



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ**  
**ΤΜΗΜΗ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**«Η δημοτικότητα ενός όρου στα μέσα κοινωνικής δικτύωσης»**

**Όνομα : Βασίλειος**

**Επώνυμο : Ζωγράφος**

**A.M.: Π14050**

**Επιβλέπων Καθηγητής**

**Δημήτριος Αποστόλου**

*Πειραιάς, Νοέμβριος 2018*

# Περίληψη

Το Twitter αποτελεί ένα κυρίαρχο μέσο κοινωνικής δικτύωσης το οποίο χρησιμοποιείται καθημερινά από εκατομμύρια χρήστες. Ως αποτέλεσμα, αμέτρητα tweets αναρτώνται ανα ώρα τα οποία απαρτίζονται συνήθως από διαφορετικό περιεχόμενο το οποίο μπορεί να κατηγοριοποιηθεί σε συγκεκριμένα θέματα που αποτελούν τα φλέγοντα θέματα.

Έτσι, στην συγκεκριμένη πτυχιακή εργασία αρχικά αναλύουμε το Twitter και τα χαρακτηριστικά του και στη συνέχεια εστιάζουμε στα φλέγοντα θέματα (δημοφιλείς όροι) που αναδύονται καθημερινά μέσα στα πλαίσια αυτού του ιστοτόπου.

Εν συνεχεία, παρουσιάζουμε τον αλγόριθμο που αναπτύξαμε για την παρούσα πτυχιακή εργασία, ο οποίος έχει ως στόχο την αξιολόγηση της δημοτικότητας των προαναφερθέντων φλεγόντων θεμάτων στην πλατφόρμα του Twitter. Αυτό το πετυχαίνει με το να αντλεί μέσω του Twitter API, διάφορα χρήσιμα δεδομένα, όπως το πλήθος των retweets, comments και favorites των σχετικών με έναν αναζητούμενο όρο tweets και να εξάγει ένα βαθμό δημοτικότητας από αυτά.

Τέλος, παρουσιάζουμε διάφορα παραδείγματα εκτέλεσης του συγκεκριμένου αλγορίθμου με σκοπό την επίδειξη της λειτουργίας του.

**Λέξεις κλειδιά:** Twitter, φλέγοντα θέματα, δημοτικότητα όρου

# Abstract

Twitter is a dominant social networking tool used by millions of users every day. As a result, countless tweets are posted per hour, which usually consist of different content that can be categorized into specific topics which are the hot topics.

Thus, in this specific thesis we first analyze Twitter and its features and then we focus on the hot topics (popular terms) that emerge daily within this site.

Next, we present the algorithm we developed within this thesis, which aims to evaluate the popularity of the aforementioned hot topics in the Twitter platform. This is accomplished by mining through the Twitter API, a number of useful data such as the number of retweets, comments and favorites related to a search term's tweets and extracting a popularity mark from them.

Finally, we present several examples of the execution of this algorithm to demonstrate its operation.

**Keywords:** *Twitter, hot topics, term popularity*

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Σελ

## Κεφάλαιο 1ο

### Το μέσο κοινωνικής δικτύωσης Twitter

1.1	Τα social media και ο ρόλος του Twitter .....	6
1.2	Τα χαρακτηριστικά της πλατφόρμας του Twitter .....	8

## Κεφάλαιο 2ο

### Τα φλέγοντα θέματα (δημοτικότητα όρων) στο Twitter

2.1	Η έννοια των φλεγόντων θεμάτων στο Twitter.....	12
2.2	Οι κατηγορίες των φλεγόντων θεμάτων του Twitter.....	15
2.2.1	Ειδήσεις.....	15
2.2.2	Τρέχοντα Γεγονότα.....	16
2.2.3	Meme.....	16
2.2.4	Αναμνηστικά.....	17
2.3	Ο κύκλος ζωής και η δημοτικότητα των φλεγόντων θεμάτων του Twitter.....	18
2.4	Τα γεγονότα και η δημοτικότητά τους στο Twitter.....	21

## Κεφάλαιο 3ο

### Ο αλγόριθμος της πτυχιακής εργασίας

3.1	Ο στόχος του αλγορίθμου.....	23
3.2	Ανάλυση αλγορίθμου.....	24
3.2.1	Αρχείο “config.js”.....	24
3.2.2	Αρχείο “utils.js”.....	26
3.2.3	Αρχείο “search-tweets.js”.....	28
3.2.4	Αρχείο “term-popularity.js”.....	31
3.2.5	Αρχείο “test.js”.....	35
3.2.6	Διαδικασία ταυτοποίησης στο Twitter API.....	37
3.3	Παραδείγματα εκτέλεσης αλγορίθμου.....	39

<b>Παράρτημα</b>	Ο κώδικας του αλγορίθμου της πτυχιακής εργασίας.....	51
------------------	--	----

## Κατάλογος Εικόνων

Εικόνα 1	Αρχείο “config.js”.....	σελ. 25
Εικόνα 2	Αρχείο “utils.js”.....	σελ. 27
Εικόνα 3	Αρχείο “search-tweets.js” (α).....	σελ. 29
Εικόνα 4	Αρχείο “search-tweets.js” (β).....	σελ. 30
Εικόνα 5	Αρχείο “term-popularity.js” (α).....	σελ. 33
Εικόνα 6	Αρχείο “term-popularity.js” (β).....	σελ. 34
Εικόνα 7	Αρχείο “test.js”.....	σελ. 36
Εικόνα 8	Παράδειγμα αναζήτησης “#PokemonGo”.....	σελ. 39
Εικόνα 9	Παράδειγμα αναζήτησης “#PokemonQuest”.....	σελ. 40
Εικόνα 10	Παράδειγμα αναζήτησης “#PokemonLetsGo” (α).....	σελ. 41
Εικόνα 11	Παράδειγμα αναζήτησης “#PokemonLetsGo” (β).....	σελ. 42
Εικόνα 12	Παράδειγμα αναζήτησης “#PokemonLetsGo” (γ).....	σελ. 43
Εικόνα 13	Παράδειγμα αναζήτησης “#NintendoSwitch”.....	σελ. 44
Εικόνα 14	Παράδειγμα αναζήτησης “#Pokemon”.....	σελ. 45
Εικόνα 15	Παράδειγμα αναζήτησης “#Zelda”.....	σελ. 46
Εικόνα 16	Παράδειγμα αναζήτησης “#MarioKart”.....	σελ. 47
Εικόνα 17	Παράδειγμα αναζήτησης “#mariomaker”.....	σελ. 48

# Κεφάλαιο 1

## Το μέσο κοινωνικής δικτύωσης Twitter

### 1.1 Τα social media και ο ρόλος του Twitter

Τα online κοινωνικά και ειδησεογραφικά μέσα δημιουργούν πλούσιες και έγκαιρες πληροφορίες σχετικά με τα κάθε είδους γεγονότα του πραγματικού κόσμου. Ωστόσο, ο τεράστιος όγκος των διαθέσιμων δεδομένων, μαζί με το εύρος της βάσης χρηστών, απαιτούν σημαντική προσπάθεια φιλτραρίσματος των πληροφοριών για την επιτυχή εμβάθυνση σε σχετικά θέματα και γεγονότα. Επομένως, η ανίχνευση του θέματος είναι ένα θεμελιώδες δομικό στοιχείο για την παρακολούθηση και την σύνοψη των πληροφοριών που προέρχονται από κοινωνικές πηγές. Υπάρχουν πολλές μέθοδοι και μεταβλητές που επηρεάζουν σημαντικά την ποιότητα των αποτελεσμάτων.

Ως πλατφόρμα παροχής πληροφοριών, το Twitter συλλέγει εκατομμύρια tweets καθημερινά. Ωστόσο, ορισμένοι χρήστες, ειδικά νέοι χρήστες, συχνά δυσκολεύονται στο να κατανοήσουν τα θέματα που παρουσιάζουν οι τάσεις στο Twitter όταν αντιμετωπίζουν τα συντριπτικά και ανοργάνωτα tweets. Το Twitter, ως δημοφιλής υπηρεσία micro-blogging, συλλέγει εκατομμύρια μηνύματα σύντομων κειμένων σε πραγματικό χρόνο (έως και 280 χαρακτήρες) κάθε

δευτερόλεπτο. Λειτουργεί όχι μόνο ως μια δημόσια πλατφόρμα για την απόσπαση μικροπραγμάτων σχετικά με την καθημερινή ζωή των χρηστών, αλλά και ως δημόσιος ανταποκριτής για ειδήσεις σε πραγματικό χρόνο. Έχει δείξει την ισχυρή ικανότητά του σε πολλές δημόσιες εκδηλώσεις, όπως οι πυρκαγιές στο Σαν Ντιέγκο και ο σεισμός στην Ιαπωνία. Παρ' όλα αυτά, μια παρενέργεια είναι ότι οι μεμονωμένοι χρήστες βομβαρδίζονται συνήθως από μια πλημμύρα εκατομμυρίων tweets.

Το Twitter είναι ένα μέσο κοινωνικής δικτύωσης στο οποίο οι χρήστες μπορούν να δημιουργούν σύντομα μηνύματα που σχετίζονται με μια ευρεία ποικιλία θεμάτων. Ορισμένα θέματα επισημαίνονται από το Twitter ως τα πιο δημοφιλή θέματα και είναι γνωστά ως φλέγοντα θέματα.

Έχει σημειώσει μεγάλη ανάπτυξη από τότε που ξεκίνησε το 2006 και έχει πάνω από 140 εκατομμύρια ενεργών χρηστών με 340 εκατομμύρια μηνύματα (tweets) την ημέρα από τον Μάρτιο του 2012. Οι χρήστες του Twitter γράφουν tweets για οποιοδήποτε θέμα και ακολουθούν άλλους χρήστες για να λαμβάνουν τα tweets τους. Ένα σημαντικό χαρακτηριστικό του Twitter είναι η πραγματικού χρόνου φύση του. Για παράδειγμα, όταν συμβαίνει ένα σημαντικό γεγονός, οι χρήστες διαδίδουν tweets στο διαδίκτυο τα οποία σχετίζονται με το συμβάν, πράγμα που καθιστά δυνατή την έγκαιρη ανίχνευση του γεγονότος με την παρατήρηση των tweets. Τα δημοφιλή γεγονότα και θέματα είναι επίσης γνωστά ως φλέγοντα θέματα, και η ανίχνευσή τους μας βοηθά να καταλάβουμε καλύτερα τι συμβαίνει στον κόσμο.

## 1.2 Τα χαρακτηριστικά της πλατφόρμας του Twitter

Το περιεχόμενο των tweets ποικίλλει ευρέως από απλό, όπως "τι κάνω τώρα", σε θεματικό, όπως "παιχνίδια ποδοσφαίρου". Υπάρχουν επίσης πολλοί διαφορετικοί τρόποι με τους οποίους οι χρήστες μπορούν να ενημερώσουν τα tweets τους, συμπεριλαμβανομένων των κινητών τους τηλεφώνων, των υπηρεσιών ανταλλαγής μηνυμάτων στο Web, κειμένων κ.λ.π. Ως υπηρεσία κοινωνικής δικτύωσης, το Twitter χρησιμοποιεί ένα κοινωνικό μοντέλο που ονομάζεται "following", στο οποίο ο χρήστης έχει τη δυνατότητα να επιλέξει όλους τους άλλους χρήστες που θέλει να ακολουθήσει χωρίς καμία άδεια ή αμοιβαίο "follow" από τον άλλο χρήστη. Αυτός που ακολουθεί έναν άλλο χρήστη ονομάζεται ακόλουθος. Το να είναι κάποιος ακόλουθος ενός άλλου χρήστη στο Twitter σημαίνει ότι λαμβάνει όλες τις ενημερώσεις των φίλων του. Υπάρχουν τρεις τύποι tweets: απάντηση, retweet και κανονικό tweet. Η απάντηση είναι ένα μήνυμα προς οποιονδήποτε χρήστη με το σύμβολο "@" ακολουθούμενο από το όνομα του χρήστη-στόχου στην αρχή του tweet. Το retweet είναι ένα μήνυμα που ο χρήστης μοιράζει στους φίλους του προς τους ακόλουθούς του και αρχίζει με το σύμβολο "RT". Τα κανονικά tweet είναι τα tweets εκτός από τις απαντήσεις και τα retweets. Οι χρήστες μπορούν επίσης να προσθέτουν ετικέτες σχετικές με τα tweets. Στο Twitter η ετικέτα ονομάζεται "hashtag" και ξεκινάει με το σύμβολο "#". Διαφορετικοί τύποι tweets και hashtags κάνουν μια ροή δεδομένων στο Twitter πιο ευανάγνωστη. Το Twitter είναι κατάλληλο σε μεγάλο βαθμό για την



διάδοση πληροφοριών και ειδικότερα για τις ειδησεογραφικές πληροφορίες. Η λειτουργία “following” το καθιστά πολύ κατάλληλο για τη διάδοση των ειδησεογραφικών περιεχομένων. Όταν ένας χρήστης συμμετέχει σε μια συζήτηση που αφορά θέματα ειδήσεων, τα σχετικά tweets θα ωθηθούν στους ακόλουθούς του. Εάν οι ακόλουθοί του ενδιαφέρονται επίσης για το θέμα αυτό, μπορούν επίσης να συμμετάσχουν στη συζήτηση. Έτσι, το θέμα των ειδήσεων διαδίδεται πολύ πιο γρήγορα. Υπάρχουν πολλά νέα γεγονότα και θέματα που συζητούνται καθημερινά στο Twitter. Ορισμένα θέματα μπορεί να πάρουν μεγάλη διάσταση και κάποια μπορεί όχι. Έτσι, κάποια στιγμή, θέλουμε απλώς να μάθουμε ποια θέματα θα αποτελέσουν φλέγοντα θέματα στο Twitter, και για ποιο λόγο. Επομένως, πρέπει να προβλέψουμε την τάση των θεμάτων και να δώσουμε κάποιες εξηγήσεις για τη σημαντική διακύμανση αυτής της τάσης και διάδοσης. Το κείμενο ενός tweet είναι σύντομο, οπότε είναι εύκολο για τους χρήστες να δημοσιεύουν τα μηνύματα των tweets τους με κινητά τηλέφωνα ή άλλα μέσα. Ως αποτέλεσμα, οι χρήστες μπορούν να δημοσιεύουν αυτό που βλέπουν σε πραγματικό χρόνο, ειδικότερα πληροφορίες ειδήσεων.

Όποιος επιλέγει να δημοσιεύσει τα tweets των άλλων στο χρονοδιάγραμμα κάποιου, ονομάζεται ακόλουθος (follower). Το Twitter έχει χρησιμοποιηθεί ως ένα μέσο για τη διάδοση πληροφοριών σε πραγματικό χρόνο και έχει χρησιμοποιηθεί σε διάφορες διαφημιστικές εκστρατείες μάρκας, εκλογές και ακόμα ως μέσο ενημέρωσης. Από την ίδρυσή του το 2006, η δημοτικότητα της χρήσης του Twitter αυξήθηκε δραματικά. Από τον Ιούνιο του 2011, δημιουργούνται περίπου 200 εκατομμύρια tweets κάθε μέρα. Όταν ένα νέο θέμα γίνει δημοφιλές στο Twitter, κατηγοριοποιείται ως ένα

φλέγων θέμα, το οποίο μπορεί να πάρει τη μορφή σύντομων φράσεων (π.χ. Michael Jackson) ή hashtags (π.χ #εκλογές). Η online υπηρεσία “What The Trend” συνηθιζόταν να χρησιμοποιείται για την παροχή μιας τακτικά επικαιροποιημένης λίστας φλεγόντων θεμάτων στον ιστότοπο του Twitter. Είναι πάρα πολύ ενδιαφέρον να γνωρίζουμε ποια θέματα είναι φλέγοντα και τι είδους άνθρωποι σε άλλα μέρη του κόσμου ενδιαφέρονται για αυτά. Ωστόσο, ένα πολύ υψηλό ποσοστό φλεγόντων θεμάτων είναι τα hashtags που μπορεί να είναι ένα όνομα ενός ατόμου, ή λέξεις σε άλλες γλώσσες και είναι συχνά επεξηγηματικά για θέματα τα οποία είναι δύσκολο να καταλάβουμε ως έχουν τι πράγμα αφορούν. Επομένως, είναι σημαντικό να ταξινομηθούν αυτά τα θέματα σε γενικές κατηγορίες για ευκολότερη κατανόηση αυτών και για την καλύτερη ανάκτηση πληροφοριών. Τα ονόματα φλεγόντων θεμάτων ενδέχεται να είναι ενδεικτικά του είδους των πληροφοριών που οι χρήστες αναρτούν στο Twitter, εκτός εάν κάποιος διαβάσει το κείμενο που συνοδεύει το φλέγων θέμα. Για παράδειγμα, το hashtag #prespesagreement δηλώνει ότι οι χρήστες αναρτούν tweets σχετικά με τη συμφωνία των Πρεσπών. Μια άλλη τάση που ονομάζεται **Boone Logan** είναι ενδεικτική ότι τα tweets αφορούν ένα άτομο που ονομάζεται Boone Logan. Όμως, όποιος δεν ακολουθεί το αμερικανικό Major League Baseball (MLB) δεν θα γνωρίζει ότι οι πληροφορίες αυτής της τάσης αφορούν τον Boone Logan, ο οποίος είναι ένας “pitcher” για τους Yankees της Νέας Υόρκης, εκτός και αν διαβάσει μερικά tweets σχετικά με αυτό το φλέγων θέμα.

