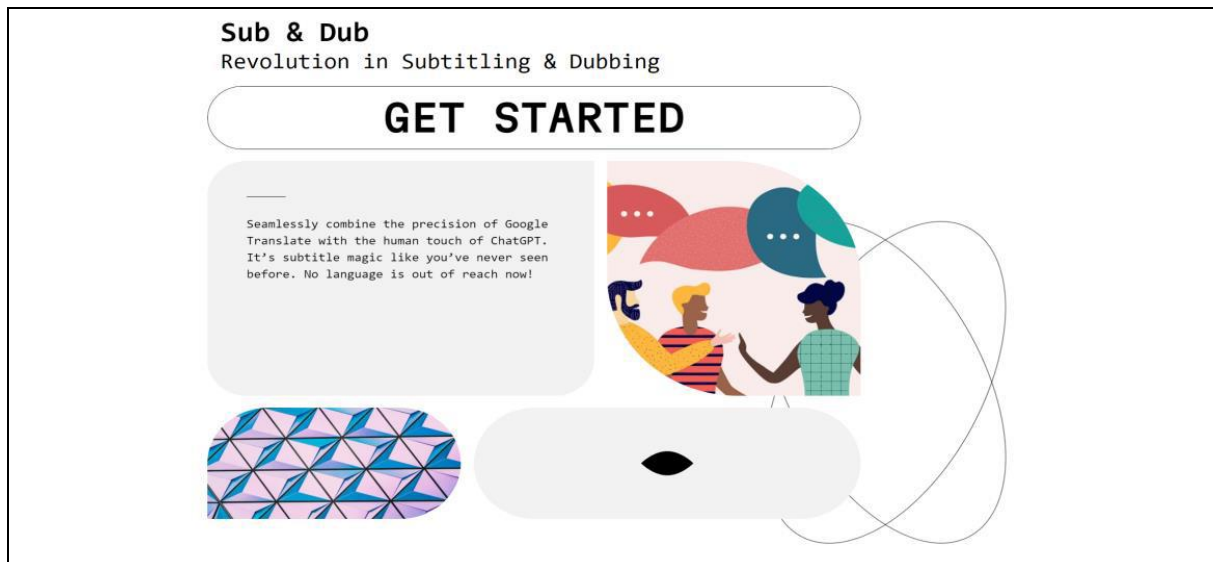


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Αναπτυξη Web Εφαρμογής για Υποβοηθούμενο  
Υποτιλισμό και Μεταγλώττιση YouTube Βιντεο»



Της φοιτήτριας  
Καραμπίδου Κωνσταντίνας  
Αρ. Μητρώου: 164674

Επιβλέπων  
Ονοματεπώνυμο Σαλαμπάσης  
Μιχάλης

**Ημερομηνία 24/01/2025**

Τίτλος Δ.Ε. Αναπτυξη Web Εφαρμογής για Υποβοηθούμενο Υποτιτλισμό και Μεταγλώττιση  
YouTube Βιντεο

Κωδικός Δ.Ε. 24147

Όνοματεπώνυμο φοιτητή/τών Καραμπίδου Κωνσταντίνα

Όνοματεπώνυμο εισηγητή Σαλαμπάσης Μιχάλης

Ημερομηνία ανάληψης Δ.Ε. 11-03-2024

Ημερομηνία περάτωσης Δ.Ε. 24-01-2025

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτήτριας Καραμπίδου Κωνσταντίνας που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Περίληψη

Η παρούσα εργασία εξετάζει τον σχεδιασμό και την ανάπτυξη μιας καινοτόμου διαδικτυακής εφαρμογής, η οποία στοχεύει στη βελτίωση της προσβασιμότητας και της εμπειρίας χρήστη κατά τη δημιουργία και διαμοιρασμό ψηφιακού περιεχομένου. Με την αυξανόμενη εξάρτηση από τις διαδικτυακές πλατφόρμες για την πρόσβαση σε πληροφορίες, η εξασφάλιση της ένταξης διαφορετικών ομάδων χρηστών γίνεται ιδιαίτερα σημαντική. Η εφαρμογή αυτή επικεντρώνεται στα βίντεο του YouTube, μια διαδεδομένη μορφή διαδικτυακού περιεχομένου, και αποσκοπεί στην κάλυψη της ανάγκης για βελτιωμένη προσβασιμότητα μέσω υποτίτλων σε πολλές γλώσσες.

Η εφαρμογή αναπτύχθηκε με σκοπό να επιτρέπει στους χρήστες να δημιουργούν, να επεξεργάζονται και να προσαρμόζουν υπότιτλους για βίντεο YouTube με έναν εύχρηστο και διαισθητικό τρόπο. Επιπλέον, προσφέρει λειτουργίες διαμοιρασμού υποτίτλων, επιτρέποντας στην κοινότητα των χρηστών να συνεργάζεται και να μοιράζεται το περιεχόμενο, ενισχύοντας τη συλλογική προσπάθεια. Με τη χρήση σύγχρονων τεχνολογιών όπως το Angular Framework, η εφαρμογή ενσωματώνεται άψογα με διάφορα APIs, δίνοντας τη δυνατότητα εισαγωγής και επεξεργασίας βίντεο απευθείας μέσα στην πλατφόρμα, χρησιμοποιώντας serverless λύσεις όπως το Firebase. Το User Interface δίνει έμφαση στην απλότητα και την αποδοτικότητα, καθιστώντας τη διαδικασία υποτιτλισμού προσιτή σε ένα ευρύ φάσμα χρηστών, ακόμα και σε εκείνους με περιορισμένες τεχνολογικές γνώσεις.

Συνοψίζοντας, η εργασία αυτή συμβάλλει σημαντικά στον τομέα της τεχνολογικής προσβασιμότητας, προσφέροντας μια πρακτική λύση για τον εύκολο υποτιτλισμό των βίντεο του YouTube και ενθαρρύνοντας τη συνεργασία και τη συμμετοχή της κοινότητας μέσω μιας φιλικής προς τον χρήστη διαδικτυακής εφαρμογής.

# «Development of Web Application for Assisted Subtitling and Dubbing of YouTube Videos»

«Konstantina Karampidou»

## **Abstract**

This thesis examines the design and development of an innovative web application aimed at improving accessibility and user experience in the creation and sharing of digital content. With the increasing reliance on online platforms for accessing information, ensuring the inclusion of diverse user groups becomes particularly important. This application focuses on YouTube videos, a widely consumed form of online content, and aims to address the need for improved accessibility through subtitles in multiple languages.

The application was developed to allow users to create, edit, and customize subtitles for YouTube videos in a user-friendly and intuitive manner. Additionally, it offers subtitle sharing features, enabling the user community to collaborate and share content, thereby enhancing collective efforts. By utilizing modern web technologies such as the Angular Framework, the application seamlessly integrates with various APIs, allowing users to directly import and edit videos within the platform, using serverless solutions like Firebase. The User Interface emphasizes simplicity and efficiency, making the subtitling process accessible to a wide range of users, including those with limited technological skills.

In conclusion, this work significantly contributes to the field of technological accessibility by offering a practical solution for easily subtitling YouTube videos and encouraging community collaboration and participation through a user-friendly web application.

# Περιεχόμενα

Περίληψη.....	iv
Abstract .....	v
Περιεχόμενα .....	vi
Κατάλογος Σχημάτων .....	ix
Συντομογραφίες.....	x
Κεφάλαιο 1ο: Γενική Επισκόπηση του Υποτιτλισμού .....	11
1.1 Εισαγωγή.....	11
1.1.1 Ορισμός και Σκοπός του Υποτιτλισμού .....	11
1.1.2 Η Σημασία του Υποτιτλισμού για την Κατανόηση Περιεχομένου.....	11
1.2 Ο Ρόλος του Υποτιτλισμού στο YouTube.....	12
1.2.1 Σημασία για την Προσβασιμότητα.....	12
1.2.2 Διευκόλυνση της Κατανόησης .....	12
1.2.3 Βελτίωση της Προσβασιμότητας για Άτομα με Προβλήματα Ακοής.....	12
1.2.4 Αυτόματη Δημιουργία Υποτίτλων .....	12
1.2.5 Συνολική Αξιολόγηση .....	12
1.3 Πρόκληση και Αναγκαιότητα του Υποτιτλισμού στη Σύγχρονη Ψηφιακή Εποχή .....	13
1.3.1 Ακρίβεια Υποτίτλων.....	13
1.3.2 Συγχρονισμός με το Βίντεο .....	13
1.3.3 Ανάγκη για Συστήματα Υποτίτλων .....	13
1.4 Στόχος της Εργασίας .....	13
Κεφάλαιο 2ο: Διαμοιρασμός και Συνεργασία στο Σύστημα Υποτίτλων.....	14
2.1 Εισαγωγή.....	14
2.1.1 Ορισμός και Χρησιμότητα του Διαμοιρασμού.....	14
2.2 Διαμοιρασμός.....	14
2.2.1 Διαμοιρασμός στον Τομέα των Υποτίτλων .....	14
2.2.2 Προκλήσεις στον Διαμοιρασμό Υποτίτλων .....	15
2.2.3 Λύσεις για Αποτελεσματικό Διαμοιρασμό και Συνεργασία.....	15
2.3 Επίπεδα Δικαιωμάτων στον Διαμοιρασμό Υποτίτλων.....	15
2.4 Επίλογος.....	16
Κεφάλαιο 3ο: Ανάλυση και Ανάπτυξη Κοινότητας Χρηστών Υποτίτλων.....	17
3.1 Εισαγωγή.....	17
3.2 Στρατηγικές για την Προσέλκυση και Διατήρηση Χρηστών .....	17

3.2.1	Εκπαίδευση και Υποστήριξη.....	17
3.2.2	Κίνητρα και Αναγνώριση .....	17
3.2.3	Εύχρηστα Εργαλεία.....	17
3.3	Χαρακτηριστικά που Ενισχύουν τη Συμμετοχή.....	17
3.3.1	Διαφάνεια και Συνεργασία .....	17
3.3.2	Κοινότητες και Φόρουμ.....	17
3.3.3	Ανατροφοδότηση και Βελτίωση.....	18
3.4	Community Subtitling και Bidding .....	18
3.5	Επίλογος.....	18
Κεφάλαιο 4ο:	Τεχνολογίες Web Development / API που χρησιμοποιήθηκαν .....	19
4.1	Angular.....	19
4.1.1	TypeScript .....	19
4.1.2	Αρχιτεκτονική και Βασικά Χαρακτηριστικά της Angular .....	19
4.1.3	Change Detection .....	20
4.1.4	Lifecycle Hooks.....	21
4.1.5	Data Binding.....	21
4.2	Firebase .....	23
4.2.1	Εισαγωγή.....	23
4.2.2	Δυνατότητες.....	23
4.2.3	Cloud Firestore και Βάσεις Δεδομένων στο Firebase .....	24
4.2.4	Κύρια Χαρακτηριστικά του Cloud Firestore:.....	24
4.2.5	Πλεονεκτήματα του Firestore.....	24
4.2.6	Μειονεκτήματα του Firestore.....	24
4.2.7	Collections.....	25
4.2.8	Where Queries .....	25
4.2.9	Χρήση των Filter και Map σε Δεδομένα του Firestore .....	25
4.2.10	Συμπέρασμα .....	26
4.3	EmailJS.....	26
4.3.1	Χρησιμότητα του EmailJS.....	26
4.3.2	Δυνατότητες του EmailJS.....	26
4.3.3	Ευκολία Χρήσης.....	27
4.3.4	Συμπέρασμα.....	27
4.4	Git.....	27
4.4.1	Github Pages.....	28
Κεφάλαιο 5ο:	Αναλυτική Περιγραφή της Εφαρμογής.....	29

5.1	Dashboard.....	29
5.2	My Videos .....	30
5.2.1	Κοινοποίηση υποτίτλου.....	31
5.2.2	Δικαιώματα χρήστη.....	38
5.2.3	Μεταβίβαση ιδιοκτησίας .....	42
5.3	My Shared Videos .....	46
5.4	Community.....	48
5.5	Messages .....	55
Κεφάλαιο 6ο:	Συμπεράσματα & Βελτιώσεις.....	59
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		61
ΠΑΡΑΡΤΗΜΑ Α : Πηγαίος Κώδικας .....		62

## Κατάλογος Σχημάτων

Σχήμα 5.1 Dashboard.....	30
Σχήμα 5.2 My Videos - Details.....	31
Σχήμα 5.3 Initial share window.....	31
Σχήμα 5.4 User rights.....	32
Σχήμα 5.5 Send notification email.....	32
Σχήμα 5.6 Email notification (EmailJS) .....	33
Σχήμα 5.7 Remove right.....	34
Σχήμα 5.8 View-only mode.....	38
Σχήμα 5.9 Comment mode.....	39
Σχήμα 5.10 Comment box.....	39
Σχήμα 5.11 Comment text on the subtitle box.....	40
Σχήμα 5.12 Editor view.....	40
Σχήμα 5.13 Κλείδωμα υπότιτλου.....	41
Σχήμα 5.14 Μήνυμα κλειδώματος υπότιτλου.....	42
Σχήμα 5.15 Right options.....	42
Σχήμα 5.16 Transfer Ownership.....	43
Σχήμα 5.17 Send invitation email (EmailJS).....	44
Σχήμα 5.18 Accept ownership invitation message.....	44
Σχήμα 5.19 Owner pending.....	45
Σχήμα 5.20 No Deadline request help.....	46
Σχήμα 5.21 Deadline request help.....	46
Σχήμα 5.22 My Shared Videos view.....	47
Σχήμα 5.23 Community search.....	48
Σχήμα 5.24 Community search by Language.....	49
Σχήμα 5.25 Community sort.....	49
Σχήμα 5.26 Place a bid without deadline .....	50
Σχήμα 5.27 Place a bid with deadline .....	51
Σχήμα 5.28 Offer message.....	51
Σχήμα 5.29 Accept message.....	52
Σχήμα 5.30 Close request message.....	53
Σχήμα 5.31 Edit request with no deadline .....	54
Σχήμα 5.32 Edit request with deadline .....	54
Σχήμα 5.33 Changed deadline message.....	55

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

## Κεφάλαιο 1ο: Γενική Επισκόπηση του Υποτιτλισμού

### 1.1 Εισαγωγή

Ο υποτιτλισμός είναι κρίσιμος για την κατανόηση και την προσβασιμότητα στις οπτικοακουστικές παραγωγές, επιτρέποντας σε θεατές που μιλούν διαφορετικές γλώσσες να απολαύσουν και να κατανοήσουν περιεχόμενο. Από την πρώιμη χρήση του στον κινηματογράφο έως τις σύγχρονες ψηφιακές εφαρμογές, έχει εξελιχθεί σε ένα σύνθετο πεδίο που συνδυάζει γλωσσικές δεξιότητες με τεχνολογικές προκλήσεις.

Σήμερα, οι υπότιτλοι διευκολύνουν όχι μόνο τη μετάφραση αλλά και την πρόσβαση για άτομα με προβλήματα ακοής, καθώς και την εκμάθηση γλωσσών. Στο κεφάλαιο αυτό, θα αναλύσουμε τον ορισμό, τον σκοπό και την εξέλιξη του υποτιτλισμού, καθώς και τη σημασία του στις σύγχρονες οπτικοακουστικές παραγωγές.

#### 1.1.1 Ορισμός και Σκοπός του Υποτιτλισμού

Ο υποτιτλισμός αποτελεί μια μέθοδο μετάφρασης που χρησιμοποιείται στις οπτικοακουστικές παραγωγές για την αναπαραγωγή του διαλόγου ή άλλων ακουστικών στοιχείων με τη μορφή γραπτών κειμένων στην οθόνη. Οι υπότιτλοι εμφανίζονται συνήθως στο κάτω μέρος της οθόνης και λειτουργούν ως μέσο κατανόησης για θεατές που δεν γνωρίζουν τη γλώσσα του πρωτότυπου ήχου ή έχουν δυσκολία στην ακοή.

Η ιστορική ανάπτυξη του υποτιτλισμού ξεκίνησε στις αρχές του 20ού αιώνα, με την εμφάνιση του κινηματογράφου. Αρχικά, οι υπότιτλοι χρησιμοποιούνταν κυρίως για τη μετάφραση ξένων ταινιών, αλλά με την πάροδο του χρόνου εξελίχθηκαν σε ένα απαραίτητο εργαλείο για τη διεθνή διανομή οπτικοακουστικών περιεχομένων. Σήμερα, οι υπότιτλοι είναι κρίσιμοι στις τηλεοπτικές εκπομπές, τις διαδικτυακές πλατφόρμες ροής, και τα βιντεοπαιχνίδια, καθώς εξασφαλίζουν την προσβασιμότητα και την κατανόηση του περιεχομένου από ένα ευρύ και διεθνές κοινό.

Ο σκοπός του υποτιτλισμού είναι πολλαπλός: αφενός διευκολύνει την επικοινωνία μεταξύ διαφορετικών γλωσσικών κοινοτήτων και αφετέρου παρέχει πρόσβαση σε άτομα με προβλήματα ακοής. Επιπλέον, οι υπότιτλοι συμβάλλουν στην εκμάθηση γλωσσών, την πολιτισμική κατανόηση και την ενίσχυση της εμπειρίας του θεατή.

#### 1.1.2 Η Σημασία του Υποτιτλισμού για την Κατανόηση Περιεχομένου

Οι υπότιτλοι παίζουν έναν καθοριστικό ρόλο στην κατανόηση οπτικοακουστικού περιεχομένου, ιδιαίτερα για άτομα που δεν είναι εξοικειωμένα με τη γλώσσα του πρωτότυπου ήχου ή προέρχονται από διαφορετική πολιτισμική προέλευση. Η κύρια λειτουργία τους είναι να προσφέρουν μια μεταφρασμένη ή αποδομένη εκδοχή του διαλόγου και άλλων ακουστικών στοιχείων, καθιστώντας το περιεχόμενο προσβάσιμο σε ένα ευρύτερο κοινό.

Για θεατές που δεν γνωρίζουν τη γλώσσα του πρωτότυπου, οι υπότιτλοι επιτρέπουν την κατανόηση των διαλόγων και των μηνυμάτων της παραγωγής χωρίς την ανάγκη για γνώση της γλώσσας. Αυτό είναι ιδιαίτερα σημαντικό για τη διεθνή διανομή ταινιών, τηλεοπτικών εκπομπών και άλλων οπτικοακουστικών έργων, όπου η γλωσσική κατανόηση μπορεί να αποτελέσει εμπόδιο.

Επιπλέον, οι υπότιτλοι συμβάλλουν στην γεφύρωση πολιτισμικών διαφορών, διευκολύνοντας τη μετάδοση πολιτισμικών εννοιών και συμπεριφορών που μπορεί να είναι ξένες ή μη προφανείς για το κοινό. Μέσω της προσεκτικής μετάφρασης και του πολιτισμικού προσανατολισμού, οι υπότιτλοι

βοηθούν στην κατανομή πολιτισμικών αναφορών, αστείων, και άλλων ιδιομορφιών που διαφορετικά θα μπορούσαν να χαθούν ή να παρεξηγηθούν.

Η σημασία των υποτίτλων επεκτείνεται επίσης στην προσβασιμότητα για άτομα με προβλήματα ακοής, καθώς παρέχουν το κείμενο του διαλόγου και άλλων ήχων που δεν μπορούν να ακουστούν. Αυτό ενισχύει την εμπειρία του θεατή και διασφαλίζει ότι το περιεχόμενο είναι διαθέσιμο σε όλους, ανεξαρτήτως ικανοτήτων.

## **1.2 Ο Ρόλος του Υποτιτλισμού στο YouTube**

### **1.2.1 Σημασία για την Προσβασιμότητα**

Οι υπότιτλοι έχουν καθοριστική σημασία στο YouTube, μια από τις μεγαλύτερες πλατφόρμες ψηφιακού περιεχομένου παγκοσμίως. Η πλατφόρμα προσφέρει βίντεο που καλύπτουν ένα ευρύ φάσμα θεμάτων, όπως εκπαιδευτικά, επαγγελματικά, ψυχαγωγικά και προσωπικά vlogs. Η χρήση υποτίτλων είναι κρίσιμη για την προσβασιμότητα και την κατανόηση αυτών των βίντεο από ένα παγκόσμιο κοινό.

### **1.2.2 Διευκόλυνση της Κατανόησης**

Οι υπότιτλοι επιτρέπουν σε χρήστες που δεν γνωρίζουν τη γλώσσα του πρωτότυπου βίντεο να κατανοήσουν το περιεχόμενο, διευκολύνοντας τη διάδοση του περιεχομένου πέρα από γλωσσικά σύνορα. Για παράδειγμα, ένα βίντεο στα αγγλικά μπορεί να προσεγγίσει θεατές που μιλούν άλλες γλώσσες μέσω της μετάφρασης των υποτίτλων, καθιστώντας το περιεχόμενο προσβάσιμο σε μεγαλύτερο κοινό. Επιπλέον, το YouTube προσφέρει εργαλεία που επιτρέπουν στους δημιουργούς να προσθέτουν υπότιτλους σε πολλές γλώσσες, διευρύνοντας ακόμη περισσότερο την προσβασιμότητα του περιεχομένου τους.

