

Σκοπός της εργασίας είναι να υλοποιήσετε ένα frontend και ένα backend τα οποία θα επιτρέπουν στον υποτιθέμενο χρήστη να διαχειρίζεται μία απομακρυσμένη βάση δεδομένων που περιέχει βιβλία.

## Frontend

Η παρουσίαση και διαχείριση όλων των λειτουργιών θα γίνεται με μία μόνο “frontend.html” σελίδα η οποία θα παίρνει δεδομένα με AJAX calls προς το API που θα υλοποιεί ο server.

Η σελίδα θα υποστηρίζει δύο λειτουργίες:

### Καταχώρηση νέου βιβλίου

Ο χρήστης θα μπορεί να καταχωρήσει δεδομένα (author, title, genre, price) σε 4 inputs. Όλα θα είναι textboxes με εξαίρεση το genre το οποίο θα είναι ένα dropdown list από το οποίο ο χρήστης θα μπορεί να επιλέξει από τα ακόλουθα:

- Science fiction
- Satire
- Drama
- Action and Adventure
- Romance
- Mystery
- Horror

Το σύστημα θα κάνει κατάλληλους ελέγχους για να βεβαιώνεται ότι ο χρήστης έχει δώσει το σωστό input στο price και θα ενημερώνει το χρήστη για τις διορθώσεις που πρέπει να κάνει.

Με το πάτημα ενός κουμπιού, ένα AJAX HTTP request θα φεύγει προς το κατάλληλο endpoint του API που υλοποιεί το backend του server για την καταχώρηση του βιβλίου και με τη λήψη του HTTP response θα ενημερώνει το χρήστη για την επιτυχία ή όχι της λειτουργίας

### Αναζήτηση βιβλίου με λέξη κλειδί

Ο χρήστης μπορεί να γράψει μία λέξη σε ένα textbox και με το πάτημα ενός κουμπιού ένα HTTP request με AJAX θα καλεί το αντίστοιχο endpoint του server. Με την επιστροφή του HTTP response, η σελίδα θα εμφανίζει δομημένα τα δεδομένα των βιβλίων που επεστράφησαν ή θα ενημερώνει για τη μη εύρεση το χρήστη.

## Backend

Το backend αποτελείται από ένα RESTful API στο server το οποίο επιτρέπει τη διαχείριση μίας MySQL (ή SQLite ή όποια άλλη θέλετε) βάσης και στην οποία αποθηκεύονται τα βιβλία. Η βάση μπορεί να βρίσκεται στο ίδιο ή άλλο μηχάνημα. Η δομή της βάσης προκύπτει με βάση το ακόλουθο SQL query:

```
CREATE TABLE `books` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
```

```
`author` VARCHAR( 25 ) NOT NULL ,  
`title` VARCHAR( 40 ) NOT NULL ,  
`genre` VARCHAR( 20 ) NOT NULL ,  
`price` FLOAT NOT NULL  
) ENGINE = MYISAM ;
```

Το API αποτελείται από 2 endpoints τα οποία υλοποιούνται στο web server:

URL: `http://<yourdomain>:<yourport>/books/{keyword}`

Μέθοδος: GET

Είσοδος: Το keyword ως μέρος του URL

Έξοδος: Ένα json που περιέχει ένα array από objects με τα στοιχεία του/των βιβλίο/ων (συμπεριλαμβανόμενου και του ID του/τους) που ο τίτλος τους περιέχει το keyword. Παράδειγμα εξόδου για keyword "rings":

```
[  
  {  
    "id" : 134  
    "author" : "J. R. R. Tolkien"  
    "title" : "Lord of the Rings"  
    "genre" : "fantasy"  
    "price" : 20.5  
  },  
  {  
    "id" : 2367  
    "author" : "W. G. Sebald"  
    "title" : "The Rings of Saturn"  
    "genre" : "fiction"  
    "price" : 13.7  
  }  
]
```

URL: `http://<yourdomain>:<yourport>/books/`

Μέθοδος: POST

Είσοδος: Ένα JSON που περιγράφει ένα object με τα στοιχεία του βιβλίου που καταχωρεί ο χρήστης.

Έξοδος: Ένα JSON μήνυμα που να αναφέρει αν η λειτουργία ήταν επιτυχής ή όχι

Παράδειγμα εισόδου:

```
{  
  "id" : 2367  
  
  "author" : "W. G. Sebald"  
  
  "title" : "The Rings of Saturn"  
  
  "genre" : "fiction"  
  
  "price" : 13.7  
}
```

## Γενικές παρατηρήσεις

- Στο server πρέπει να χρησιμοποιήσετε node.js.
- Στον client θα πρέπει να χρησιμοποιήσετε αποκλειστικά Javascript και καμία άλλη βιβλιοθήκη (jQuery, Angular, κλπ)
- Ο κώδικας θα πρέπει να τρέχει τουλάχιστον σε Firefox και Chrome.
- Θα πρέπει να «στήσετε» ένα web και έναν MySQL server
- Η εργασία είναι ατομική.
- Η βάση δεδομένων πρέπει να είναι ακριβώς όπως αναφέρεται πιο πάνω (SQL Query)
- Τα παραδοτέα της εργασίας είναι ο κώδικας που γράψατε και ένα txt αρχείο το οποίο θα περιέχει το URL από το οποίο μπορεί κανείς να δοκιμάσει το σύστημα. ΔΕ χρειάζονται τα nodeJS modules που χρησιμοποιείτε από το npm. Μόνο τα .js, .html, .css και το .txt που γράψατε εσείς. Τα συμπιέζετε με το ZIP (ΌΧΙ rar, 7z, tar ή οτιδήποτε) και φτιάχνετε ένα αρχείο με όνομα τον ΑΜ και κατάληξη ".zip". Π.χ. 12345.zip, αν ο ΑΜ είναι 12345.
- ΠΡΟΣΟΧΗ! Οι οδηγίες και όροι που αναφέρονται στον Οδηγό Υποβολής Εργασιών θα τηρηθούν αυστηρά. Μη συμμόρφωση ακόμα και στη μικρότερη λεπτομέρεια θα συνεπάγεται βαθμούς ποινή ή μη αποδοχή της εργασίας.
- Η παράδοση μετά την προθεσμία επιτρέπεται με κόστος -2 βαθμούς για κάθε 24ωρο καθυστέρησης.

Μπορείτε να χρησιμοποιήσετε είτε το δικό σας webserver είτε κάποιο cloud service provider.