Οι χρήστες του Twitter μπορούν να χρησιμοποιήσουν μια μορφή σχολιασμού hashtag, όπως #Trump, για να υποδείξουν το περιεχόμενο των μηνυμάτων τους (π.χ. "American Elections #Trump"). Επιπλέον, το Twitter προσφέρει πολλούς τρόπους ώστε οι χρήστες να μπορούν να

συνομιλούν και να αλληλεπιδρούν μέσω της μεταξύ τους αναφοράς σε μηνύματα χρησιμοποιώντας το σύμβολο @. Το Twitter χρησιμοποιεί επί του παρόντος έναν ιδιωτικό αλγόριθμο για την εμφάνιση τάσεων, αποτελούμενων από όρους και φράσεις που παρουσιάζουν την συμπεριφορά των τάσεων. Αν και τα φλέγοντα θέματα του Twitter συχνά αντανakλούν τα τρέχοντα γεγονότα (π.χ. "Παγκόσμιο Κύπελλο"), συχνά περιλαμβάνουν λέξεις-κλειδιά για δημοφιλή θέματα συζήτησης (π.χ. "#MacedonianIssue", "prespes agreement"), χωρίς διάκριση μεταξύ των διαφόρων τύπων περιεχομένου.

Οι αναρτήσεις που γίνονται από έναν χρήστη εμφανίζονται αυτόματα στη σελίδα προφίλ του χρήστη, καθώς και στους ακόλουθούς του. Ένα retweet είναι μια ανάρτηση που έχει αρχικά δημιουργηθεί από έναν χρήστη η οποία προωθείται από έναν άλλο χρήστη. Τα φλέγοντα θέματα παρουσιάζονται ως μια λίστα από το Twitter στον κύριο ιστότοπο Twitter.com. Αποτελούν εκείνες τις λέξεις-κλειδιά οι οποίες εμφανίζονται συχνότερα στην πιο πρόσφατη ροή tweets από αυτήν που θα περίμενε κανείς από μια ανάλυση συχνότητας όρων ενός εγγράφου όπως το TFIDF. Η λίστα των φλεγόντων θεμάτων ενημερώνεται κάθε λίγα λεπτά καθώς νέα θέματα γίνονται πιο δημοφιλή.

## Κεφάλαιο 2

### Το φλέγοντα θέματα (δημοτικότητα όρων) στο Twitter

#### **2.1 Η έννοια των φλεγόντων θεμάτων στο Twitter**

Το Twitter συνοψίζει τα πολλά μηνύματα που αποστέλλονται από τους χρήστες σε μορφή φλεγόντων θεμάτων που αντικατοπτρίζουν τις κορυφαίες συνομιλίες που συζητούνται σε μια δεδομένη χρονική στιγμή. Αυτά τα φλέγοντα θέματα τείνουν να συνδέονται με τις τρέχοντα συμβάντα. Διαφορετικά γεγονότα είναι αυτά που μπορούν να οδηγήσουν στην εμφάνιση αυτών των θεμάτων. Για παράδειγμα, ένα αθλητικό γεγονός που μεταδίδεται μέσω της τηλεόρασης ή ένα ιογενές meme που εισάγεται από μια κοινότητα χρηστών. Η ανίχνευση του τύπου προέλευσης μπορεί να διευκολύνει το φιλτράρισμα των πληροφοριών, να βελτιώσει την επεξεργασία των δεδομένων σε πραγματικό χρόνο και να βελτιώσει την εμπειρία του χρήστη.

Τα tweets στο Twitter τείνουν να εξαπλώνονται σε μεγάλο αριθμό χρηστών σε πολύ μικρό χρονικό διάστημα. Οι χρήστες στο Twitter όχι μόνο κάνουν tweet για τα προσωπικά τους θέματα ή τα τοπικά γεγονότα, αλλά και για γενικότερα θέματα ή ειδήσεις. Λόγω των

μεγάλων ποσοτήτων και ποικιλίας των πληροφοριών σε πραγματικό χρόνο που περιέχονται στον ιστότοπο, το Twitter παραθέτει ένα πρόσφατα ενημερωμένο σύνολο φλεγόντων θεμάτων. Τα φλέγοντα θέματα περιλαμβάνουν τους κορυφαίους όρους που συζητούνται επί του παρόντος στο Twitter. Αυτή η λίστα των κορυφαίων όρων, η οποία ενημερώνεται σε πραγματικό χρόνο, αντικατοπτρίζει τα τρέχοντα κυριότερα ενδιαφέροντα της κοινότητας, δηλαδή τις συζητήσεις που συζητήθηκαν πιο συχνά αυτή τη χρονική στιγμή. Έχει γίνει ένα ελκυστικό χαρακτηριστικό για τους χρήστες του Twitter, τους προγραμματιστές εφαρμογών σε πραγματικό χρόνο και τους ερευνητές των ερευνητών των μέσων κοινωνικής δικτύωσης, χάρη στην ικανότητά τους να ανιχνεύουν τα φλέγοντα θέματα στο αρχικό τους στάδιο. Τα συμφέροντα της μεγάλης κοινότητας των χρηστών του Twitter περιλαμβάνουν ένα τεράστιο αριθμό θεμάτων και τύπων μηνυμάτων. Ως εκ τούτου, δεν υπάρχει κανένας λόγος που να κάνει ένα θέμα να γίνει φλέγων θέμα. Πιστεύεται ότι η ανίχνευση του τύπου ενός τρέχοντος θέματος συμβάλλει στην καλύτερη κατανόηση της προέλευσης που το έκανε να εξελιχθεί. Αυτή η ανίχνευση καθιστά δυνατή την καλύτερη αξιοποίηση δεδομένων σε πραγματικό χρόνο, δεδομένου ότι κάθε είδος καυτού θέματος μπορεί να είναι χρήσιμο για διαφορετικό σκοπό. Η διάκριση των φλεγόντων θεμάτων ανά τύπο πυροδότησης σε πραγματικό χρόνο θα μπορούσε να είναι χρήσιμη στο να φιλτράρει τα ανεπιθύμητα θέματα μέσα σε συστήματα φιλτραρίσματος πληροφοριών ή για να βελτιώσει την επεξεργασία δεδομένων σε πραγματικό χρόνο, αντιμετωπίζοντας τα ξεχωριστά.

Επίσης, το Twitter είναι ένα ταχέως αναπτυσσόμενο εργαλείο μέσων κοινωνικής δικτύωσης σε πραγματικό χρόνο. Καθώς το Twitter εξελίσσεται, όλο και περισσότεροι άνθρωποι συμμετέχουν στην κοινή

χρήση όσων συμβαίνουν σε όλο τον κόσμο μέσω διαφόρων εφαρμογών Twitter. Η χρήση του hashtag έχει γίνει μια μοναδική συνήθεια πραγματοποίησης tagging για να βοηθήσει τη συσχέτιση των μηνυμάτων του Twitter με τα συγκεκριμένα συμβάντα ή περιβάλλοντα. Προτεταμένο από ένα σύμβολο # μαζί με μια λέξη-κλειδί, ένα hashtag του Twitter χρησιμεύει ως σύμβολο προσθήκης ετικετών που προτείνει ο χρήστης από κάτω προς τα πάνω. Συνεισφέρει επίσης στη συμμετοχή των χρηστών στη διαδικασία της καινοτομίας του hashtag, ειδικά επειδή αφορά καθήκοντα οργάνωσης της πληροφορίας.

Ένα από τα κύρια χαρακτηριστικά που βρίσκονται στην αρχική σελίδα του Twitter εμφανίζει ανά πάσα στιγμή μια λίστα με τους κορυφαίους όρους που ονομάζονται φλέγοντα θέματα. Αυτοί οι όροι αντικατοπτρίζουν τα θέματα που συζητούνται περισσότερο στα τελευταία λεπτά μιας ροής από tweets του ιστότοπου. Προκειμένου να αποφευχθούν θέματα που είναι πάντα δημοφιλή, το Twitter επικεντρώνεται σε θέματα που συζητούνται πολύ περισσότερο από το συνηθισμένο, δηλαδή σε θέματα που πρόσφατα υπέστησαν αύξηση χρήσης, έτσι ώστε για κάποιο λόγο να αποτελούν φλέγοντα θέματα. Τα φλέγοντα θέματα έχουν προκαλέσει μεγάλο ενδιαφέρον όχι μόνο στους ίδιους τους χρήστες αλλά και στα άτομα που αναζητούν πληροφορίες όπως δημοσιογράφοι, προγραμματιστές εφαρμογών σε πραγματικό χρόνο και ερευνητές μέσω κοινωνικής δικτύωσης. Η ικανότητα του να γνωρίζει κάποιος τις κορυφαίες συνομιλίες που συζητούνται σε μια συγκεκριμένη χρονική στιγμή βοηθά στην ενημέρωσή του για τις τρέχουσες υποθέσεις και στην ανακάλυψη των κύριων ανησυχιών της κοινότητας. Το Twitter ορίζει τα φλέγοντα θέματα αυτά που είναι άμεσα δημοφιλή, παρά τα θέματα που έχουν γίνει δημοφιλή για μικρό χρονικό διάστημα ή σε καθημερινή βάση. Ένα

φλέγων θέμα αποτελείται από το ίδιο το θέμα -δηλαδή, ο όρος που έγινε τάση- και από μια ροή tweets που περιέχει αυτό το θέμα.

## 2.2 Οι κατηγορίες των φλεγόντων θεμάτων του Twitter

Διαφορετικά γεγονότα, είτε στον κόσμο, είτε στην τηλεόραση είτε στο διαδίκτυο, μπορούν να παρακινήσουν τους χρήστες να συζητήσουν για ένα θέμα. Σύμφωνα με μια έρευνα, τα φλέγοντα θέματα μπορούν να χωριστούν στις εξής τέσσερις κατηγορίες:

### 2.2.1 Ειδήσεις

**Ειδήσεις:** οι χρήστες που μοιράζονται έκτακτες ειδήσεις εξυψώνουν αμέσως ένα φλέγων θέμα. Έχει αποδειχθεί ότι μερικές φορές τα νέα αναρτούνται στο Twitter πριν από άλλα μέσα ενημέρωσης στο διαδίκτυο. Έτσι, η ανίχνευση των τρεχόντων θεμάτων που παράγονται από ειδήσεις μπορεί να είναι εξαιρετικά ενδιαφέρουσα για την ανακαλύψη έκτακτων ειδήσεων. Η έγκαιρη ανίχνευση ειδήσεων μπορεί επίσης να βοηθήσει την παροχή υπηρεσιών επιμέλειας ειδήσεων και να ενημερώσει τις μηχανές αναζήτησης σχετικά με τους δείκτες που συνδέονται με το θέμα.

## 2.2.2 Τρέχοντα Γεγονότα

**Τρέχοντα γεγονότα:** οι χρήστες που σχολιάζουν ένα γεγονός που πραγματοποιείται επί του παρόντος αξιοποιούν ένα φλέγων θέμα σε πολλές περιπτώσεις. Οι χρήστες τείνουν να αναρτούν tweets σχετικά με ένα γεγονός που ακολουθούν (π.χ. ένα παιχνίδι ποδοσφαίρου ή μια τηλεοπτική εκπομπή) ή παρακολουθούν (π.χ. ένα φεστιβάλ ή μια διάσκεψη). Η ανίχνευση αυτών των τρεχόντων θεμάτων μπορεί να βοηθήσει στην ανακάλυψη γεγονότων που συμβαίνουν τώρα ή σε γεγονότα για τα οποία οι χρήστες ενδιαφέρονται περισσότερο.

## 2.2.3 Meme

**Meme:** τα memes μπορεί επίσης να γίνουν ένα φλέγων θέμα στο Twitter. Τα Memes είναι ιδέες που διαδίδονται μέσω ενός κοινωνικού δικτύου. Συνήθως, είναι μια ιδέα που μια κοινότητα ή ένας μοναδικός (συνήθως ο επηρεάζων) χρήστης αποφασίζει να ξεκινήσει για να το κάνει να αναπτυχθεί γρήγορα και να το κάνει ιογενές. Η ιδέα πίσω από ένα meme μπορεί να ποικίλει. Μπορεί να είναι η επιθυμία μιας κοινότητας για κάτι να συμβεί, ή μια αστεία ιδέα που πρότεινε κάποιος. Η ανίχνευση αυτών των φλεγόντων θεμάτων είναι χρήσιμη για να μελετήσουμε την ιογένεια των ιδεών στο Διαδίκτυο και για τους χρήστες που αναζητούν αστεία πράγματα.



## 2.2.4 Αναμνηστικά

**Αναμνηστικά:** τα αναμνηστικά tweets μπορούν επίσης να αποτελέσουν ένα φλέγων θέμα. Ένα φλέγων θέμα θεωρείται ως αναμνηστικό θέμα όταν οι χρήστες συγχαίρουν μια διασημότητα για τα γενέθλιά της, μοιράζονται την επέτειο ενός γεγονότος ή είναι μια μέρα μνήμης. Αυτά τα φλέγοντα θέματα μπορούν μερικές φορές να προβλεφθούν, καθώς οι επερχόμενες γιορτές και εορτασμοί μπορούν προηγουμένως να γίνουν γνωστοί. Ωστόσο, η ανίχνευσή τους μπορεί να είναι χρήσιμη για την ανάλυση του τρόπου με τον οποίο δημιουργούνται, του τρόπου με τον οποίο εξαπλώνονται και ποιες απόψεις ή σχόλια αποδίδουν οι χρήστες.

## 2.3 Ο κύκλος ζωής και η δημοτικότητα των φλεγόντων θεμάτων του Twitter

Τα μέσα κοινωνικής δικτύωσης εξελίσσονται με εκρηκτικό ρυθμό, με εκατομμύρια ανθρώπους σε όλο τον κόσμο να δημιουργούν και να μοιράζονται περιεχόμενο σε μια κλίμακα που κανείς δεν θα μπορούσε να φανταστεί πριν από λίγα χρόνια. Αυτή η ευρεία παραγωγή και κατανάλωση περιεχομένου δημιούργησε ένα εξαιρετικά ανταγωνιστικό ηλεκτρονικό περιβάλλον όπου διάφοροι τύποι περιεχομένου συνυπάρχουν λόγω της περιορισμένης προσοχής της κοινότητας των χρηστών. Παρά την φαινομενικά χαοτική τάση με την οποία πραγματοποιούνται όλες αυτές οι αλληλεπιδράσεις, ορισμένα θέματα καταφέρνουν να προσελκύσουν μια υπέρμετρη ποσότητα προσοχής, ανεβάζοντάς τα έτσι στην κορυφή, όσον αφορά τη δημοτικότητα. Μέσω της ορατότητάς τους, αυτά τα δημοφιλή θέματα συμβάλλουν στη συλλογική συνειδητοποίηση των τάσεων και, κατά περιόδους, μπορούν επίσης να επηρεάσουν τη δημόσια ατζέντα της κοινότητας. Προς το παρόν, δεν υπάρχει σαφής εικόνα για το τι οδηγεί αυτά τα θέματα στο να γίνουν εξαιρετικά δημοφιλή, ούτε για το πως κάποια από αυτά παραμένουν στο φως της δημοσιότητας για μεγαλύτερο χρονικό διάστημα από άλλα. Υπάρχουν σημαντικές ενδείξεις ότι μια πτυχή που προκαλεί την αποσύνθεση των θεμάτων με την πάροδο του χρόνου είναι η πρωτοτυπία τους (Wu και Huberman 2007). Ένας άλλος παράγοντας υπεύθυνος για τη φθορά τους είναι ο

ανταγωνιστικός χαρακτήρας του μέσου. Καθώς το περιεχόμενο αρχίζει να διαδίδεται μέσω ενός κοινωνικού δικτύου, μπορεί να συσσωρεύσει τις θέσεις προηγούμενων ενδιαφερόντων θεμάτων και λόγω της περιορισμένης προσοχής των χρηστών σύντομα καθίσταται αόρατο από κάποιο νεότερο περιεχόμενο. Ακόμη ένας άλλος λόγος για τη δημοτικότητα ορισμένων θεμάτων είναι η επιρροή των μελών του δικτύου στη διάδοση του περιεχομένου. Μερικοί χρήστες δημιουργούν περιεχόμενο που αντηχεί πολύ έντονα με τους ακόλουθούς τους, προκαλώντας έτσι τη διάδοση του περιεχομένου και την αύξηση της δημοτικότητάς του (Romero κ.α, 2011). Η πηγή αυτού του περιεχομένου μπορεί να προέρχεται από κλασικά μέσα μαζικής ενημέρωσης ή από χρήστες που δημιουργούν θέματα τα οποία τελικά αποτελούν μέρος των τάσεων και συγκεντρώνουν την προσοχή μεγάλων κοινοτήτων. Και στις δύο περιπτώσεις, το γεγονός ότι ένα μικρό σύνολο θεμάτων εντάσσεται στο τρέχον σύνολο σημαίνει ότι θα προσελκύσουν την προσοχή ενός μεγάλου κοινού για ένα μικρό χρονικό διάστημα, συμβάλλοντας έτσι σε κάποιο βαθμό στην δημόσια ατζέντα. Όταν τα θέματα προέρχονται από τα μέσα μαζικής ενημέρωσης, το κοινωνικό μέσο λειτουργεί ως ένα φίλτρο και ως ένας ενισχυτής από ό,τι παράγουν τα κλασικά μέσα και συμβάλλει έτσι στους μηχανισμούς ρύθμισης της ατζέντας που έχουν μελετηθεί διεξοδικά για περισσότερο από τρεις δεκαετίες (McCombs και Shaw, 1993). Τα κυρίαρχα θέματα, τα οποία εμφανίζονται στον κύριο ιστότοπο, αντιπροσωπεύουν εκείνα τα κομμάτια περιεχομένου που αναδύονται στην επιφάνεια του Twitter εξαιτίας συχνών αναφορών από την κοινότητα.