### **1.2.3 Βελτίωση της Προσβασιμότητας για Άτομα με Προβλήματα Ακοής**

Οι υπότιτλοι επίσης βελτιώνουν την προσβασιμότητα για άτομα με προβλήματα ακοής. Στο YouTube, οι υπότιτλοι μπορούν να παρέχουν μια ακριβή καταγραφή του διαλόγου και των ήχων, εξασφαλίζοντας ότι όλοι οι χρήστες έχουν τη δυνατότητα να κατανοήσουν το περιεχόμενο των βίντεο, ανεξαρτήτως ικανότητας ακοής.

### **1.2.4 Αυτόματη Δημιουργία Υποτίτλων**

Επιπλέον, η δυνατότητα αυτόματης δημιουργίας υποτίτλων που προσφέρει το YouTube ενισχύει την εμπειρία του χρήστη, καθιστώντας πιο εύκολη την αναζήτηση και την κατανόηση περιεχομένου σε διαφορετικές γλώσσες. Αυτός ο αυτοματισμός είναι χρήσιμος για τη γρήγορη πρόσβαση σε περιεχόμενο, αν και η ακρίβεια των αυτόματων υποτίτλων μπορεί να ποικίλλει, απαιτώντας συχνά επεξεργασία από τους δημιουργούς για την καλύτερη ποιότητα.

### **1.2.5 Συνολική Αξιολόγηση**

Συνολικά, οι υπότιτλοι στο YouTube είναι απαραίτητοι για την παγκόσμια διάδοση του περιεχομένου, διευκολύνοντας την επικοινωνία και τη συμπερίληψη σε μια πλατφόρμα που συνδέει ανθρώπους από όλο τον κόσμο.

### **1.3 Πρόκληση και Αναγκαιότητα του Υποτιτλισμού στη Σύγχρονη Ψηφιακή Εποχή**

#### **1.3.1 Ακρίβεια Υποτίτλων**

Στη σύγχρονη ψηφιακή εποχή, η ακρίβεια των υποτίτλων είναι μία από τις βασικές προκλήσεις. Οι υπότιτλοι πρέπει να μεταφέρουν το ακριβές νόημα του διαλόγου και των ήχων, κάτι που μπορεί να είναι δύσκολο λόγω των διαφορετικών πολιτισμικών συμφραζομένων και των τεχνικών περιορισμών. Η σωστή μετάφραση και προσαρμογή είναι απαραίτητες για να διασφαλιστεί ότι οι θεατές κατανοούν σωστά το περιεχόμενο.

#### **1.3.2 Συγχρονισμός με το Βίντεο**

Μια άλλη σημαντική πρόκληση είναι ο συγχρονισμός των υποτίτλων με το βίντεο. Οι υπότιτλοι πρέπει να εμφανίζονται και να εξαφανίζονται ακριβώς τη στιγμή που τα λόγια και οι ήχοι συμβαίνουν. Ο λανθασμένος συγχρονισμός μπορεί να προκαλέσει σύγχυση και να αποδυναμώσει την εμπειρία του θεατή, καθώς μπορεί να μην ακολουθήσουν την πλοκή σωστά.

#### **1.3.3 Ανάγκη για Συστήματα Υποτίτλων**

Η ανάγκη για συστήματα που διευκολύνουν την παραγωγή και τον διαμοιρασμό υποτίτλων είναι επίσης σημαντική. Με την ανάπτυξη του ψηφιακού περιεχομένου, η συμμετοχή της κοινότητας στη δημιουργία και επεξεργασία υποτίτλων μπορεί να ενισχύσει την προσβασιμότητα και να καλύψει τα κενά που προκύπτουν από επαγγελματικές υπηρεσίες. Εργαλεία και πλατφόρμες που επιτρέπουν στους χρήστες να προσθέτουν, να διορθώνουν και να μοιράζονται υπότιτλους συμβάλλουν στην επίτευξη υψηλότερης ποιότητας και ακριβούς περιεχομένου.

### **1.4 Στόχος της Εργασίας**

Η εργασία στοχεύει στην ανάπτυξη μιας web εφαρμογής που θα αντιμετωπίσει κενά και περιορισμούς στις υπάρχουσες πλατφόρμες υποτιτλισμού, όπως το YouTube Studio. Η εφαρμογή θα βελτιώσει τη διαδικασία υποτιτλισμού για βίντεο στο YouTube προσφέροντας δυνατότητες όπως η αυτοματοποιημένη δημιουργία χρονισμών υποτίτλων και η άμεση μετάφραση υποτίτλων από γλώσσα σε γλώσσα.

Επιπλέον, η εφαρμογή θα ενσωματώσει λειτουργίες διαμοιρασμού συστήματος, επιτρέποντας την εύκολη ανταλλαγή υποτίτλων και την συμμετοχή της κοινότητας στη διαδικασία δημιουργίας και επεξεργασίας υποτίτλων. Θα υποστηρίζει επίσης τη δυνατότητα bidding, επιτρέποντας στους χρήστες να προσφέρουν για την επιμέλεια και τη βελτίωση υποτίτλων, ενισχύοντας έτσι την προσβασιμότητα και τη συνεργασία μεταξύ των χρηστών.

## **Κεφάλαιο 2ο: Διαμοιρασμός και Συνεργασία στο Σύστημα Υποτίτλων**

### **2.1 Εισαγωγή**

Στη σημερινή ψηφιακή εποχή, η ανάγκη για πολυγλωσσικό οπτικοακουστικό περιεχόμενο είναι πιο έντονη από ποτέ, με τους υπότιτλους να διαδραματίζουν κρίσιμο ρόλο στη διάδοση πληροφοριών παγκοσμίως. Καθώς ο όγκος των βίντεο αυξάνεται συνεχώς, η δημιουργία και διαχείριση υποτίτλων γίνεται όλο και πιο πολύπλοκη.

Ένα από τα βασικά στοιχεία που ενισχύει τη διαδικασία αυτή είναι ο διαμοιρασμός και η συνεργασία μεταξύ των χρηστών. Ωστόσο, παρά τα σημαντικά πλεονεκτήματα που προσφέρει ο διαμοιρασμός υποτίτλων, προκύπτουν και αρκετές προκλήσεις που πρέπει να αντιμετωπιστούν για να διασφαλιστεί η ποιότητα και η συνέπεια των υποτίτλων.

Σε αυτό το κεφάλαιο, θα εξετάσουμε τον ορισμό και τη σημασία του διαμοιρασμού υποτίτλων, τις προκλήσεις που σχετίζονται με τη συνεργασία σε αυτόν τον τομέα, καθώς και τις τεχνολογικές λύσεις που έχουν αναπτυχθεί για την αποτελεσματική αντιμετώπιση αυτών των προβλημάτων.

#### **2.1.1 Ορισμός και Χρησιμότητα του Διαμοιρασμού**

Ο διαμοιρασμός, γενικά, αναφέρεται στη διαδικασία κατά την οποία πόροι, πληροφορίες ή δεδομένα γίνονται διαθέσιμα σε άλλους χρήστες ή συστήματα για την επίτευξη κοινού στόχου. Στο πλαίσιο της ψηφιακής εποχής, ο διαμοιρασμός αποτελεί κρίσιμο εργαλείο που διευκολύνει τη συνεργασία, την καινοτομία και την αποτελεσματική διαχείριση πληροφοριών. Η χρησιμότητα του διαμοιρασμού έγκειται στην ενίσχυση της παραγωγικότητας, την προώθηση της γνώσης και την εξοικονόμηση πόρων, καθώς οι χρήστες μπορούν να μοιράζονται εργαλεία και δεδομένα, αποφεύγοντας την επανάληψη της εργασίας και αξιοποιώντας τη συλλογική τεχνογνωσία.

### **2.2 Διαμοιρασμός**

#### **2.2.1 Διαμοιρασμός στον Τομέα των Υποτίτλων**

Στον τομέα των υποτίτλων, ο διαμοιρασμός αποκτά ιδιαίτερη σημασία, καθώς επιτρέπει σε ομάδες ή κοινότητες χρηστών να συνεργάζονται για τη δημιουργία, επεξεργασία και διανομή υποτίτλων. Μέσω του διαμοιρασμού υποτίτλων, οι δημιουργοί μπορούν να διανέμουν το έργο τους σε μια ευρύτερη κοινότητα, επιτρέποντας τη βελτίωση και προσαρμογή τους από άλλους χρήστες. Αυτό όχι μόνο αυξάνει την προσβασιμότητα και την ποιότητα του περιεχομένου, αλλά και επιταχύνει τη διαδικασία παραγωγής υποτίτλων, ιδιαίτερα σε πολυγλωσσικά περιβάλλοντα. Εργαλεία και πλατφόρμες που υποστηρίζουν το διαμοιρασμό υποτίτλων επιτρέπουν τη συνεργασία μεταξύ επαγγελματιών και ερασιτεχνών, συμβάλλοντας στη δημιουργία πιο ολοκληρωμένων και ακριβών υποτίτλων.

## 2.2.2 Προκλήσεις στον Διαμοιρασμό Υποτίτλων

Παρά τα πλεονεκτήματα, ο διαμοιρασμός υποτίτλων συνοδεύεται από αρκετές προκλήσεις. Η διατήρηση της συνέπειας και της ποιότητας των υποτίτλων μπορεί να είναι δύσκολη όταν πολλοί χρήστες εργάζονται ταυτόχρονα πάνω στο ίδιο έργο. Η διαχείριση διαφορετικών εκδοχών και η διασφάλιση ότι όλες οι αλλαγές ενσωματώνονται σωστά απαιτούν ισχυρά συστήματα ελέγχου εκδόσεων. Επιπλέον, ο συγχρονισμός μεταξύ των χρηστών, ιδιαίτερα όταν αυτοί βρίσκονται σε διαφορετικές γεωγραφικές περιοχές, αποτελεί σημαντικό εμπόδιο που πρέπει να αντιμετωπιστεί μέσω κατάλληλων τεχνολογικών λύσεων.

## 2.2.3 Λύσεις για Αποτελεσματικό Διαμοιρασμό και Συνεργασία

Για την αντιμετώπιση των προκλήσεων που σχετίζονται με τον διαμοιρασμό και τη συνεργασία στον υποτιτλισμό, έχουν αναπτυχθεί καινοτόμες λύσεις, που βασίζονται σε προηγμένες τεχνολογίες και συνεργατικά μοντέλα. Πλατφόρμες υποτιτλισμού που βασίζονται στην κοινότητα, όπως το Amara, επιτρέπουν σε χρήστες από όλο τον κόσμο να συμβάλλουν στη δημιουργία και τη βελτίωση υποτίτλων, ενισχύοντας το community subtitling. Αυτές οι πλατφόρμες συνδυάζουν την προσπάθεια της κοινότητας με επαγγελματικές υπηρεσίες, διασφαλίζοντας τόσο την ποιότητα όσο και την ευρεία γλωσσική κάλυψη.

## 2.3 Επίπεδα Δικαιωμάτων στον Διαμοιρασμό Υποτίτλων

Στον διαμοιρασμό υποτίτλων, η κατανομή ρόλων και δικαιωμάτων είναι κρίσιμη για τη διαχείριση και την ποιότητα του περιεχομένου. Τα επίπεδα δικαιωμάτων που μπορούν να αποδοθούν σε χρήστες περιλαμβάνουν:

- **Owner:** Ο ιδιοκτήτης έχει πλήρη έλεγχο στον υποτιτλισμό, με δυνατότητα δημιουργίας, επεξεργασίας, διαγραφής και διαμοιρασμού του περιεχομένου, καθώς και διαχείρισης των δικαιωμάτων άλλων χρηστών.
- **Editor:** Οι συντάκτες μπορούν να επεξεργάζονται τους υπότιτλους, να τροποποιούν το περιεχόμενο και τους χρονισμούς, και να κάνουν άμεσες αλλαγές που είναι ορατές σε όλους.
- **Commenter:** Οι σχολιαστές έχουν δικαίωμα να προσθέτουν σχόλια και προτάσεις χωρίς να επεμβαίνουν απευθείας στους υπότιτλους, διευκολύνοντας τη συνεργασία και την ανατροφοδότηση.
- **Viewer:** Οι θεατές μπορούν να παρακολουθούν τους υπότιτλους χωρίς δυνατότητα επεξεργασίας, κατάλληλοι για έγκριση ή απλή αναφορά.

Οι ρόλοι και τα επίπεδα δικαιωμάτων στον διαμοιρασμό υποτίτλων είναι παρόμοια με αυτά που χρησιμοποιούνται στο Google Docs [1]. Στο Google Docs, οι χρήστες μπορούν να έχουν ρόλους όπως "Owner", "Editor", "Commenter" και "Viewer", παρέχοντας τον ίδιο τύπο έλεγχου και συνεργασίας για έγγραφα, που είναι κρίσιμος και για την επεξεργασία υποτίτλων. Αυτή η προσέγγιση διευκολύνει τη διαχείριση της περιεχομένου και την επικοινωνία μεταξύ των χρηστών.

## **2.4 Επίλογος**

Στο κεφάλαιο αυτό, αναλύσαμε τις προκλήσεις και τις λύσεις που σχετίζονται με τον διαμοιρασμό και τη συνεργασία στον υποτιτλισμό. Εξετάσαμε τα διάφορα επίπεδα δικαιωμάτων (Owner, Editor, Commenter, Viewer) που διευκολύνουν την αποτελεσματική διαχείριση των υποτίτλων. Επίσης, αναδείξαμε την αξία των τεχνολογιών και εργαλείων συνεργασίας, καθώς και τη σημασία της συμμετοχής της κοινότητας μέσω πλατφορμών όπως το Amara, για τη βελτίωση της προσβασιμότητας και της ποιότητας του περιεχομένου.

## **Κεφάλαιο 3ο: Ανάλυση και Ανάπτυξη Κοινότητας Χρηστών Υποτίτλων**

### **3.1 Εισαγωγή**

Η ανάπτυξη και διατήρηση μιας ενεργής κοινότητας χρηστών υποτίτλων είναι κρίσιμη για την επιτυχία των πλατφορμών υποτιτλισμού. Σε αυτό το κεφάλαιο, θα αναλύσουμε τις στρατηγικές που χρησιμοποιούνται για την προσέλκυση και διατήρηση χρηστών, καθώς και τα χαρακτηριστικά που ενισχύουν τη συμμετοχή στην κοινότητα υποτίτλων.

### **3.2 Στρατηγικές για την Προσέλκυση και Διατήρηση Χρηστών**

#### **3.2.1 Εκπαίδευση και Υποστήριξη**

Οι πλατφόρμες παρέχουν εκπαιδευτικό υλικό και οδηγούς για νέους χρήστες, προκειμένου να τους βοηθήσουν να κατανοήσουν πώς να χρησιμοποιούν την πλατφόρμα και να συνεισφέρουν αποτελεσματικά. Η υποστήριξη μέσω forums και συχνών ερωτήσεων (FAQ) ενισχύει την εμπειρία των χρηστών.

#### **3.2.2 Κίνητρα και Αναγνώριση**

Οι πλατφόρμες συχνά προσφέρουν κίνητρα όπως πιστοποιήσεις, βραβεία ή αναγνώριση των καλύτερων συνεισφορών για να ενθαρρύνουν τη συμμετοχή. Αυτές οι ενέργειες δημιουργούν μια αίσθηση επιβράβευσης και κίνητρο για συνεχόμενη συμμετοχή.

#### **3.2.3 Εύχρηστα Εργαλεία**

Η παροχή εργαλείων που είναι εύχρηστα και αποτελεσματικά για τη δημιουργία και επεξεργασία υποτίτλων μειώνει την πολυπλοκότητα της διαδικασίας, καθιστώντας την πιο προσιτή σε νέους χρήστες.

### **3.3 Χαρακτηριστικά που Ενισχύουν τη Συμμετοχή**

#### **3.3.1 Διαφάνεια και Συνεργασία**

Εργαλεία που επιτρέπουν τη συνεργασία σε πραγματικό χρόνο και παρέχουν διαφάνεια στη διαδικασία επεξεργασίας ενισχύουν την αλληλεπίδραση μεταξύ των χρηστών και διευκολύνουν την επίλυση προβλημάτων.

#### **3.3.2 Κοινότητες και Φόρουμ**

Δημιουργία κοινοτήτων και φόρουμ όπου οι χρήστες μπορούν να συζητούν, να ανταλλάσσουν ιδέες και να ζητούν βοήθεια προάγει την αίσθηση του ανήκειν και την ενεργή συμμετοχή στην κοινότητα.

### **3.3.3 Ανατροφοδότηση και Βελτίωση**

Η συλλογή ανατροφοδότησης από τους χρήστες και η ενσωμάτωσή της στη διαδικασία βελτίωσης της πλατφόρμας βοηθά στη συνεχή αναβάθμιση των εργαλείων και των διαδικασιών, ενισχύοντας την ικανοποίηση των χρηστών.

### **3.4 Community Subtitling και Bidding**

Το community subtitling επιτρέπει σε χρήστες από διάφορα μέρη του κόσμου να συμβάλλουν στην παραγωγή και την επεξεργασία υποτίτλων. Οι συμμετοχές σε αυτό το είδος υποτιτλισμού είναι συχνά οργανωμένες μέσω bidding συστημάτων, όπου οι χρήστες μπορούν να "δικδικήσουν" ή να προτείνουν έργα που θέλουν να υποτιτλίσουν ή να επεξεργαστούν. Αυτή η διαδικασία επιτρέπει την ευέλικτη κατανομή εργασίας και την ενεργή συμμετοχή της κοινότητας [2].

### **3.5 Επίλογος**

Η ανάπτυξη μιας επιτυχημένης κοινότητας χρηστών υποτίτλων απαιτεί στρατηγικές προσέλκυσης, κίνητρα και εύχρηστα εργαλεία, καθώς και χαρακτηριστικά που προάγουν τη συμμετοχή και τη συνεργασία. Οι πλατφόρμες που υιοθετούν αυτές τις στρατηγικές και χαρακτηριστικά μπορούν να ενισχύσουν τη συμμετοχή των χρηστών και να διασφαλίσουν την ποιότητα και την αποτελεσματικότητα της δημιουργίας υποτίτλων.

## Κεφάλαιο 4ο: Τεχνολογίες Web Development / API που χρησιμοποιήθηκαν

Η ανάπτυξη της web εφαρμογής έγινε χρησιμοποιώντας ένα συνδυασμό τεχνολογιών από το σύγχρονο οικοσύστημα των web frameworks, το οποίο έχει γνωρίσει σημαντική ανάπτυξη τα τελευταία χρόνια. Ειδικότερα, για το Frontend της εφαρμογής επιλέχθηκε το Angular, ένα framework βασισμένο στη TypeScript, που αναπτύσσεται από την Google. Στο Backend χρησιμοποιήθηκε το Firebase, μια cloud πλατφόρμα επίσης από την Google, η οποία παρέχει μια πληθώρα υπηρεσιών για την ανάπτυξη ολοκληρωμένων Full-stack εφαρμογών. Επιπλέον, αξιοποιήθηκαν διάφορα διαθέσιμα API για τη χρήση υπαρχόντων δεδομένων και υπηρεσιών, όπως το YouTube, το Google και το OpenAI.

### 4.1 Angular

Η **Angular** είναι ένα δημοφιλές framework ανοιχτού κώδικα για την ανάπτυξη web εφαρμογών, το οποίο αναπτύχθηκε από την Google. Είναι ένα πλήρες περιβάλλον για τη δημιουργία εφαρμογών μονής σελίδας (SPA - Single Page Applications), οι οποίες προσφέρουν μια πιο ρευστή και διαδραστική εμπειρία χρήστη σε σύγκριση με τις παραδοσιακές εφαρμογές. Η Angular είναι χτισμένη πάνω σε μια ισχυρή αρχιτεκτονική και προσφέρει λειτουργίες που καλύπτουν σχεδόν όλες τις ανάγκες μιας μοντέρνας web εφαρμογής [3].

#### 4.1.1 TypeScript

**TypeScript** είναι μια γλώσσα προγραμματισμού που αναπτύχθηκε από τη Microsoft και είναι μια υπερσύνολο της JavaScript. Προσθέτει στατική τυποποίηση στον κώδικα JavaScript, κάτι που βοηθά στον εντοπισμό σφαλμάτων κατά τη διάρκεια της ανάπτυξης. Η TypeScript χρησιμοποιείται ευρέως σε μεγάλα έργα λόγω της δυνατότητας της να εντοπίζει σφάλματα νωρίτερα και να κάνει τον κώδικα πιο ευανάγνωστο και ευκολότερο στη συντήρηση [5].

#### 4.1.2 Αρχιτεκτονική και Βασικά Χαρακτηριστικά της Angular

Η αρχιτεκτονική της Angular βασίζεται σε μια δομή που διευκολύνει τη δημιουργία, συντήρηση και επέκταση των εφαρμογών με σαφή διαχωρισμό των ευθυνών.

