



Μηχανική Λογισμικού 2 (2025 – 2026) - Εργασία Μαθήματος

1. Σκοπός της εργασίας

Σκοπός της εργασίας είναι η εφαρμογή των εννοιών που διδάσκονται στα πλαίσια των μαθημάτων Μηχανική Λογισμικού I και Μηχανική Λογισμικού II. Πέρα από τις έννοιες που παρουσιάστηκαν στα πλαίσια της Μηχανικής Λογισμικού I (καθορισμός απαιτήσεων χρηστών και λογισμικού, καθορισμός προδιαγραφών, σχεδίαση συστήματος), η παρούσα εργασία εστιάζει στη διαχείριση του έργου (project management), τις διαδικασίες συνεχούς ενσωμάτωσης και ανάπτυξης (Continuous Integration – Continuous Deployment, CI/CD), τον έλεγχο του λογισμικού (testing), αλλά και την ενσωμάτωση των μετρικών ποιότητας στη διαδικασία βελτίωσης του έργου.

2. Έργο λογισμικού

Για την εργασία του μαθήματος θα χρησιμοποιηθούν ως βάση οι εργασίες του μαθήματος “Μηχανική Λογισμικού I”. Στις εργασίες αυτές είχατε καθορίσει τις απαιτήσεις χρηστών και λογισμικού στο οποίο επιλέξατε να εργαστείτε, και στη συνέχεια σχεδιάσατε τις προδιαγραφές της εφαρμογής σας, τόσο του backend όσο και του frontend, μέσα από μία σειρά assets που κληθήκατε να δημιουργήσετε, όπως για παράδειγμα Mockups, Swagger Files, User Stories, Requirements, Activity Diagrams, κλπ. Με βάση αυτά τα assets, τα οποία μπορείτε πλέον να κατεβάσετε συνολικά ανά ενότητα μέσω του Usereq¹, καλείστε να αναπτύξετε ένα web application το οποίο θα αποτελείται από ένα λειτουργικό backend και ένα λειτουργικό frontend που θα καλεί τα endpoints του backend και θα αντλεί mock/hardcoded πληροφορία η οποία θα εμφανίζεται στο frontend. Τόσο για το backend, όσο και για το frontend, δεν είναι απαραίτητο να αναπτύξετε την εφαρμογή στο σύνολό της, όπως ακριβώς την περιγράψατε στο πλαίσιο της ΜΛ1, αλλά θα πρέπει να πληρούνται οι παρακάτω προδιαγραφές:

Backend:

- Θα πρέπει να υπάρχουν τουλάχιστον 10 (δέκα) διαθέσιμα routes/API calls που ένας εξωτερικός χρήστης μπορεί να χτυπήσει και να περιμένει απάντηση.
- Θα πρέπει να υπάρχει τουλάχιστον ένα route τύπου GET, ένα route τύπου POST, ένα route τύπου PUT και ένα route τύπου DELETE. Ιδανικά, σε ένα resource, ώστε να μπορεί να ελεγχθεί πλήρως η λειτουργικότητα.



- Θα πρέπει να εμπεριέχονται τουλάχιστον 3 (τρεις) διαφορετικές οντότητες (π.χ. χρήστες-βιβλία-βιβλιοθήκες), οι οποίες αλληλοεπιδρούν μεταξύ τους στα διάφορα routes του συστήματος (π.χ. ένα route καταγράφει τον δανεισμό ενός βιβλίου από έναν χρήστη, ένα route ρωτάει αν υπάρχει διαθεσιμότητα ενός βιβλίου σε μια βιβλιοθήκη κλπ).
- Για τις ανάγκες των routes, και επειδή κάποια από τα δεδομένα θα πρέπει να εμφανίζονται και στο frontend, θα πρέπει να χρησιμοποιηθούν mock δεδομένα που θα είναι hardcoded μέσα στον κώδικα του backend, ώστε να είναι άμεσα διαθέσιμα σε κάθε εκτέλεση του server. Προαιρετικά, μπορείτε να συνδέσετε τον backend σας με μια βάση δεδομένων MongoDB², εφόσον επιθυμείτε τα δεδομένα να αποθηκεύονται μόνιμα. Τα mock δεδομένα μπορούν να είναι απλά παραδείγματα αντικειμένων (π.χ. λίστες χρηστών, βιβλίων ή βιβλιοθηκών), αρκεί να εξυπηρετούν τις οιθόνες του frontend και να επιτρέπουν την επίδειξη της βασικής λειτουργικότητας της εφαρμογής δεν απαιτείται να είναι ρεαλιστικά ή μεγάλης κλίμακας.