Ένας σημαντικός λόγος για να μελετήσει κανείς τα φλέγοντα θέματα στο Twitter είναι να καταλάβει γιατί κάποια από αυτά παραμένουν στην επικαιρότητα ενώ άλλα χάνουν τη δημοτικότητά τους πιο γρήγορα. Σύμφωνα με μια μελέτη σχετικά με το γενικό πρότυπο συμπεριφοράς των θεμάτων στο Twitter, παρουσιάζεται η διάρκεια ζωής τους στο πέρας του χρόνου. Περίπου το 34% των θεμάτων εμφανίζονται σε περισσότερες από μία ροές tweets. Αυτό σημαίνει ότι η δημοτικότητα των θεμάτων θα μειωθεί για μια ορισμένη χρονική περίοδο προτού αρχίσει ξανά η άνοδός της. Αυτό θα μπορούσε να οφείλεται στο γεγονός ότι εμπλέκονται οι διαφορετικές χρονικές ζώνες. Για παράδειγμα, εάν ένα θέμα αποτελεί μία είδηση για τους αναγνώστες της Βόρειας Αμερικής, μπορεί να εμφανιστεί μια τάση στην Ανατολική ζώνη ώρας και 3 ώρες αργότερα στη ζώνη ώρας του Ειρηνικού. Ομοίως, μια τάση μπορεί να αυξηθεί το επόμενο πρωί αν αποτελούσε μια τάση το προηγούμενο βράδυ, όταν οι περισσότεροι χρήστες συνδεθούν ξανά στους λογαριασμούς τους μετά το πέρας της νύχτας. Δεδομένου ότι πολλά θέματα δεν εμφανίζονται συνεχώς, μπορεί να παρατηρηθεί ότι πρόκειται για νόμο του ισχυρού που σημαίνει ότι οι περισσότερες ροές θεμάτων είναι σύντομες και μερικά θέματα διαρκούν για πολύ μεγάλο χρονικό διάστημα. Αυτό θα μπορούσε να οφείλεται στο γεγονός ότι υπάρχουν πολλά θέματα που ανταγωνίζονται μεταξύ τους για την προσοχή του κοινού. Έτσι, τα θέματα που φτάνουν στην κορυφή (της λίστας τάσεων) διαρκούν για μικρό χρονικό διάστημα.

## 2.4 Τα γεγονότα και η δημοτικότητα τους στο Twitter

Οι ιστότοποι μέσω κοινωνικής δικτύωσης όπως το Twitter, το Facebook και το YouTube έχουν αναδειχθεί ως ισχυρό μέσο επικοινωνίας για τα άτομα που αναζητούν, να μοιράζονται και να ανταλλάσσουν πληροφορίες σχετικά με μια μεγάλη ποικιλία πραγματικών γεγονότων. Αυτά τα γεγονότα κυμαίνονται από δημοφιλείς, ευρέως γνωστές, (για παράδειγμα, μια συναυλία από μιας δημοφιλούς μουσικής μπάντας) ως σε μικρότερες κλίμακες, τοπικές εκδηλώσεις, (για παράδειγμα, μια τοπική κοινωνική συγκέντρωση, μια διαμαρτυρία ή ένα ατύχημα). Τα σύντομα μηνύματα που δημοσιεύονται σε ιστότοπους κοινωνικής δικτύωσης όπως το Twitter, μπορούν συνήθως να αντικατοπτρίζουν αυτά τα γεγονότα καθώς συμβαίνουν. Για το λόγο αυτό, το περιεχόμενο αυτών των ιστότοπων μέσω κοινωνικής δικτύωσης είναι ιδιαίτερα χρήσιμο για την ταυτοποίηση των γεγονότων του πραγματικού κόσμου σε πραγματικό χρόνο και των συναφών με τους χρήστες μηνυμάτων. Τα μηνύματα στο Twitter αντικατοπτρίζουν χρήσιμες πληροφορίες γεγονότων για ποικίλα γεγονότα διαφόρων τύπων και κλίμακας. Αυτά τα μηνύματα που αναφέρονται σε γεγονότα μπορούν να παρέχουν ένα σύνολο μοναδικών προοπτικών, ανεξάρτητα από τον τύπο του συμβάντος, που αντανακλά τις απόψεις των χρηστών οι οποίοι ενδιαφέρονται ή συμμετέχουν σε μια εκδήλωση. Ειδικότερα, όσον αφορά τις απρογραμμάτιστες εκδηλώσεις, όπως για παράδειγμα τις

διαμαρτυρίες του Ιράν και τους σεισμούς, οι χρήστες του Twitter ανάρτησαν μερικές φορές νέα tweets πριν από τα παραδοσιακά μέσα ενημέρωσης. Ακόμη και σχετικά με προγραμματισμένες εκδηλώσεις όπως το συνέδριο των Apple Developers για το έτος 2010, οι χρήστες του Twitter συχνά δημοσιεύουν μηνύματα εν αναμονή της εκδήλωσης.

## Κεφάλαιο 3

### Ο αλγόριθμος της πτυχιακής εργασίας

#### 3.1 Ο στόχος του αλγορίθμου

Ο συγκεκριμένος αλγόριθμος αναπτύχθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας και για τον εμπλουτισμό της λειτουργικότητας του εργαλείου για δημοσιογράφους INJECT (σύνδεσμος ιστοτόπου: <https://injectproject.eu>). Παρ'όλα αυτά μπορεί εύκολα να χρησιμοποιηθεί και σε οποιαδήποτε web και mobile εφαρμογή μιας και ως είσοδο απαιτεί μόνο τον όρο προς αναζήτηση και τα απαραίτητα στοιχεία ταυτοποίησης για το Twitter API.

Συγκεκριμένα, δέχεται ως είσοδο μια λέξη που αποτελεί το φλέγων θέμα (trending topic) που θέλουμε να αναζητήσουμε και μετατρέπεται σε όρο τύπου hashtag (π.χ #term). Έπειτα, υπολογίζεται η δημοτικότητα του όρου που έχει αναζητηθεί και παρατίθενται διάφορα στοιχεία όπως το πλήθος των retweets (αναδημοσιεύσεις), comments (σχόλια) και favorites (ενδιαφερόμενοι χρήστες) και άλλα.

## 3.2 Ανάλυση αλγορίθμου

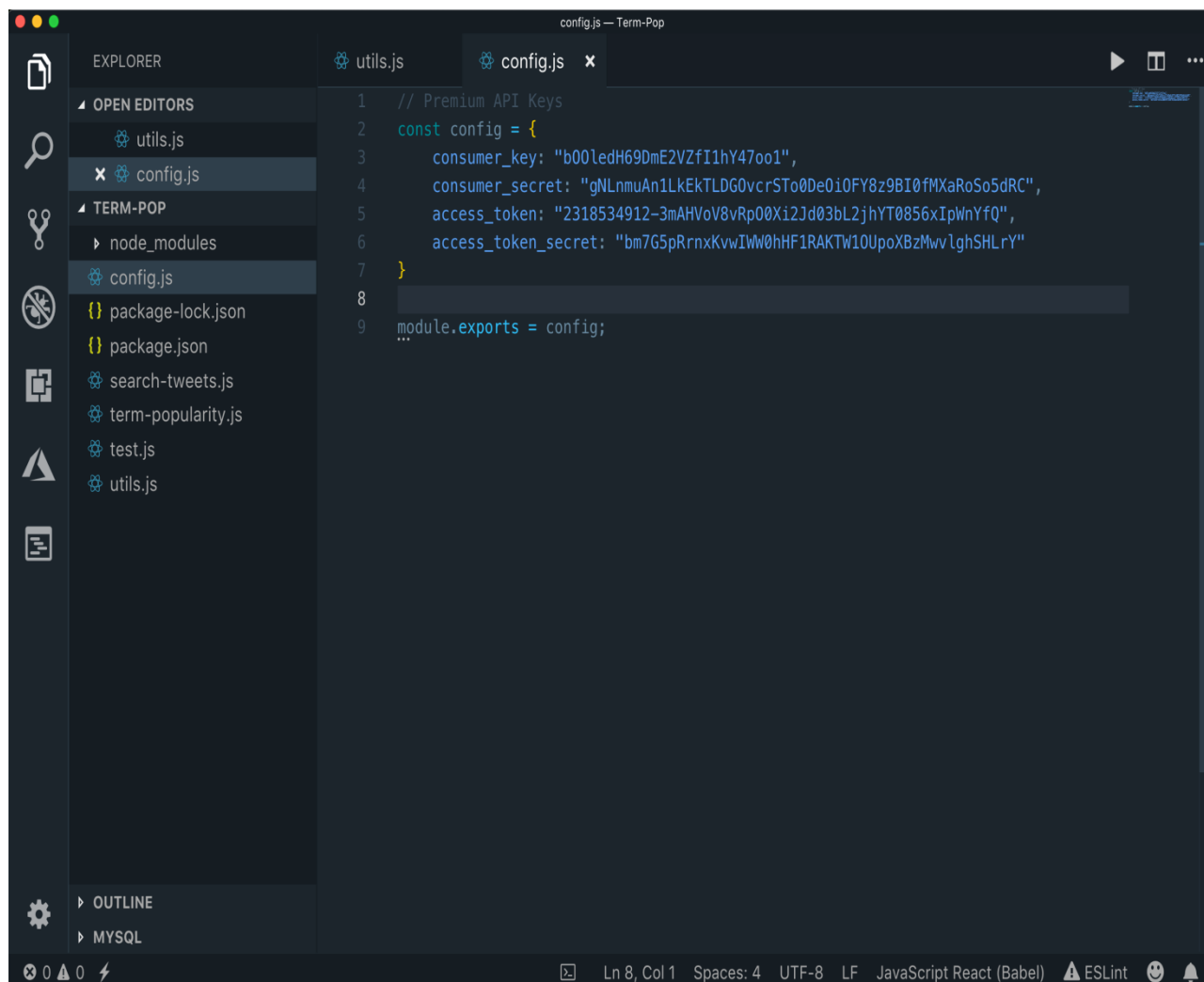
Παρακάτω περιγράφονται αναλυτικά όλες οι λειτουργίες του αλγορίθμου της πτυχιακής εργασίας καθώς αναλύονται όλα τα αρχεία που τον απαρτίζουν ένα προς ένα καθώς οι αντίστοιχες συναρτήσεις τους.

### 3.2.1 Αρχείο “config.js”

Το αρχείο “config.js” περιέχει απαραίτητα στοιχεία για την χρήση του Twitter API. Τα στοιχεία που μας χρειάστηκαν είναι τα εξής: consumer key, consumer secret, access token και access token secret και τα έχουμε αποθηκεύσει μέσα σε ένα object με όνομα config. Με τη βοήθεια αυτών των στοιχείων ένας χρήστης του Twitter API μπορεί να έχει διαφορετικά επίπεδα πρόσβασης στα δεδομένα του Twitter API: Standard, Premium και Enterprise. Εμείς έχουμε “ξεκλειδώσει” το δεύτερο επίπεδο πρόσβασης.

Για περισσότερες πληροφορίες σχετικά με την διαδικασία της ταυτοποίησης για το Twitter API δείτε την ενότητα **3.2.6 Διαδικασία ταυτοποίησης στο Twitter API**.





Εικόνα 1: Αρχείο “config.js”

### 3.2.2 Αρχείο “utils.js”

Στο αρχείο “utils.js” έχουμε ορίσει τις συναρτήσεις `findPopularity`, `sumOfArray`, `getOldDate` και `getNewDate` οι οποίες μας βοηθούν στην εύρεση tweets και στον υπολογισμό της δημοτικότητας ενός όρου στο Twitter και χρησιμοποιούνται στο αρχείο `term-popularity.js`. Για τις συναρτήσεις `getOldDate` και `getNewDate` χρησιμοποιήσαμε την βιβλιοθήκη: “moment.js” η οποία βοηθάει στην επεξεργασία `Date` object.

Παρακάτω θα αναλύσουμε τον κώδικα των συναρτήσεων `findPopularity`, `sumOfArray`, `getOldDate` και `getNewDate`.

#### **Συνάρτηση *findPopularity*:**

Παίρνει ως είσοδο τον συνολικό αριθμό των favorites, τον συνολικό αριθμό των retweets και τον συνολικό αριθμό των tweets που περιέχουν έναν όρο και επιστρέφει έναν βαθμό ο οποίος δείχνει τη δημοτικότητα του όρου. Δίνεται ίδια βαρύτητα και στις τρεις εισόδους για τον υπολογισμό του βαθμού.

#### **Συνάρτηση *sumOfArray*:**

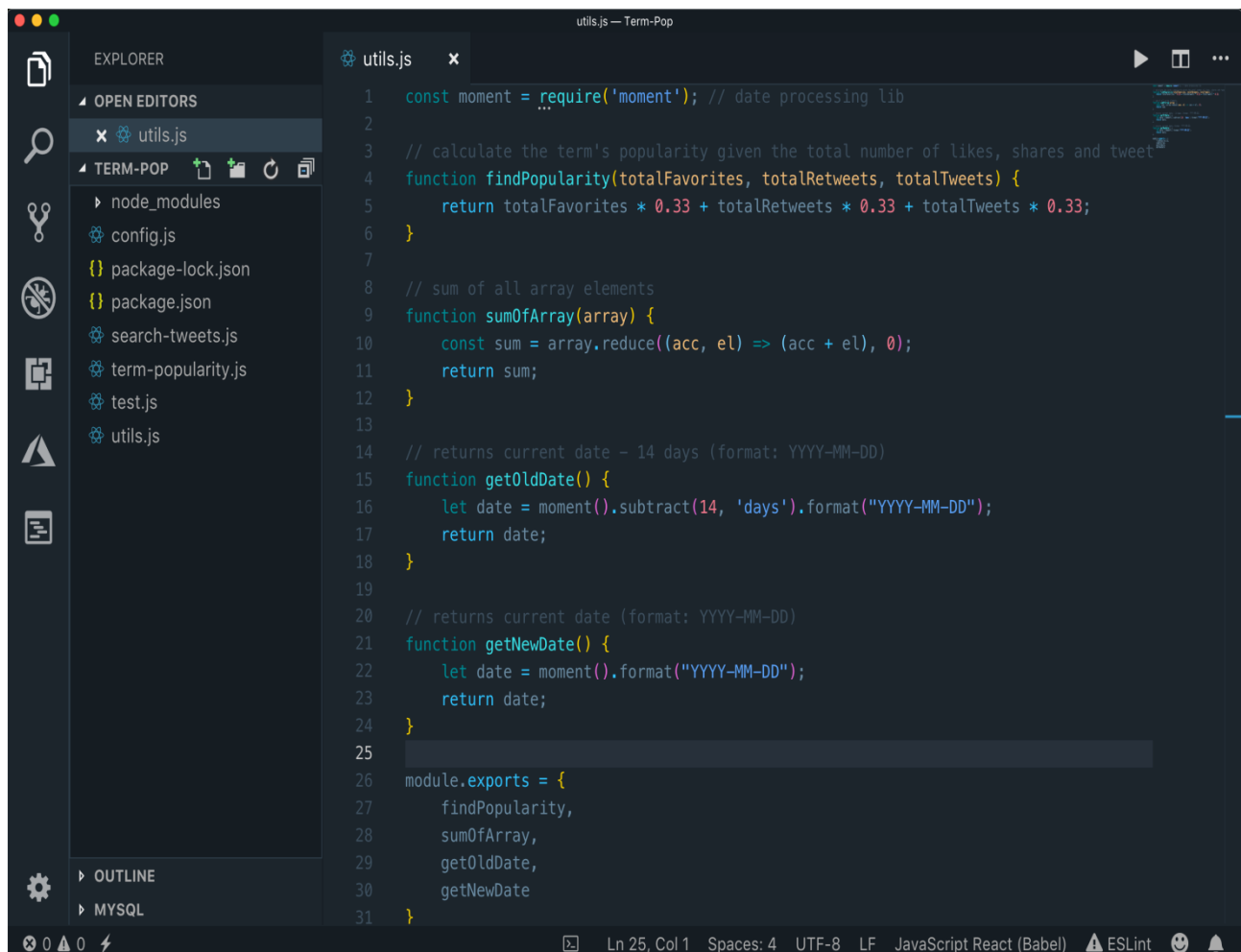
Παίρνει ως είσοδο έναν πίνακα ο οποίος περιέχει μόνο αριθμούς και υπολογίζει το άθροισμα των στοιχείων του πίνακα.

