**EΣΤΟΙΧΕΙΑ ΟΜΑΔΑΣ**

1ο μέλος: Διονυσία Ψυρρή, ΑΜ:1080424, 4ο έτος, email: up1080424@upnet.gr

**ΕΡΩΤΗΜΑ 1**

**Εργαλεία εγκατάστασης**: pip install elasticsearch(vsc)

Python: 3.7.9

Elasticsearch: 8.5.0

pip install pandas:numpy-1.21.6 pandas-1.3.5 python-dateutil-2.8.2 pytz-2022.6 six-1.16.0

Graphical user interface, text

Description automatically generatedΈπειτα απο εγκατάσταση της έκδοσης της elastic που ανεφέρεται επάνω για να μπορέσουμε να συνδεθούμε εκτελούμε τις παραπάνω εντολές στο command prompt:

Graphical user interface, text, application

Description automatically generatedΈπειτα ανοίγουμε εναν browser και πληκτρολογούμε την εντολή localhost:9200:

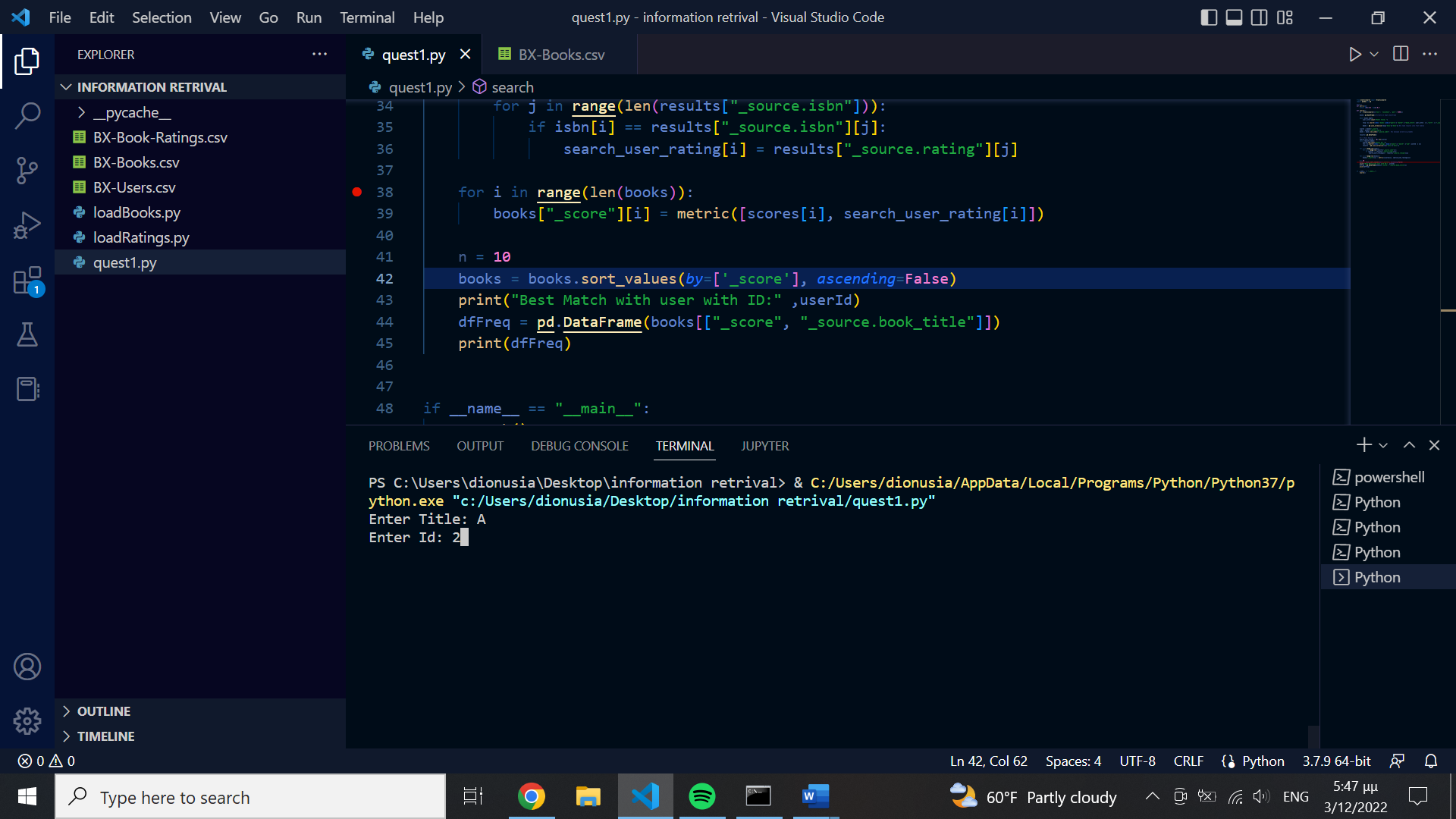
Έπειτα συνδεθήκαμε επιτυχώς στην elasticsearch και προχωρούμε στα επόμενα βήματα.