##### Modules

Τα modules είναι τα βασικά δομικά στοιχεία στην Angular, που οργανώνουν το application σε αναγνωρίσιμες και επαναχρησιμοποιήσιμες μονάδες. Κάθε module περιλαμβάνει components, directives, pipes, services και άλλα resources που αφορούν μια συγκεκριμένη λειτουργικότητα. Το κύριο module ονομάζεται **AppModule**, και είναι το κεντρικό σημείο εκκίνησης της εφαρμογής.

##### Components

Τα components είναι υπεύθυνα για την εμφάνιση και τη λογική της διεπαφής χρήστη. Κάθε component έχει ένα συνδεδεμένο template (HTML) και μπορεί να περιέχει CSS για styling. Ένα component αποτελείται από:

- **Component Class:** Περιλαμβάνει την επιχειρησιακή λογική.
- **Template:** Περιγράφει τη δομή του UI.
- **Styles:** Περιλαμβάνει την εμφάνιση και το στυλ του component.

### Services

Οι υπηρεσίες στην Angular είναι κλάσεις που περιλαμβάνουν λογική που δεν σχετίζεται άμεσα με την εμφάνιση, όπως πρόσβαση σε δεδομένα από ένα API ή διαχείριση κατάστασης. Οι υπηρεσίες εισάγονται στα components μέσω Dependency Injection (DI).

### Dependency Injection

Το Dependency Injection είναι ένας θεμελιώδης μηχανισμός της Angular, που επιτρέπει την εύκολη εισαγωγή εξαρτήσεων, όπως services, σε classes ή components, χωρίς να χρειάζεται ο προγραμματιστής να τις δημιουργεί ή να τις διαχειρίζεται χειροκίνητα.

### Routing

Η δρομολόγηση (Routing) επιτρέπει τη διαχείριση των URLs και τη φόρτωση διαφορετικών components ή views ανάλογα με τη διαδρομή του URL. Το Angular Router είναι υπεύθυνο για αυτή τη λειτουργία, διασφαλίζοντας την πλοήγηση εντός της εφαρμογής χωρίς ανανέωση της σελίδας.

#### 4.1.3 Change Detection

Η ανίχνευση αλλαγών (Change Detection) είναι η διαδικασία με την οποία η Angular παρακολουθεί και ενημερώνει το DOM (Document Object Model) με τις αλλαγές στα δεδομένα της εφαρμογής. Όταν τα δεδομένα σε ένα component αλλάξουν, η Angular ελέγχει αν πρέπει να ενημερώσει το UI για να αντικατοπτρίζει αυτές τις αλλαγές.

#### Τύποι Ανίχνευσης Αλλαγών:

##### 1. Default Change Detection Strategy:

- Η προεπιλεγμένη στρατηγική παρακολουθεί κάθε αλλαγή και ενημερώνει το DOM άμεσα. Αυτό επιτυγχάνεται μέσα από τον μηχανισμό "zone.js", που καθιστά δυνατή την αναγνώριση όλων των γεγονότων που μπορεί να επηρεάσουν το state της εφαρμογής.

##### 2. OnPush Change Detection Strategy:

- Η OnPush στρατηγική απαιτεί λιγότερους πόρους καθώς ενημερώνει το DOM μόνο όταν οι τιμές των @Input() properties ενός component αλλάξουν ή όταν εκτελείται μια ρητή εντολή, όπως το detectChanges().

#### 4.1.4 Lifecycle Hooks

Τα components της Angular περνούν από μια σειρά από φάσεις κατά τη διάρκεια της ζωής τους, και σε κάθε φάση μπορούν να εκτελούν συγκεκριμένες ενέργειες μέσω των λεγόμενων lifecycle hooks. Αυτά τα hooks είναι μέθοδοι που μπορούν να υλοποιηθούν σε ένα component για να παρακολουθούν και να αντιδρούν σε αλλαγές στον κύκλο ζωής του component.

##### Κυριότεροι Lifecycle Hooks:

1. **ngOnInit():**
  - Καλείται μετά την αρχικοποίηση του component και είναι κατάλληλο για αρχικοποίηση δεδομένων.
2. **ngOnChanges():**
  - Ενεργοποιείται κάθε φορά που αλλάζουν οι δεσμευμένες ιδιότητες @Input().
3. **ngOnDestroy():**
  - Καλείται αμέσως πριν την καταστροφή του component, και χρησιμοποιείται για καθαρισμό (π.χ., αποδέσμευση πόρων, ακύρωση συνδρομών).
  - Παρακάτω είναι ένα παράδειγμα με τις μεθόδους ngOnInit() και ngOnDestroy():

#### 4.1.5 Data Binding

Το **Data Binding** είναι μία από τις πιο ισχυρές δυνατότητες της Angular, επιτρέποντας τη σύνδεση των δεδομένων της εφαρμογής με τα στοιχεία του User Interface (UI) με διάφορους τρόπους. Με το Data Binding, οι προγραμματιστές μπορούν να συγχρονίζουν τις αλλαγές στα δεδομένα της εφαρμογής με την προβολή στο UI και αντίστροφα, χωρίς να χρειάζεται να χειρίζονται άμεσα το DOM (Document Object Model). Αυτό καθιστά τη διαχείριση των δεδομένων πιο εύκολη και αποδοτική. Η Angular υποστηρίζει τέσσερις κύριους τύπους data binding:

##### Interpolation

Η διαπλοκή είναι η πιο βασική μορφή data binding και χρησιμοποιείται για την εισαγωγή τιμών από την κλάση ενός component σε ένα template. Αυτό επιτυγχάνεται με την περιγραφή των δεδομένων μέσα σε διπλά αγκύλες `{{ }}`.

```
@Component({
  selector: 'app-hello',
  template: `

# Hello, {{ name }}!</h1>` }) export class HelloComponent { name = 'Angular'; }


```

Στο παραπάνω παράδειγμα, η τιμή της μεταβλητής `name` μέσα στο `component` εισάγεται δυναμικά στο `template` και εμφανίζεται στο `UI`.

### Property Binding

Το `Property Binding` επιτρέπει την εισαγωγή τιμών από την κλάση ενός `component` σε ιδιότητες (`attributes`) `HTML` στοιχείων στο `template`. Χρησιμοποιείται το `[]` για να δεσμεύσει την ιδιότητα του `HTML` στοιχείου με μια τιμή από το `component`.

```
@Component({
  selector: 'app-image',
  template: `<img [src]="imageUrl">`
})
export class ImageComponent {
  imageUrl = 'https://example.com/image.png';
}
```

Εδώ, η ιδιότητα `src` του `HTML` στοιχείου `img` δεσμεύεται στην τιμή της μεταβλητής `imageUrl` του `component`.

### Event Binding

Το `Event Binding` επιτρέπει στο `component` να ανταποκρίνεται σε ενέργειες του χρήστη, όπως κλικ σε κουμπιά ή πληκτρολόγηση. Χρησιμοποιείται το `()` για να δεσμεύσει ένα συμβάν σε μια μέθοδο του `component`.

### Two-Way Data Binding

Η Δέσμευση Δεδομένων Διπλής Κατεύθυνσης επιτρέπει τη συγχρονισμένη ενημέρωση μεταξύ του `component` και του `template`, διασφαλίζοντας ότι οι αλλαγές στα δεδομένα του `component` αντικατοπτρίζονται στο `UI` και οι αλλαγές στο `UI` ενημερώνουν τα δεδομένα του `component`. Χρησιμοποιεί τη σύνταξη `[(ngModel)]`.

```
@Component({
  selector: 'app-input',
  template: `<input [(ngModel)]="name"><p>Your name is {{ name }}</p>`
})
export class InputComponent {
  name = "";
}
```

Εδώ, κάθε φορά που ο χρήστης πληκτρολογεί στο πεδίο εισαγωγής, η τιμή της μεταβλητής `name` ενημερώνεται και το περιεχόμενο του παραγράφου αλλάζει αντίστοιχα.

#### 4.1.5.1 Πλεονεκτήματα του Data Binding

- **Αυτοματοποιημένη Διαχείριση UI:** Οι αλλαγές στα δεδομένα του component αντικατοπτρίζονται αυτόματα στο UI, χωρίς χειροκίνητη παρέμβαση στον κώδικα DOM.
- **Ευκολία στην Ανάγνωση και Συντήρηση Κώδικα:** Το Data Binding καθιστά τον κώδικα πιο ευανάγνωστο και διαχειρίσιμο, καθώς οι developers δεν χρειάζεται να ανησυχούν για τη διαχείριση των ενημερώσεων του UI.
- **Αυξημένη Παραγωγικότητα:** Επιτρέπει στους προγραμματιστές να επικεντρώνονται περισσότερο στη λογική της εφαρμογής παρά στις λεπτομέρειες του UI.

## 4.2 Firebase

### 4.2.1 Εισαγωγή

Το **Firestore** είναι μια πλατφόρμα που παρέχει μια σειρά από εργαλεία και υπηρεσίες για την ανάπτυξη εφαρμογών, που κυρίως εστιάζουν σε εφαρμογές για κινητές συσκευές και διαδικτυακές εφαρμογές [9]. Η πλατφόρμα δημιουργήθηκε από την Firebase Inc. το 2011 και εξαγοράστηκε από την Google το 2014. Το Firestore είναι δημοφιλές λόγω της ευκολίας χρήσης του και της ολοκληρωμένης σειράς εργαλείων που προσφέρει, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στην ανάπτυξη των λειτουργιών της εφαρμογής χωρίς να χρειάζεται να διαχειριστούν τις υποδομές του backend.

### 4.2.2 Δυνατότητες

Το Firestore προσφέρει μια σειρά από δυνατότητες και υπηρεσίες που καθιστούν την ανάπτυξη εφαρμογών πιο εύκολη και γρήγορη. Οι βασικές δυνατότητες περιλαμβάνουν:

1. **Authentication (Πιστοποίηση Χρηστών):** Παρέχει έτοιμες λύσεις για την αυθεντικοποίηση χρηστών μέσω email/κωδικού πρόσβασης, Google, Facebook, Twitter, GitHub, καθώς και με custom providers.
2. **Realtime Database:** Ένα NoSQL database που αποθηκεύει δεδομένα σε μορφή JSON και τα συγχρονίζει σε πραγματικό χρόνο με τους συνδεδεμένους clients.
3. **Cloud Firestore:** Μια εξελιγμένη βάση δεδομένων NoSQL, που επιτρέπει την αποθήκευση, την αναζήτηση και τον συγχρονισμό δεδομένων σε πραγματικό χρόνο, προσφέροντας πιο πλούσια δυνατότητα ερωτημάτων σε σχέση με το Realtime Database.
4. **Cloud Storage:** Ένας τρόπος αποθήκευσης αρχείων (όπως εικόνες, βίντεο κ.λπ.) που επιτρέπει τον εύκολο διαμοιρασμό τους με τους χρήστες της εφαρμογής.
5. **Cloud Functions:** Επιτρέπει την εκτέλεση server-side κώδικα σε απάντηση σε γεγονότα που λαμβάνουν χώρα στην εφαρμογή, όπως η δημιουργία ενός νέου χρήστη.
6. **Analytics:** Παρέχει λεπτομερή ανάλυση χρήσης της εφαρμογής, με δυνατότητα προσαρμογής των συμβάντων που παρακολουθούνται.
7. **Crashlytics:** Ένα εργαλείο που βοηθά στον εντοπισμό και τη διαχείριση σφαλμάτων και crashes της εφαρμογής.

8. **Hosting:** Παρέχει ασφαλή φιλοξενία στατικών αρχείων της εφαρμογής, με ενσωματωμένα χαρακτηριστικά όπως HTTPS και Global CDN.

### 4.2.3 Cloud Firestore και Βάσεις Δεδομένων στο Firebase

Το **Cloud Firestore** είναι η επόμενη γενιά της βάσης δεδομένων της Firebase μετά το Realtime Database. Ενώ και οι δύο βάσεις δεδομένων είναι NoSQL και λειτουργούν σε πραγματικό χρόνο, το Firestore προσφέρει περισσότερη ευελιξία, καλύτερη υποστήριξη ερωτημάτων και βελτιωμένες επιδόσεις.

### 4.2.4 Κύρια Χαρακτηριστικά του Cloud Firestore:

- **Δομή Δεδομένων:** Το Firestore αποθηκεύει δεδομένα σε έγγραφα (documents), τα οποία οργανώνονται σε συλλογές (collections). Κάθε έγγραφο είναι ένα σύνολο από πεδία κλειδί-τιμή, όπως και σε μια JSON δομή, και μπορεί να περιέχει υπο-συλλογές.
- **Πραγματικός Χρόνος:** Το Firestore συγχρονίζει δεδομένα σε πραγματικό χρόνο με όλους τους συνδεδεμένους clients, κάτι που το καθιστά ιδανικό για εφαρμογές που απαιτούν ενημερώσεις σε πραγματικό χρόνο, όπως chat apps.
- **Πλούσια Ερωτήματα:** Σε αντίθεση με το Realtime Database, το Firestore υποστηρίζει σύνθετα ερωτήματα, όπως συνδυαστικές αναζητήσεις, ταξινόμηση και φίλτρα, καθιστώντας την διαχείριση δεδομένων πιο ευέλικτη και ισχυρή.
- **Αυτόματη Κλιμάκωση:** Το Firestore κλιμακώνεται αυτόματα για να χειριστεί μεγάλο όγκο δεδομένων και αυξημένη κίνηση, χωρίς να απαιτείται διαχείριση από τον προγραμματιστή.
- **Ασφάλεια και Διαχείριση Πρόσβασης:** Το Firestore παρέχει ευέλικτους κανόνες ασφαλείας για τον έλεγχο της πρόσβασης στα δεδομένα, επιτρέποντας ή αποκλείοντας την πρόσβαση σε χρήστες με βάση το authentication state.

### 4.2.5 Πλεονεκτήματα του Firestore

- **Πιο Ευέλικτη Δομή:** Η οργάνωση των δεδομένων σε συλλογές και υπο-συλλογές επιτρέπει μεγαλύτερη ευελιξία στον τρόπο που αποθηκεύονται τα δεδομένα.
- **Καλύτερη Υποστήριξη Ερωτημάτων:** Τα πιο σύνθετα και πλούσια ερωτήματα που υποστηρίζει το Firestore παρέχουν μεγαλύτερο έλεγχο και ευελιξία στην ανάκτηση δεδομένων.
- **Ασφάλεια:** Τα εύκολα προσαρμόσιμα πρότυπα ασφαλείας και κανόνων πρόσβασης επιτρέπουν τη λεπτομερή διαχείριση των δικαιωμάτων χρήστη.

### 4.2.6 Μειονεκτήματα του Firestore

- **Περιορισμοί Συναλλαγών:** Οι συναλλαγές και οι λειτουργίες παρτίδας (batch operations) στο Firestore έχουν ορισμένους περιορισμούς σε σύγκριση με τις παραδοσιακές SQL βάσεις δεδομένων.
- **Περιορισμένη Λογική Στον Client:** Παρά την ευελιξία του Firestore, η πολυπλοκότητα ορισμένων ερωτημάτων μπορεί να αναγκάσει τον προγραμματιστή να υλοποιήσει λογική στον client, κάτι που μπορεί να επιβραδύνει την απόδοση.

#### 4.2.7 Collections

**Collections** αποτελούν τη βασική μονάδα οργάνωσης δεδομένων στο Cloud Firestore. Τα δεδομένα στο Firestore αποθηκεύονται σε έγγραφα (documents), τα οποία ομαδοποιούνται σε συλλογές (collections). Κάθε συλλογή μπορεί να περιέχει ένα ή περισσότερα έγγραφα, και κάθε έγγραφο αποτελείται από ένα σύνολο από κλειδιά (keys) και τιμές (values), δηλαδή πεδία (fields). Επιπλέον, τα έγγραφα μπορούν να περιέχουν υπο-συλλογές (subcollections), επιτρέποντας την εύκολη δημιουργία πολύπλοκων και πολυεπίπεδων δομών δεδομένων.

Οι συλλογές είναι ζωτικής σημασίας για την οργάνωση και την κλιμάκωση των δεδομένων σε μια εφαρμογή. Η ιεραρχική δομή του Firestore, με συλλογές και υπο-συλλογές, διευκολύνει την κατηγοριοποίηση των δεδομένων με τρόπο που αντικατοπτρίζει τη λογική δομή της εφαρμογής. Αυτή η δομή επιτρέπει επίσης την αποδοτική αναζήτηση και φιλτράρισμα των δεδομένων, καθώς και την εύκολη διαχείριση των δεδομένων σε κλιμακούμενες εφαρμογές.

Για παράδειγμα, σε μια εφαρμογή διαχείρισης καταστημάτων, μπορείτε να έχετε μια συλλογή stores, η οποία περιέχει έγγραφα που αναπαριστούν καταστήματα. Κάθε έγγραφο καταστήματος μπορεί να περιέχει υπο-συλλογές όπως products (για τα προϊόντα του καταστήματος) και orders (για τις παραγγελίες).

#### 4.2.8 Where Queries

Τα **where queries** είναι μια ισχυρή δυνατότητα του Firestore που επιτρέπει τον φιλτράρισμα των εγγράφων μέσα σε μια συλλογή βάσει συγκεκριμένων κριτηρίων. Με τα where queries, μπορείτε να ανακτήσετε έγγραφα που πληρούν μία ή περισσότερες συνθήκες, όπως για παράδειγμα έγγραφα που έχουν μια συγκεκριμένη τιμή σε ένα πεδίο.

Η χρήση των where queries επιτρέπει την αποδοτική ανάκτηση μόνο των δεδομένων που είναι απαραίτητα για μια συγκεκριμένη διεργασία ή λειτουργία της εφαρμογής. Αυτά τα queries μπορούν να συνδυαστούν με άλλες λειτουργίες όπως η ταξινόμηση (orderBy), η περιορισμένη ανάκτηση (limit) και η χρήση φίλτρων (filters) για ακόμα πιο ακριβείς αναζητήσεις.

#### 4.2.9 Χρήση των Filter και Map σε Δεδομένα του Firestore

Το **filter** και το **map** είναι μερικές από τις πιο χρήσιμες συναρτήσεις στη JavaScript όταν θέλετε να χειριστείτε δεδομένα που έχετε ανακτήσει από το Firestore.

Η συνάρτηση filter επιτρέπει την δημιουργία ενός νέου πίνακα που περιλαμβάνει μόνο τα στοιχεία του αρχικού πίνακα που πληρούν μια συγκεκριμένη συνθήκη. Αυτό είναι ιδιαίτερα χρήσιμο όταν έχετε ανακτήσει πολλά δεδομένα από το Firestore και θέλετε να διατηρήσετε μόνο εκείνα που πληρούν ορισμένα κριτήρια.

Η συνάρτηση map χρησιμοποιείται για να δημιουργήσει έναν νέο πίνακα βασισμένο στα στοιχεία ενός άλλου πίνακα, εκτελώντας μια λειτουργία σε κάθε στοιχείο. Στο πλαίσιο του Firestore, μπορεί να χρησιμοποιηθεί για να διαμορφώσει τα δεδομένα πριν τα εμφανίσει στον χρήστη.

#### 4.2.10 Συμπέρασμα

Το Firebase αποτελεί μια ισχυρή και ολοκληρωμένη πλατφόρμα για την ανάπτυξη σύγχρονων εφαρμογών, ειδικά για εφαρμογές που απαιτούν πραγματικό χρόνο και mobile-first προσέγγιση. Το Cloud Firestore, με την εξελιγμένη NoSQL αρχιτεκτονική του, αποτελεί ένα από τα πιο ισχυρά εργαλεία της πλατφόρμας, προσφέροντας μεγάλη ευελιξία και απόδοση. Ωστόσο, πρέπει να λαμβάνονται υπόψη και οι περιορισμοί και τα μειονεκτήματα του, ιδιαίτερα όταν εξετάζεται η κλιμάκωση της εφαρμογής ή η πολυπλοκότητα της λογικής του backend.

### 4.3 EmailJS

Το **EmailJS** είναι μια υπηρεσία που επιτρέπει την αποστολή email απευθείας από τις εφαρμογές web και κινητών χωρίς την ανάγκη για διακομιστές (servers). Αντί να ρυθμίζετε έναν δικό σας διακομιστή email και να ανησυχείτε για τις περίπλοκες ρυθμίσεις SMTP (Simple Mail Transfer Protocol), το EmailJS προσφέρει έναν εύκολο τρόπο για την αποστολή email χρησιμοποιώντας μόνο JavaScript. Αυτό το καθιστά ιδιαίτερα χρήσιμο για προγραμματιστές που επιθυμούν να ενσωματώσουν λειτουργίες αποστολής email σε εφαρμογές τους χωρίς τον επιπλέον φόρτο διαχείρισης διακομιστών.