#### **Συνάρτηση *getNewDate*:**

Βρήσκει την τρέχουσα ημερομηνία την μετατρέπει στην εξής μορφή: “YYYY-MM-DD” και την επιστρέφει.

## Συνάρτηση *getOldDate*:

Βρήσκει την ημερομηνία πρίν δεκατέσσερις μέρες την μετατρέπει στην μορφή: “YYYY-MM-DD” και την επιστρέφει.



```
1  const moment = require('moment'); // date processing lib
2
3  // calculate the term's popularity given the total number of likes, shares and tweet
4  function findPopularity(totalFavorites, totalRetweets, totalTweets) {
5      return totalFavorites * 0.33 + totalRetweets * 0.33 + totalTweets * 0.33;
6  }
7
8  // sum of all array elements
9  function sumOfArray(array) {
10     const sum = array.reduce((acc, el) => (acc + el), 0);
11     return sum;
12 }
13
14 // returns current date - 14 days (format: YYYY-MM-DD)
15 function getOldDate() {
16     let date = moment().subtract(14, 'days').format("YYYY-MM-DD");
17     return date;
18 }
19
20 // returns current date (format: YYYY-MM-DD)
21 function getNewDate() {
22     let date = moment().format("YYYY-MM-DD");
23     return date;
24 }
25
26 module.exports = {
27     findPopularity,
28     sumOfArray,
29     getOldDate,
30     getNewDate
31 }
```

Εικόνα 2: Αρχείο “utils.js”

### 3.2.3 Αρχείο “search-tweets.js”

Έχουμε ορίσει την συνάρτηση *searchTweets* η οποία παίρνει ως είσοδο το config object, μια string συνθήκη αναζήτησης στο Twitter API (query) και μια συνάρτηση που θα επεξεργαστεί τα tweets (callback). Ο σκοπός της είναι να βρεί tweets που να ικανοποιούν την συνθήκη αναζήτησης και να τα περάσει στην συνάρτηση callback.

Η string συνθήκη που θα δωθεί στην συνάρτηση θα περιέχει τον όρο που θέλουμε να βρούμε (στη μορφή #όρος), την ημερομηνία από την συνάρτηση *getOldDate* και την ημερομηνία από την συνάρτηση *getNewDate* για να βρούμε τα tweets των τελευταίων δεκατεσσάρων ημερών.

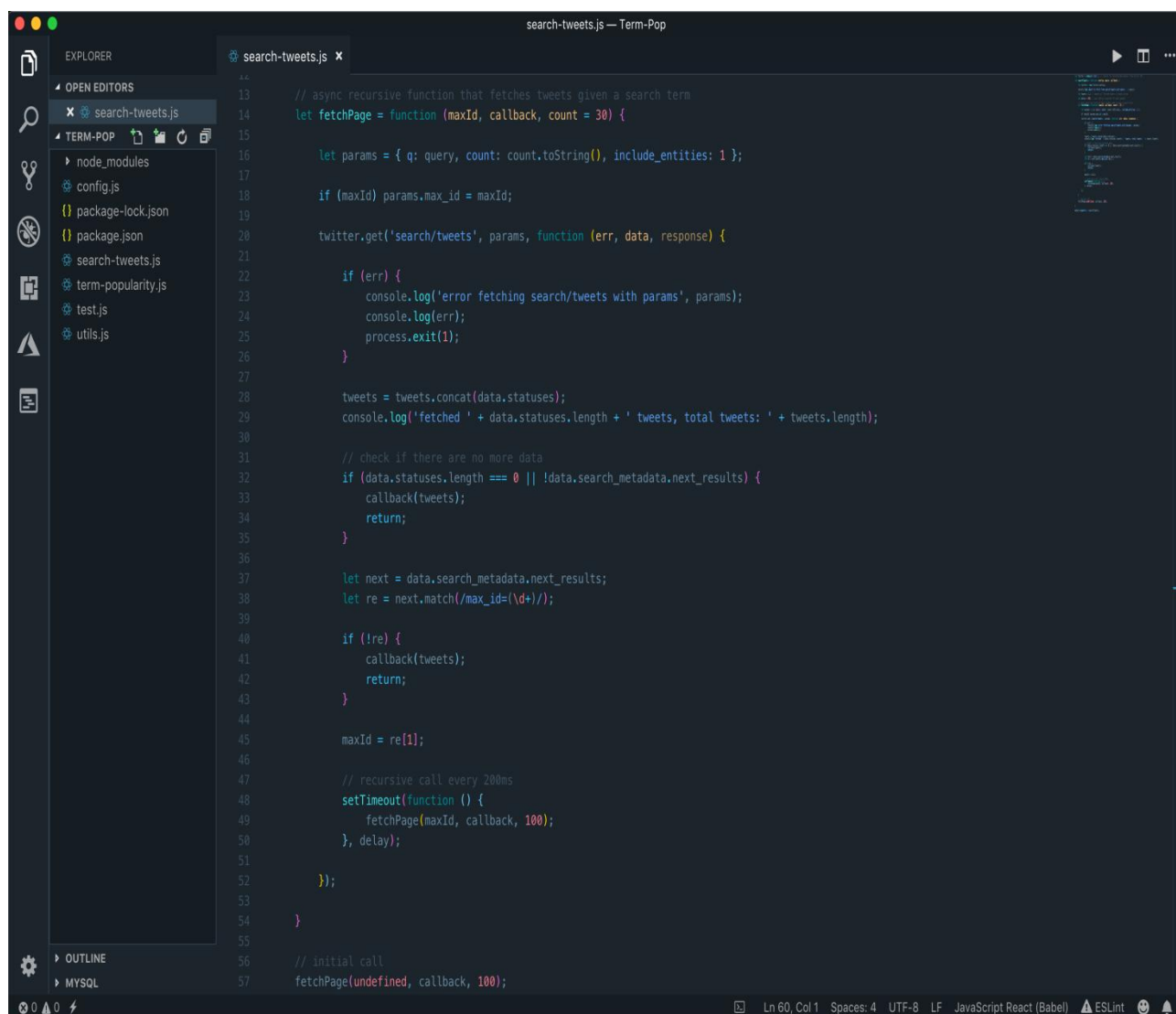
Μέσα στη συνάρτηση *searchTweets* έχουμε ορίσει μια άλλη ασύγχρονη και αναδρομική συνάρτηση την *fetchPage* η οποία κάθε 200 milliseconds παίρνει τα επόμενα *N* tweets (αν υπάρχουν) από τα tweets που έχουμε βρεί μέσω του Twitter API.

Η συνάρτηση *fetchPage* παίρνει ως είσοδο ένα id το οποίο δείχνει τα επόμενα *N* tweets (μπορεί να είναι ίσο με null/ undefined), μια συνάρτηση που θα επεξεργάζεται τα τρέχοντα *N* tweets και τον αριθμό των tweets που θελούμε να πέρνουμε ανά 200 milliseconds.

Για την εύκολη εύρεση των tweets χρησιμοποιήσαμε την βιβλιοθήκη *twit*. Αυτή η βιβλιοθήκη χρησιμοποιεί το config object για να επικοινωνήσει με το Twitter API.

```
1 let Twitter = require('twit'); // library for fetching data easier from Twitter API
2
3 let searchTweets = function (config, query, callback) {
4
5     let twitter = new Twitter(config);
6
7     console.log('about to fetch from search/tweets with query: ' + query);
8
9     let tweets = []; // append all fetched tweets in tweets array
10
11     let delay = 200; // wait 200 milliseconds for each request
12
13     // async recursive function that fetches tweets given a search term
14     let fetchPage = function (maxId, callback, count = 30) {~
15         // ...
16     }
17
18     // initial call
19     fetchPage(undefined, callback, 100);
20
21 }
22
23 module.exports = searchTweets;
```

Εικόνα 3: Αρχείο “search-tweets.js” (α)



```
13 // async recursive function that fetches tweets given a search term
14 let fetchPage = function (maxId, callback, count = 30) {
15
16     let params = { q: query, count: count.toString(), include_entities: 1 };
17
18     if (maxId) params.max_id = maxId;
19
20     twitter.get('search/tweets', params, function (err, data, response) {
21
22         if (err) {
23             console.log('error fetching search/tweets with params', params);
24             console.log(err);
25             process.exit(1);
26         }
27
28         tweets = tweets.concat(data.statuses);
29         console.log('fetched ' + data.statuses.length + ' tweets, total tweets: ' + tweets.length);
30
31         // check if there are no more data
32         if (data.statuses.length === 0 || !data.search_metadata.next_results) {
33             callback(tweets);
34             return;
35         }
36
37         let next = data.search_metadata.next_results;
38         let re = next.match(/max_id=(\d+)/);
39
40         if (!re) {
41             callback(tweets);
42             return;
43         }
44
45         maxId = re[1];
46
47         // recursive call every 200ms
48         setTimeout(function () {
49             fetchPage(maxId, callback, 100);
50         }, delay);
51     });
52 }
53
54 // initial call
55 fetchPage(undefined, callback, 100);
```

Εικόνα 4: Αρχείο “search-tweets.js” (β)

### 3.2.4 Αρχείο “term-popularity.js”

Στο αρχείο *term-popularity.js* έχουμε ορίσει την συνάρτηση *termPopularity* η οποία βρίσκει *tweets* των τελευταίων δεκατεσσάρων ημερών που περιέχουν ένα συγκεκριμένο όρο, υπολογίζει τη δημοτικότητα του μέσω του υπολογισμού ενός βαθμού μεταξύ 0 και 10 και επιστρέφει τον βαθμό αυτό μαζί με τον συνολικό αριθμό *tweets*, *retweets* και *favorites*.

Η συνάρτηση *termPopularity* παίρνει ως είσοδο έναν όρο (της μορφής #όρος) και μία συνάρτηση η οποία θα επεξεργαστεί τον βαθμό δημοτικότητας του όρου, τον συνολικό αριθμό *tweets*, *retweets* και *favorites*.

Μέσα στη συνάρτηση *termPopularity* χρησιμοποιούμε την συνάρτηση *searchTweets* από το αρχείο *search-tweets.js* το *config object* από το αρχείο *config.js* και τις συναρτήσεις *findPopularity*, *sumOfArray*, *getOldDate* και *getNewDate* από το αρχείο *utils.js*.

Μέσω των συναρτήσεων *getOldDate* και *getNewDate* του όρου που δίνεται ως είσοδος στη συνάρτηση *termPopularity* δημιουργούμε το *string* που θα περιέχει τη κατάλληλη συνθήκη αναζήτησης.

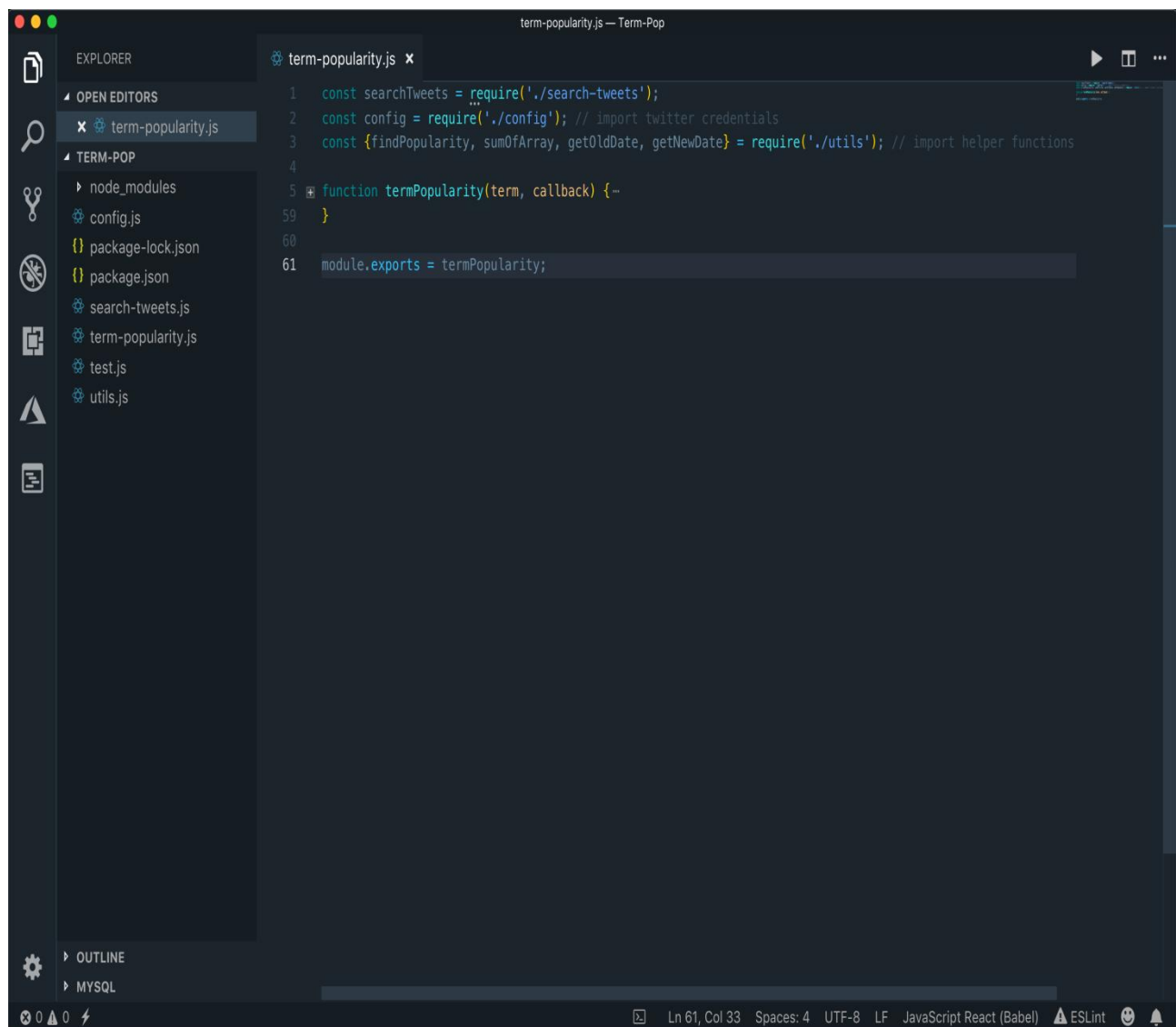
Μέσω της συνάρτησης *searchTweets* βρίσκουμε τα *tweets* των τελευταίων δεκατεσσάρων ημερών που περιέχουν τον όρο που δώσαμε στη συνάρτηση *termPopularity*.

Όταν βρούμε τα *tweets* των τελευταίων δεκατεσσάρων ημερών φιλτράρουμε τα *tweets* που περιέχουν τον όρο που δώσαμε στη συνάρτηση *termPopularity* και μέσω αυτών υπολογίζουμε τον συνολικό αριθμό *tweets*, *retweets* και *favorites*.