Frontend:

- Θα πρέπει να αποτελείται από τουλάχιστον 4 οιθόνες (screens), όπως για παράδειγμα τη σελίδα αναζήτησης βιβλίων, τη σελίδα προβολής περισσότερων πληροφοριών για ένα βιβλίο, τη σελίδα εισαγωγής ενός νέου βιβλίου και τη σελίδα δανεισμού του βιβλίου. Προτείνεται οι οιθόνες να αντιστοιχούν σε κάποια από τα mockups που υλοποιήσατε στην εργασία ΜΛ1, μπορείτε όμως να συμπεριλάβετε και οιθόνες που δεν υπήρχαν σε αυτή, εφόσον κρίνεται απαραίτητο.
- Θα πρέπει να καλούνται τουλάχιστον 5 από τα διαθέσιμα endpoints του backend.

3. Παραγωγή Κώδικα

Για την παραγωγή του κώδικα συστήνεται ακράδαντα η χρήση generative (LLM) μοντέλων. Μπορείτε να χρησιμοποιήσετε οποιοδήποτε μοντέλο επιθυμείτε. Η μεθοδολογία παραγωγής κώδικα που προτείνεται είναι η παρακάτω:

1. Αρχικά θα κατεβάσετε από το usereq τα assets της ΜΛ1 εργασίας σας που θεωρείτε απαραίτητα.
2. Έπειτα, χρησιμοποιώντας τα prompt templates που σας δίνονται μαζί με αυτήν την περιγραφή, θα παράξετε τον κώδικα του backend και του frontend.
3. Επιπλέον prompts πιθανώς να χρειαστούν για να γίνει ο κώδικας λειτουργικός.

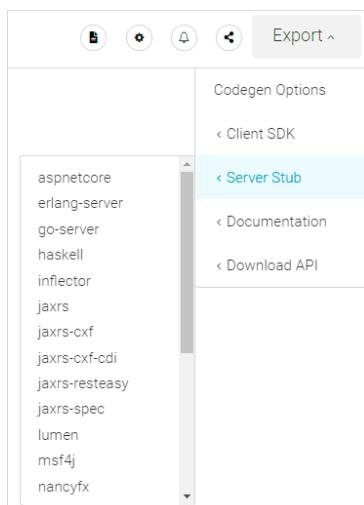
Εναλλακτικά, και εφόσον το κρίνετε απαραίτητο, μπορείτε να χρησιμοποιήσετε την παρακάτω διαδικασία για να παράξετε μία βασική δομή του backend και στη συνέχεια, με χρήση generative μοντέλων, να κάνετε τις

² <https://www.mongodb.com/products/platform/atlas-database>



απαραίτητες αλλαγές ώστε το backend να πληροί τις προαναφερθείσες προϋποθέσεις. Η επιλογή αυτής της προσέγγισης, ή της παραγωγής του backend από την αρχή με generative μοντέλα αφήνεται στην ευχέρειά σας.