Ο κώδικας για το διάβασμα και την φόρτωση των δεδομένων είναι στα αρχεία **loadBooks.py**, **loadRatings.py**, διαβάζουμε τα αρχεία με την pandas, τα μετατρέπουμε σε json file(για την αναπαράσταση και αποθήκευση δεδομένων) και έπειτα αυτα τα json τα μετατρέπουμε σε python dictionary(καθώς ένα λεξικό Python είναι η πραγματική δομή δεδομένων (αντικείμενο) που διατηρείται στη μνήμη ενώ εκτελείται ένα πρόγραμμα Python) ώστε να κάνουμε μαζική εισαγωγή δεδομένων στην elasticsearch με την helpers.bulk της οποίας το πλεονέκτημα που την χρησιμοποιήσαμε αντί για ενα for loop για παράδειγμα είναι οτι με μια κλήση κάνει μαζική εισαγωγή πολλαπλών εγγραφών χωρίς να υπολογίζει εκ νέου κάθε φορά όπως θα γινόταν εάν χρησιμοποιούσαμε for loop

Συντομη περιγραφη του κώδικα για το ερώτημα, ο κώδικας είναι στο αρχείο **quest1.py**: συνδεόμαστε στην elasticsearch για να πάρουμε τα δεδομένα που χρειαζόμαστε, δημιουργήσαμε δυο άδεια dataframe ένα για να αποθηκεύσουμε τα δεδομένα του query για το match του string που έδωσε ο χρήστης με τους τίτλους και ένα για το query για το αναγνωριστικό του χρήστη και στις δυο περιπτώσεις χρησιμοποιήσαμε την συνάρτηση json\_normalize() ώστε τα αποτελέσματα μας να τα πάρουμε σε επίπεδους πίνακες, όσο για το ταίριασμα του uid και του book\_title, και στα δυο query παίρνουν το κοινό field isbn ώστε έπειτα και απο την ολοκλήρωση και των δυο query με ένα for loop ελέγχουμε τα κοινά isbn ώστε να έχουμε κάνει το match που θέλουμε και να αποθηκεύσουμε τα ζητούμενα ratings στην λίστα search\_user\_rating. Τέλος την επιστροφή του 10% των βιβλίων με το καλύτερο ταίριασμα το πετύχαμε με την εντολή**: print(dfFreq.head(int(len(dfFreq)\*(n/100)))),** όπου ορίσαμε το n = 10 και μέσω της head() η οποία μας δίνει την δυνατότητα να εμφανίσουμε τον αριθμό των σειρών της επιλογής μας, επιστρέψαμε το 10%

Παραθέτουμε ένα παράδειγμα για να δείξουμε το 10%:

Χωρίς την εντολή head():



A screenshot of a computer

Description automatically generated with medium confidence

Με την εντολή head():

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated