.
17. **Cron**: Για τη δημιουργία χρονοπρογραμματισμένων εργασιών (cron jobs).
18. **Axios**: Για HTTP αιτήματα.
19. **Chalk**: Για χρωματική κωδικοποίηση των console logs.

* **Frontend:** React.js, χρησιμοποιήθηκε για την ανάπτυξη του frontend, επιτρέποντας τη δημιουργία μιας διαδραστικής και δυναμικής διεπαφής χρήστη.
* **Database:** MongoDB, επιλέχθηκε ως βάση δεδομένων λόγω της ικανότητάς της να διαχειρίζεται μεγάλες ποσότητες μη δομημένων δεδομένων.
* **Γλώσσες Προγραμματισμού:** JavaScript, Python (για την υλοποίηση του αλγορίθμου συστάσεων)

**2.2 Δομή Βάσης Δεδομένων**

Η βάση δεδομένων MongoDB αποτελείται από τις εξής συλλογές:

* **users:** Περιλαμβάνει τα στοιχεία των χρηστών όπως όνομα, email, κωδικό πρόσβασης, κλπ.
* **articles:** Περιλαμβάνει πληροφορίες για τα άρθρα που δημοσιεύουν οι χρήστες.
* **interactions:** Περιλαμβάνει τις αλληλεπιδράσεις των χρηστών με άρθρα και αγγελίες εργασίας.
* **joblistings:** Περιλαμβάνει πληροφορίες για τις αγγελίες εργασίας που δημοσιεύονται.
* **sessions:** Περιλαμβάνει πληροφοριές για τις περιόδους σύνδεσης.
* **recommendations:** Περιλαμβάνει πληροφορίες για προτάσεις σε δουλειές και άρθρα που γίνονται στους χρήστες.
* **conversations:** Περιλαμβάνει τις συζητήσεις ανάμεσα στους χρήστες.
* **connectionrequests:** Περιλαμβάνει τις αιτήσεις σύνδεσης μεταξύ χρηστών.

**2.3 Αρχιτεκτονική Εφαρμογής**

Η εφαρμογή αναπτύχθηκε σύμφωνα με το μοντέλο-θέα-ελεγκτής (MVC), όπου το frontend (React.js) αποτελεί τη θέα, το backend (Node.js/Express.js) λειτουργεί ως ο ελεγκτής, και η MongoDB ως το μοντέλο. Οι χρήστες αλληλεπιδρούν με την εφαρμογή μέσω του frontend, το οποίο επικοινωνεί με το backend μέσω API κλήσεων. Το backend χειρίζεται την επιχειρησιακή λογική και τη διαχείριση δεδομένων, ενώ η MongoDB αποθηκεύει τα δεδομένα των χρηστών και των άρθρων.

**2.4 Ασφάλεια και Κρυπτογράφηση**

Για την προστασία των δεδομένων, η εφαρμογή υλοποιεί τα εξής:

* SSL με Self-Signed Πιστοποιητικό: Η χρήση HTTPS με self-signed SSL πιστοποιητικό διασφαλίζει την κρυπτογράφηση της επικοινωνίας μεταξύ πελάτη και διακομιστή.
* TLS 1.2: Επιλέχθηκε το TLS 1.2 ως το πρωτόκολλο ασφαλείας για την κρυπτογράφηση της επικοινωνίας, εξασφαλίζοντας την ασφάλεια και την ακεραιότητα των δεδομένων που μεταδίδονται.

**3. Παραδοχές και Περιορισμοί**

Κατά την ανάπτυξη της εφαρμογής, έγιναν οι εξής παραδοχές:

• Τεχνητά Δεδομένα: Δεδομένου ότι δεν υπήρχαν διαθέσιμα πραγματικά δεδομένα χρηστών, δημιουργήσαμε τεχνητά δεδομένα για τις ανάγκες της ανάπτυξης και του ελέγχου της εφαρμογής.