Η διαδικασία παραγωγής μέρους του backend από swagger αρχείο απαιτεί τη χρήση SwaggerHub² ή εναλλακτικά open-access issel swagger editor³. Εκεί, αφού κάνετε log in με λογαριασμό της επιλογής σας, μπορείτε να δημιουργήσετε ένα νέο project, στο οποίο θα επικολλήσετε το περιεχόμενο του swagger αρχείου σας (ο issel swagger editor δεν απαιτεί λογαριασμό). Αφού ελέγξετε την ορθότητα του παραγόμενου API (ελέγχοντας τα πιθανά routes που παράγει το SwaggerHub), μπορείτε να παράξετε και να κατεβάσετε τον αντίστοιχο κώδικα που υλοποιεί το API σας, χρησιμοποιώντας την επιλογή “Export”, “Server Stub” και “nodejs-server” (ή εναλλακτικά “Generate Server”, “nodejs-server” για τον issel swagger editor), όπως φαίνεται στην εικόνα παρακάτω:



Στο zip αρχείο που θα κατέβει στον υπολογιστή σας, μπορείτε να βρείτε την υλοποίηση του API server σας, μαζί με τις απαραίτητες βιβλιοθήκες που θα πρέπει να εγκατασταθούν. Εδώ μπορείτε να κάνετε τις απαραίτητες τροποποιήσεις με generative μοντέλα, εφόσον απαιτούνται, ώστε το backend να πληροί τις προϋποθέσεις αυτού του παραδοτέου.

4. Παραδοτέα

Η εργασία αποτελείται από τρία διαφορετικά παραδοτέα, κάθε ένα εκ των οπίων καλύπτει συγκεκριμένα τμήματα από τη διδαχθείσα ύλη του μαθήματος. Στην παρακάτω ενότητα περιγράφεται η δομή, ο σκοπός και



τα ζητούμενα του 1ου παραδοτέου, καθώς και η ημερομηνία παράδοσής του. Περισσότερες πληροφορίες για τα αρχεία που θα πρέπει να παραδώσετε σε κάθε περίπτωση, θα δίνονται κατά τη διάρκεια του εξαμήνου.

4.1. Παραδοτέο 1 (Παράδοση: 23/11/2025)

Στο πρώτο παραδοτέο, το οποίο έχει ημερομηνία παράδοσης την Κυριακή 23/11/2025, σκοπός είναι το αρχικό στήσιμο του έργου, τόσο σε επίπεδο κώδικα, όσο και σε επίπεδο διαχείρισης έργου (project management). Για αυτόν τον σκοπό, το συγκεκριμένο παραδοτέο θα πρέπει να περιέχει:

- Δημιουργία αρχείων του web application (frontend και backend)**

Αναπτύξτε το frontend και το backend της εφαρμογής σας χρησιμοποιώντας την παραπάνω μεθοδολογία. Εφόσον μόνο κρίνεται απαραίτητο, χρησιμοποιείστε το έτοιμο backend που παράγεται από το swaggerHub. Ανεξαρτήτως της μεθοδολογία παραγωγής του κώδικα που θα επιλέξετε, βεβαιωθείτε ότι και οι δύο εφαρμογές (frontend & backend) συμμορφώνονται με τις γενικές απαιτήσεις της εργασίας.

- Αποθετήριο Λογισμικού**

Ανεβάστε τον παραγόμενο κώδικα σε δύο νέα αποθετήρια στο GitHub³ (ένα για το frontend και ένα για το backend), στο οποίο θα πρέπει να προστεθούν όλα τα μέλη της ομάδας, καθώς και οι διδάσκοντες (usernames: asymeon, karanikio, dcnatsos, GSiach13).