Το EmailJS επιτρέπει στις εφαρμογές να στέλνουν email χρησιμοποιώντας email templates που έχουν προδημιουργηθεί από τον χρήστη. Αυτή η διαδικασία δεν απαιτεί backend ή διακομιστή από την πλευρά του χρήστη, καθιστώντας την εξαιρετικά ελαφριά και εύκολη στην ενσωμάτωση. Το μόνο που απαιτείται είναι η χρήση της JavaScript για την ενεργοποίηση της αποστολής email. Το EmailJS λειτουργεί με τα περισσότερα εργαλεία front-end, όπως Angular, React, Vue.js, ή ακόμα και απλό HTML και JavaScript, κάνοντάς το ευέλικτο και προσαρμόσιμο σε διάφορα είδη έργων.

#### 4.3.1 Χρησιμότητα του EmailJS

Το EmailJS είναι ιδιαίτερα χρήσιμο σε περιπτώσεις όπου χρειάζεται η αποστολή email για ειδοποιήσεις, επιβεβαιώσεις, αναφορές ή οποιαδήποτε άλλη μορφή επικοινωνίας μέσα από την εφαρμογή σας, χωρίς να χρειάζεται να στήσετε έναν διακομιστή email. Για παράδειγμα, μια φόρμα επικοινωνίας σε μια ιστοσελίδα μπορεί να χρησιμοποιήσει το EmailJS για να στέλνει email με τις πληροφορίες που εισάγει ο χρήστης απευθείας στον ιδιοκτήτη της ιστοσελίδας [6].

#### 4.3.2 Δυνατότητες του EmailJS

Το EmailJS προσφέρει πληθώρα δυνατοτήτων, όπως:

- **Αποστολή Email Χωρίς Backend:** Το EmailJS εξαλείφει την ανάγκη για τη δημιουργία backend κώδικα για την αποστολή email. Όλα γίνονται μέσω του front-end της εφαρμογής.
- **Υποστήριξη Προσαρμόσιμων Templates:** Οι χρήστες μπορούν να δημιουργήσουν και να αποθηκεύσουν προσαρμόσιμα email templates, τα οποία μπορούν να συμπληρωθούν δυναμικά με δεδομένα από την εφαρμογή πριν αποσταλούν [7].
- **Ενσωμάτωση με Υπηρεσίες Email:** Το EmailJS λειτουργεί με πολλούς γνωστούς παρόχους email, όπως το Gmail, το Yahoo, και το Outlook, επιτρέποντας την ευέλικτη χρήση διαφόρων λογαριασμών email.
- **Υποστήριξη Αρχείων:** Η υπηρεσία επιτρέπει την προσθήκη συνημμένων αρχείων στα email, γεγονός που είναι εξαιρετικά χρήσιμο για την αποστολή αναφορών, τιμολογίων, και άλλων εγγράφων.
- **Εύκολη Διαχείριση και Ενσωμάτωση:** Η διαχείριση των email templates, των κλειδιών API, και των ρυθμίσεων είναι εύκολη μέσω ενός φιλικού περιβάλλοντος χρήστη.

### 4.3.3 Ευκολία Χρήσης

Η ευκολία χρήσης του EmailJS είναι ένα από τα πιο ισχυρά του σημεία. Απαιτεί ελάχιστη εγκατάσταση και ρύθμιση. Μπορείτε να αρχίσετε να στέλνετε email με μερικές γραμμές κώδικα JavaScript, καθιστώντας το ιδιαίτερα ελκυστικό για νέους προγραμματιστές ή για μικρά έργα που δεν απαιτούν πολύπλοκα συστήματα διαχείρισης email. Η διαχείριση και ενσωμάτωση των υπηρεσιών είναι τόσο απλή που μπορεί να ολοκληρωθεί σε λίγα λεπτά.

### 4.3.4 Συμπέρασμα

Το EmailJS αποτελεί μια εξαιρετική επιλογή για την αποστολή email μέσω εφαρμογών χωρίς την ανάγκη για backend υποδομή. Με τις ευέλικτες δυνατότητες και την ευκολία ενσωμάτωσης, προσφέρει έναν γρήγορο και αξιόπιστο τρόπο για την ενίσχυση της επικοινωνίας μέσω email σε κάθε είδους εφαρμογή.

## 4.4 Git

Το Git είναι ένα καταναμημένο σύστημα version control που επιτρέπει στους προγραμματιστές να παρακολουθούν τις αλλαγές στο codebase τους, να συνεργάζονται απρόσκοπτα και να διαχειρίζονται διαφορετικές εκδόσεις των έργων τους [8]. Δημιουργήθηκε από τον Linus Torvalds το 2005, και πλέον το Git έχει γίνει το de facto πρότυπο για τον version control στην ανάπτυξη λογισμικού λόγω της ευελιξίας, της ταχύτητας και της αποτελεσματικότητάς του. Επιτρέπει στους προγραμματιστές να εργάζονται σε έργα συλλογικά παρέχοντας έναν τρόπο συγχώνευσης αλλαγών από πολλούς συνεισφέροντες, παρακολούθησης του ιστορικού των τροποποιήσεων κώδικα και εύκολης επαναφοράς σε προηγούμενες καταστάσεις εάν είναι απαραίτητο.

Το GitHub το οποίο χρησιμοποιήθηκε για το versioning της εφαρμογής, είναι μια διαδικτυακή πλατφόρμα που βασίζεται στο Git και παρέχει πρόσθετες δυνατότητες συνεργασίας και hosting. Το GitHub, το οποίο ιδρύθηκε το 2008, επιτρέπει στους προγραμματιστές να αποθηκεύουν τα αποθετήρια Git τους στο cloud, διευκολύνοντας την συνεργασία μεταξύ των μελών μιας ομάδας ανεξάρτητα από τη φυσική τους τοποθεσία. Το GitHub περιλαμβάνει λειτουργίες όπως παρακολούθηση προβλημάτων, pull requests και wikis, βελτιώνοντας τη διαχείριση έργων και την επικοινωνία. Λειτουργεί επίσης ως κεντρικός κόμβος για open source repos, ενισχύοντας μια προσέγγιση με γνώμονα την κοινότητα στην ανάπτυξη λογισμικού[10].

Οι προγραμματιστές χρησιμοποιούν το Git τοπικά στους υπολογιστές τους για να διαχειριστούν τις αλλαγές και στη συνέχεια να προωθήσουν αυτές τις αλλαγές σε ένα απομακρυσμένο αποθετήριο που φιλοξενείται σε πλατφόρμες όπως το GitHub. Το GitHub έχει γίνει ένα κρίσιμο εργαλείο τόσο για έργα ανοιχτού κώδικα όσο και για ιδιωτικά έργα, παρέχοντας μια φιλική προς το χρήστη διεπαφή, παρακολούθηση των issues και του code collaboration. Ο συνδυασμός Git και GitHub έχει εξορθολογίσει σημαντικά τη διαδικασία ανάπτυξης λογισμικού, ενισχύοντας τη συνεργασία, τη διαφάνεια και την αποτελεσματικότητα στον κόσμο του προγραμματισμού.

### 4.4.1 Github Pages

Οι GitHub Pages είναι μια ισχυρή δυνατότητα που παρέχεται από το GitHub και επιτρέπει στους προγραμματιστές να φιλοξενούν και να δημοσιεύουν αβίαστα στατικές ιστοσελίδες απευθείας από τα αποθετήρια GitHub τους. Το GitHub Pages που κυκλοφόρησε το 2008 έχει γίνει μια δημοφιλής επιλογή για φιλοξενία προσωπικών ιστολογίων, projects και οποιουδήποτε άλλου στατικού περιεχομένου. Το feature αυτό χρησιμοποιήθηκε για το deployment της εφαρμογής στο public domain του GitHub. Παρακάτω είναι κάποια βασικά χαρακτηριστικά των GitHub Pages:

- Δωρεάν hosting: Το GitHub Pages προσφέρει δωρεάν φιλοξενία για στατικούς ιστότοπους, καθιστώντας το μια εξαιρετική επιλογή για προσωπικά project, και ιστότοπους μικρής κλίμακας. Οι χρήστες μπορούν να επωφεληθούν από αυτήν την υπηρεσία χωρίς να επιβαρυνθούν με επιπλέον κόστος.
- Ενσωμάτωση με το Git: Οι Σελίδες GitHub αξιοποιούν το υποκείμενο version control σύστημα Git, επιτρέποντας στους χρήστες να δημοσιεύουν τους ιστότοπούς τους κάνοντας “push” απλώς τον κώδικά τους σε ένα dedicated branch, που συνήθως ονομάζεται gh-pages ή main.
- Custom domains: Οι χρήστες μπορούν να συσχετίσουν ένα custom domain με τον ιστότοπο του GitHub Pages, παρέχοντας μια επαγγελματική και επώνυμη διεύθυνση ιστού. Αυτή η δυνατότητα επιτρέπει στους προγραμματιστές να χρησιμοποιούν τα δικά τους domain name αντί να βασίζονται στα προεπιλεγμένα subdomain του GitHub.
- Automatic Builds: Το GitHub Pages δημιουργεί αυτόματα και ενημερώνει τον δημοσιευμένο ιστότοπο κάθε φορά που οι αλλαγές γίνεται push στο σχετικό repository. Αυτό διασφαλίζει ότι η πιο πρόσφατη έκδοση του ιστότοπου είναι πάντα διαθέσιμη στους επισκέπτες χωρίς να χρειάζεται manual παρέμβαση από τον maintainer του project.

Το GitHub Pages απλοποιεί τη διαδικασία φιλοξενίας στατικών ιστοτόπων, προσφέροντας μια λύση χωρίς προβλήματα με την ενσωμάτωσή του στο Git και την υποστήριξη προσαρμοσμένων τομέων. Είτε είστε προγραμματιστής, σχεδιαστής ή δημιουργός περιεχομένου, οι Σελίδες GitHub παρέχουν έναν απλό και αποτελεσματικό τρόπο για να μοιραστείτε την εργασία σας με τον κόσμο.

## Κεφάλαιο 5ο: Αναλυτική Περιγραφή της Εφαρμογής

Στο κεφάλαιο αυτό η παρουσίαση της εφαρμογής θα επικεντρωθεί στην ανάδειξη δύο βασικών λειτουργιών: τον διαμοιρασμό υποτίτλων και το σύστημα προσφορών (bidding) στο πλαίσιο της κοινότητας. Αρχικά, θα εξετάσουμε τον τρόπο με τον οποίο οι χρήστες μπορούν να συνεργάζονται για τη δημιουργία και τη διαχείριση υποτίτλων, αναδεικνύοντας τα διαφορετικά επίπεδα δικαιωμάτων και τη συμβολή της κοινότητας. Στη συνέχεια, θα παρουσιάσουμε τη διαδικασία του bidding, όπου τα μέλη της κοινότητας υποβάλλουν προσφορές για να αναλάβουν συγκεκριμένα έργα υποτιτλισμού, ενισχύοντας τη συμμετοχή και την ποιότητα του περιεχομένου.

Η υλοποίηση του συστήματος διαμοιρασμού υποτίτλων βασίστηκε στη δομή και τη λειτουργικότητα που συναντάται στα Google Docs, όπου οι χρήστες μπορούν να έχουν διαφορετικά επίπεδα πρόσβασης και δικαιωμάτων. Συγκεκριμένα, οι ρόλοι όπως ο "Ιδιοκτήτης" (Owner), ο "Συντάκτης" (Editor), ο "Σχολιαστής" (Commenter), και ο "Θεατής" (Viewer) μεταφέρθηκαν στον σχεδιασμό του συστήματος υποτιτλισμού, επιτρέποντας μια ομαλή και αποτελεσματική συνεργασία μεταξύ των χρηστών. Αυτό το μοντέλο διαχείρισης δικαιωμάτων εξασφαλίζει ότι οι υπότιτλοι μπορούν να δημιουργούνται, να τροποποιούνται και να κοινοποιούνται με ακρίβεια και ασφάλεια, διασφαλίζοντας την ποιότητα του περιεχομένου.

### 5.1 Dashboard

Έχοντας κάνει successful login στην εφαρμογή, ο χρήστης αποκτά πρόσβαση στο dashboard του

Σε αυτό το section υπάρχει η δυνατότητα να αναζητήσει ο χρήστης κάποιο βίντεο με βάση τον τίτλο του. Αυτή η λειτουργία επιτρέπει την ταχύτερη και πιο αποδοτική εύρεση των βίντεο, ιδιαίτερα σε περιπτώσεις όπου η συλλογή περιέχει μεγάλο αριθμό αποθηκευμένων αρχείων.

Η αναζήτηση πραγματοποιείται μέσω μιας απλής και φιλικής προς τον χρήστη γραμμής αναζήτησης, όπου μπορεί να πληκτρολογηθεί είτε ο πλήρης τίτλος του βίντεο είτε μέρος του. Το σύστημα στη συνέχεια προτείνει αποτελέσματα που ταιριάζουν με την αναζήτηση, διευκολύνοντας την επιλογή του επιθυμητού βίντεο. Με αυτόν τον τρόπο, ο χρήστης μπορεί να εντοπίσει γρήγορα το περιεχόμενο που χρειάζεται, εξοικονομώντας χρόνο και αποφεύγοντας την περιήγηση σε μεγάλες λίστες αρχείων.

## Κεφάλαιο 5

The screenshot displays a user's dashboard with three main sections:

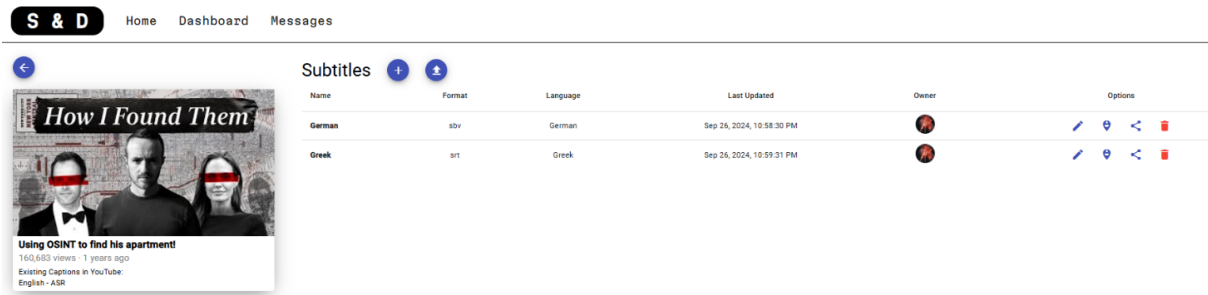
- My Videos (7):** A row of seven video cards. From left to right: 'Angular in 100 Seconds' (800,437 views, 4 years ago), 'JavaScript Best Practices to Write Better & Efficient...' (728 views, 1 year ago), 'Hacking Websites with SQL Injection - Computerphile...' (2,435,313 views, 11 years ago), 'I started a blog... in 2024 (why you should too)' (189,632 views, 26 days ago), '100+ JavaScript Concepts you Need to Know' (2,382,832 views, 2 years ago), 'Mindset of Successful Programmers' (1,178,200 views, 1 year ago), and 'Firebase in 100 Seconds' (748,648 views, 2 years ago).
- My Shared Videos (3):** A row of three video cards: 'Firebase in 100 Seconds', 'Angular in 100 Seconds', and 'JavaScript Best Practices to Write Better & Efficient...'. Each card includes a 'Request for BID' icon.
- Request for BID: someone needs your help!** A section with a search bar and a row of five video cards: 'JavaScript Best Practices to Write Better & Efficient...', 'Angular in 100 Seconds', 'I started a blog... in 2024 (why you should too)', 'Firebase in 100 Seconds', and 'Mindset of Successful Programmers'. Each card shows detailed metadata like creator name, view count, language, and creation date.

Σχήμα 5.1 Dashboard

Εκτός από τα βίντεο του χρήστη, υπάρχει το section My Shared Videos, στο οποίο εμφανίζονται τα βίντεο στα οποία έχει κάποιο δικαίωμα ο χρήστης και ένα section που ονομάζεται Request for BID, στο οποίο εμφανίζει όλα τα βίντεο τα οποία κάποιος χρήστης της εφαρμογής έχει ζητήσει βοήθεια από το community για τον υποτιτλισμό ενός δικού του βίντεο. Σε αυτά τα βίντεο ο current χρήστης της εφαρμογής μπορεί να βοηθήσει κάποιον άλλον user, βοηθώντας τον να δημιουργήσει υπότιτλους για κάποια συγκεκριμένη γλώσσα. Αυτή η διαδικασία δημιουργεί την αίσθηση του community μεταξύ των χρηστών.

### 5.2 My Videos

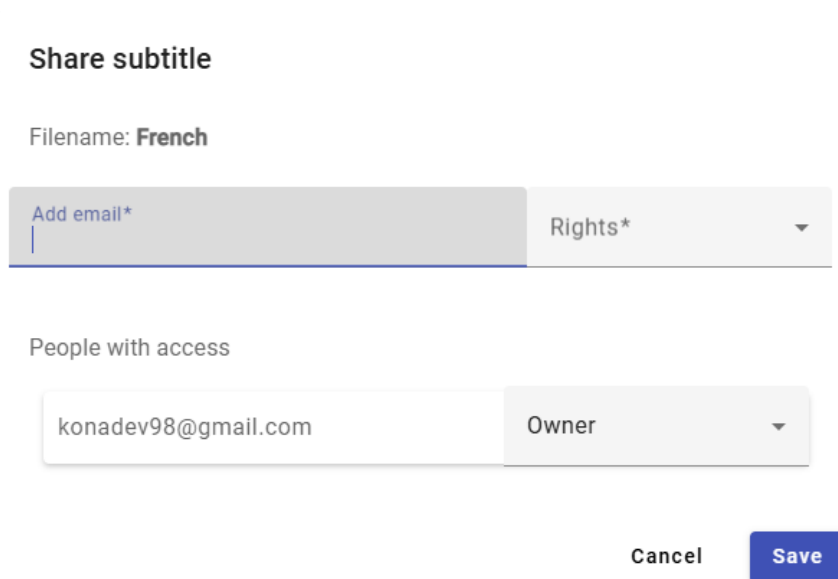
Κάνοντας click σε ένα video card από το My Videos view, η εφαρμογή θα κάνει τον χρήστη navigate στα details αυτού του video. Εκεί ο χρήστης έχει τη δυνατότητα να προσθέσει νέους υπότιτλους και να δει αναλυτικά τους υπότιτλους που έχει ήδη δημιουργήσει για το συγκεκριμένο βίντεο. Για κάθε υπότιτλο εμφανίζεται το μοναδικό όνομά του, το format του αρχείου (όπως sbn ή srt), η γλώσσα στην οποία ανήκει, ο ιδιοκτήτης του και η ημερομηνία/ώρα της τελευταίας ενημέρωσης. Σε κάθε υπότιτλο παρέχεται η επιλογή “request community help”, το οποίο κάνει flag την συγκεκριμένη γλώσσα για το ίδιο βίντεο και την προσθέτει στο Request for BID section του Dashboard. Επιπλέον, υπάρχει και η δυνατότητα κοινοποίησης των υπότιτλων σε άλλους χρήστες, διευκολύνοντας τη συνεργασία και τη διαχείριση.



Σχήμα 5.2 My Videos – Details

### 5.2.1 Κοινοποίηση υποτίτλου

Επιλέγοντας την κοινοποίηση εμφανίζεται ένα παράθυρο στο οποίο φαίνονται οι χρήστες που έχουν δικαιώματα για τον υπότιτλο αυτό. Ο κάθε υπότιτλος έχει έναν και μοναδικό ιδιοκτήτη, που παίρνει το δικαίωμα “Owner” όταν ο υπότιτλος δημιουργείται.



Σχήμα 5.3 Initial share window

### Share subtitle

Filename: **French**

Add email*	Rights*
People with access	<ul style="list-style-type: none"> <li>Editor</li> <li>Viewer</li> <li>Commenter</li> </ul>
konadev98@gmail.com	
<span>Cancel</span> <span>Save</span>	

Σχήμα 5.4 User rights

Ο ιδιοκτήτης του υποτίτλου μπορεί να κοινοποιήσει στους χρήστες που επιθυμεί τον υπότιτλο προσθέτοντας το email τους και το δικαίωμα που θα τους αντιστοιχεί. Τα δικαιώματα που επιτρέπονται είναι “Editor”, “Viewer” και “Commenter”. Καθώς συμπληρώνονται τα απαραίτητα πεδία δίνετε η δυνατότητα να σταλεί ειδοποίηση στο email που συμπληρώθηκε. Η διαδικασία γίνεται με τη βοήθεια του EmailJS. Έχουν δημιουργηθεί δύο templates για αποστολή email στον χρήστη, ένα όταν παίρνει κάποιο δικαίωμα σε έναν υπότιτλο και ένα άλλο όταν ο ιδιοκτήτης ενός υποτίτλου στέλνει πρόσκληση για μεταβίβαση την ιδιοκτησίας του υποτίτλου.