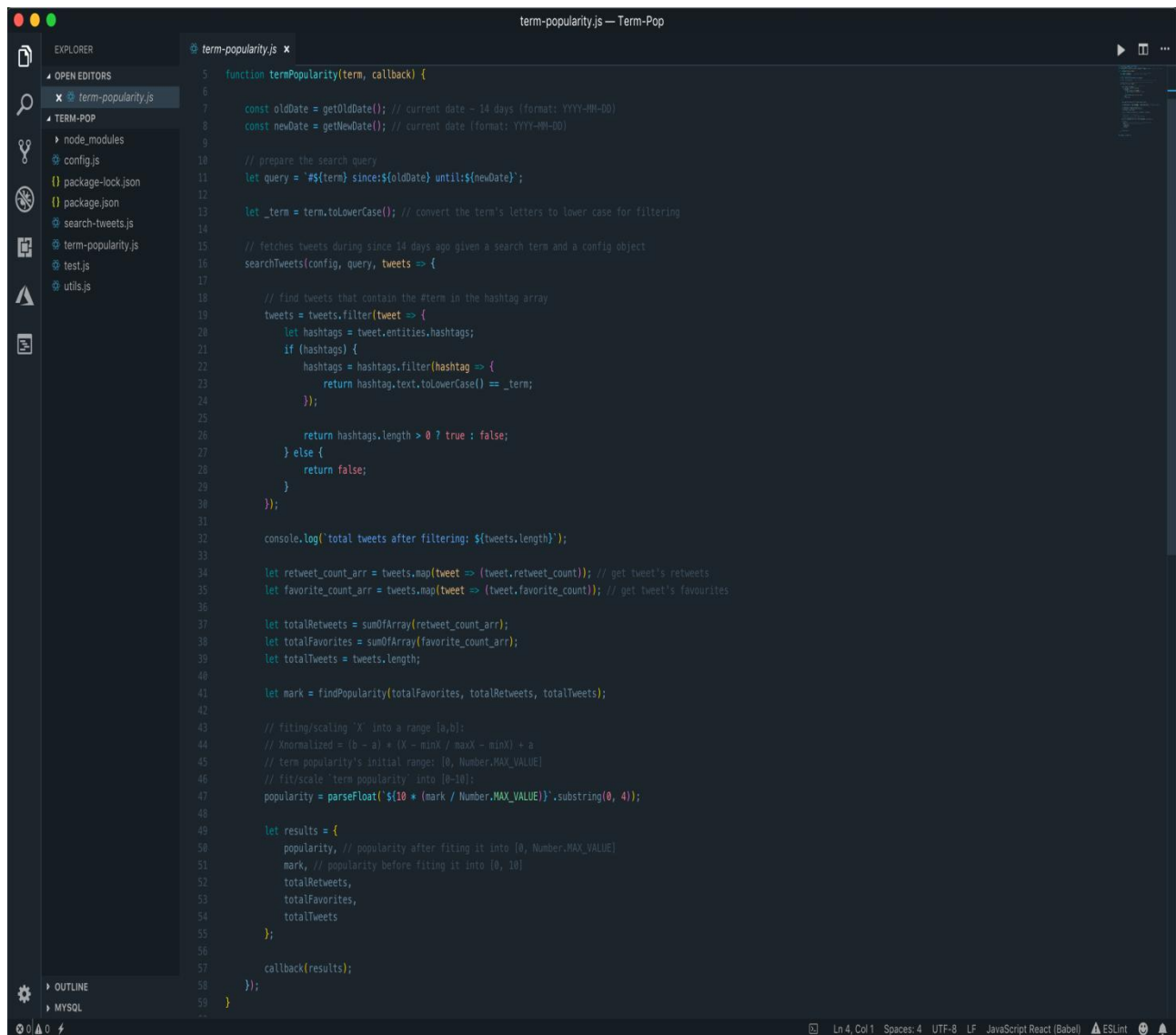
Μέσω του υπολογισμού των συνολικών αριθμών *tweets*, *retweets* και *favorites* μπορούμε μετά να βρούμε τον βαθμό που θα δώσουμε στον όρο (ο βαθμός αναπαριστά τη δημοτικότητα του όρου). Ο βαθμός αυτός αρχικά ορίζεται στο διάστημα [0 - Μέγιστος Ακέραιος Αριθμός] και τον μετατρέπουμε σε έναν αριθμό που θα ανήκει στο διάστημα [0-10] για ευκολότερη αξιολόγηση της δημοτικότητάς του όρου.

Όταν υπολογίσουμε και την δημοτικότητα του όρου καλούμε τη συνάρτηση που δώθηκε ως δεύτερη παράμετρος στη συνάρτηση *termPopularity* και δίνουμε ως είσοδο την αρχική δημοτικότητα του όρου την επεξεργασμένη δημοτικότητα του όρου και τον συνολικό αριθμό *tweets*, *retweets* και *favorites*.





Εικόνα 5: Αρχείο “term-popularity.js” (α)



```
5 function termPopularity(term, callback) {
6
7   const oldDate = getOldDate(); // current date - 14 days (format: YYYY-MM-DD)
8   const newDate = getNewDate(); // current date (format: YYYY-MM-DD)
9
10  // prepare the search query
11  let query = `#${term} since:${oldDate} until:${newDate}`;
12
13  let _term = term.toLowerCase(); // convert the term's letters to lower case for filtering
14
15  // fetches tweets during since 14 days ago given a search term and a config object
16  searchTweets(config, query, tweets => {
17
18    // find tweets that contain the #term in the hashtag array
19    tweets = tweets.filter(tweet => {
20      let hashtags = tweet.entities.hashtags;
21      if (hashtags) {
22        hashtags = hashtags.filter(hashtag => {
23          return hashtag.text.toLowerCase() == _term;
24        });
25
26        return hashtags.length > 0 ? true : false;
27      } else {
28        return false;
29      }
30    });
31
32    console.log('total tweets after filtering: ${tweets.length}');
33
34    let retweet_count_arr = tweets.map(tweet => (tweet.retweet_count)); // get tweet's retweets
35    let favorite_count_arr = tweets.map(tweet => (tweet.favorite_count)); // get tweet's favourites
36
37    let totalRetweets = sumOfArray(retweet_count_arr);
38    let totalFavorites = sumOfArray(favorite_count_arr);
39    let totalTweets = tweets.length;
40
41    let mark = findPopularity(totalFavorites, totalRetweets, totalTweets);
42
43    // fitting/scaling 'X' into a range [a,b]:
44    // Xnormalized = (b - a) * (X - minX / maxX - minX) + a
45    // term popularity's initial range: [0, Number.MAX_VALUE]
46    // fit/scale 'term popularity' into [0-10]:
47    popularity = parseFloat(`${10 * (mark / Number.MAX_VALUE)}`.substring(0, 4));
48
49    let results = {
50      popularity, // popularity after fitting it into [0, Number.MAX_VALUE]
51      mark, // popularity before fitting it into [0, 10]
52      totalRetweets,
53      totalFavorites,
54      totalTweets
55    };
56
57    callback(results);
58  });
59 }
```

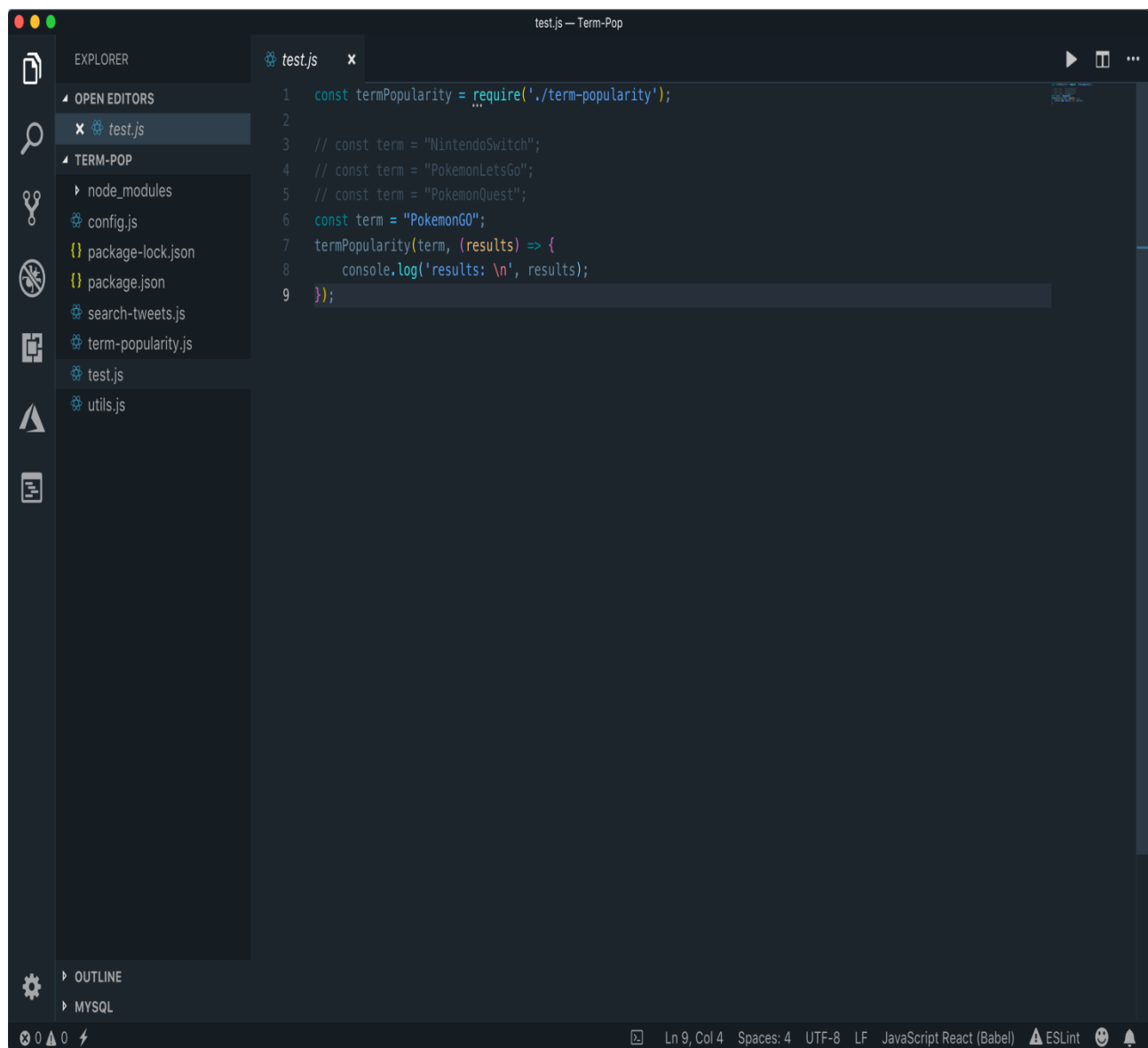
Εικόνα 6: Αρχείο “term-popularity.js” (β)

### 3.2.5 Αρχείο “test.js”

Σε αυτό το αρχείο δοκιμάζουμε τη συνάρτηση *termPopularity* που έχει οριστεί στο αρχείο *term-popularity.js*.

Συγκεκριμένα έχουμε δημιουργήσει τέσσερα παραδείγματα προς χρήση. Σε κάθε παράδειγμά έχουμε δημιουργήσει διαφορετικό όρο. Όλοι σχεδόν οι όροι εκτός από έναν έχουν τοποθετηθεί σε σχόλιο.

Όταν εκτελεστεί το αρχείο *test.js* και κληθεί η συνάρτηση *termPopularity* θα τυπωθούν στην οθόνη η αρχική δημοτικότητα του όρου, η επεξεργασμένη δημοτικότητα του όρου, ο συνολικός αριθμός *tweets*, *retweets* και *favourites* ο αριθμός των *tweets* που βρήκε η συνάρτηση *searchTweets* και τον τρέχων αριθμό *tweets* που πήρε η συνάρτηση *searchTweets* από το *Twitter API*.



Εικόνα 7: Αρχείο “test.js”

### 3.2.6 Διαδικασία ταυτοποίησης στο Twitter API

Το Open Authentication (OAuth) είναι ένα ανοικτό πρότυπο για τον έλεγχο ταυτότητας, το οποίο υιοθετήθηκε από το Twitter για να παρέχει πρόσβαση σε προστατευμένες πληροφορίες. Οι κωδικοί πρόσβασης είναι ιδιαίτερα ευάλωτοι για κλοπή και το OAuth παρέχει μια ασφαλέστερη εναλλακτική λύση σε σχέση με τις παραδοσιακές προσεγγίσεις ελέγχου ταυτότητας, χρησιμοποιώντας μια τριφασική «χειραψία». Επίσης, βελτιώνει την εμπιστοσύνη του χρήστη στην εφαρμογή καθώς ο κωδικός πρόσβασής του για το λογαριασμό του στο Twitter δεν μοιράζεται ποτέ με εφαρμογές τρίτων κατασκευαστών. Ο έλεγχος ταυτότητας των αιτήσεων API στο Twitter πραγματοποιείται χρησιμοποιώντας το OAuth. Τα API του Twitter μπορούν να είναι προσβάσιμα μόνο από εφαρμογές. Παρακάτω περιγράφονται λεπτομερώς τα βήματα για την πραγματοποίηση μιας κλήσης API από μια εφαρμογή Twitter με χρήση του OAuth:

- Οι αιτήσεις είναι επίσης γνωστές ως καταναλωτές και απαιτείται από όλες τις εφαρμογές να έχουν πραγματοποιήσει εγγραφή στο Twitter. Μέσω αυτής της διαδικασίας, εκδίδεται για την εφαρμογή ένα δημόσιο κλειδί για τον καταναλωτή και ένα μυστικό που η εφαρμογή πρέπει να χρησιμοποιήσει για να πιστοποιήσει την ταυτότητά της στο Twitter.
- Η εφαρμογή χρησιμοποιεί το κλειδί του πελάτη και το μυστικό κλειδί της για να δημιουργήσει ένα μοναδικό σύνδεσμο “Twitterlink” στο οποίο ο χρήστης κατευθύνεται για έλεγχο ταυτότητας. Ο χρήστης

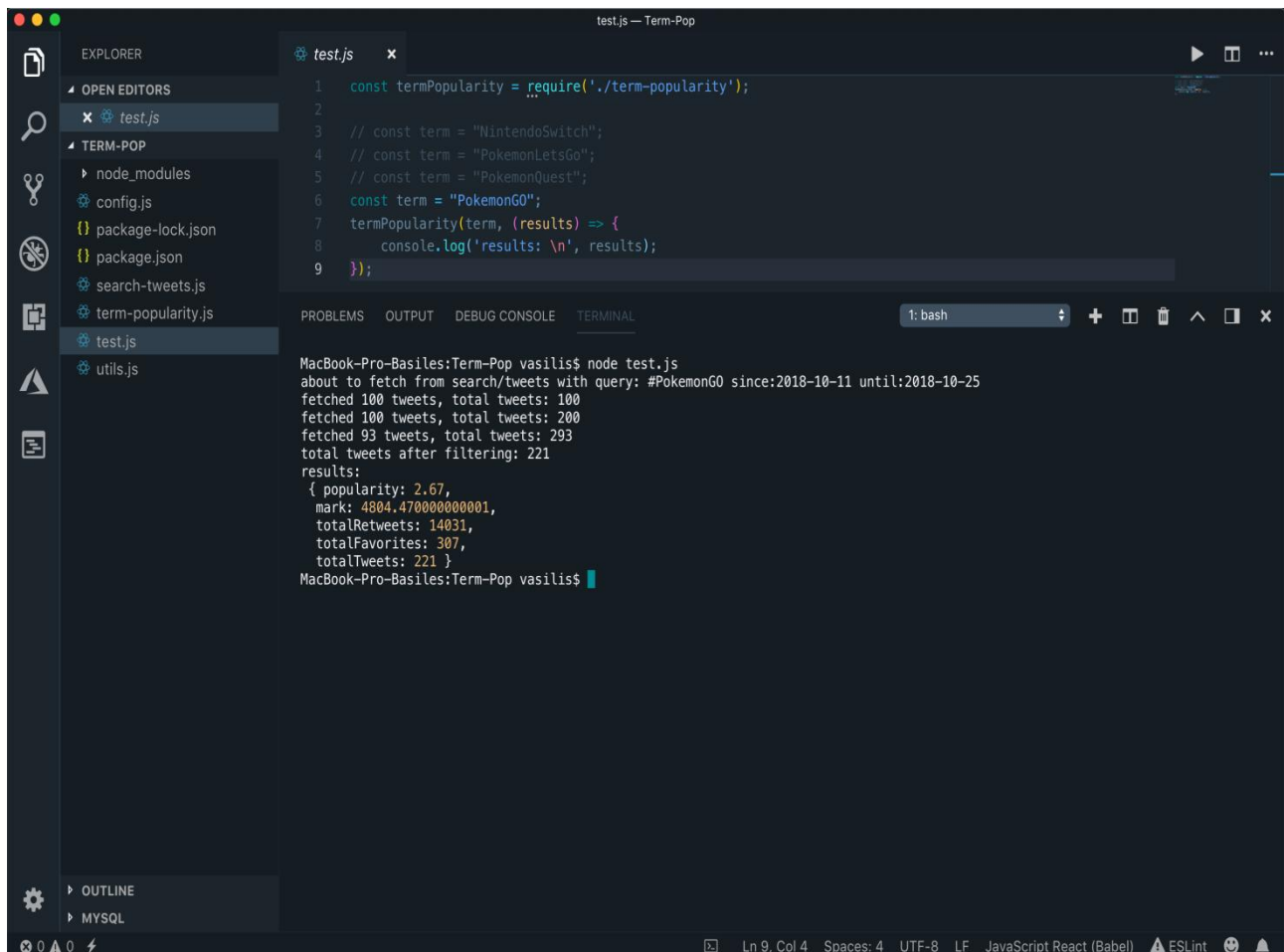
εξουσιοδοτεί την εφαρμογή επαληθεύοντας τον εαυτό του στο Twitter. Έπειτα, το Twitter επαληθεύει την ταυτότητα του χρήστη και εκδίδει έναν επαληθευτή OAuth που επίσης ονομάζεται PIN.