- Διαχείριση έργου**

Για το Usereq project σας θα δημιουργήσετε στην πλατφόρμα της Cyclopt⁴ ένα project, σε ομάδα αντίστοιχη με αυτήν που έχετε δηλώσει στο elearning. Για να γίνει αυτό αρκεί ένας από κάθε ομάδα να δημιουργήσει ένα project στη Cyclopt και να προσθέσει τα υπόλοιπα μέλη. Στο παραπάνω project θα περάσουν τα υπάρχοντα features του συστήματός σας (η διαδικασία του Integration θα συζητηθεί στο μάθημα), ενώ στη συνέχεια θα πρέπει να περάσετε και τα μελλοντικά tasks, αυτά δηλαδή που απαιτούνται για την υλοποίηση της εργασίας. Όλα τα tasks θα πρέπει να χρονοπρογραμματιστούν με βάση τη λογική που θα συζητήσουμε στο μάθημα. Θυμίζουμε πως η εργασία αποτελεί ένα agile project, κατά συνέπεια το Cyclopt board του project σας πρέπει να παραμένει ενημερωμένο καθ' όλη τη διάρκεια του εξαμήνου (assessment θα γίνεται σε διεβδομαδιαία βάση).

Επιπλέον, κατά τη διάρκεια του έργου μπορείτε να παρακολουθείτε την ποιότητα του κώδικα του github repo σας (το κάθε μέλος της ομάδας ξεχωριστά) μέσω του εργαλείου Cyclopt Companion⁵, το οποίο προσφέρει αναλυτικές πληροφορίες για όλο το commit history του εκάστοτε Github

³ <https://github.com/>

⁴ <https://platform.cyclopt.com>

⁵ <https://companion.cyclopt.com/>



αποθετηρίου. Αυτό προτείνεται απλά για εξοικείωση και δεν αποτελεί υποχρέωση σε αυτό το παραδοτέο.

- **Βαθμολόγηση**

Η αξιολόγηση του πρώτου παραδοτέου θα βασιστεί στους παρακάτω άξονες. Συγκεκριμένα, θα εξεταστεί η τελική έκδοση του Swagger αρχείου που περιγράφει την εφαρμογή σας, καθώς και το backend και το frontend, με βάση το κατά πόσο τηρούν τις απαιτήσεις που έχουν τεθεί. Για τον λόγο αυτό, θα πρέπει να ανεβάσετε το τελικό Swagger αρχείο στο repository του backend στο GitHub, μέσα σε φάκελο /docs.

Αν και στο πρώτο παραδοτέο το frontend και το backend πρέπει να λειτουργούν και να επικοινωνούν μεταξύ τους (δηλαδή το frontend να αντλεί δεδομένα από τα endpoints του backend και να τα εμφανίζει), η αξιολόγηση δεν θα βασιστεί στο πόσο καλαίσθητο είναι το frontend ούτε στο αν η επικοινωνία των δύο είναι απολύτως απροβλημάτιστη. Μικρά σφάλματα ή δυσλειτουργίες στη διασύνδεση δεν θα επηρεάσουν τη βαθμολογία. Ωστόσο, συνιστάται και τα δύο μέρη (frontend και backend) να είναι πλήρως λειτουργικά, καθώς στο επόμενο παραδοτέο θα χρειαστεί να υλοποιήσετε tests, τα οποία προϋποθέτουν σωστή επικοινωνία και ορθή λειτουργία του συστήματος συνολικά (περισσότερα για αυτό στο 2ο παραδοτέο).

Επιπλέον, στο πρώτο παραδοτέο θα βαθμολογηθεί και η δημιουργία δύο GitHub repositories (ένα για το backend και ένα για το frontend), καθώς και η σωστή οργάνωση των branches (π.χ. ένα branch για το production και ένα για κάθε μέλος της ομάδας). Τέλος, θα αξιολογηθεί και η δημιουργία και ενημέρωση του πλάνου διαχείρισης έργου στην πλατφόρμα της Cyclopt.

Για περισσότερες πληροφορίες:

Ανδρέας Λ. Συμεωνίδης

Καθηγητής

symeonid@ece.auth.gr

Καρανικιώτης Θωμάς

Μεταδιδακτορικός
Ερευνητής

karanikio@ece.auth.gr

Νάτσος Δημοσθένης

Υποψήφιος Διδάκτορας

dcnatsos@ece.auth.gr

Σιαχάμης Γεώργιος

Υποψήφιος Διδάκτορας

gsiacha@ece.auth.gr