### Share subtitle

Filename: **Greek**

Add email*	Rights*
konakara74@gmail.com	Viewer

Send notification email

People with access

konadev98@gmail.com	Owner
---------------------	-------

Cancel Save

Σχήμα 5.5 Send notification email

**Code snippet:**

```
<h3 *ngIf="email.value && right.value"><input type="checkbox"
[(ngModel)]="sendNotificationValue" (ngModelChange)="sendNotification =
sendNotificationValue ? true : false">Send notification email</h3>
```

Μόνο όταν συμπληρωθούν το email και το right στα πεδία εμφανίζεται το checkbox με το Send notification email. Αυτό συμβαίνει καθώς υπάρχει το \*ngIf.

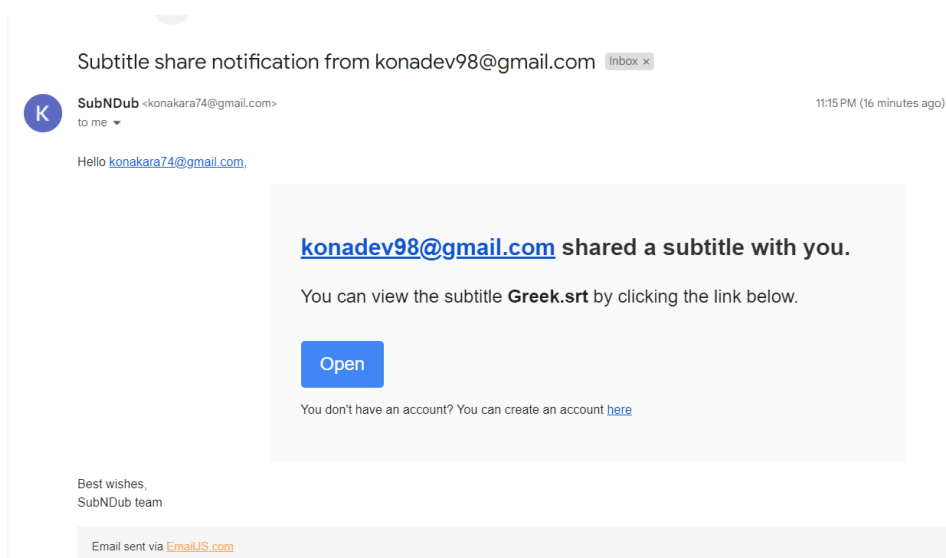
Το \*ngIf είναι ένα δομικό directive του Angular, που χρησιμοποιείται για να εμφανίζει ή να αποκρύπτει ένα HTML στοιχείο με βάση μία συνθήκη.

Το ngModel είναι ένα directive του Angular που χρησιμοποιείται για δεσμό δεδομένων (data binding) ανάμεσα στο HTML και στο TypeScript component. Συγκεκριμένα, το [(ngModel)] υποστηρίζει δεσμό δύο κατευθύνσεων (two-way data binding), πράγμα που σημαίνει ότι οποιαδήποτε αλλαγή στο πεδίο sendNotificationValue θα ενημερώνει το component και αντίστροφα. Έτσι, όταν ο χρήστης κάνει κλικ στο checkbox, η τιμή sendNotificationValue θα ενημερωθεί αυτόματα στο component.

Το ngModelChange είναι ένα event του Angular που ενεργοποιείται κάθε φορά που αλλάζει η τιμή του ngModel

**Email Notification**

Όταν ο ιδιοκτήτης του υπότιτλου επιλέξει το Send notification email, ο χρήστης λαμβάνει ένα email σαν ειδοποίηση μέσω του EmailJS. Στο email υπάρχει ένα link, το οποίο πατώντας το ο χρήστης ανακατευθύνεται στο dashboard του application. Το email έχει την παρακάτω μορφή:



Σχήμα 5.6 Email notification (EmailJS)

Τα δικαιώματα των χρηστών μπορούν να αλλάξουν ανά πάσα στιγμή. Είναι δυνατή και η οριστική αφαίρεση δικαιώματος κάποιου χρήστη από τον υπότιτλο, όπως φαίνεται και στο σχήμα[5.7] με

διαγραμμένο το κείμενο του email. Πατώντας το κουμπί Save αποθηκεύονται όλες οι αλλαγές για τον υπότιτλο, ενημερώνοντας τον πίνακα των δικαιωμάτων του υπότιτλου.

**Share subtitle**

Filename: **Danish**

Add email*	Rights*
konadev98@gmail.com	Owner
konkar.dev98@gmail.com	Remove right

People with access

Cancel Save

Σχήμα 5.7 Remove right

Παρακάτω είναι μια επεξήγηση ενός μέρους του details-view component, το οποίο διαχειρίζεται την λειτουργία της κοινοποίησης ενός υποτίτλου:

Ο χρήστης κάνοντας click στο εικονίδιο της κοινοποίησης ενός υποτίτλου, του ανοίγει ένα dialog prompt για να εισάγει το email και το right ενός χρήστη:

```

this.dialog.open(ShareSubtitleDialogComponent,{width:'600px', id: 'shared-
dialog',data: {filename, usersRights, videoId:this.videoId, ISOcode, language,
owner_text, format, videoTitle, subtitleId,
userEmail:this.user$.value.email}}).afterClosed().pipe(take(1)).subscribe(dial
og => {
    if (dialog === null || dialog === undefined )){
        this.dialog.closeAll();
    }else if(dialog){
        if (dialog.email && dialog.right) {
            this.detailsViewService.shareSubtitle(this.videoId, ISOcode,
language, this.user$.value.uid, filename, format, dialog.email, dialog.right,
subtitleId, dialog.sendNotification);
        }else if((dialog.email == "" && dialog.right) || (dialog.email
&& dialog.right == undefined)){
            this.snackbar.open('Both email and right have to be filled',
'DISMISS', {duration:3000});
        }else {

```

```

        this.detailsViewService.updateSharedSubtitleRights(this.videoId,
        ISOcode, language, this.user$.value.uid, filename, format, usersRights,
        subtitleId);
    }
}

```

Μόλις ο χρήστης συμπληρώσει το email και το αντίστοιχο right και κάνοντας click στο Save button, τρέχει το service function shareSubtitle(), το οποίο ξεκινάει την διαδικασία προσθήκης δικαιώματος ενός χρήστη. Υπάρχει όμως και η περίπτωση που ο ιδιοκτήτης του υπότιτλου να θέλει να αλλάξει το δικαίωμα κάποιου υπάρχοντος χρήστη ή και να του αφαιρέσει την πρόσβαση από τον υπότιτλο. Στην περίπτωση αυτή με το Save button, τρέχει το service function updateSharedSubtitleRights(), το οποίο κάνει ενημέρωση των δικαιωμάτων των χρηστών.

Με την shareSubtitle() καλείται η addUserRightOnSub() function. Η addUserRightOnSub() προσθέτει νέα δικαιώματα σε χρήστες για την πρόσβαση σε υπότιτλους βίντεο, ενημερώνει τις αντίστοιχες καταχωρήσεις στη βάση δεδομένων, και αποστέλλει προαιρετικά ειδοποιήσεις.

```

addUserRightOnSub(videoId: string, ISOcode: string, language: string, userId:
string, name: string, format: string, email: string, right: string,
subtitleId: any, sendNotification: boolean): void {
    const sharedVideoRef: AngularFireStoreCollection =
this.firestore.collection('sharedVideos');
    const id = this.firestore.createId();

    const data = {
        id: id,
        lastUpdated: Date.now(),
        fileName: name,
        fullFilename: `${name}.${format}`,
        format: format,
        language: language,
        iso: ISOcode,
        videoId: videoId,
        usersRights: [],
        requestOwnerEmail: ""
    };

    const subtitleRef =
this.firestore.collection('users').doc(userId).collection('videos').doc(video
Id)
        .collection('subtitleLanguages').doc(ISOcode).collection('subtitles').doc(
name);

    subtitleRef.get().toPromise().then((docSnapshot) => {

        const currentRights = docSnapshot.exists ?
docSnapshot.data().usersRights || [] : [];

```

```

    const existingRight = currentRights.find((right) => right.userEmail ===
email);
    const subtitleSharedId = docSnapshot.data().subtitleSharedId;
    const ownerEmail = currentRights.find((right) => right.right ===
"Owner");
    const isUsed = docSnapshot.data().isUsed;

    if (!existingRight) {
      currentRights.push({
        right: right,
        userEmail: email
      });
    }
    if(docSnapshot.exists) {
      subtitleRef.update({usersRights: currentRights});
      if (isUsed == undefined || isUsed==null)
        docSnapshot.ref.update({ isUsed: false });
      data.usersRights = currentRights;
      var sharedQuery;

      if(subtitleId == undefined){
        if(subtitleSharedId != undefined){
          subtitleId = subtitleSharedId;
        } else {
          subtitleId = "";
        }
      }

      sharedQuery = this.firestore.collection(`sharedVideos`, ref =>
ref.where('videoId', '==', videoId).where('iso', '==',
ISOcode).where('language', '==', language)
        .where('fileName', '==', name).where("id","==", subtitleId));
      sharedQuery.get()
        .subscribe((querySnapshot) => {

          if (!querySnapshot.empty) {

            const sharedVideoRef = querySnapshot.docs[0].ref;
            sharedVideoRef.update({
              usersRights: currentRights
            }).then(() => {
              this.notifier.showNotification("User rights have been
added.", "OK");
            }).catch((error) => {
              this.notifier.showNotification("Error updating user rights:
" + error.message, "DIMISS");
            });
          } else {
            sharedVideoRef.add(data);
            subtitleRef.update({subtitleSharedId: data.id});
          }
        }

```

```

        if (isUsed !== undefined)
            subtitleRef.update({isUsed: isUsed});
        this.notifier.showNotification("User rights have been
added.", "OK");
    }
});
if(sendNotification === true)
    this.emailService.sendShareSubEmail(ownerEmail.userEmail, email,
name+"."+format, videoId);
}
} else {
    this.notifier.showNotification("User already exist with a
right.", "DIMISS");
}
});
}
}

```

Αναλυτικά το τι πραγματοποιεί ο παραπάνω κώδικας:

- Δημιουργεί αναφορά στη συλλογή sharedVideos και έναν μοναδικό id για το νέο δικαίωμα.
- Ορίζει το αντικείμενο data με πληροφορίες του υπότιτλου, όπως το όνομα αρχείου, τη γλώσσα και τα δικαιώματα χρήστη.
- Εντοπίζει την αναφορά του συγκεκριμένου υπότιτλου στη βάση δεδομένων Firestore.
- Ελέγχει αν το email του χρήστη έχει ήδη καταχωρημένο δικαίωμα:
  - Αν δεν υπάρχει, προσθέτει το νέο δικαίωμα και ενημερώνει τη βάση δεδομένων.
- Αν ο υπότιτλος δεν έχει subtitleId, ορίζει καινούργιο ή χρησιμοποιεί το υπάρχον subtitleSharedId.
- Αν υπάρχει καταχώρηση για τον υπότιτλο στη συλλογή sharedVideos, ενημερώνει το πεδίο usersRights με τα νέα δικαιώματα: αν όχι, προσθέτει νέα καταχώρηση και αποθηκεύει το subtitleSharedId.
- Αν το sendNotification είναι true, αποστέλλει ειδοποίηση στον ιδιοκτήτη του υπότιτλου μέσω email.
- Εμφανίζει ειδοποίηση επιτυχίας ή αποτυχίας, ανάλογα με το αποτέλεσμα της ενημέρωσης στη βάση δεδομένων.

Στην άλλη περίπτωση που τρέχει η updateSharedSubtitleRights έχουμε 2 περιπτώσεις:

```

updateSharedSubtitleRights(videoId: string, ISOcode: string, language: string,
userId: string, name: string, format: string, usersrights: string[],
subtitleId: any): void {

    const removeUserRights = usersrights.some(user => user['right'] ===
"Remove right");
    if (removeUserRights){
        this.removeUserRightFromSub(videoId, ISOcode, language, name, usersrights,
subtitleId);
    }else {
        this.updateSharedVideosUserRights(videoId, ISOcode, language, name,
usersrights, subtitleId);
    }
}
}

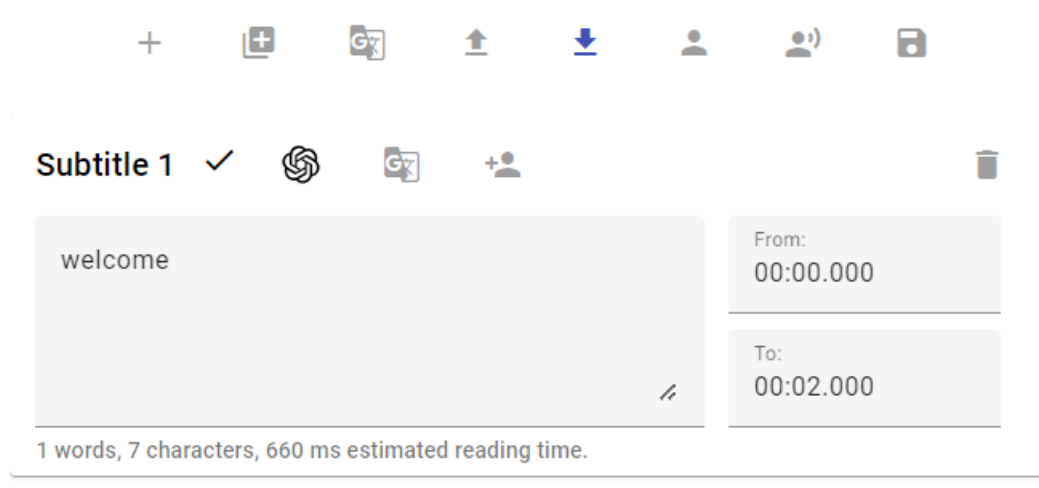
```

Χρησιμοποιώντας τη μέθοδο `some` μπορούμε να ελέγξουμε αν έχει επιλεχτεί έστω μία φορά το “Remove right” από τα δικαιώματα των χρηστών, οπότε τότε τρέχει το function `removeUserRightFromSub()`, στο οποίο γίνεται η οριστική αφαίρεση των δικαιωμάτων των χρηστών, αλλιώς το `updateSharedVideoUserRights()` function, στο οποίο ενημερώνονται κατάλληλα τα δικαιώματα των χρηστών. Έτσι, η μέθοδος λειτουργεί για τη διαχείριση και την ενημέρωση των δικαιωμάτων των χρηστών πάνω σε συγκεκριμένους υπότιτλους, ανάλογα με το αν πρέπει να αφαιρεθούν ή να ενημερωθούν δικαιώματα.

### 5.2.2 Δικαιώματα χρήστη

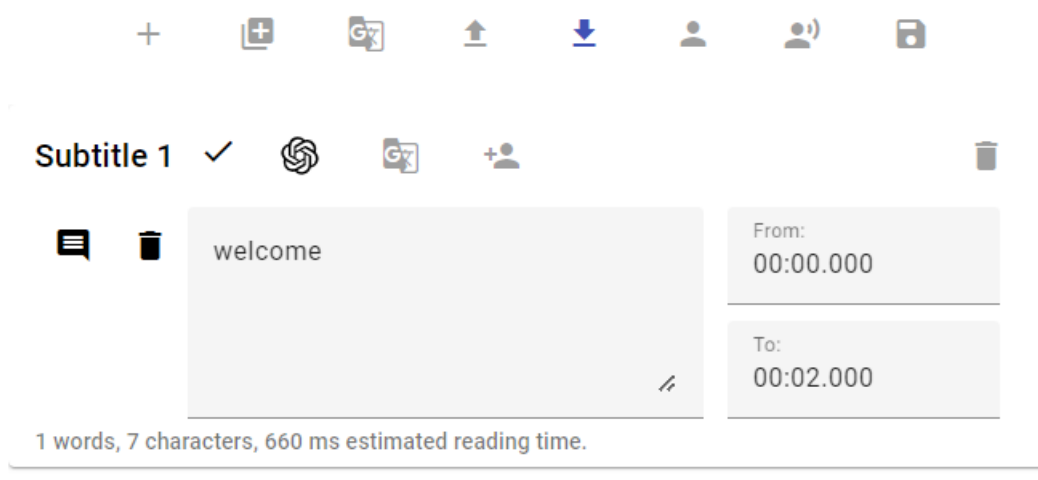
Ο χρήστης ανάλογα με το δικαίωμα (`right`) που έχει στον υπότιτλο έχει πρόσβαση και στις διάφορες λειτουργίες του που υπάρχουν στο `details view`.

Έχοντας το δικαίωμα “Viewer” ο χρήστης έχει πρόσβαση σε περιορισμένες λειτουργίες του υποτίτλου. Συγκεκριμένα μπορεί να κάνει `download` ένα υπότιτλο.



Σχήμα 5.8 View-only mode

Στη περίπτωση που ο χρήστης είναι “Commenter” έχει τις ίδιες λειτουργίες με τον “Viewer” με την διαφορά ότι μπορεί να γράψει κάποιο σχόλιο στο `dialog box` του υποτίτλου και να το διαγράψει επίσης.



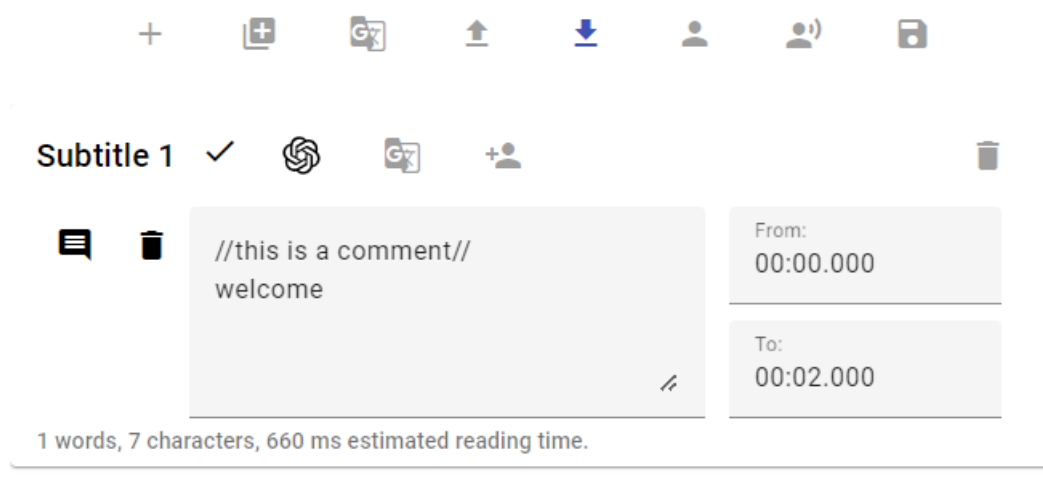
Σχήμα 5.9 Comment mode

Επιλέγοντας το εικονίδιο με το σχόλιο, στην αριστερή πλευρά του dialog box του υποτίτλου, ανοίγει ένα απλό dialog prompt για να συμπληρώσει ο χρήστης το σχόλιο του.



Σχήμα 5.10 Comment box

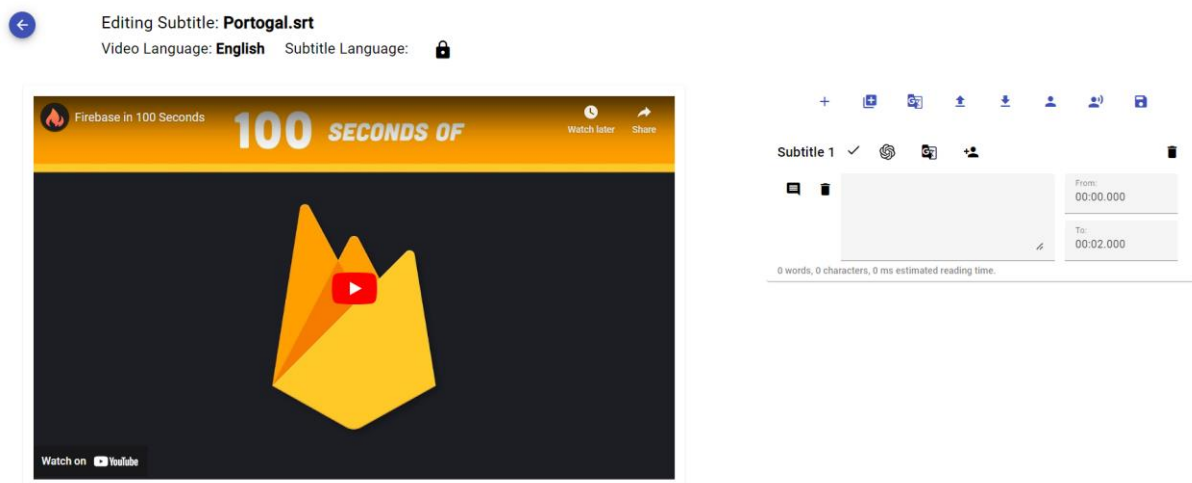
Μόλις ολοκληρώσει το σχόλιο του ο χρήστης και πατήσει Add comment, το κείμενο το οποίο συμπλήρωσε τοποθετείται στην αρχή του συγκεκριμένου dialog box του υποτίτλου με τον χαρακτήρα “/” (2 καθέτους) στην αρχή και στο τέλος της λέξης/πρότασης του.