- Χρησιμοποιώντας τα "Access Token" και "Access Secret", η εφαρμογή επιβεβαιώνει τον χρήστη στο Twitter και εκδίδει κλήσεις API για λογαριασμό του χρήστη.
- Ο χρήστης παρέχει αυτό το PIN στην εφαρμογή. Η εφαρμογή χρησιμοποιεί αυτό το PIN για να ζητήσει ένα "Access Token" και "Access Secret" μοναδικό για το χρήστη. Τα "Token Access" και "Access Secret" για κάθε χρήστη και δεν αλλάζουν και ούτε μπορούν να αποθηκευτούν από την εφαρμογή για μελλοντικά αιτήματα. Έτσι, αυτή η διαδικασία πρέπει να εκτελείται μόνο μία φορά και μπορεί εύκολα να επιτευχθεί με τη χρήση της μεθόδου Get Key Access Key Secret.

### 3.3 Παραδείγματα εκτέλεσης αλγορίθμου

Παρακάτω παρατίθενται διάφορα παραδείγματα εκτέλεσης του αλγορίθμου της πτυχιακής εργασίας προκειμένου να παρουσιαστεί ο τρόπος λειτουργίας του.

#### Παράδειγμα 1ο: #PokemonGO

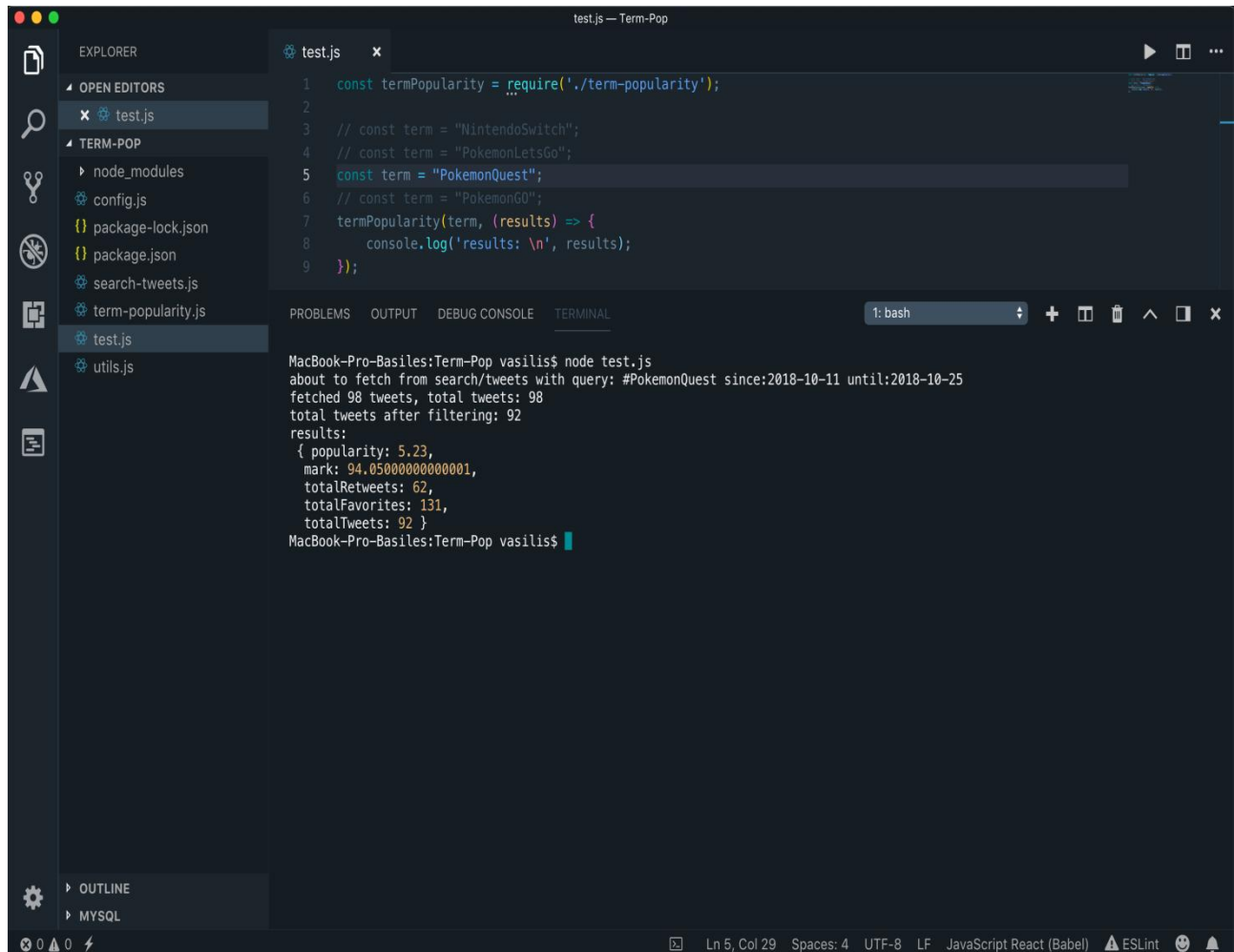


```
test.js — Term-Pop
1  const termPopularity = require('./term-popularity');
2
3  // const term = "NintendoSwitch";
4  // const term = "PokemonLetsGo";
5  // const term = "PokemonQuest";
6  const term = "PokemonGO";
7  termPopularity(term, (results) => {
8    console.log('results: \n', results);
9  });
```

```
MacBook-Pro-Basiles:Term-Pop vasilis$ node test.js
about to fetch from search/tweets with query: #PokemonGO since:2018-10-11 until:2018-10-25
fetched 100 tweets, total tweets: 100
fetched 100 tweets, total tweets: 200
fetched 93 tweets, total tweets: 293
total tweets after filtering: 221
results:
{ popularity: 2.67,
  mark: 4804.470000000001,
  totalRetweets: 14031,
  totalFavorites: 307,
  totalTweets: 221 }
```

Εικόνα 8: Παράδειγμα αναζήτησης “#PokemonGO”

## Παράδειγμα 2ο: #PokemonQuest



The screenshot shows a VS Code editor window with a file named `test.js` open. The file contains the following JavaScript code:

```
1 const termPopularity = require('./term-popularity');
2
3 // const term = "NintendoSwitch";
4 // const term = "PokemonLetsGo";
5 const term = "PokemonQuest";
6 // const term = "PokemonGO";
7 termPopularity(term, (results) => {
8   console.log('results: \n', results);
9 });
```

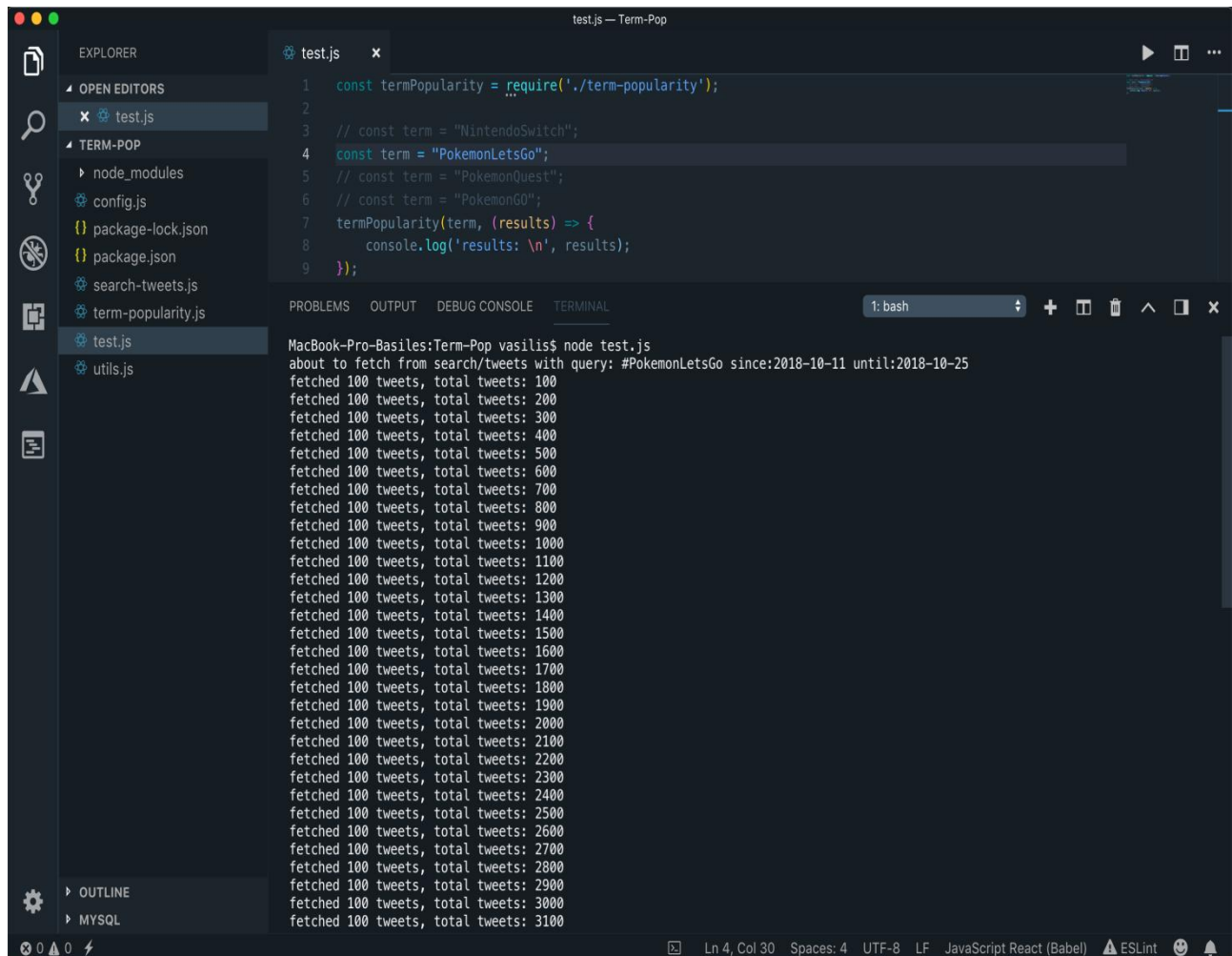
The terminal output shows the execution of `node test.js` on a MacBook-Pro-Basiles. The output is as follows:

```
MacBook-Pro-Basiles:Term-Pop vasilis$ node test.js
about to fetch from search/tweets with query: #PokemonQuest since:2018-10-11 until:2018-10-25
fetched 98 tweets, total tweets: 98
total tweets after filtering: 92
results:
{ popularity: 5.23,
  mark: 94.05000000000001,
  totalRetweets: 62,
  totalFavorites: 131,
  totalTweets: 92 }
MacBook-Pro-Basiles:Term-Pop vasilis$
```

Εικόνα 9: Παράδειγμα αναζήτησης “#PokemonQuest”



## Παράδειγμα 3ο: #PokemonLetsGo



```
test.js — Term-Pop
1  const termPopularity = require('./term-popularity');
2
3  // const term = "NintendoSwitch";
4  const term = "PokemonLetsGo";
5  // const term = "PokemonQuest";
6  // const term = "PokemonGO";
7  termPopularity(term, (results) => {
8    console.log('results: \n', results);
9  });
```

MacBook-Pro-Basiles:Term-Pop vasilis\$ node test.js  
about to fetch from search/tweets with query: #PokemonLetsGo since:2018-10-11 until:2018-10-25  
fetched 100 tweets, total tweets: 100  
fetched 100 tweets, total tweets: 200  
fetched 100 tweets, total tweets: 300  
fetched 100 tweets, total tweets: 400  
fetched 100 tweets, total tweets: 500  
fetched 100 tweets, total tweets: 600  
fetched 100 tweets, total tweets: 700  
fetched 100 tweets, total tweets: 800  
fetched 100 tweets, total tweets: 900  
fetched 100 tweets, total tweets: 1000  
fetched 100 tweets, total tweets: 1100  
fetched 100 tweets, total tweets: 1200  
fetched 100 tweets, total tweets: 1300  
fetched 100 tweets, total tweets: 1400  
fetched 100 tweets, total tweets: 1500  
fetched 100 tweets, total tweets: 1600  
fetched 100 tweets, total tweets: 1700  
fetched 100 tweets, total tweets: 1800  
fetched 100 tweets, total tweets: 1900  
fetched 100 tweets, total tweets: 2000  
fetched 100 tweets, total tweets: 2100  
fetched 100 tweets, total tweets: 2200  
fetched 100 tweets, total tweets: 2300  
fetched 100 tweets, total tweets: 2400  
fetched 100 tweets, total tweets: 2500  
fetched 100 tweets, total tweets: 2600  
fetched 100 tweets, total tweets: 2700  
fetched 100 tweets, total tweets: 2800  
fetched 100 tweets, total tweets: 2900  
fetched 100 tweets, total tweets: 3000  
fetched 100 tweets, total tweets: 3100

Εικόνα 10: Παράδειγμα αναζήτησης “#PokemonLetsGo” (α)

```
test.js — Term-Pop
1  const termPopularity = require('./term-popularity');
2
3  // const term = "NintendoSwitch";
4  const term = "PokemonLetsGo";
5  // const term = "PokemonQuest";
6  // const term = "PokemonGO";
7  termPopularity(term, (results) => {
8    console.log('results: \n', results);
9  });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: bash

```

fetched 100 tweets, total tweets: 3200
fetched 100 tweets, total tweets: 3300
fetched 100 tweets, total tweets: 3400
fetched 100 tweets, total tweets: 3500
fetched 100 tweets, total tweets: 3600
fetched 100 tweets, total tweets: 3700
fetched 100 tweets, total tweets: 3800
fetched 100 tweets, total tweets: 3900
fetched 100 tweets, total tweets: 4000
fetched 100 tweets, total tweets: 4100
fetched 100 tweets, total tweets: 4200
fetched 100 tweets, total tweets: 4300
fetched 100 tweets, total tweets: 4400
fetched 100 tweets, total tweets: 4500
fetched 100 tweets, total tweets: 4600
fetched 100 tweets, total tweets: 4700
fetched 100 tweets, total tweets: 4800
fetched 100 tweets, total tweets: 4900
fetched 100 tweets, total tweets: 5000
fetched 100 tweets, total tweets: 5100
fetched 100 tweets, total tweets: 5200
fetched 100 tweets, total tweets: 5300
fetched 100 tweets, total tweets: 5400
fetched 100 tweets, total tweets: 5500
fetched 100 tweets, total tweets: 5600
fetched 100 tweets, total tweets: 5700
fetched 100 tweets, total tweets: 5800
fetched 100 tweets, total tweets: 5900
fetched 100 tweets, total tweets: 6000
fetched 100 tweets, total tweets: 6100
fetched 86 tweets, total tweets: 6186
total tweets after filtering: 4091
results:
```

Εικόνα 11: Παράδειγμα αναζήτησης “#PokemonLetsGo” (β)

```
test.js — Term-Pop
1  const termPopularity = require('./term-popularity');
2
3  // const term = "NintendoSwitch";
4  const term = "PokemonLetsGo";
5  // const term = "PokemonQuest";
6  // const term = "PokemonGO";
7  termPopularity(term, (results) => {
8    console.log('results: \n', results);
9  });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: bash