• Ασφάλεια: Για την ασφάλεια των κωδικών πρόσβασης, χρησιμοποιήθηκε η βιβλιοθήκη bcrypt. Χρησιμοποιείται το πρωτόκολλο HTTPS και επίσης το SSL certificate είναι self-signed και πρέπει να εισαχθεί χειροκίνητα στον web browser. Όλες οι συνδέσεις στο σύστημα προστατεύονται με TLS.

• Περιορισμοί: Το σύστημα συστάσεων βασίζεται αποκλειστικά στις αλληλεπιδράσεις χρηστών με άρθρα και αγγελίες. Στην τρέχουσα υλοποίηση, δεν συμπεριλαμβάνεται υποστήριξη για προτάσεις βασισμένες σε εξωτερικά δεδομένα ή προτιμήσεις.

**4. Λεπτομέρειες Υλοποίησης**

**4.1 Σχεδιασμός Διεπαφής Χρήστη**

Η διεπαφή χρήστη σχεδιάστηκε με στόχο την απλότητα και τη λειτουργικότητα. Χρησιμοποιήθηκαν HTML, CSS, και JavaScript για την ανάπτυξη των σελίδων, και η βιβλιοθήκη React για τη διαχείριση της κατάστασης και την απόδοση των στοιχείων στο frontend.

**4.2 Υλοποίηση Συστήματος Προτάσεων**

Το σύστημα προτάσεων βασίστηκε στον αλγόριθμο Matrix Factorization, ο οποίος υλοποιήθηκε χρησιμοποιώντας Python. Οι αλληλεπιδράσεις χρηστών με τα άρθρα και τις αγγελίες αναλύονται για να δημιουργηθούν διανύσματα που αναπαριστούν τους χρήστες και τα αντικείμενα (άρθρα/αγγελίες). Οι προτάσεις δημιουργούνται με βάση την ομοιότητα αυτών των διανυσμάτων.

**4.3 Διαχείριση Αλληλεπιδράσεων**

Οι αλληλεπιδράσεις των χρηστών με τα άρθρα (όπως likes, σχόλια, και δηλώσεις ενδιαφέροντος) καταγράφονται στη συλλογή interactions. Αυτές οι αλληλεπιδράσεις χρησιμοποιούνται για να υπολογιστεί η προσαρμοσμένη πρόταση περιεχομένου για τον κάθε χρήστη.

**5. Εγκατάσταση και Εκτέλεση**

**5.1 Οδηγίες Εγκατάστασης**

1. **Εγκατάσταση των απαιτούμενων προγραμμάτων:**

Εγκατάσταση MongoDB (<https://www.mongodb.com/docs/manual/installation/>)

Εγκατάσταση Node.js (<https://nodejs.org/en/download/package-manager>)

Εγκατάσταση Python (<https://www.python.org/downloads/>)

Εγκατάσταση NumPy: pip install numpy

1. **Ρύθμιση της βάσης δεδομένων MongoDB:**
   * Δημιουργήστε μια νέα βάση δεδομένων MongoDB με όνομα “worknow”
   * Εισαγωγή του self-signed SSL (cert.pem) στο: chrome//settings/security ->Trusted Root Certification Authorities
   * Δημιουργήστε τα παρακάτω collections:
     1. articles
     2. connectionrequests
     3. conversations
     4. interactions
     5. joblistings
     6. recommendations
     7. sessions
     8. users
2. **Εκτέλεση του backend:**

Τρέξτε τις εντολές:

1. npm install axios bcrypt connect-mongo cors crypto express express-session fs http-errors mongod mongoose multer node-cron nodemailer socket.io socket.io-client chalk
2. npm install --save-dev nodemon concurrently

4. **Εκτέλεση του frontend:**

Τρέξτε την εντολή στο directory worknow/client:

* npm install @testing-library/jest-dom @testing-library/react @testing-library/user-event axios multer react react-dom react-router-dom react-scripts socket.io socket.io-client web-vitals