Σχήμα 5.11 Comment text on the subtitle box

Οι χαρακτήρες αυτοί προστίθενται για να διαφοροποιείται το σχόλιο από το υπόλοιπο κείμενο του υπότιτλου.

Η τελευταία περίπτωση είναι ο χρήστης να έχει το δικαίωμα του “Editor”. Εδώ ο χρήστης έχει πρόσβαση σε όλες τις λειτουργίες που υπάρχουν για τους υπότιτλους.



Σχήμα 5.12 Editor view

Μερικές από τις λειτουργίες που υπάρχουν και μπορεί να χρησιμοποιήσει είναι οι εξής:

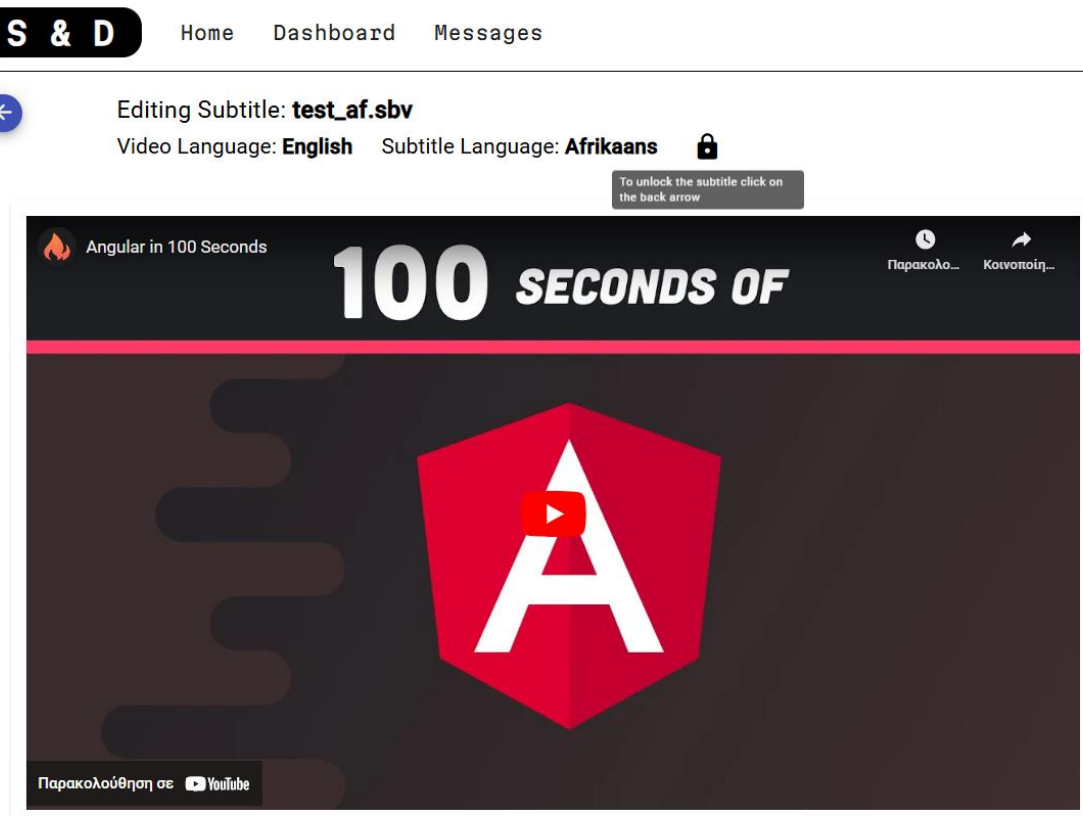
- Προσθήκη κειμένου στο κάθε subtitle box
- Προσθήκη κάποιου σχολίου στο κάθε subtitle box
- Επιλογή ChatGPT Helper, δίνοντας την επιλογή για αλλαγή τόνου/ύφους του κειμένου
- Μετάφραση του υπότιτλου στην target γλώσσα ή σε άλλη γλώσσα που επιθυμεί ο χρήστης
- Ανάθεση προσώπου στον υπότιτλο

Επιπλέον, μπορεί ο χρήστης να διαχειριστεί χαρακτήρες και φωνές ηθοποιών, στους οποίους θα αναθέσει τους υπότιτλους.

Είναι δυνατή και η παραγωγή φωνής/ομιλίας για τους υπότιτλους.

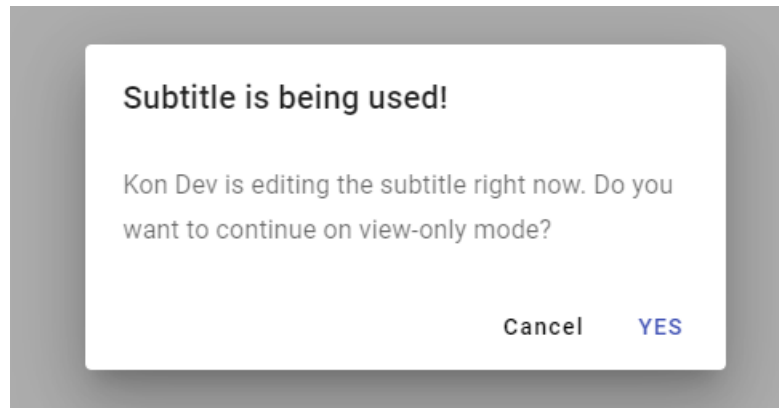
### 5.2.2.1 Κλείδωμα υπότιτλου

Όταν ένας χρήστης κάνει edit έναν υπότιτλο εμφανίζεται η εικόνα μιας κλειδαριάς πάνω από το βίντεο, η οποία υποδηλώνει ότι ο υπότιτλος έχει κλειδώσει και δεν μπορεί να τον επεξεργαστεί κάποιος άλλος χρήστης.



Σχήμα 5.13 Κλείδωμα υπότιτλου

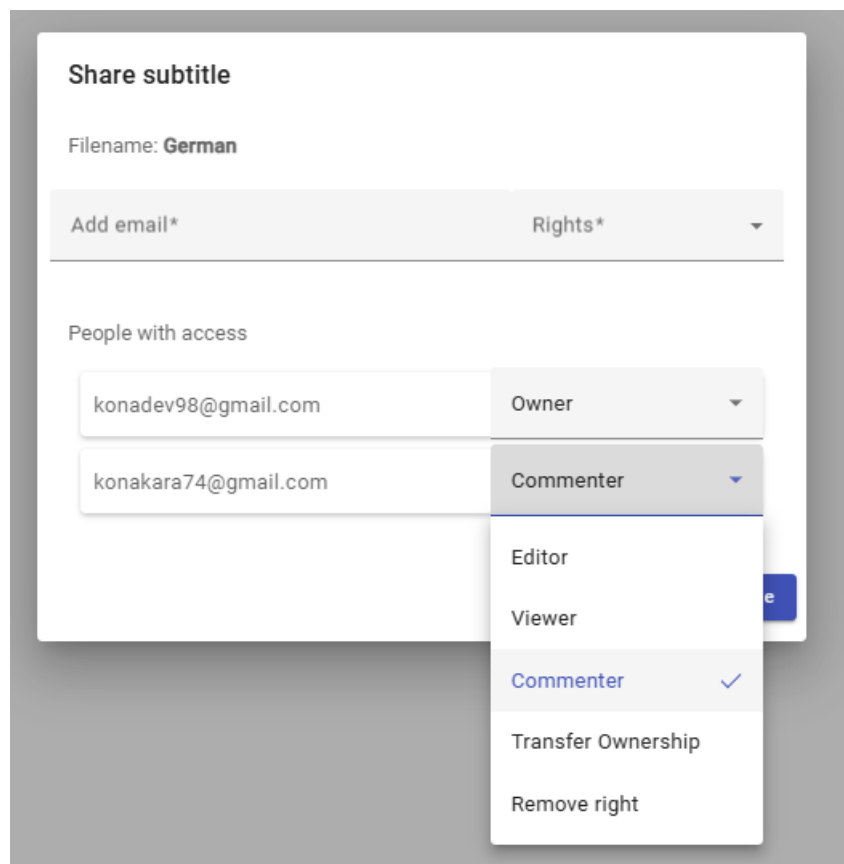
Αν κάποιος άλλος χρήστης κάνει κλικ για να επεξεργαστεί τον ίδιο υπότιτλο, του εμφανίζεται ένα μήνυμα ότι ο συγκεκριμένος υπότιτλος επεξεργάζεται από άλλο χρήστη, του οποίου το όνομα εμφανίζεται και στο μήνυμα. Ο χρήστης μπορεί να συνεχίσει μόνο σε view-only mode, δηλαδή σαν Viewer.



Σχήμα 5.14 Μήνυμα κλειδώματος υπότιτλου

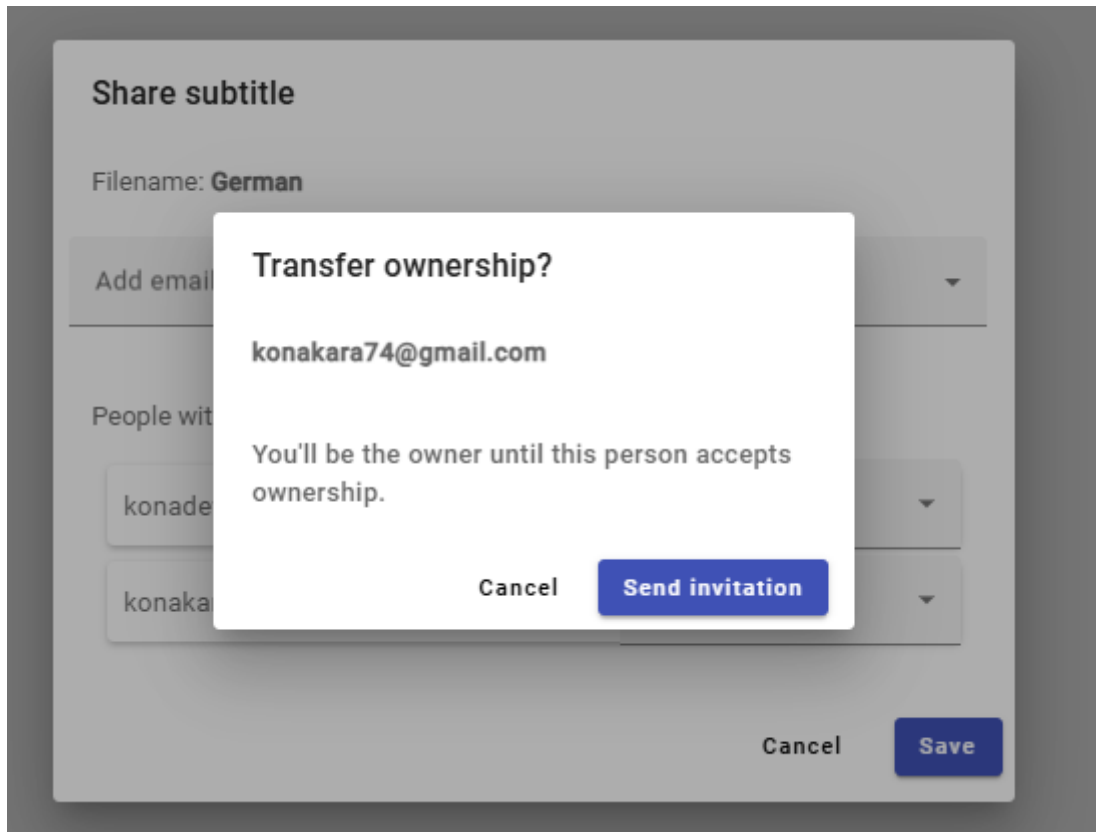
### 5.2.3 Μεταβίβαση ιδιοκτησίας

Μία από τις πιο σημαντικές επιλογές που παρέχονται στα δικαιώματα διαχείρισης είναι η δυνατότητα μεταβίβασης της ιδιοκτησίας (Transfer Ownership). Αυτή η λειτουργία δίνει στον τρέχοντα ιδιοκτήτη τη δυνατότητα να παραχωρήσει την κυριότητα ενός συγκεκριμένου υποτίτλου σε έναν άλλο χρήστη. Ο νέος ιδιοκτήτης πρέπει να είναι κάποιος από τους χρήστες που έχουν ήδη προστεθεί και έχουν πρόσβαση στο συγκεκριμένο περιεχόμενο. Όταν ο ιδιοκτήτης επιλέξει την εντολή “Transfer Ownership” από το dropdown μενού, ανοίγει ένα νέο παράθυρο διαλόγου με το επιλεγμένο email για τη μεταβίβαση.



Σχήμα 5.15 Right options

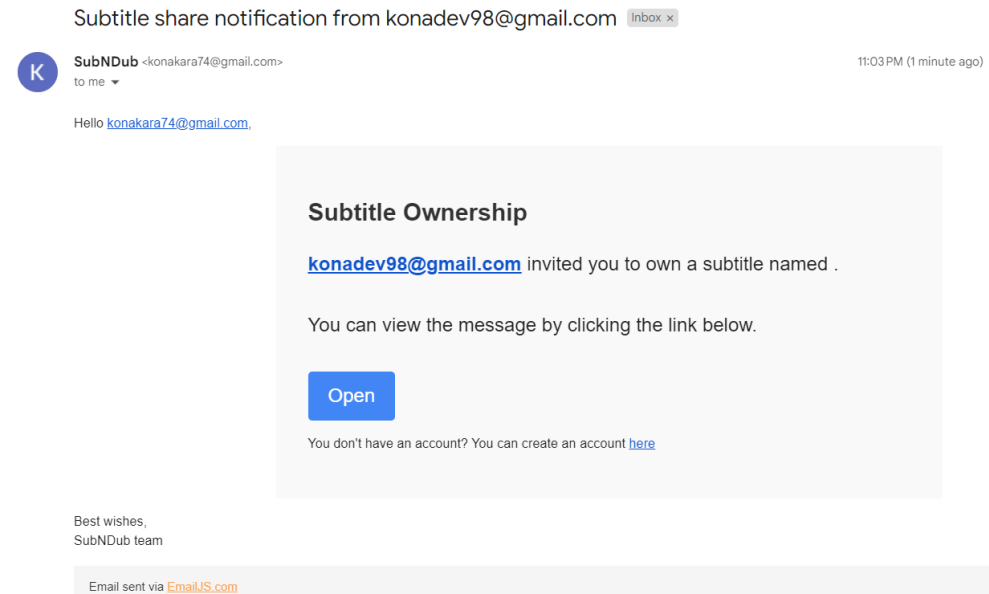
Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη σε περιπτώσεις όπου ο αρχικός ιδιοκτήτης επιθυμεί να μεταφέρει τις ευθύνες διαχείρισης σε κάποιον άλλο χρήστη, για παράδειγμα λόγω αλλαγής ρόλων ή ανακατανομής καθηκόντων.



Σχήμα 5.16 Transfer Ownership

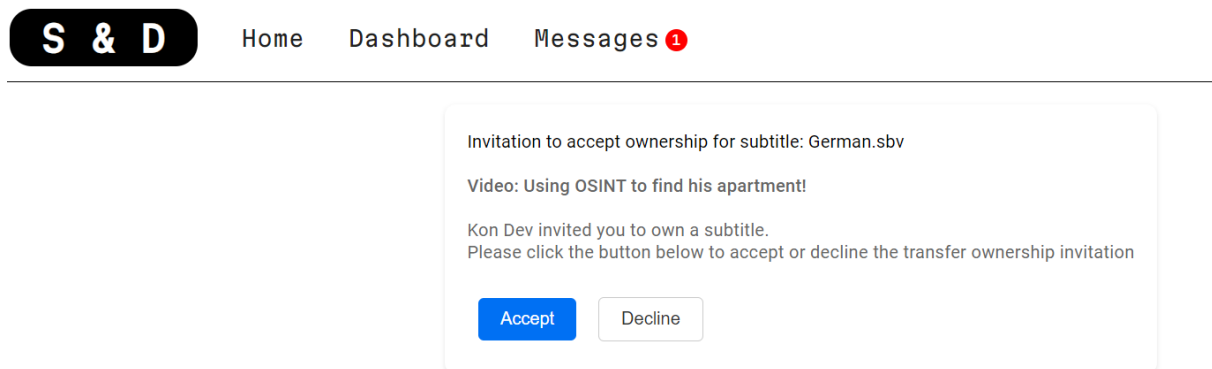
Πατώντας “Send invitation” ο χρήστης με το επιλεγμένο email λαμβάνει μία πρόσκληση για να δεχτεί την κυριότητα του υποτίτλου. Στο email που θα λάβει υπάρχει ένα link που κάνει ανακατεύθυνση στο Messages tab, όπου και εμφανίζονται τα μηνύματα του χρήστη. Μέχρι να αποδεχτεί την κυριότητα ο χρήστης αυτός, παραμένει ιδιοκτήτης αυτός που ήταν.

Το function `transferOwnership()`, το οποίο καλείται όταν δεχτεί ο χρήστης την μεταβίβαση ιδιοκτησίας, μεταβιβάζει την ιδιοκτησία ενός υποτίτλου από έναν χρήστη (`from_email`) σε έναν άλλον (`to_email`). Ξεκινά με την αποστολή ειδοποίησης μέσω email στον νέο κάτοχο του υποτίτλου. Στη συνέχεια, ανακτά το όνομα του χρήστη που μεταβιβάζει την ιδιοκτησία. Εντοπίζει την καταχώρηση του υπότιτλου στη συλλογή `sharedVideos` στη βάση δεδομένων και ενημερώνει το πεδίο `requestOwnerEmail` με το email του νέου κατόχου. Έπειτα, δημιουργεί ένα μήνυμα πρόσκλησης για τη μεταβίβαση, το οποίο προστίθεται στη συλλογή μηνυμάτων του νέου κατόχου. Αν υπάρχει παλιό μήνυμα στο λογαριασμό του προηγούμενου κατόχου, αυτό διαγράφεται για να μην προκαλείται σύγχυση. Τέλος, εμφανίζεται μια ειδοποίηση στον χρήστη για την επιτυχή ή αποτυχημένη αποστολή της πρόσκλησης.



Σχήμα 5.17 Send invitation email (EmailJS)

Το μήνυμα που λαμβάνει ο χρήστης στο tab Messages έχει την παρακάτω μορφή:



Σχήμα 5.18 Accept ownership invitation message

Μόλις ο χρήστης αποδεχτεί την κυριότητα του υποτίτλου, το βίντεο με τον υπότιτλο εμφανίζονται στο My Videos και στο My Shared Videos, καθώς ο προηγούμενος ιδιοκτήτης έχει πλέον το δικαίωμα “Editor”.

Στην περίπτωση που δεν έχει αποδεχτεί ή απορρίψει την κυριότητα του υποτίτλου ο χρήστης, εμφανίζεται η ένδειξη “Owner pending” στο email του.

**Share subtitle**

Filename: **Portogal**

Add email*	Rights* ▼
------------	-----------

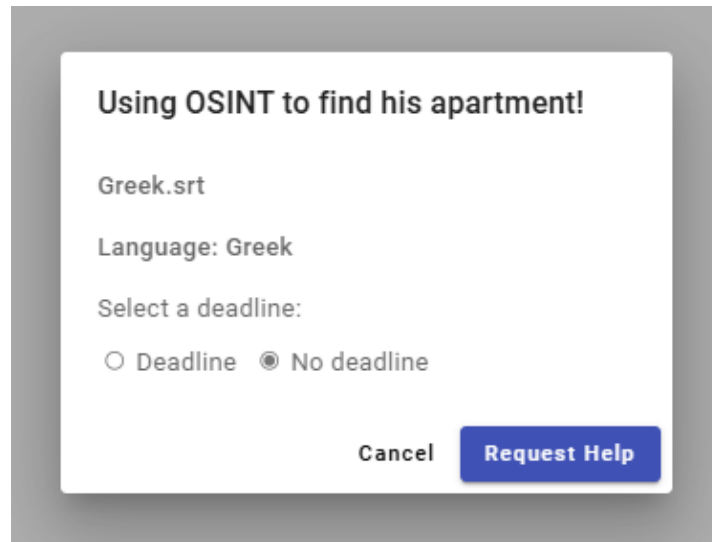
People with access

konadev98@gmail.com	Owner ▼
konkar.dev98@gmail.com	Viewer ▼
Owner pending	

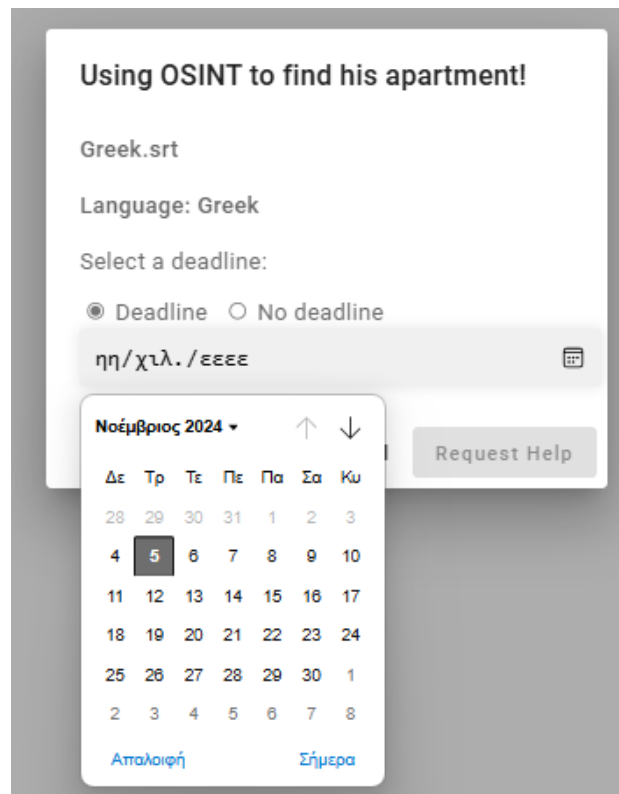
Σχήμα 5.19 Owner pending

Στα details του βίντεο, ο ιδιοκτήτης έχει τη δυνατότητα να χρησιμοποιήσει την επιλογή "Request for Help", με την οποία το βίντεο με τον υπότιτλο εμφανίζεται στο Community view και τίθενται στην διάθεση των χρηστών να κάνουν μία προσφορά στον ιδιοκτήτη για να δημιουργήσουν τον υπότιτλο στην γλώσσα που ζητήθηκε.

