



# Μάθημα: Προηγμένος Προγραμματισμός Full Stack JavaScript Frameworks

## Ατομική

### Απαλλακτική εργασία 8 μονάδων

#### Θέμα:

Στο πλαίσιο της εργασίας σας, θα αναπτύξετε ένα API που υποστηρίζει λειτουργίες **έξυπνου τουρισμού (Smart Tourism)**. Τα δεδομένα παρέχονται σε αρχείο JSON, το οποίο βρίσκεται στον φάκελο του project, και περιλαμβάνει πληροφορίες για διάφορες χώρες. Το αρχείο περιέχει τα εξής πεδία:

- **Country:** Όνομα χώρας (π.χ., "Greece").
- **Quality of Life:** Δείκτης ποιότητας ζωής (ranking: όσο μικρότερη η τιμή, τόσο καλύτερη).
- **Adventure:** Δείκτης περιπέτειας (ranking: όσο μικρότερη η τιμή, τόσο καλύτερη).
- **Heritage:** Δείκτης πολιτισμικής κληρονομιάς (ranking: όσο μικρότερη η τιμή, τόσο καλύτερη).
- **Cost of Living Index:** Δείκτης κόστους ζωής (όσο μεγαλύτερη η τιμή, τόσο ακριβότερη η χώρα).
- **Restaurant Price Index:** Δείκτης κόστους εστίασης (όσο μεγαλύτερη η τιμή, τόσο ακριβότερη η χώρα).

Και έχουν εξαχθεί από τις παρακάτω πηγές:

1. [Cost of Living Index by Country 2024](#)
2. [Best Countries Rankings 2023](#)

#### Παράδειγμα Εγγραφής από το JSON

```
{  
  "Country": "Greece",  
  "Quality of Life": 28,  
  "Adventure": 3,  
  "Heritage": 2,  
  "Cost of Living Index": 52.0,  
  "Restaurant Price Index": 51.3  
}
```

Καλείστε να κάνετε τα παρακάτω.



## Βήματα Υλοποίησης

### 1. Αποθήκευση Αρχικών Δεδομένων στη Βάση Δεδομένων:

- Διαβάστε το αρχείο JSON που περιέχει τα δεδομένα των χωρών.
- Αποθηκεύστε τα δεδομένα αυτά σε μία βάση δεδομένων MongoDB, χρησιμοποιώντας τη Mongoose.

### 2. Λειτουργίες API (CRUD):

Στη συνέχεια, θα συνδέεστε στη MongoDB βάση σας στην οποία θα κάνετε crud operations. Πιο συγκεκριμένα, δημιουργήστε endpoints για τις παρακάτω λειτουργίες:

#### i. Ανάγνωση Δεδομένων (Read Requests):

- Επιστροφή όλων των χωρών.
- Φιλτράρισμα και ταξινόμηση χωρών με βάση:
  - Κόστος ζωής (π.χ., φθηνότερες ή ακριβότερες χώρες).
  - Κόστος εστίασης (π.χ., οικονομικές ή ακριβές χώρες).
  - Quality of Life, Adventure, Heritage (όσο μικρότερο το ranking, τόσο καλύτερο).

#### ii. Ενημέρωση Δεδομένων (Update Request):

Τροποποίηση των δεδομένων μιας χώρας, όπως ενημέρωση κόστους ζωής ή εστίασης.

#### iii. Προσθήκη Νέων Δεδομένων (Post Request):

Εισαγωγή νέας χώρας (μπορείτε να βάλετε δεδομένα που δεν είναι απαραίτητα αληθινά)

#### iv. Διαγραφή Δεδομένων (Delete Request):

Διαγραφή μιας χώρας από τη βάση δεδομένων.

## Οδηγίες Υλοποίησης

### 1. Δομή του Project:

- Οργανώστε την εφαρμογή σας σε αρχιτεκτονική **MVC**.
- Περιλαμβάνετε ένα αρχείο package.json με τις εξαρτήσεις και τις ρυθμίσεις του project.

### 2. Παράμετροι API:

- Τα requests θα δέχονται παραμέτρους όπως:
  - criterion για το πεδίο φιλτραρίσματος (π.χ., Quality of Life, Cost of Living Index).
  - type για το είδος ταξινόμησης (π.χ., lowest, highest).



- limit για τον αριθμό των εγγραφών που επιστρέφονται.

### 3. Παραδείγματα Λειτουργιών:

- ο Επιστροφή όλων των χωρών:

GET /api/v1/countries

- ο Φιλτράρισμα φθηνότερων χωρών:

GET/api/v1/countries/filter?criterion=Cost of Living  
Index&type=lowest&limit=5

### 4. Επεξήγηση Φίλτρων:

- ο Τα πεδία Quality of Life, Adventure, Heritage είναι **rankings**: όσο μικρότερη η τιμή, τόσο καλύτερα.
- ο Τα πεδία Cost of Living Index και Restaurant Price Index είναι **δείκτες κόστους**: όσο μεγαλύτερη η τιμή, τόσο ακριβότερα.

### Προαιρετικό Bonus (2 Μονάδες):

Υλοποιήστε μία απλή διεπαφή front-end που επιτρέπει στους χρήστες να βλέπουν και να φιλτράρουν τις πληροφορίες του API.

#### Επιπλέον:

- Η εργασία **πρέπει** να αναπτυχθεί χρησιμοποιώντας το **express** (<https://expressjs.com/>)
- Για τη χρήση του παρόντος API, **δεν** απαιτείται αυθεντικοποίηση (authentication)
- **Δεν** απαιτείται να δημιουργηθεί front end.
- Για τη βάση καλείστε να χρησιμοποιήσετε την **MongoDB** και τη **mongoose**
- Αν θεωρείτε ότι χρειάζεται, να μου **στείλετε** και **readme** αρχείο
- Το project σας θα πρέπει να είναι οργανωμένο σε **MVC αρχιτεκτονική**.
- Η εφαρμογή θα πρέπει να περιέχει **ένα package.json** αρχείο. Εκεί θα ορίζεται από που γίνεται η εκτέλεση της εφαρμογής. Στο terminal εκτελώντας την εντολή **npm start** θα πρέπει να γίνεται εκκίνηση της εφαρμογής.
- Να χρησιμοποιούνται **μεταβλητές περιβάλλοντος** στην εφαρμογή σας.
- Μπορείτε να αποστείλετε το postman collection



- Μαζί με τον κώδικα **καλείστε** να παραδώσετε και ένα **έγγραφο** στο οποίο θα περιλαμβάνονται ως απαντήσεις στα παραπάνω ζητούμενα, τα requests που κάνετε, και τα responses που λαμβάνετε.

#### Σημαντικό:

- ✓ Σε περίπτωση που δεν καταφέρετε να υλοποιήσετε ολόκληρη την εργασία, μπορείτε να στείλετε μέρος της (το οποίο βέβαια να είναι λειτουργικό) και θα βαθμολογηθείτε για αυτό!
- ✓ Το παραδοτέο σας θα είναι ένα αρχείο .zip το οποίο θα περιλαμβάνει τον κώδικα, το έγγραφο και ότι άλλο κρίνετε απαραίτητο (πχ postman collection) το οποίο πρέπει να το ανεβάσετε στο χώρο των εργασιών του e-class εντός της προθεσμίας.

Καλή επιτυχία!