Τροποποίηση directories στο worknow/client:

* "start": "set HTTPS=true&& set SSL\_CRT\_FILE=C:\\Users\\UserName\\Desktop\\worknow\\cert.pem&& set SSL\_KEY\_FILE=C:\\Users\\UserName\\Desktop\\worknow\\key.pem&& react-scripts start",

**5.2 Οδηγίες Εκτέλεσης**

Μετά την εγκατάσταση, μπορείτε να εκτελέσετε την εφαρμογή τρέχοντας στο directory worknow “npm run start:both” και θα ανακατευθυνθείτε αυτόματα στη διεύθυνση https://localhost:3000 από τον περιηγητή σας. Οι χρήστες μπορούν να εγγραφούν, να δημοσιεύσουν άρθρα και να αλληλοεπιδράσουν με το περιεχόμενο. Το σύστημα προτάσεων θα ενημερώνεται αυτόματα με βάση τις αλληλεπιδράσεις. Επιπλεόν ο διαχειριστής μπορεί να εξάγει δεδομένα χρηστών, να τους διαγράψει, να διαγράψει άρθρα και να διαγράψει τις αγγελίες εργασίας.

**6. Επίλογος**

Κατά την ανάπτυξη της εφαρμογής, αντιμετώπισα αρκετές προκλήσεις, όπως η υλοποίηση μιας διεπαφής φιλικής προς τον καθημερινό χρήστη, η διαχείριση των αλληλεπιδράσεων σε πραγματικό χρόνο καθώς θεωρούσα ότι είναι πολύ σημαντική η αμεσότητα της αλληλεπίδρασης ανάμεσα στους χρήστες, η διασφάλιση της ασφάλειας των δεδομένων των χρηστών και της απόδοσης του συστήματος. Η μεγαλύτερη πρόκληση θεωρώ πως ήταν η βελτιστοποίηση της απόδοσης του συστήματος και παρόλο που έχω ασχοληθεί αρκετά με το παρόν ζήτημα πιστεύω ότι ακόμα χρήζει βελτίωση αφού δεν είμαι σίγουρος εάν θα λειτουργεί αποτελεσματικά με μεγάλα σύνολα δεδομένων. Επιπλέον, η υλοποίηση ασφαλών μηχανισμών επαναφοράς κωδικών μέσω OTP και η διασφάλιση της επικοινωνίας μέσω HTTPS και TLS 1.2 συνέβαλαν στην ενίσχυση της συνολικής ασφάλειας της πλατφόρμας, τα όποια επίσης ήταν χρονοβόρα να υλοποιηθούν καθώς και ήταν περίπλοκο να βρω ένα email API και να δημιουργήσω ένα self-signed certificate και να κάνω να μην βγάζει σφάλμα ο browser. Η χρήση του αλγορίθμου Matrix Factorization μέσω του recommendation\_service.py βελτίωσε την εμπειρία των χρηστών προσφέροντας προσωποποιημένες προτάσεις άρθρων και θέσεων εργασίας και χρειάστηκε εκτενές αναζήτηση, διάβασμα και σκέψη ώστε να κατανοήσω και να ενσωματώσω.

Μέσα από αυτή τη διαδικασία, απέκτησα πολύτιμες γνώσεις για την ανάπτυξη, από το μετωπιαίο έως και το νωτιαίο άκρο, μιας ιστοσελίδας, στην ανάπτυξη συστημάτων προτάσεων και στη διαχείριση δεδομένων χρηστών. Τέλος συνειδητοποίησα πόσο μεγάλος είναι ο κόσμος του web development και πόσο ενδιαφέρον έχει.

**7. Βιβλιογραφία**

* MongoDB Documentation. Available at: <https://docs.mongodb.com/>
* React.js Documentation. Available at: https://reactjs.org/docs/getting-started.html
* Node.js Documentation. Available at: https://nodejs.org/en/docs/
* Python Documentation. Available at: <https://docs.python.org/3/>