Κατά την ενεργοποίηση αυτής της επιλογής, εμφανίζεται ένα νέο παράθυρο διαλόγου που επιτρέπει στον ιδιοκτήτη να διαμορφώσει τις λεπτομέρειες του αιτήματος. Σε αυτό το παράθυρο, μπορεί να καθορίσει αν θα ορίσει συγκεκριμένο χρονοδιάγραμμα για την ολοκλήρωση του υποτίτλου. Υπάρχουν δύο επιλογές: **“No Deadline”**, που σημαίνει ότι το αίτημα θα είναι ενεργό χωρίς χρονικό περιορισμό, ή **“Deadline”**, που επιτρέπει στον ιδιοκτήτη να επιλέξει μια συγκεκριμένη ημερομηνία μέχρι την οποία οι ενδιαφερόμενοι χρήστες θα πρέπει να ολοκληρώσουν την εργασία. Αυτό είναι ιδιαίτερα χρήσιμο όταν υπάρχει ανάγκη για έγκαιρη ολοκλήρωση.



Σχήμα 5.20 No Deadline request help



Σχήμα 5.21 Deadline request help

### 5.3 My Shared Videos

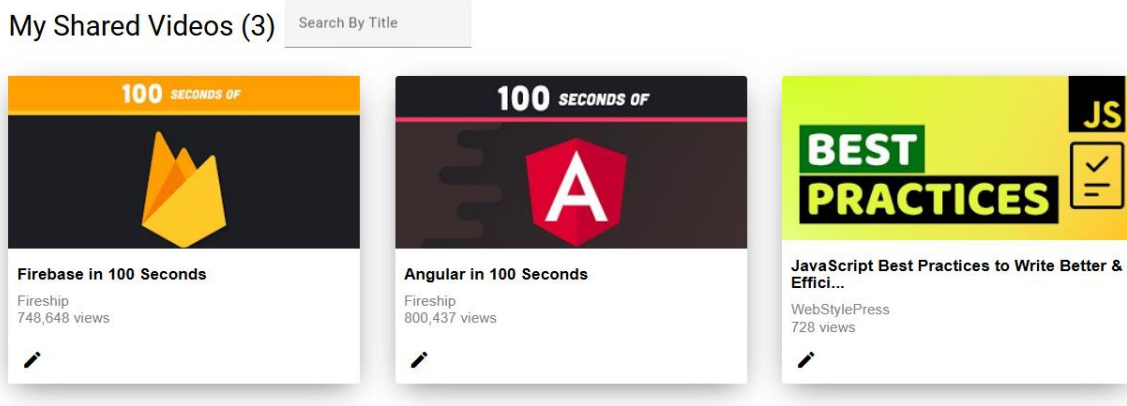
Στο My Shared Videos εμφανίζονται όλα εκείνα τα βίντεο που ο current user έχει κάποιο δικαίωμα και τα δικά του βίντεο στα οποία έχει δώσει σε άλλους χρήστες δικαιώματα.

Σε αυτό το section, παρέχεται επίσης στον χρήστη η δυνατότητα να αναζητήσει βίντεο με βάση τον τίτλο τους, διευκολύνοντας έτσι την εύρεση του επιθυμητού περιεχομένου. Η διαδικασία αναζήτησης

γίνεται μέσω μιας εύχρηστης και απλής γραμμής αναζήτησης. Με βάση την εισαγωγή, το σύστημα εμφανίζει προτάσεις που ταιριάζουν με τα κριτήρια της αναζήτησης, διευκολύνοντας την άμεση επιλογή του επιθυμητού αρχείου.

Στα details τώρα του κάθε βίντεο υπάρχουν σχεδόν τα ίδια στοιχεία με τα details του My Videos, με την εξαίρεση ότι από εδώ δεν μπορεί ο χρήστης να κάνει request για το Community και να διαγράψει τον υπότιτλο.

Η διαφορά στο κομμάτι της κοινοποίησης εδώ είναι ότι όταν ο χρήστης είναι ο ιδιοκτήτης, μπορεί να προσθέσει και από εδώ νέους χρήστες, είτε να τροποποιήσει τα δικαιώματα των χρηστών που υπάρχουν. Όταν όμως ο χρήστης δεν είναι ο ιδιοκτήτης αλλά έχει απλά κάποιο δικαίωμα στον υπότιτλο, τότε μπορεί μόνο να δει ποιοι άλλοι χρήστες έχουν δικαίωμα για τον συγκεκριμένο υπότιτλο, χωρίς να μπορεί να τροποποιήσει κάποια πληροφορία.



Σχήμα 5.22 My Shared Videos view

#### Code snippet:

```
getSharedVideos(email: any): Observable<SharedVideo[]> {
  return
  this.firestore.collection<SharedVideo>('sharedVideos').snapshotChanges().pipe(
    map(actions => {
      return actions.map(a => {
        const data = a.payload.doc.data() as SharedVideo;
        const id = a.payload.doc.id;
        const owner = data.usersRights.find(right => right.right ===
'Owner')?.userEmail;
        return { id,...data, owner };
      }).filter(video => {
        return video.usersRights.some(right => right.userEmail === email);
      });
    }),
    map(videos => [...new Map(videos.map(video => [video.videoId,
video])).values()])
  );
}
```

}

Με την μέθοδο `getSharedVideos()` εμφανίζονται όλα τα βίντεο στα οποία ο `current user` έχει δικαιώματα. Πιο αναλυτικά:

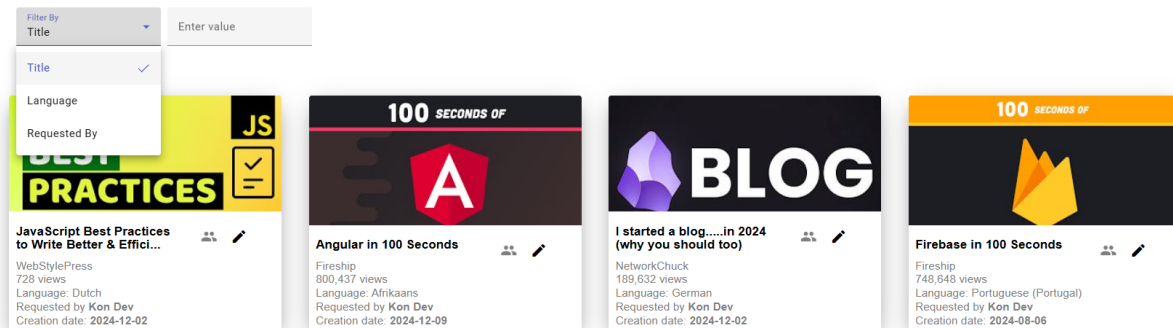
- Η μέθοδος `getSharedVideos` δέχεται το `email` ενός χρήστη και επιστρέφει έναν `Observable` με λίστα βίντεο `SharedVideo` από τη συλλογή `sharedVideos`.
- Χρησιμοποιεί την `snapshotChanges()` για να παρακολουθεί τις αλλαγές στη συλλογή σε πραγματικό χρόνο.
- Χρησιμοποιεί `map` για να δημιουργήσει αντικείμενα που περιέχουν `id`, όλα τα δεδομένα του `SharedVideo`, και το `owner`, που είναι το `email` του ιδιοκτήτη.
- Διατηρεί μόνο τα βίντεο στα οποία ο χρήστης έχει πρόσβαση (όπου `userEmail` ταιριάζει με το `email`).
- Χρησιμοποιεί έναν `Map` για να εξασφαλίσει ότι κάθε `videoId` εμφανίζεται μόνο μία φορά.
- Επιστρέφει το τελικό φιλτραρισμένο και μοναδικό σύνολο βίντεο ως `Observable`.

## 5.4 Community

Το τρίτο `view` στο `dashboard` είναι το `Request for BID` ή αλλιώς `Community`. Σε αυτό το `view` εμφανίζονται όλα τα βίντεο για τα οποία έχει ζητηθεί από τον ιδιοκτήτη τους βοήθεια για μετάφραση στην επιθυμητή γλώσσα.

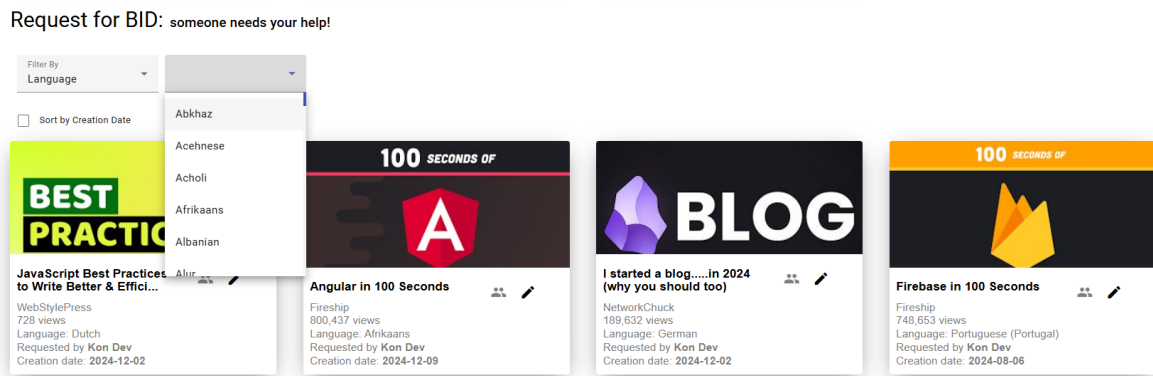
Στο τρίτο αυτό `view`, υπάρχει μια `dropdown` λίστα που επιτρέπει στον χρήστη να επιλέξει τον τρόπο αναζήτησης των βίντεο, παρέχοντας μεγαλύτερη ευελιξία και ακρίβεια στη διαδικασία. Οι διαθέσιμες επιλογές είναι τρεις: **‘Title’**, **‘Language’** και **‘Requested By’**.

Request for BID: someone needs your help!



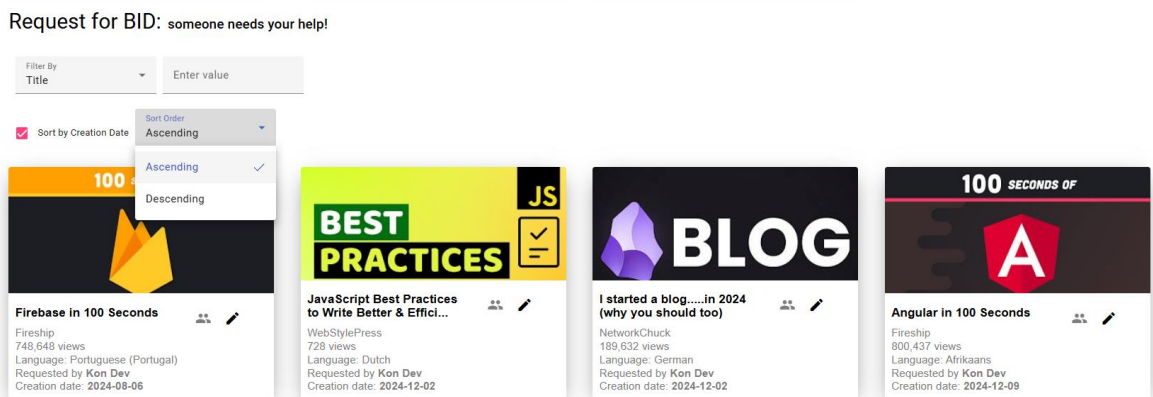
Σχήμα 5.23 Community search

Με την επιλογή **‘Language’**, εμφανίζεται μια δεύτερη `dropdown` λίστα από την οποία ο χρήστης μπορεί να διαλέξει τη γλώσσα που επιθυμεί, φιλτράροντας έτσι τα αποτελέσματα με βάση τη γλώσσα των υποτίτλων.



Σχήμα 5.24 Community search by Language

Επιπλέον, έχει προστεθεί η δυνατότητα ταξινόμησης των βίντεο με βάση την ημερομηνία δημιουργίας του υπότιτλου. Ο χρήστης μπορεί να επιλέξει αν η ταξινόμηση θα γίνει σε **αύξουσα** ή **φθίνουσα** σειρά, ανάλογα με τις ανάγκες του.

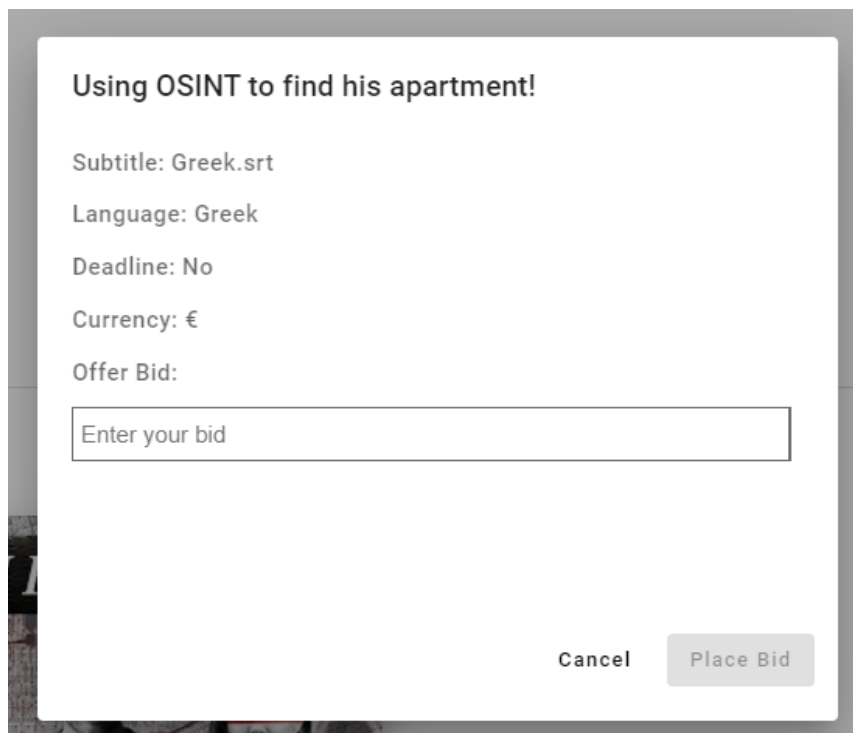


Σχήμα 5.25 Community sort

Κάνοντας κλικ στο βίντεο ο χρήστης μπορεί να δώσει μία προσφορά και να θέσει το χρόνο (deadline) που θα του πάρει για να ολοκληρώσει τη μετάφραση του υπότιτλου. Υπάρχει και η περίπτωση ο χρόνος να είναι αόριστος, να μην έχει συγκεκριμένο χρονικά περιθώριο, στην περίπτωση αυτή ο χρήστης προσθέτει μόνο τη χρηματική προσφορά του, όπως φαίνεται στις εικόνες παρακάτω.

Πατώντας στο εικονίδιο με το group ανοίγει ένα dialog prompt για να εισάγει την προσφορά του ο χρήστης:

```
openPlaceABid(requestDetails: any, videoTitle: string, filename: string): void
{
    this.dialog.open(PlaceABidComponent,{width:'550px', height: '470px', data:
    { language:requestDetails.language, videoTitle, videoId: this.videoId,
    requestedByID: requestDetails.requestedByID, filename, requestDetails,
    userEmail:this.userEmail}}).afterClosed().pipe(take(1)).subscribe(dialog => {
        if (dialog === (null || undefined )){
            this.dialog.closeAll();
        }else if (dialog && dialog.yourBidAmount>0) {
            this.notifier.showNotification("Offer has been successfully sent to
            the requestor.", "OK");
        }
    });
}
```



Σχήμα 5.26 Place a bid without deadline

**Angular in 100 Seconds**

Subtitle: Danish.srt

Language: Danish

Deadline: 2024-08-15

Currency: €

Offer Deadline:

ηη/μμ/2024

Offer Bid:

Enter your bid

Cancel Place Bid

Σχήμα 5.27 Place a bid with deadline

Όταν ο χρήστης συμπληρώσει τη φόρμα της προσφοράς και πατήσει “Place Bid” , στέλνεται μήνυμα/ειδοποίηση στον ιδιοκτήτη του υποτίτλου ότι ο συγκεκριμένος χρήστης έχει κάνει μια προσφορά. Στο μήνυμα εμφανίζονται το όνομα του χρήστη, του βίντεο και του υπότιτλου, η χρηματική προσφορά και το deadline.

Messages 1

Offer for helping with subtitle: Danish.srt

Video: Angular in 100 Seconds

Kon place the below bid for this subtitle.  
 Bid: 10€  
 Deadline: No

Accept Offer
Decline Offer

Σχήμα 5.28 Offer message

Ο ιδιοκτήτης έχει την επιλογή να απορρίψει την προσφορά ή να την δεχτεί. Απορρίπτοντας την προσφορά, διαγράφεται το μήνυμα από το Messages. Στην περίπτωση που δεχτεί κάποια προσφορά, ο χρήστης που έκανε αυτή την προσφορά λαμβάνει μήνυμα ότι έγινε δεκτή η προσφορά του και έχει πλέον το δικαίωμα “Editor” για το συγκεκριμένο υπότιτλο στο αναφερόμενο βίντεο. Το βίντεο αυτό θα εμφανίζεται στο Dashboard στο view My Shared Videos. Μήνυμα για το κλείσιμο της προσφοράς λαμβάνουν και όλοι όσοι έχουν στείλει κάποια προσφορά.