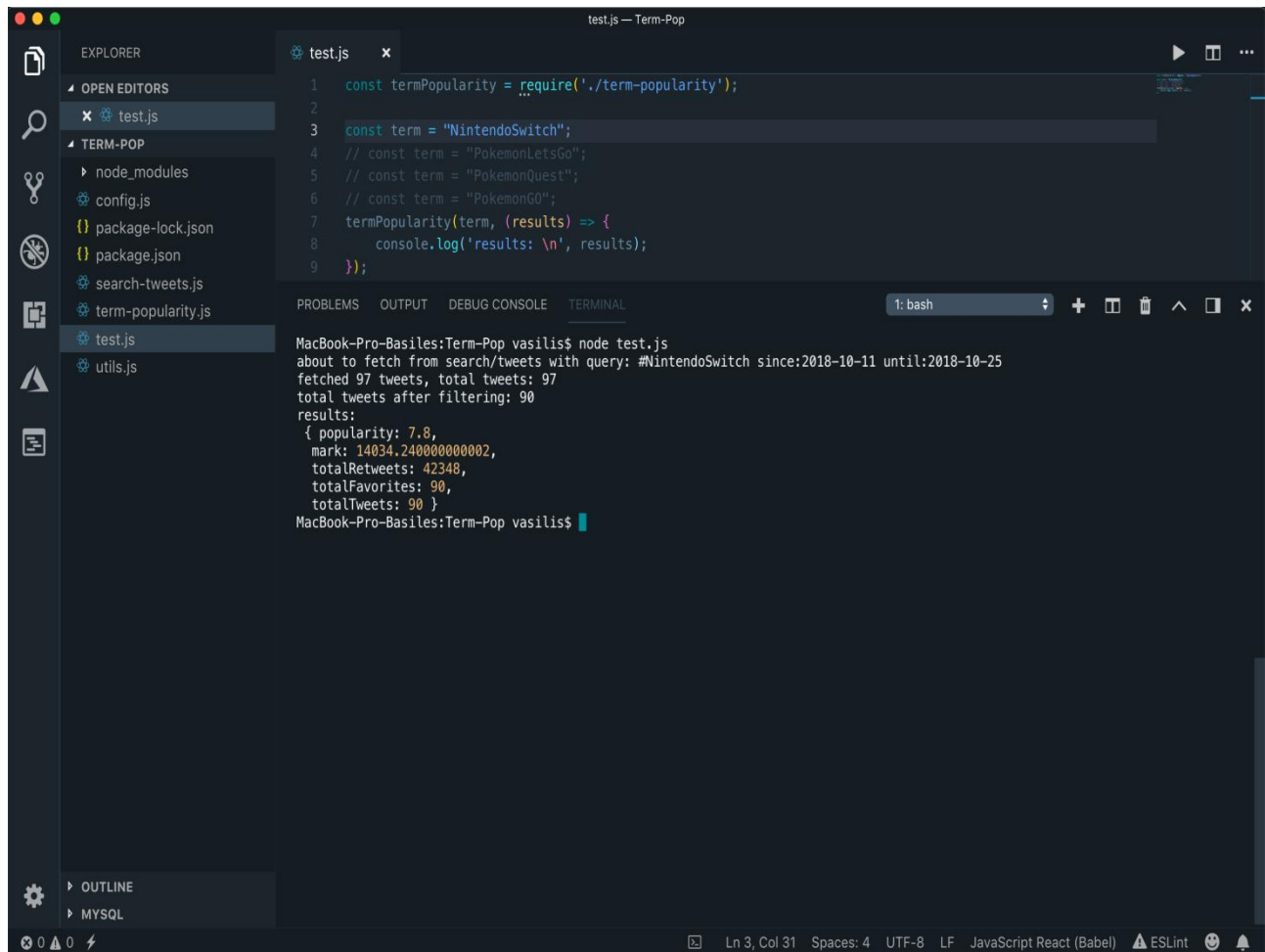
```

fetched 100 tweets, total tweets: 3800
fetched 100 tweets, total tweets: 3900
fetched 100 tweets, total tweets: 4000
fetched 100 tweets, total tweets: 4100
fetched 100 tweets, total tweets: 4200
fetched 100 tweets, total tweets: 4300
fetched 100 tweets, total tweets: 4400
fetched 100 tweets, total tweets: 4500
fetched 100 tweets, total tweets: 4600
fetched 100 tweets, total tweets: 4700
fetched 100 tweets, total tweets: 4800
fetched 100 tweets, total tweets: 4900
fetched 100 tweets, total tweets: 5000
fetched 100 tweets, total tweets: 5100
fetched 100 tweets, total tweets: 5200
fetched 100 tweets, total tweets: 5300
fetched 100 tweets, total tweets: 5400
fetched 100 tweets, total tweets: 5500
fetched 100 tweets, total tweets: 5600
fetched 100 tweets, total tweets: 5700
fetched 100 tweets, total tweets: 5800
fetched 100 tweets, total tweets: 5900
fetched 100 tweets, total tweets: 6000
fetched 100 tweets, total tweets: 6100
fetched 86 tweets, total tweets: 6186
total tweets after filtering: 4091
results:
{ popularity: 6.45,
  mark: 1159565.88,
  totalRetweets: 3501143,
  totalFavorites: 8602,
  totalTweets: 4091 }
```

MacBook-Pro-Basiles:Term-Pop vasilis\$

Εικόνα 12: Παράδειγμα αναζήτησης “#PokemonLetsGo” (γ)

## Παράδειγμα 4ο: #NintendoSwitch

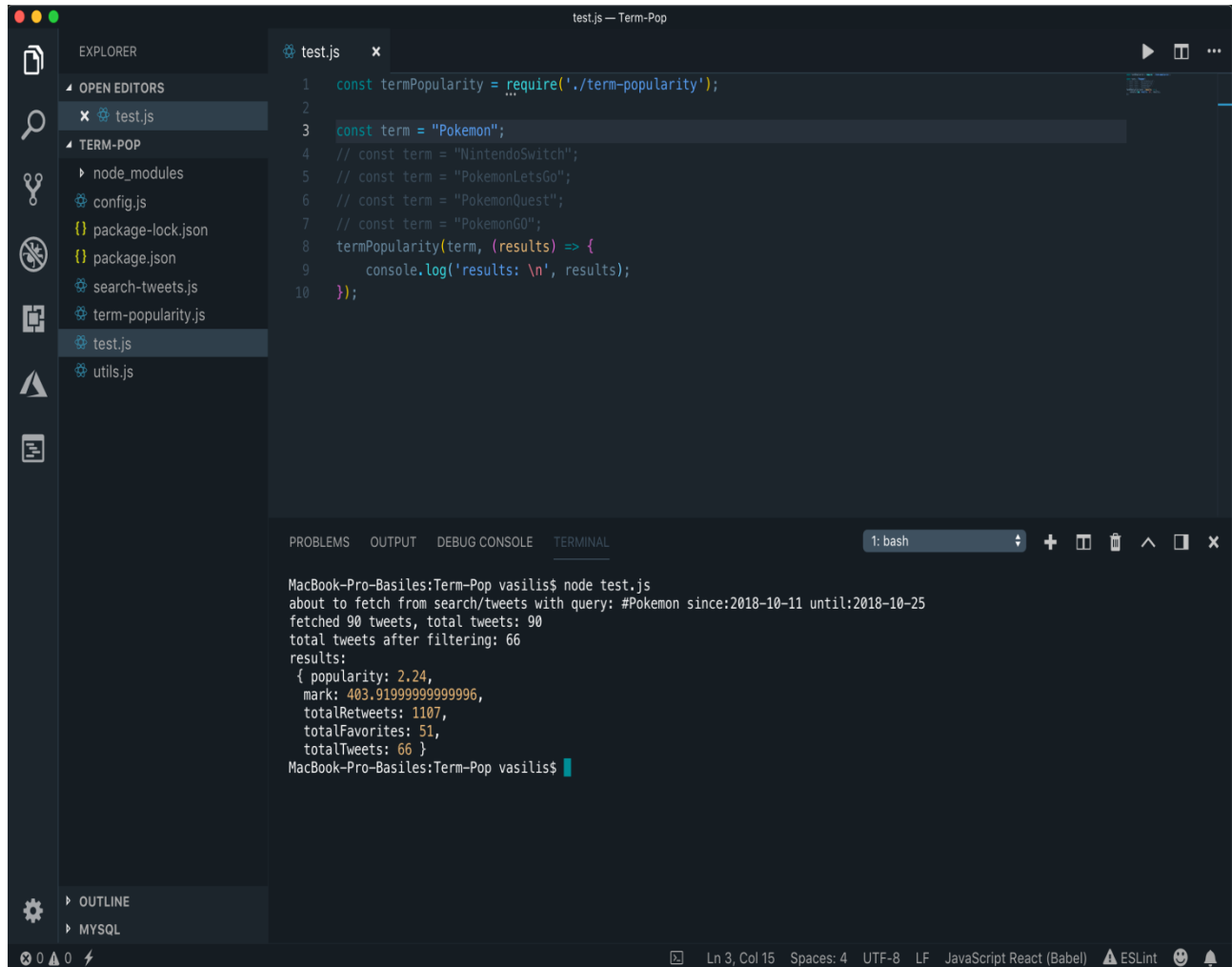


```
test.js — Term-Pop
1  const termPopularity = require('./term-popularity');
2
3  const term = "NintendoSwitch";
4  // const term = "PokemonLetsGo";
5  // const term = "PokemonQuest";
6  // const term = "PokemonGO";
7  termPopularity(term, (results) => {
8    console.log('results: \n', results);
9  });
```

```
MacBook-Pro-Basiles:Term-Pop vasilis$ node test.js
about to fetch from search/tweets with query: #NintendoSwitch since:2018-10-11 until:2018-10-25
fetched 97 tweets, total tweets: 97
total tweets after filtering: 90
results:
{ popularity: 7.8,
  mark: 14034.240000000002,
  totalRetweets: 42348,
  totalFavorites: 90,
  totalTweets: 90 }
MacBook-Pro-Basiles:Term-Pop vasilis$
```

Εικόνα 13: Παράδειγμα αναζήτησης “#NintendoSwitch”

## Παράδειγμα 5ο: #Pokemon



The screenshot shows a VS Code editor window titled "test.js — Term-Pop". The Explorer sidebar on the left shows the file structure of a project named "TERM-POP", including files like config.js, package-lock.json, package.json, search-tweets.js, term-popularity.js, test.js, and utils.js. The main editor area displays the content of test.js:

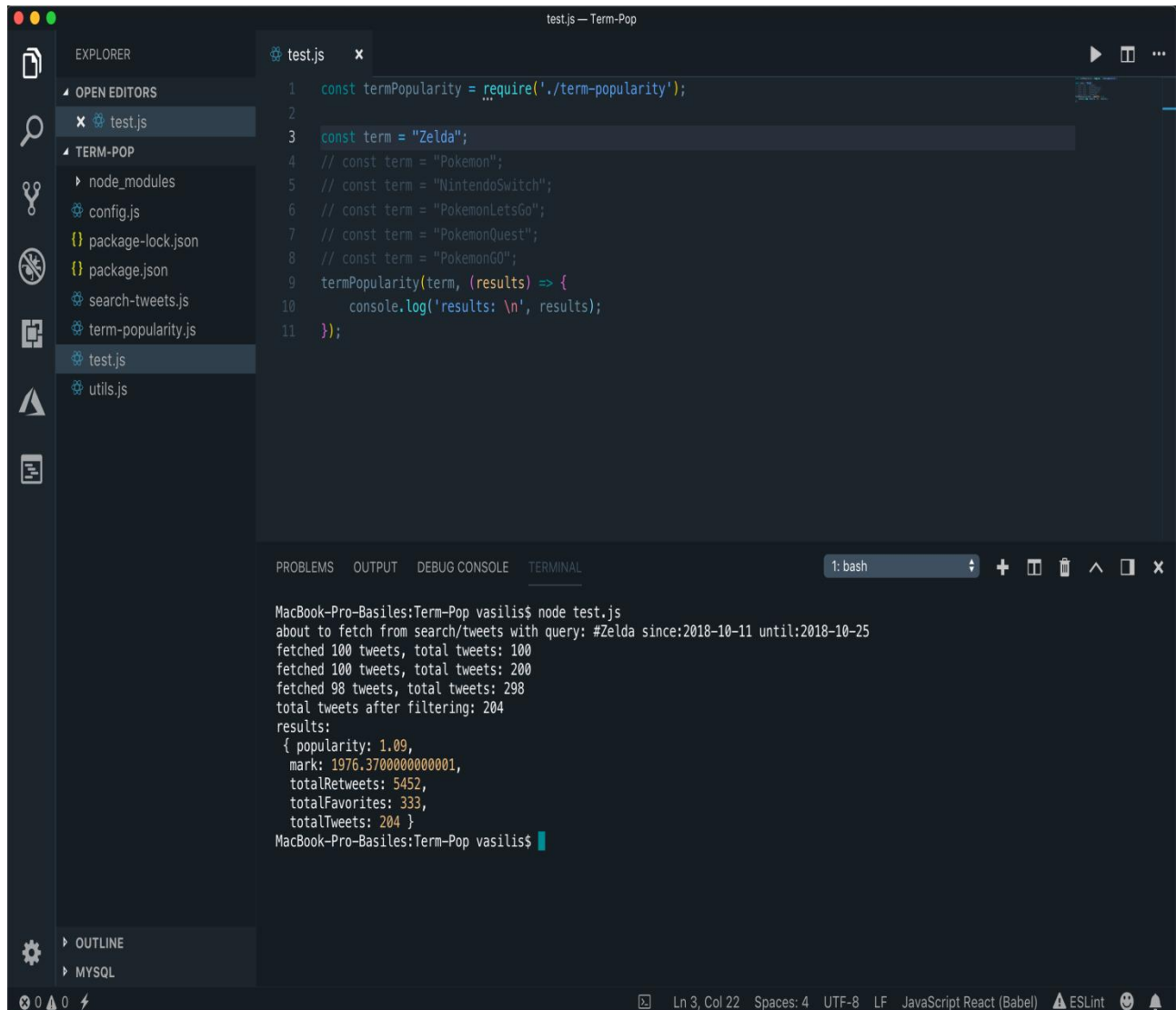
```
1  const termPopularity = require('./term-popularity');
2
3  const term = "Pokemon";
4  // const term = "NintendoSwitch";
5  // const term = "PokemonLetsGo";
6  // const term = "PokemonQuest";
7  // const term = "PokemonGO";
8  termPopularity(term, (results) => {
9      console.log('results: \n', results);
10 });
```

Below the editor, the TERMINAL panel shows the output of running the command `node test.js` in a bash shell:

```
MacBook-Pro-Basiles:Term-Pop vasilis$ node test.js
about to fetch from search/tweets with query: #Pokemon since:2018-10-11 until:2018-10-25
fetched 90 tweets, total tweets: 90
total tweets after filtering: 66
results:
{ popularity: 2.24,
  mark: 403.91999999999996,
  totalRetweets: 1107,
  totalFavorites: 51,
  totalTweets: 66 }
```

Εικόνα 14: Παράδειγμα αναζήτησης “#Pokemon”

## Παράδειγμα 6ο: #Zelda

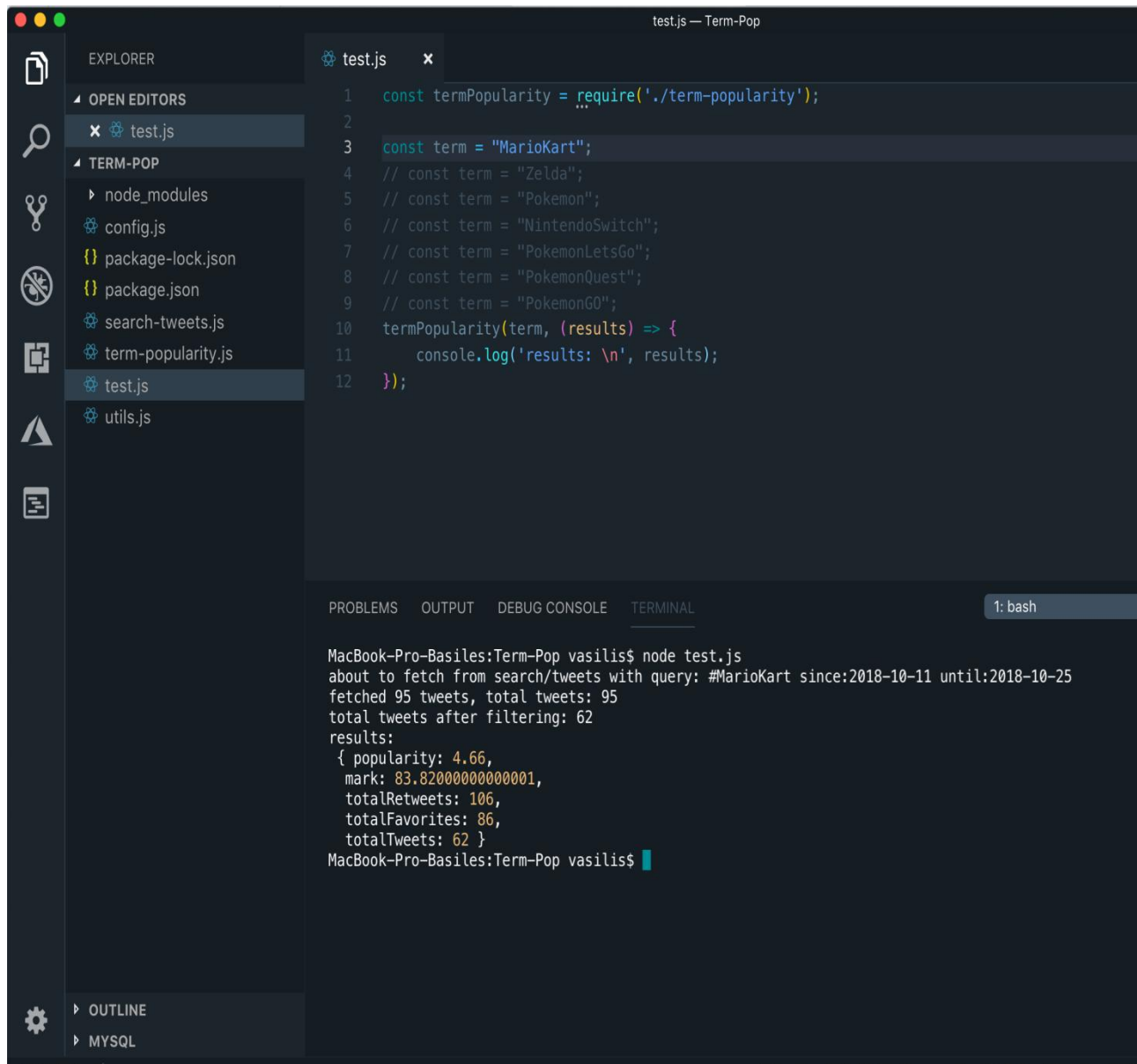


```
test.js — Term-Pop
1  const termPopularity = require('./term-popularity');
2
3  const term = "Zelda";
4  // const term = "Pokemon";
5  // const term = "NintendoSwitch";
6  // const term = "PokemonLetsGo";
7  // const term = "PokemonQuest";
8  // const term = "PokemonGO";
9  termPopularity(term, (results) => {
10     console.log('results: \n', results);
11 });
```

```
MacBook-Pro-Basiles:Term-Pop vasilis$ node test.js
about to fetch from search/tweets with query: #Zelda since:2018-10-11 until:2018-10-25
fetched 100 tweets, total tweets: 100
fetched 100 tweets, total tweets: 200
fetched 98 tweets, total tweets: 298
total tweets after filtering: 204
results:
{ popularity: 1.09,
  mark: 1976.3700000000001,
  totalRetweets: 5452,
  totalFavorites: 333,
  totalTweets: 204 }
MacBook-Pro-Basiles:Term-Pop vasilis$
```