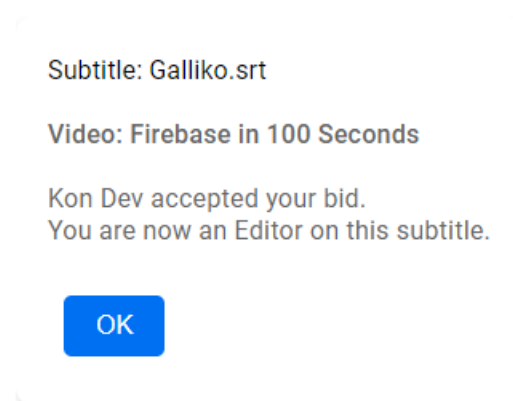
### Code snippet:

Το function αυτό χρησιμοποιείται για να διαγράψει ένα συγκεκριμένο μήνυμα από τα μηνύματα ενός χρήστη στη βάση δεδομένων Firestore. Δέχεται το userid του χρήστη, καθώς και στοιχεία για το βίντεο και τον υπότιτλο (όπως videoId, ISOcode, language, name, και subtitleId). Η μέθοδος εντοπίζει τα μηνύματα που ταιριάζουν με τα συγκεκριμένα κριτήρια στο Firestore μέσα στη συλλογή messages του χρήστη. Στη συνέχεια, αναζητά τα έγγραφα που πληρούν τις συνθήκες και διαγράφει κάθε ένα από αυτά, εξασφαλίζοντας έτσι ότι το μήνυμα που αφορά τον συγκεκριμένο υπότιτλο αφαιρείται από τον λογαριασμό του χρήστη.

```
removeMessageFromUser(userid: string, videoId:string, ISOcode:string,
language:string, name:string, subtitleId: any){
  const messageRef =
this.firestore.collection('users').doc(userid).collection('messages', ref=>
ref.where('videoId', '==', videoId).where('iso', '==',
ISOcode).where('language', '==', language)
  .where('subtitle_name', '==', name).where("subtitleId", "==",
subtitleId));

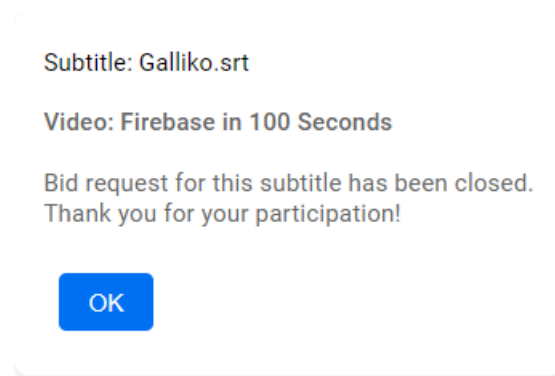
  messageRef.get().toPromise().then(querySnapshot => {
    querySnapshot.forEach(doc => {
      doc.ref.delete();
    });
  });
}
```

## Messages 1



Σχήμα 5.29 Accept message

## Messages 1

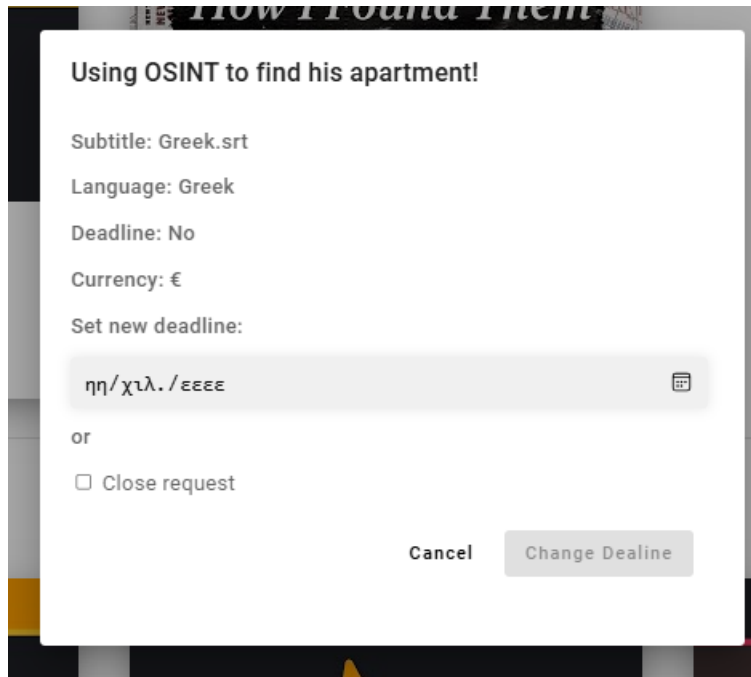


Σχήμα 5.30 Close request message

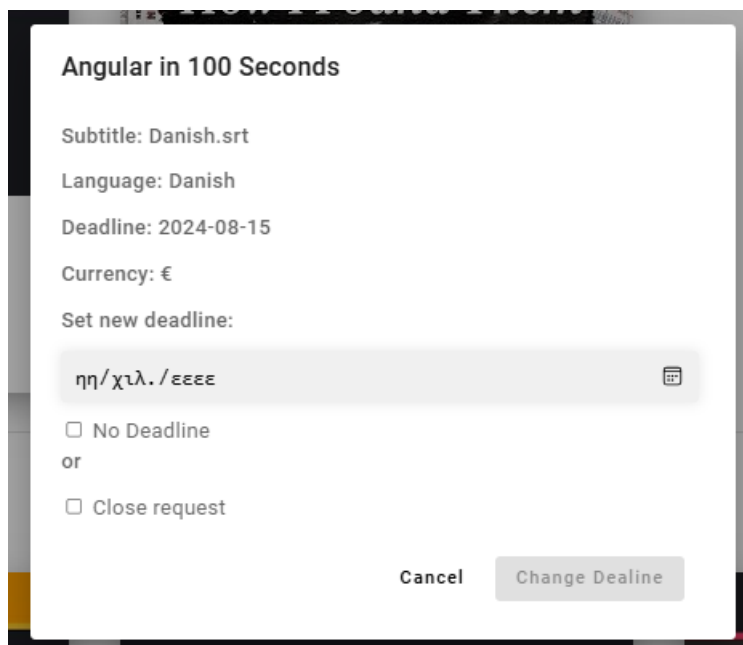
Ο ιδιοκτήτης του βίντεο έχει τη δυνατότητα να επεξεργαστεί το deadline του υποτίτλου, είτε με το να αλλάξει την ημερομηνία του deadline είτε με το να αφαιρέσει το deadline. Έχει ακόμα τη δυνατότητα να κλείσει και το request αυτό. Σε κάθε περίπτωση ο κάθε χρήστης που έχει στείλει προσφορά για το βίντεο για το συγκεκριμένο υπότιτλο θα ενημερωθεί με το αντίστοιχο μήνυμα στο Messages.

Κάνοντας click για επεξεργασία του βίντεο στο Request for BID view γίνεται η ανάκτηση του deadline του υποτίτλου με την παρακάτω μέθοδο:

```
ngOnInit(): void {
  this.firestore.collection('helpRequests', ref => {
    return ref.where('videoId', '==', this.videoId)
      .where('filename', '==', this.fileName)
      .where('language', '==', this.languageRequested)
      .where('requestedByID', '==', this.requestedByID);
  })
  .get()
  .subscribe((querySnapshot) => {
    querySnapshot.forEach((doc) => {
      this.deadline = doc.get('deadline');
    });
  });
}
```



Σχήμα 5.31 Edit request with no deadline

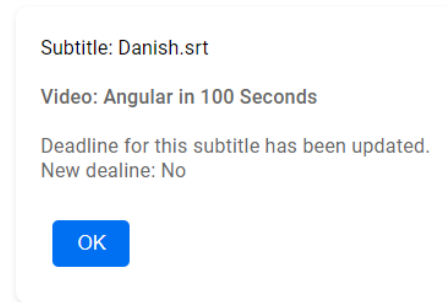


Σχήμα 5.32 Edit request with deadline

Κάνοντας click στο Change Deadline, που ενεργοποιείται μόνο όταν επιλεχτεί μία από τις τρεις επιλογές που υπάρχουν για το deadline, εκτελείται η μέθοδος `changeDeadline()`. Στη μέθοδο αυτή δημιουργούνται τα κατάλληλα μηνύματα που θα σταλούν στους χρήστες, ανάλογα με το τι έχει επιλέξει να αλλάξει ο ιδιοκτήτης του υπότιτλου.

Ένα από τα μηνύματα που εμφανίζεται για την αλλαγή του deadline είναι το παρακάτω:

Home Dashboard Messages 1



Σχήμα 5.33 Changed deadline message

## 5.5 Messages

Στο tab αυτό εμφανίζονται όλα τα μηνύματα του χρήστη. Πιο συγκεκριμένα με την μέθοδο `ngOnInit()` γίνεται η αρχικοποίηση των μηνυμάτων:

```
ngOnInit(): void {
  this.authService.user.subscribe((id) => {
    this.userid = id.uid;
    this.userEmail = id.email;

    const messagesCollection =
this.messagesService.getUserMessages(this.userid);
    const messagesQuery = messagesCollection.stateChanges();
    this.messagesSubscription = messagesQuery.subscribe(changes => {
      changes.forEach(change => {
        if (change.type === 'added') {
          const data = change.payload.doc.data();
          this.messages.push({
            id: change.payload.doc.id,
            sender: data.sender,
            recipient: data.recipient,
            subject: data.subject,
            createdAt: data.createdAt,
            subtitle_name: data.subtitle_name,
            body: data.body,
            videoId: data.videoId,
            status: data.status,
            iso: data.iso,
```

```

        language: data.language,
        format: data.format,
        videoTitle: data.videoTitle,
        subtitleId: data.subtitleId
    } as Message);
} else if (change.type === 'removed') {
    const index = this.messages.findIndex(message => message.id ===
change.payload.doc.id);
    if (index >= 0) {
        this.messages.splice(index, 1);
    }
} else if (change.type === 'modified') {
    const data = change.payload.doc.data();
    const index = this.messages.findIndex(message => message.id ===
change.payload.doc.id);
    if (index >= 0) {
        this.messages[index] = {
            id: change.payload.doc.id,
            sender: data.sender,
            recipient: data.recipient,
            subject: data.subject,
            createdAt: data.createdAt,
            subtitle_name: data.subtitle_name,
            body: data.body,
            videoId: data.videoId,
            status: data.status,
            iso: data.iso,
            language: data.language,
            format: data.format,
            videoTitle: data.videoTitle,
            subtitleId: data.subtitleId
        } as Message;
    }
}
});
});
});
}

```

Η μέθοδος `ngOnInit` αρχικοποιεί τη διαδικασία φόρτωσης και παρακολούθησης των μηνυμάτων του τρέχοντος χρήστη όταν το `component` δημιουργείται. Ακολουθεί τα κύρια βήματα που εκτελεί:

1. **Παρακολούθηση Ταυτότητας Χρήστη:**
  - Αρχικά, συνδέεται με το `authService.user` και λαμβάνει τα στοιχεία του τρέχοντος συνδεδεμένου χρήστη (`userid` και `userEmail`).
2. **Ανάκτηση Μηνυμάτων Χρήστη:**
  - Χρησιμοποιεί το `messagesService` για να ανακτήσει την αναφορά στα μηνύματα του χρήστη μέσω της `getUserMessages` μεθόδου, βασισμένη στο `userid`.
3. **Παρακολούθηση Αλλαγών Στα Μηνύματα:**

- Δημιουργεί ένα `messagesQuery` με `stateChanges()` για να παρακολουθεί σε πραγματικό χρόνο τις αλλαγές στη συλλογή μηνυμάτων (όπως νέες προσθήκες, διαγραφές ή τροποποιήσεις).
4. **Ενημέρωση Κατάστασης των Μηνυμάτων:**
- Συνδρομή στη `messagesSubscription` για την παρακολούθηση των αλλαγών:
    - **Προσθήκη** (`added`): Προσθέτει ένα νέο μήνυμα στον τοπικό πίνακα `messages`.
    - **Διαγραφή** (`removed`): Εντοπίζει και αφαιρεί το μήνυμα από τον πίνακα `messages`.
    - **Τροποποίηση** (`modified`): Εντοπίζει και ενημερώνει ένα υπάρχον μήνυμα με τα νέα δεδομένα.

Η μέθοδος διασφαλίζει ότι ο τοπικός πίνακας `messages` διατηρείται συγχρονισμένος με τα δεδομένα στη `Firestore`, επιτρέποντας την προβολή των τελευταίων αλλαγών των μηνυμάτων στον χρήστη σε πραγματικό χρόνο.

Η μέθοδος `getMessages` ανακτά και φιλτράρει τα μηνύματα ενός συγκεκριμένου χρήστη από μια `Firestore` συλλογή, εμφανίζοντας μόνο τα μηνύματα που έχουν αποδέκτη το `email` του χρήστη (`userEmail`).

#### Code snippet:

```
getMessages(userid: string, userEmail: string) {
  this.messagesService.getUserMessages(userid)
    .snapshotChanges()
    .pipe(
      switchMap(actions => {
        this.messages = actions.map(action => {
          const data = action.payload.doc.data();
          const id = action.payload.doc.id;
          return {
            id,
            sender: data.sender,
            recipient: data.recipient,
            subject: data.subject,
            createdAt: data.createdAt,
            subtitle_name: data.subtitle_name,
            body: data.body,
            videoId: data.videoId,
            status: data.status,
            iso: data.iso,
            language: data.language,
            subtitleId: data.subtitleId
          } as Message;
        });
        return of(actions).filter(message => message.recipient === userEmail);
      })
    )
    .subscribe();
}
```

Αρχικά, καλεί τη μέθοδο `getUserMessages(userid)` του `messagesService` για να αποκτήσει μια αναφορά στα μηνύματα που σχετίζονται με το συγκεκριμένο χρήστη (`userid`). Στη συνέχεια, χρησιμοποιεί τη μέθοδο `snapshotChanges()` για να παρακολουθεί σε πραγματικό χρόνο τις αλλαγές στα μηνύματα του χρήστη στη βάση δεδομένων.

Αφού τα δεδομένα περάσουν μέσα από έναν `pipe` με `switchMap`, κάθε αλλαγή (`action`) επεξεργάζεται ώστε να εξάγονται τα περιεχόμενα του μηνύματος, όπως `sender`, `recipient`, `subject`, `createdAt`, `subtitle_name`, και άλλες πληροφορίες, μαζί με το μοναδικό `id` του εγγράφου. Στη συνέχεια, δημιουργεί ένα νέο αντικείμενο `Message` με τα συγκεκριμένα δεδομένα και χρησιμοποιεί ένα `filter` για να κρατήσει μόνο εκείνα τα μηνύματα όπου ο `recipient` ταιριάζει με το `userEmail`. Έτσι, φιλτράρει τα δεδομένα ώστε να διατηρείται στη μεταβλητή `this.messages` μόνο το σύνολο των μηνυμάτων που αφορούν τον τρέχοντα χρήστη.

Η μέθοδος ολοκληρώνεται με ένα `subscribe()`, το οποίο εξασφαλίζει ότι η `this.messages` ενημερώνεται άμεσα με κάθε νέα προσθήκη, τροποποίηση ή διαγραφή μηνύματος, κρατώντας τη σε πραγματικό χρόνο συγχρονισμένη με τις αλλαγές στη `Firestore`.

## Κεφάλαιο 6ο: Συμπεράσματα & Βελτιώσεις

Η παρούσα πτυχιακή εργασία επικεντρώθηκε στην ανάπτυξη και αξιολόγηση μιας καινοτόμου διαδικτυακής εφαρμογής, σχεδιασμένης για τη διευκόλυνση και τη διαχείριση της κοινοποίησης υποτίτλων σε βίντεο YouTube, προσφέροντας μια εμπειρία συνεργασίας παρόμοια με αυτή του Google Docs. Στόχος της εφαρμογής ήταν η βελτίωση της προσβασιμότητας και της αμεσότητας στον υποτιτλισμό, προσαρμοσμένη για την κάλυψη των ποικίλων αναγκών των χρηστών. Μέσα από τον σχεδιασμό και την υλοποίηση μιας φιλικής προς τον χρήστη πλατφόρμας, αναπτύχθηκε ένα εργαλείο που όχι μόνο υποστηρίζει τη διαδικασία δημιουργίας υποτίτλων αλλά και ενθαρρύνει τη συνεργασία και την αίσθηση της κοινότητας, κάνοντας τη διαδικασία του υποτιτλισμού πιο αποτελεσματική και προσβάσιμη.

Βασική συνιστώσα της εφαρμογής ήταν η διαχείριση των δικαιωμάτων πρόσβασης, δίνοντας στους χρήστες τη δυνατότητα να μοιράζονται, να συνεργάζονται και να συνεισφέρουν στο κομμάτι του υποτιτλισμού.

Μέσω ενός προηγμένου συστήματος ελέγχου δικαιωμάτων, οι χρήστες μπορούσαν να ορίσουν με ακρίβεια ποιοι είχαν δικαιώματα προβολής, επεξεργασίας ή πλήρους διαχείρισης των υποτίτλων τους. Παρόμοια με την προσέγγιση του Google Docs, η εφαρμογή υποστήριξε την απόδοση διαφορετικών επιπέδων δικαιωμάτων με δυνατότητα προσαρμογής από τους δημιουργούς των υποτίτλων. Αυτό διευκόλυνε τη συνεργασία, ενθαρρύνοντας χρήστες από διάφορα μέρη να συνεισφέρουν συλλογικά στη διαδικασία υποτιτλισμού και να προσαρμόζουν τους υποτίτλους σε διάφορες γλώσσες, με στόχο τη δημιουργία ενός περιβάλλοντος χωρίς γλωσσικά και πολιτισμικά εμπόδια.

Σημαντική ήταν η συμβολή των σύγχρονων τεχνολογιών στην ανάπτυξη της εφαρμογής, με το Angular Framework και το Firebase να διαδραματίζουν κομβικό ρόλο. Το Angular Framework διευκόλυνε τη διαχείριση των εισαγωγών δεδομένων και τη δυναμική αλληλεπίδραση με τον χρήστη μέσω των βιβλιοθηκών του και των Observables, που διαχειρίζονται αποτελεσματικά τα ασύγχρονα δεδομένα. Επίσης, η δημιουργία και η χρήση υπηρεσιών (services) για την ανάκτηση δεδομένων από εξωτερικές πηγές όπως το Firebase επέτρεψε την εύκολη υλοποίηση και διαχείριση σύνθετων λειτουργιών με κεντρική εστίαση στον χρήστη. Από την άλλη πλευρά, το Firebase παρείχε τις κατάλληλες υποδομές για το “backend” μέρος της εφαρμογής, προσφέροντας λύσεις για την αποθήκευση δεδομένων μέσω του Firestore (μια NoSQL βάση δεδομένων), καθώς και το Cloud Storage για την αποθήκευση και επεξεργασία των αρχείων.

Η εφαρμογή απέδειξε με επιτυχία τις δυνατότητές της, επιτρέποντας στους χρήστες να δημιουργούν, να τροποποιούν και να προσαρμόζουν υποτίτλους με ευκολία, ενσωματώνοντας τη δυνατότητα κοινοποίησης σε άλλους χρήστες της πλατφόρμας. Η δυνατότητα αυτή καλλιέργησε μια αίσθηση κοινότητας, επιτρέποντας στους χρήστες να συνεισφέρουν συλλογικά και να μοιράζονται τους υποτίτλους τους με άλλους. Το σύστημα κοινοποίησης έδωσε στους χρήστες την ευελιξία να εκχωρούν διαφορετικά δικαιώματα, δίνοντας τη δυνατότητα σε συνεργάτες να προσαρμόζουν υποτίτλους σε άλλες γλώσσες ή να κάνουν διορθώσεις και τροποποιήσεις.

Φυσικά, υπάρχουν ακόμα βελτιώσεις που μπορούν να γίνουν στην εφαρμογή οι οποίες θα διευκόλυναν τον χρήστη.

Αρχικά, θα μπορούσε να προστεθεί η δυνατότητα μεταφοράς του χρηματικού ποσού άμεσα στην πλατφόρμα σε περίπτωση έγκρισης της προσφοράς τους από τον ιδιοκτήτη του υποτίτλου. Επιπλέον, θα μπορούσε να βελτιωθεί ο τρόπος με τον οποίο ενημερώνονται οι χρήστες με μεγαλύτερη ακρίβεια στα μηνύματα. Οι χρήστες θα μπορούσαν να ενημερώνονται άμεσα για τυχόν νέες συνεργασίες, προτάσεις ή ενημερώσεις στους υποτίτλους που μοιράζονται, χωρίς να απαιτείται να ελέγχουν συνεχώς την εφαρμογή, ίσως με την αποστολή email.

Συνολικά, η διαδικτυακή εφαρμογή που αναπτύχθηκε αποτελεί ένα ισχυρό εργαλείο για την κοινότητα των χρηστών που ασχολούνται με τον υποτιτλισμό, προσφέροντας όχι μόνο τις βασικές λειτουργίες υποτιτλισμού, αλλά και δυνατότητες κοινοποίησης και συνεργασίας, οι οποίες είναι θεμελιώδεις για ένα διαρκώς εξελισσόμενο ψηφιακό περιβάλλον. Η εφαρμογή αυτή στοχεύει να υποστηρίξει τη δημιουργία μιας πολυπολιτισμικής και προσβάσιμης πλατφόρμας υποτιτλισμού, με ιδιαίτερη έμφαση στην αίσθηση της κοινότητας και της συνεργασίας, εμπνέοντας μελλοντικές βελτιώσεις που θα κάνουν την εμπειρία αυτή ακόμη πιο άμεση, αποτελεσματική και προσιτή για όλους.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Google Workspace Help. (n.d.). *Sharing and permissions in Google Docs*. Retrieved from Google Docs Help Center
- [2] García, I. (2021). *Collaborative and Crowdsourced Translation in the Digital Age*. John Benjamins Publishing Company.
- [3] **Angular - The modern web developer's platform**. Retrieved from [Angular Official Website](#).
- [4] **Data Binding in Angular**. Retrieved from Angular Developer Guide
- [5] **What is TypeScript?** Retrieved from [TypeScript Official Website](#)
- [6] EmailJS Documentation – Introduction <https://www.emailjs.com/docs/>
- [7] **EmailJS Documentation - Email Templates: [Creating email templates | EmailJS](#)**
- [8] Pro Git του Scott Chacon και Ben Straub (<https://git-scm.com/book>)
- [9] Google Inc., “Firebase Documentation”, <https://firebase.google.com/> [Online]. Available: <https://firebase.google.com/docs> (accessed Jan. 8, 2024).
- [10] GitHub , <https://github.com/> [Online]. Available: <https://docs.github.com/en> (accessed Jan. 8, 2024).

## **ΠΑΡΑΡΤΗΜΑ Α : Πηγαίος Κώδικας**

Ο πηγαίος κώδικας (source code) και το deployed application βρίσκονται διαθέσιμα στο GitHub Repository <https://github.com/StamChatzis/SubNDub>