Εικόνα 15: Παράδειγμα αναζήτησης “#Zelda”

## Παράδειγμα 7ο: #MarioKart



The screenshot shows a Visual Studio Code editor window with a file named `test.js` open. The file contains the following JavaScript code:

```
1  const termPopularity = require('./term-popularity');
2
3  const term = "MarioKart";
4  // const term = "Zelda";
5  // const term = "Pokemon";
6  // const term = "NintendoSwitch";
7  // const term = "PokemonLetsGo";
8  // const term = "PokemonQuest";
9  // const term = "PokemonGO";
10 termPopularity(term, (results) => {
11   console.log('results: \n', results);
12 });
```

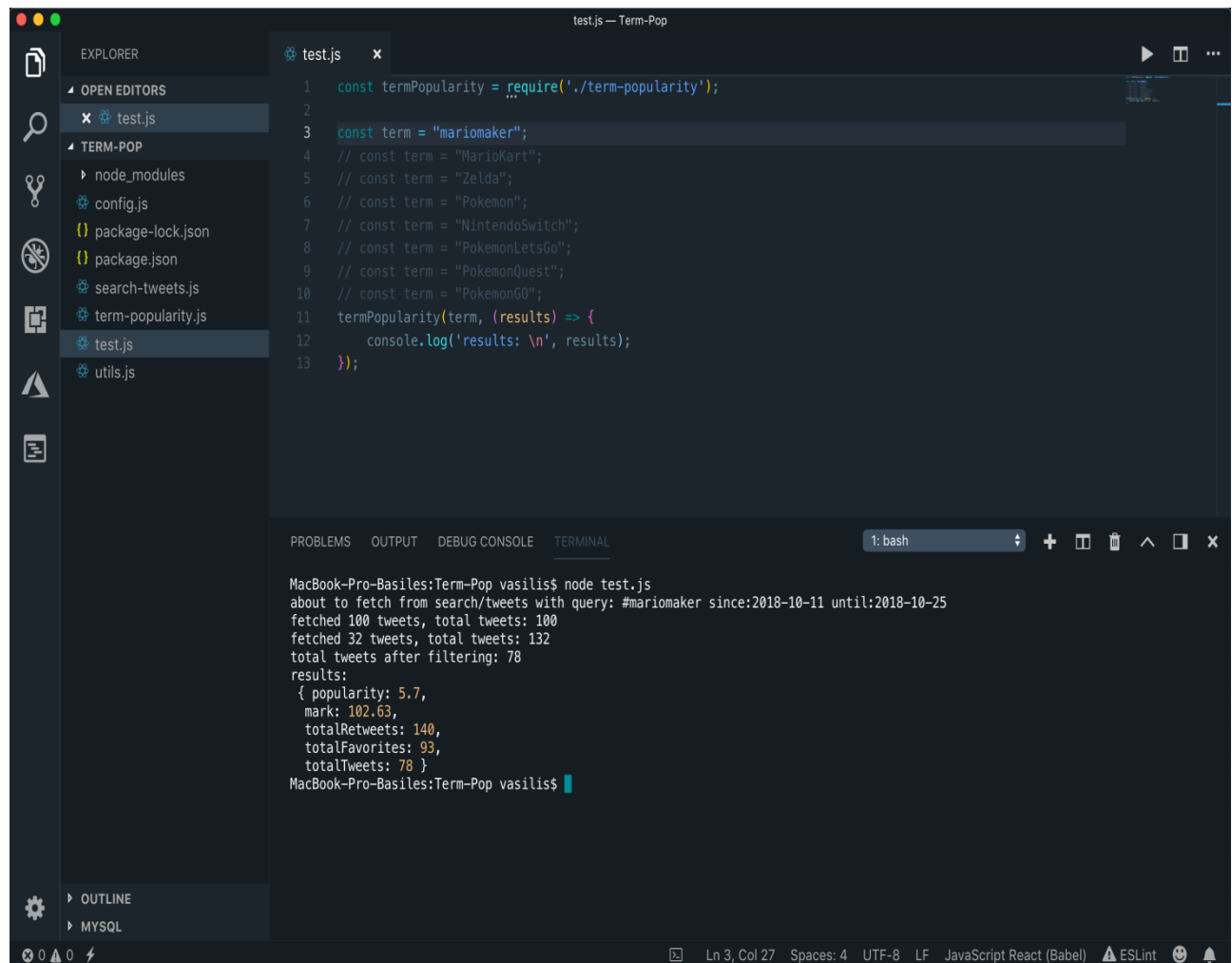
The Explorer sidebar on the left shows the project structure, including `node_modules`, `config.js`, `package-lock.json`, `package.json`, `search-tweets.js`, `term-popularity.js`, `test.js`, and `utils.js`.

The Terminal panel at the bottom shows the output of running `node test.js` in a bash shell:

```
MacBook-Pro-Basiles:Term-Pop vasilis$ node test.js
about to fetch from search/tweets with query: #MarioKart since:2018-10-11 until:2018-10-25
fetched 95 tweets, total tweets: 95
total tweets after filtering: 62
results:
{ popularity: 4.66,
  mark: 83.82000000000001,
  totalRetweets: 106,
  totalFavorites: 86,
  totalTweets: 62 }
```

Εικόνα 16: Παράδειγμα αναζήτησης “#MarioKart”

## Παράδειγμα 8ο: #mariomaker



```
test.js — Term-Pop
1  const termPopularity = require('./term-popularity');
2
3  const term = "mariomaker";
4  // const term = "MarioKart";
5  // const term = "Zelda";
6  // const term = "Pokemon";
7  // const term = "NintendoSwitch";
8  // const term = "PokemonLetsGo";
9  // const term = "PokemonQuest";
10 // const term = "PokemonGO";
11 termPopularity(term, (results) => {
12   console.log('results: \n', results);
13 });
```

```
MacBook-Pro-Basiles:Term-Pop vasilis$ node test.js
about to fetch from search/tweets with query: #mariomaker since:2018-10-11 until:2018-10-25
fetched 100 tweets, total tweets: 100
fetched 32 tweets, total tweets: 132
total tweets after filtering: 78
results:
{ popularity: 5.7,
  mark: 102.63,
  totalRetweets: 140,
  totalFavorites: 93,
  totalTweets: 78 }
MacBook-Pro-Basiles:Term-Pop vasilis$
```

Εικόνα 17: Παράδειγμα αναζήτησης “#mariomaker”



## Βιβλιογραφία:

- 1) “Sequential Summarization: A Full View of Twitter Trending Topics”, Dehong Gao, Wenjie Li, Xiaoyan Cai, Renxian Zhang, και You Ouyang, 2014
- 2) “A New Perspective on Twitter Hashtag Use: Diffusion of Innovation Theory”, Hsia-Ching Chang, 2010
- 3) “Beyond Trending Topics: Real-World Event Identification on Twitter”, Hila Becker, Mor Naaman και Luis Gravano, 2011
- 4) “Classifying Trending Topics: A Typology of Conversation Triggers on Twitter”, Arkaitz Zubiaga, Damiano Spina, Víctor Fresno και Raquel Martínez, 2011
- 5) “Sensing trending topics in Twitter”, Luca Maria Aiello, Georgios Petkos, Carlos Martin, David Corney, Symeon Papadopoulos, Ryan Skraba, Ayse Goker, Yiannis Kompatsiaris και Alejandro Jaimes, 2013
- 6) “Trends in Social Media: Persistence and Decay”, Sitaram Asur, Bernardo A. Huberman, Gabor Szabo και Chunyan Wang, 2011
- 7) “Twitter Trending Topic Classification”, Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md. Mostofa Ali Patwary, Ankit Agrawal, και Alok Choudhary, 2011

- 8) “Trend Visualization on Twitter: What’s Hot and What’s Not?”, Sandjai Bhulai, Peter Kampstra, Lidewij Kooiman, Ger Koole, Marijn Deurloo και Bert Kok, 2012
- 9) “TWITTER TREND ANALYSIS USING R”, Udeh Tochukwu Livinus, 2015
- 10) “Trend Analysis of News Topics on Twitter”, Rong Lu και Qing Yang, 2012

# ΠΑΡΑΡΤΗΜΑ:

## Ο κώδικας του αλγορίθμου της πτυχιακής

### εργασίας

#### Αρχείο “config.js”:

```
// Standard API Keys
// const config = {
//     consumer_key: 'nWFOYiDwURNw8hvety2UXg9',
//     consumer_secret:
// 'trb3NETdthh6Ymr4SkH7F8g1Rqhm4HBFCgSKdeBstAPv36L0QK',
//     access_token: '871120857577525248-
// uToYqbCL9u29x990sAMYH93FnzpSZnw',
//     access_token_secret:
// 'MApnpeCpY0XHTTN9ZJidly901CLUriN0yPTT15gAVsk2H'
// };

// Premium API Keys
const config = {
    consumer_key: "b00ledH69DmE2VZfI1hY47oo1",
    consumer_secret:
    "gNLnmuAn1LkEkTLDGOvcrSTo0De0i0FY8z9BI0fMXaRoSo5dRC",
```

```
    access_token: "2318534912-
3mAHVoV8vRp00Xi2Jd03bL2jhYT0856xIpWnYfQ",

    access_token_secret:
"bm7G5pRrnXKvwIWW0hHF1RAKTW10UpoXBzMwvlghSHLrY"
}
```

```
module.exports = config;
```

### **Αρχείο “utils.js”:**

```
const moment = require('moment'); // date processing lib

// calculate the term's popularity given the total number of likes,
shares and tweets

function findPopularity(totalFavorites, totalRetweets, totalTweets) {
    return totalFavorites * 0.33 + totalRetweets * 0.33 + totalTweets
    * 0.33;
}

// sum of all array elements

function sumOfArray(array) {
    const sum = array.reduce((acc, el) => (acc + el), 0);
    return sum;
}
```

```
// returns current date - 14 days (format: YYYY-MM-DD)
function getOldDate() {
    let date = moment().subtract(14, 'days').format("YYYY-MM-DD");
    return date;
}

// returns current date (format: YYYY-MM-DD)
function getNewDate() {
    let date = moment().format("YYYY-MM-DD");
    return date;
}

module.exports = {
    findPopularity,
    sumOfArray,
    getOldDate,
    getNewDate
}
```

## Αρχείο “search-tweets.js”:

```
let Twitter = require('twit'); // library for fetching data easier  
from Twitter API
```

```
let searchTweets = function (config, query, callback) {
```

```
    let twitter = new Twitter(config);
```

```
    console.log('about to fetch from search/tweets with query: ' +  
query);
```

```
    let tweets = []; // append all fetched tweets in tweets array
```

```
    let delay = 200; // wait 200 milliseconds for each request
```

```
    // async recursive function that fetches tweets given a search  
term
```

```
    let fetchPage = function (maxId, callback, count = 30) {
```

```
        let params = { q: query, count: count.toString(),  
include_entities: 1 };
```

```
        if (maxId) params.max_id = maxId;
```

```

        twitter.get('search/tweets', params, function (err, data,
response) {

        if (err) {

            console.log('error fetching search/tweets with
params', params);

            console.log(err);

            process.exit(1);

        }

        tweets = tweets.concat(data.statuses);

        console.log('fetched ' + data.statuses.length + ' tweets,
total tweets: ' + tweets.length);

        // check if there are no more data

        if (data.statuses.length === 0 ||
!data.search_metadata.next_results) {

            callback(tweets);

            return;

        }

        let next = data.search_metadata.next_results;

        let re = next.match(/max_id=(\d+)/);

        if (!re) {

            callback(tweets);

```

```

        return;
    }

    maxId = re[1];

    // recursive call every 200ms
    setTimeout(function () {
        fetchPage(maxId, callback, 100);
    }, delay);

});

}

// initial call
fetchPage(undefined, callback, 100);

}

module.exports = searchTweets;

```



## Αρχείο “term-popularity.js”:

```
const searchTweets = require('./search-tweets');

const config = require('./config'); // import twitter credentials

const {findPopularity, sumOfArray, getOldDate, getNewDate} =
require('./utils'); // import helper functions

function termPopularity(term, callback) {

    const oldDate = getOldDate(); // current date - 14 days (format:
YYYY-MM-DD)

    const newDate = getNewDate(); // current date (format: YYYY-MM-DD)

    // prepare the search query
    let query = `#${term} since:${oldDate} until:${newDate}`;

    let _term = term.toLowerCase(); // convert the term's letters to
lower case for filtering

    // fetches tweets during since 14 days ago given a search term and
a config object

    searchTweets(config, query, tweets => {

        // find tweets that contain the #term in the hashtag array
        tweets = tweets.filter(tweet => {

            let hashtags = tweet.entities.hashtags;
```

```

    if (hashtags) {
        hashtags = hashtags.filter(hashtag => {
            return hashtag.text.toLowerCase() == _term;
        });

        return hashtags.length > 0 ? true : false;
    } else {
        return false;
    }
});

console.log(`total tweets after filtering: ${tweets.length}`);

let retweet_count_arr = tweets.map(tweet =>
(tweet.retweet_count)); // get tweet's retweets

let favorite_count_arr = tweets.map(tweet =>
(tweet.favorite_count)); // get tweet's favourites

let totalRetweets = sumOfArray(retweet_count_arr);
let totalFavorites = sumOfArray(favorite_count_arr);
let totalTweets = tweets.length;

let mark = findPopularity(totalFavorites, totalRetweets,
totalTweets);

// fitting/scaling `X` into a range [a,b]:

```

```

    // Xnormalized = (b - a) * (X - minX / maxX - minX) + a
    // term popularity's initial range: [0, Number.MAX_VALUE]
    // fit/scale `term popularity` into [0-10]:
    popularity = parseFloat(`${10 * (mark /
Number.MAX_VALUE)} `.substring(0, 4));

    let results = {
        popularity, // popularity after fitting it into [0,
Number.MAX_VALUE]
        mark, // popularity before fitting it into [0, 10]
        totalRetweets,
        totalFavorites,
        totalTweets
    };

    callback(results);
});
}

module.exports = termPopularity;

```

## Αρχείο “tests.js”:

```
const termPopularity = require('./term-popularity');

const term = "mariomaker";
// const term = "MarioKart";
// const term = "Zelda";
// const term = "Pokemon";
// const term = "NintendoSwitch";
// const term = "PokemonLetsGo";
// const term = "PokemonQuest";
// const term = "PokemonGO";
termPopularity(term, (results) => {
    console.log('results: \n', results);
});
```